

Merging symbolic and data-driven AI for robot autonomy

Edited by

Nikos Katzouris, Simona Perri, Mohan Sridharan and Daniele Meli

Published in

Frontiers in Robotics and AI



FRONTIERS EBOOK COPYRIGHT STATEMENT

The copyright in the text of individual articles in this ebook is the property of their respective authors or their respective institutions or funders. The copyright in graphics and images within each article may be subject to copyright of other parties. In both cases this is subject to a license granted to Frontiers.

The compilation of articles constituting this ebook is the property of Frontiers.

Each article within this ebook, and the ebook itself, are published under the most recent version of the Creative Commons CC-BY licence. The version current at the date of publication of this ebook is CC-BY 4.0. If the CC-BY licence is updated, the licence granted by Frontiers is automatically updated to the new version.

When exercising any right under the CC-BY licence, Frontiers must be attributed as the original publisher of the article or ebook, as applicable.

Authors have the responsibility of ensuring that any graphics or other materials which are the property of others may be included in the CC-BY licence, but this should be checked before relying on the CC-BY licence to reproduce those materials. Any copyright notices relating to those materials must be complied with.

Copyright and source acknowledgement notices may not be removed and must be displayed in any copy, derivative work or partial copy which includes the elements in question.

All copyright, and all rights therein, are protected by national and international copyright laws. The above represents a summary only. For further information please read Frontiers' Conditions for Website Use and Copyright Statement, and the applicable CC-BY licence.

ISSN 1664-8714
ISBN 978-2-8325-6678-7
DOI 10.3389/978-2-8325-6678-7

Generative AI statement

Any alternative text (Alt text) provided alongside figures in the articles in this ebook has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

Merging symbolic and data-driven AI for robot autonomy

Topic editors

Nikos Katzouris — Institute of Informatics and Telecommunications, National Centre of Scientific Research Demokritos, Greece

Simona Perri — University of Calabria, Italy

Mohan Sridharan — University of Edinburgh, United Kingdom

Daniele Meli — University of Verona, Italy

Citation

Katzouris, N., Perri, S., Sridharan, M., Meli, D., eds. (2025). *Merging symbolic and data-driven AI for robot autonomy*. Lausanne: Frontiers Media SA.
doi: 10.3389/978-2-8325-6678-7

Table of contents

- 04 **Editorial: Merging symbolic and data-driven AI for robot autonomy**
Daniele Meli, Mohan Sridharan, Simona Perri and Nikos Katzouris
- 06 **Integrating Non-monotonic Logical Reasoning and Inductive Learning With Deep Learning for Explainable Visual Question Answering**
Heather Riley and Mohan Sridharan
- 26 **Symbolic Representation and Learning With Hyperdimensional Computing**
Anton Mitrokhin, Peter Sutor, Douglas Summers-Stay, Cornelia Fermüller and Yiannis Aloimonos
- 37 **In Search of Trustworthy and Transparent Intelligent Systems With Human-Like Cognitive and Reasoning Capabilities**
Nikhil R. Pal
- 46 **From Continuous Observations to Symbolic Concepts: A Discrimination-Based Strategy for Grounded Concept Learning**
Jens Nevens, Paul Van Eecke and Katrien Beuls
- 66 **Few-Shot Induction of Generalized Logical Concepts via Human Guidance**
Mayukh Das, Nandini Ramanan, Janardhan Rao Doppa and Sriraam Natarajan
- 79 **Commonsense knowledge in cognitive robotics: a systematic literature review**
Jan-Philipp Töberg, Axel-Cyrille Ngonga Ngomo, Michael Beetz and Philipp Cimiano
- 97 **Evaluation of co-speech gestures grounded in word-distributed representation**
Kosuke Sasaki, Jumpei Nishikawa and Junya Morita
- 112 **Deep neural network-based robotic visual servoing for satellite target tracking**
Shayan Ghasvand, Wen-Fang Xie and Abolfazl Mohebbi
- 125 **Synthetic consciousness architecture**
Konstantyn Spasokukotskiy
- 135 **Hybrid intelligence systems for reliable automation: advancing knowledge work and autonomous operations with scalable AI architectures**
Allan Grosvenor, Anton Zemlyansky, Abdul Wahab, Kyrylo Bohachov, Aras Dogan and Dwyer Deighan



OPEN ACCESS

EDITED AND REVIEWED BY
Chenguang Yang,
University of Liverpool, United Kingdom

*CORRESPONDENCE
Daniele Meli,
✉ daniele.meli@univr.it

RECEIVED 09 July 2025
ACCEPTED 15 July 2025
PUBLISHED 23 July 2025

CITATION
Meli D, Sridharan M, Perri S and Katzouris N
(2025) Editorial: Merging symbolic and
data-driven AI for robot autonomy.
Front. Robot. AI 12:1662674.
doi: 10.3389/frobt.2025.1662674

COPYRIGHT
© 2025 Meli, Sridharan, Perri and Katzouris.
This is an open-access article distributed
under the terms of the [Creative Commons
Attribution License \(CC BY\)](#). The use,
distribution or reproduction in other forums is
permitted, provided the original author(s) and
the copyright owner(s) are credited and that
the original publication in this journal is cited,
in accordance with accepted academic
practice. No use, distribution or reproduction
is permitted which does not comply with
these terms.

Editorial: Merging symbolic and data-driven AI for robot autonomy

Daniele Meli^{1*}, Mohan Sridharan², Simona Perri³ and Nikos Katzouris⁴

¹Department of Computer Science, University of Verona, Verona, Italy, ²School of Informatics, University of Edinburgh, Edinburgh, United Kingdom, ³Department of Mathematics and Computer Science (DeMaCS), University of Calabria, Arcavacata di Rende, Italy, ⁴National Centre of Scientific Research "Demokritos", Agia Paraskevi, Greece

KEYWORDS

neurosymbolic AI, probabilistic reasoning, reasoning under uncertainty, hybrid AI, robotics

Editorial on the Research Topic

Merging symbolic and data-driven AI for robot autonomy

Robots are increasingly being deployed to assist humans in many applications such as medicine, navigation, and industrial automation. To truly collaborate with humans in complex environments, robots require advanced cognitive capabilities, including the ability to reason with domain-specific commonsense knowledge and the noisy observations obtained in the presence of partial observability and non-deterministic action outcomes. Research in Artificial Intelligence (AI) has resulted in sophisticated symbolic formalisms based on logics to represent commonsense domain knowledge, as well as probabilistic and data-driven frameworks that quantitatively represent uncertainty in the decisions made by robots.

By themselves, symbolic or stochastic AI methods have limitations when applied to robots in complex scenarios. Symbolic AI methods reason with relational descriptions of the attributes of the domain and the robot to guide the robot's behavior. At the same time, they tend to require extensive prior knowledge about the domain and the robot. They also make it computationally expensive to operate at the level of granularity required for precise interaction with the physical world, or to reason about uncertainty quantitatively. Probabilistic and data-driven AI methods, on the other hand, elegantly represent uncertainty quantitatively, and provide mechanisms for reasoning and acting at the level of granularity required for interaction with the physical world. These methods, however, offer limited expressiveness for complex cognitive concepts, and it is not always meaningful to reason about uncertainty quantitatively. With the increasing use of AI and robots in different applications, there has been renewed interest in hybrid and neurosymbolic AI frameworks that combine symbolic and data-driven methods. The 10 contributions in this Research Topic highlight the promise and potential of such frameworks in the context of robotics.

Describing a vision for the future, [Spasokukotskiy](#) states that the next-generation AI systems should not only be endowed with autonomy but also "morality" that secures alignment in large systems, i.e., they should operate safely within the values of human society. Instead of being in full control of AI, humans would then cooperate and communicate

with intelligent systems. Extending this idea, Pal explains *the relevance of transparency, explainability, learning from a few examples, and the trustworthiness of an AI system*, exploring how insights into human reasoning can be a crucial ingredient for achieving reliable operation with embodied AI systems. In addition, Toberg et al. provide a systematic review of robot systems that represent, reason with, learn, and/or use commonsense knowledge in a wide range of application domains. Symbolic AI methods can play a crucial role in the design of such AI/robotics systems, providing the expressivity for elegantly representing human-level concepts and effectively modeling logical reasoning capabilities. These methods can also support more efficient and transparent learning, and the use of human guidance to generate symbolic abstractions. Das et al. describe a framework that extends an inductive logic program learner to demonstrate this capability on multiple benchmark domains, one of which focuses on planning the assembly of mechanical structures, a core task in industrial automation.

In addition to reasoning with prior knowledge that includes cognitive theories, robots that interact with the physical world process a large amount of continuous multi-modal inputs from different information sources, including humans and other agents. In this context, data-driven AI methods, particularly recent advancements in deep learning, have exhibited groundbreaking performance and established themselves as the state of the art for problems in computer vision, natural language processing, and complex decision-making. For example, Mitrokhin et al. describe a hybrid framework for image-based context awareness, training a hash neural network on images to show that *hyperdimensional vectors can be constructed such that vector-symbolic inference arises naturally out of their output*. This enhances the robustness and explainability of the classification process, achieving state-of-the-art accuracy on real-world image datasets such as the popular CIFAR-10.

Acquiring symbol abstractions from raw continuous inputs, i.e., symbol grounding, and decision-making become particularly challenging with the high-dimensional inputs received by robots. Despite the impressive results achieved by deep neural networks and foundation models, their direct use in robots becomes inefficient, hinders transparency, and provides arbitrary responses in novel situations. Hybrid frameworks can address these limitations by leveraging the complementary strengths of symbolic and data-driven AI systems. For example, the framework of Nevens et al. uses symbolic AI to enable an agent to construct *a conceptual system in which meaningful concepts are formed based on human-interpretable feature channels*. They use a dataset of images for manipulating blocks to illustrate how concepts acquired from limited data points can be combined and generalized to unseen instances. Sasaki et al. show that grounding robotic gestures *with quantitative meaning calculated from word-distributed representations constructed from a large corpus of text* enable robots to display behavior that humans perceive to be natural. Riley et al. describe a framework that supports non-monotonic logical reasoning with abstractions of prior commonsense knowledge and information extracted by deep neural networks from relevant image regions; they show substantial

performance improvement compared with state of the art for visual question answering, and vision-based planning and diagnostics. Furthermore, Grosvenor et al. and Ghiasvand et al. document examples of real-world integration of similar ideas in the context of knowledge-enhanced deep visual tracking of satellites, and a comprehensive architecture for space robotic mission planning and control, respectively.

In summary, the contributions to this topic highlight the importance of merging symbolic and data-driven AI methods in the context of robotics (and AI). These papers demonstrate how such hybrid frameworks enable robots to reason with complex cognitive theories and noisy multimodal sensor observations to achieve reliable, efficient, and transparent scene understanding, planning, diagnostics, and human-robot collaboration in complex simulated and physical domains. The papers also draw attention to the fundamental open problems that need to be addressed to leverage the full potential of robots in practical applications. We hope that these papers will foster further collaboration between the related research communities toward achieving societal benefits.

Author contributions

DM: Writing – original draft. MS: Writing – review and editing. SP: Writing – review and editing. NK: Writing – review and editing.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.



Integrating Non-monotonic Logical Reasoning and Inductive Learning With Deep Learning for Explainable Visual Question Answering

Heather Riley¹ and Mohan Sridharan^{2*}

¹ Electrical and Computer Engineering, The University of Auckland, Auckland, New Zealand, ² Intelligent Robotics Lab, School of Computer Science, University of Birmingham, Birmingham, United Kingdom

OPEN ACCESS

Edited by:

Fabrizio Riguzzi,
University of Ferrara, Italy

Reviewed by:

Stefania Costantini,
University of L'Aquila, Italy
Cristian Molinaro,
University of Calabria, Italy

*Correspondence:

Mohan Sridharan
m.sridharan@bham.ac.uk

Specialty section:

This article was submitted to
Computational Intelligence in
Robotics,
a section of the journal
Frontiers in Robotics and AI

Received: 18 September 2019

Accepted: 05 November 2019

Published: 11 December 2019

Citation:

Riley H and Sridharan M (2019)
Integrating Non-monotonic Logical
Reasoning and Inductive Learning
With Deep Learning for Explainable
Visual Question Answering.
Front. Robot. AI 6:125.
doi: 10.3389/frobt.2019.00125

State of the art algorithms for many pattern recognition problems rely on data-driven deep network models. Training these models requires a large labeled dataset and considerable computational resources. Also, it is difficult to understand the working of these learned models, limiting their use in some critical applications. Toward addressing these limitations, our architecture draws inspiration from research in cognitive systems, and integrates the principles of commonsense logical reasoning, inductive learning, and deep learning. As a motivating example of a task that requires explainable reasoning and learning, we consider Visual Question Answering in which, given an image of a scene, the objective is to answer explanatory questions about objects in the scene, their relationships, or the outcome of executing actions on these objects. In this context, our architecture uses deep networks for extracting features from images and for generating answers to queries. Between these deep networks, it embeds components for non-monotonic logical reasoning with incomplete commonsense domain knowledge, and for decision tree induction. It also incrementally learns and reasons with previously unknown constraints governing the domain's states. We evaluated the architecture in the context of datasets of simulated and real-world images, and a simulated robot computing, executing, and providing explanatory descriptions of plans and experiences during plan execution. Experimental results indicate that in comparison with an "end to end" architecture of deep networks, our architecture provides better accuracy on classification problems when the training dataset is small, comparable accuracy with larger datasets, and more accurate answers to explanatory questions. Furthermore, incremental acquisition of previously unknown constraints improves the ability to answer explanatory questions, and extending non-monotonic logical reasoning to support planning and diagnostics improves the reliability and efficiency of computing and executing plans on a simulated robot.

Keywords: nonmonotonic logical reasoning, inductive learning, deep learning, visual question answering, commonsense reasoning, human-robot collaboration

1. INTRODUCTION

Deep neural network architectures and the associated algorithms represent the state of the art for many perception and control problems in which their performance often rivals that of human experts. These architectures and algorithms are increasingly being used for a variety of tasks such as object recognition, gesture recognition, object manipulation, and obstacle avoidance, in domains such as healthcare, surveillance, and navigation. Common limitations of deep networks are that they are computationally expensive to train, and require a large number of labeled training samples to learn an accurate mapping between input(s) and output(s) in complex domains. It is not always possible to satisfy these requirements, especially in dynamic domains where previously unseen situations often change the mapping between inputs and outputs over time. Also, it is challenging to understand or provide an explanatory description of the observed behavior of a learned deep network model. Furthermore, it is difficult to use domain knowledge to improve the computational efficiency of learning these models or the reliability of the decisions made by these models. Consider a self-driving car on a busy road. Any error made by the car, e.g., in recognizing or responding to traffic signs, can result in serious accidents and make humans more reluctant to use such cars. In general, it is likely that humans interacting with a system designed for complex domains, with autonomy in some components, will want to know why and how the system arrived at particular conclusions; this “explainability” will help designers improve the underlying algorithms and their performance. Understanding the operation of these systems will also help human users build trust in the decisions made by these systems. Despite considerable research in recent years, providing explanatory descriptions of decision making and learning continues to be an open problem in AI.

We consider Visual Question Answering (VQA) as a motivating example of a complex task that inherently requires explanatory descriptions of reasoning and learning. Given a scene and a natural language question about an image of the scene, the objective of VQA is to provide an accurate answer to the question. These questions can be about the presence or absence of particular objects in the image, the relationships between these objects, or the potential outcome of executing particular actions on objects in the scene. For instance, a system recognizing and responding to traffic signs on a self-driving car may be posed questions such as “what is the traffic sign in the image?” or “what is the meaning of this traffic sign?” and a system controlling a robot arm constructing stable arrangements of objects on a tabletop may be asked “why is this structure unstable?” or “what would make the structure stable?” We assume that any such questions are provided as (or transcribed into) text, and that answers to questions are also generated as text (that may be converted to speech) using existing software. Deep networks represent the state of the art for VQA, but are characterized by the known limitations described above. We seek to address these limitations by drawing inspiration from research in cognitive systems, which indicates that reliable, efficient, and explainable reasoning and learning

can be achieved in complex problems by jointly reasoning with commonsense domain knowledge and learning from experience. Specifically, the architecture described in this paper tightly couples knowledge representation, reasoning, and learning, and exploits the complementary strengths of deep learning, inductive learning, and non-monotonic logical reasoning with incomplete commonsense domain knowledge. We describe the following characteristics of the architecture:

- For any input image of a scene of interest, Convolutional Neural Networks (CNNs) extract concise visual features characterizing the image.
- Non-monotonic logical reasoning with the extracted features and incomplete commonsense domain knowledge is used to classify the input image, and to provide answers to explanatory questions about the classification and the scene.
- Feature vectors that the non-monotonic logical reasoning is unable to classify are used to train a decision tree classifier that is also used to answer questions about the classification during testing.
- Feature vectors not classified by non-monotonic logical reasoning, along with the output of the decision tree classifier, train a Recurrent Neural Network (RNN) that is used to answer explanatory questions about the scene during testing.
- Feature vectors not classified by non-monotonic logical reasoning are also used to inductively learn, and subsequently reason with, constraints governing domain states; and
- Reasoning with commonsense knowledge is expanded (when needed) to support planning, diagnostics, and the ability to answer related explanatory questions.

This architecture builds on our prior work on combining commonsense inference with deep learning (Riley and Sridharan, 2018a; Mota and Sridharan, 2019) by introducing the ability to learn and reason with constraints governing domain states, and extending explainable inference with commonsense knowledge to also support planning and diagnostics to achieve any given goal.

Although we use VQA as a motivating example, it is not the main focus of our work. State of the art algorithms for VQA focus on generalizing to images from different domains, and are evaluated on benchmark datasets of several thousand images drawn from different domains (Shrestha et al., 2019). Our focus, on the other hand, is on transparent reasoning and learning in any given domain in which a large, labeled dataset is not readily available. Toward this objective, our approach explores the interplay between non-monotonic logical reasoning, incremental inductive learning, and deep learning. We thus neither compare our architecture and algorithms with state of the art algorithms for VQA, nor use large benchmark VQA datasets for evaluation. Instead, we evaluate our architecture’s capabilities in the context of: (i) estimating the stability of configurations of simulated blocks on a tabletop; (ii) recognizing different traffic signs in a benchmark dataset of images; and (iii) a simulated robot delivering messages to the intended recipients at different locations. The characteristics of these tasks and domains match our objective. In both domains, we focus on answering explanatory questions about images of scenes and the

underlying classification problems (e.g., recognizing traffic signs). In addition, we demonstrate how our architecture can be adapted to enable a robot assisting humans to compute and execute plans, and to answer questions about these plans. Experimental results show that in comparison with an architecture based only on deep networks, our architecture provides: (i) better accuracy on classification problems when the training dataset is small, and comparable accuracy on larger datasets; and (ii) significantly more accurate answers to explanatory questions about the scene. We also show that the incremental acquisition of state constraints improves the ability to answer explanatory questions, and to compute minimal and correct plans.

We begin with a discussion of related work in section 2. The architecture and its components are described in section 3, with the experimental results discussed in section 4. Section 5 then describes the conclusions and directions for further research.

2. RELATED WORK

State of the art approaches for VQA are based on deep learning algorithms (Jiang et al., 2015; Masuda et al., 2016; Malinowski et al., 2017; Pandhre and Sodhani, 2017; Zhang et al., 2017; Shrestha et al., 2019). These algorithms use labeled data to train neural network architectures with different arrangements of layers and connections between them, capturing the mapping between the inputs (e.g., images, text descriptions) and the desired outputs (e.g., class labels, text descriptions). Although deep networks have demonstrated the ability to model complex non-linear mappings between inputs and outputs for different pattern recognition tasks, they are computationally expensive and require large, labeled training datasets. They also make it difficult to understand and explain the internal representations, identify changes that will improve performance, or to transfer knowledge acquired in one domain to other related domains. In addition, it is challenging to accurately measure performance or identify dataset bias, e.g., deep networks can answer questions about images using question-answer training samples without even reasoning about the images (Jabri et al., 2016; Teney and van den Hengel, 2016; Zhang et al., 2017). There is on-going research on each of these issues, e.g., to explain the operation of deep networks, reduce training data requirements and bias, reason with domain knowledge, and incrementally learn the domain knowledge. We review some of these approaches below, primarily in the context of VQA.

Researchers have developed methods to understand the internal reasoning of deep networks and other machine learning algorithms. Selvaraju et al. (2017) use the gradient in the last convolutional layer of a CNN to compute the relative contribution (importance weight) of each neuron to the classification decision made. However, the weights of neurons do not provide an intuitive explanation of the CNN's operation or its internal representation. Researchers have also developed general approaches for understanding the predictions of any given machine learning algorithm. For instance, Koh and Liang (2017) use second-order approximations of influence diagrams to trace any model's prediction through a learning algorithm

back to the training data in order to identify training samples most responsible for any given prediction. Ribeiro et al. (2016) developed a framework that analyzes any learned classifier model by constructing an interpretable simpler model that captures the essence of the learned model. This framework formulates the task of explaining the learned model, based on representative instances and explanations, as a submodular optimization problem. In the context of VQA, Norcliffe-Brown et al. (2018) provide interpretability by introducing prior knowledge of scene structure as a graph that is learned from observations based on the question under consideration. Object bounding boxes are graph nodes while edges are learned using an attention model conditioned on the question. Mascharka et al. (2018) augment a deep network architecture with an image-space attention mechanism based on a set of composable visual reasoning primitives that help examine the intermediate outputs of each module. Li et al. (2018) introduce a captioning model to generate an image's description, reason with the caption and the question to construct an answer, and use the caption to explain the answer. However, these algorithms do not support the use of commonsense reasoning to (i) provide meaningful explanatory descriptions of learning and reasoning; (ii) guide learning to be more efficient; or (iii) provide reliable decisions when large training datasets are not available.

The training data requirements of a deep network can be reduced by directing attention to data relevant to the tasks at hand. In the context of VQA, Yang et al. (2016) use a Long Short-Term Memory (LSTM) network to map the question to an encoded vector, extract a feature map from the input image using a CNN, and use a neural network to compute weights for feature vectors based on their relevance to the question. A stacked attention network is trained to map the weighted feature vectors and question vector to the answer, prioritizing feature vectors with greater weights. Schwartz et al. (2017) use learned higher-order correlations between various data modalities to direct attention to elements in the data modalities that are relevant to the task at hand. Lu et al. (2016) use information from the question to identify relevant image regions and uses information from the image to identify relevant words in the question. A co-attentional model jointly and hierarchically reasons about the image and the question at three levels, embedding words in a vector space, using one-dimensional CNNs to model information at the phrase level, and using RNNs to encode the entire question. A generalization of this work, a Bilinear Attention Network, considers interactions between all region proposals in the image with all words in the (textual) question (Kim et al., 2018). A Deep Attention Neural Tensor Network for VQA, on the other hand, uses tensor-based representations to discover joint correlations between images, questions, and answers (Bai et al., 2018). The attention module is based on a discriminative reasoning process, and regression with KL-divergence losses improves scalability of training and convergence. Recent work by Anderson et al. (2018) combines top-down and bottom-up attention mechanisms, with the top-down mechanism providing an attention distribution over object proposals provided by the bottom-up mechanism.

In addition to reducing the training data requirements, researchers have focused on reducing the number of annotated

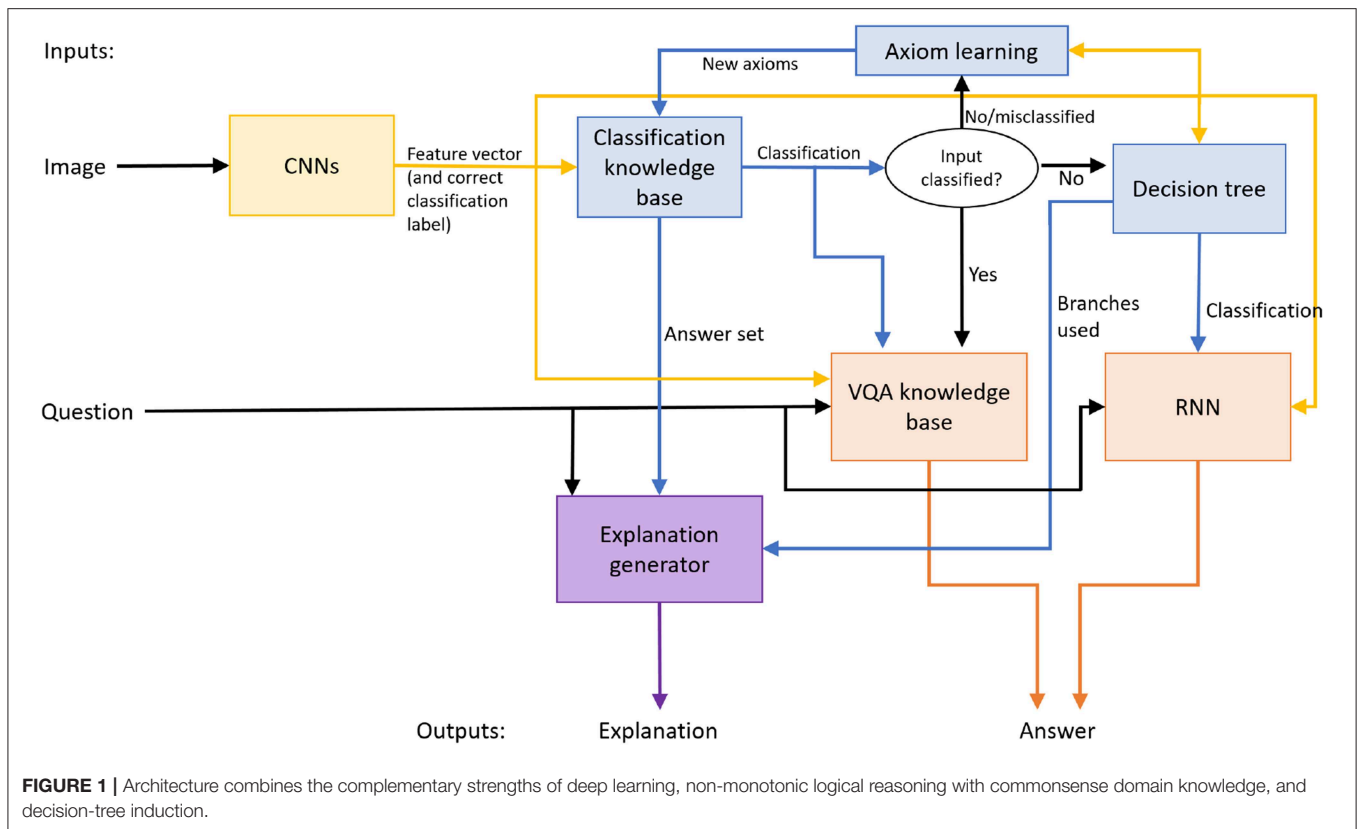
samples needed for training, and on minimizing the bias in deep network models. In the context of VQA, Lin et al. (2014) iteratively revise a model trained on an initial training set by expanding the training set with image-question pairs involving concepts it is uncertain about, with an “oracle” (human annotator) providing the answers. This approach reduces annotation time, but the database includes just as many images and questions as before. Goyal et al. (2017) provide a balanced dataset with each question associated with a pair of images that require different answers, and provide a counterexample based explanation for each image-question pair. Agrawal et al. (2018), on the other hand, separate the recognition of visual concepts in an image from the identification of an answer to any given question, and include inductive biases to prevent the learned model from relying predominantly on priors in the training data.

In computer vision, robotics and other applications, learning from data can often be made more efficient by reasoning with prior knowledge about the domain. In the context of VQA, Wang et al. (2017) reason with knowledge about scene objects to answer common questions about these objects, significantly expanding the range of natural language questions that can be answered without making the training data requirements impractical. However, this approach does not reduce the amount of data required to train the deep network. Furbach et al. (2010) directly use a knowledge base to answer questions and do not consider the corresponding images as inputs. Wagner et al. (2018), on the other hand, use physics engines and prior knowledge of domain objects to realistically simulate and explore different situations. These simulations guide the training of deep network models that anticipate action outcomes and answer questions about all situations. Based on the observation that VQA often requires reasoning over multiple steps, Wu et al. (2018) construct a chain of reasoning for multi-step and dynamic reasoning with relations and objects. This approach iteratively forms new relations between objects using relational reasoning operations, and forms new compound objects using object refining operations, to improve VQA performance. Given the different components of a VQA system, Teney and van den Hengel (2018) present a meta learning approach to separate question answering from the information required for the task, reasoning at test time over example questions and answers to answer any given question. Two meta learning methods adapt a VQA model without the need for retraining, and demonstrate the ability to provide novel answers and support vision and language learning. Rajani and Mooney (2018) developed an ensemble learning approach, Stacking With Auxiliary Features, which combines the results of multiple models using features of the problem as context. The approach considers four categories of auxiliary features, three of which are inferred from image-question pairs while the fourth uses model-specific explanations.

Research in cognitive systems indicates that reliable, efficient, and explainable reasoning and learning can be achieved by reasoning with domain knowledge and learning from experience. Early work by Gil (1994) enabled an agent to reason with first-order logic representations and incrementally refined action operators. In such methods, it is difficult to perform non-monotonic reasoning, or to merge new, unreliable information

with existing beliefs. Non-monotonic logic formalisms have been developed to address these limitations, e.g., Answer Set Prolog (ASP) has been used in cognitive robotics (Erdem and Patoglu, 2012) and other applications (Erdem et al., 2016). ASP has been combined with inductive learning to monotonically learn causal laws (Otero, 2003), and methods have been developed to learn and revise domain knowledge represented as ASP programs (Balduccini, 2007; Law et al., 2018). Cognitive architectures have also been developed to extract information from perceptual inputs to revise domain knowledge represented in first-order logic (Laird, 2012), and to combine logic and probabilistic representations to support reasoning and learning in robotics (Zhang et al., 2015; Sarathy and Scheutz, 2018). However, approaches based on classical first-order logic are not expressive enough, e.g., modeling uncertainty by attaching probabilities to logic statements is not always meaningful. Logic programming methods, on the other hand, do not support one or more of the desired capabilities such as efficient and incremental learning of knowledge, reasoning efficiently with probabilistic components, or generalization as described in this paper. These challenges can be addressed using interactive task learning, a general knowledge acquisition framework that uses labeled examples or reinforcement signals obtained from observations, demonstrations, or human instructions (Laird et al., 2017; Chai et al., 2018). Sridharan and Meadows (2018) developed such a framework to combine non-monotonic logical reasoning with relational reinforcement learning and inductive learning to learn action models to be used for reasoning or learning in dynamic domains. In the context of VQA, there has been interesting work on reasoning with learned symbolic structure. For instance, Yi et al. (2018) present a neural-symbolic VQA system that uses deep networks to infer structural object-based scene representation from images, and to generate a hierarchical (symbolic) program of functional modules from the question. An executor then runs the program on the representation to answer the question. Such approaches still do not (i) integrate reasoning and learning such that they inform and guide each other; or (ii) use the rich domain-specific commonsense knowledge that is available in any application domain.

In summary, deep networks represent the state of the art for VQA and many other pattern recognition tasks. Recent surveys on VQA methods indicate that despite considerable research, it is still difficult to use these networks to support efficient learning, intuitive explanations, or generalization to simulated and real-world images (Pandhre and Sodhani, 2017; Shrestha et al., 2019). Our architecture draws on principles of cognitive systems to address these limitations. It tightly couples deep networks with components for non-monotonic logical reasoning with commonsense domain knowledge, and for learning incrementally from samples over which the learned model makes errors. This work builds on our proof of concept architecture that integrated deep learning with commonsense inference for VQA (Riley and Sridharan, 2018a). It also builds on work in our research group on using commonsense inference and learned state constraints to guide deep networks that estimate object stability and occlusion in images (Mota and Sridharan, 2019). In comparison with our prior work, we introduce a



new component for incrementally learning constraints governing domain states, expand reasoning with commonsense knowledge to support planning and diagnostics, explore the interplay between the architecture's components, and discuss detailed experimental results.

3. ARCHITECTURE

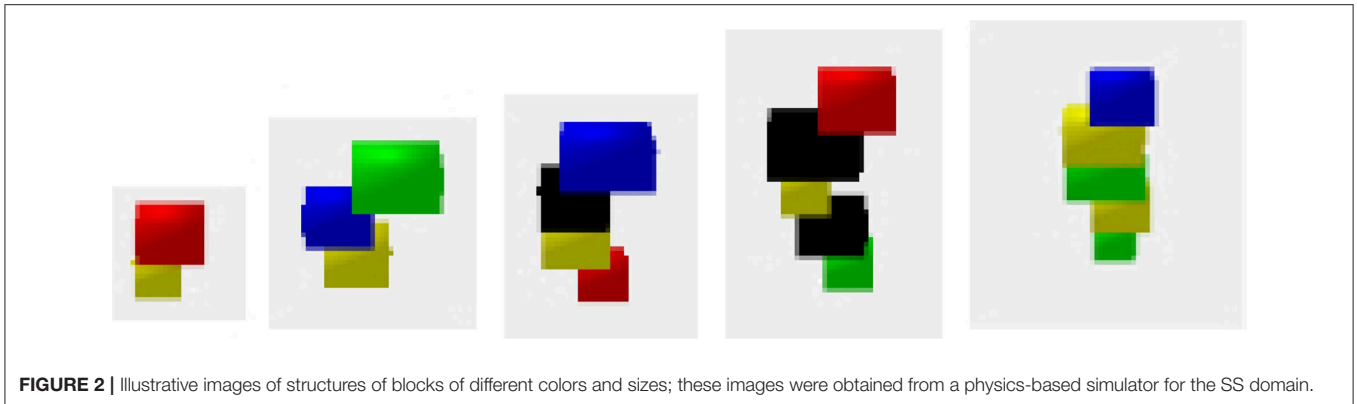
Figure 1 is an overview of our architecture that provides answers to explanatory questions about images of scenes and an underlying classification problem. The architecture seeks to improve accuracy and reduce training effort, i.e., reduce training time and the number of training samples, by embedding non-monotonic logical reasoning and inductive learning in a deep network architecture. We will later demonstrate how the architecture can be adapted to address planning problems on a simulated robot—see section 3.5. The architecture may be viewed as having four key components that are tightly coupled with each other.

1. A component comprising CNN-based feature extractors, which are trained and used to map any given image of a scene under consideration to a vector of image features.
2. A component that uses one of two methods to classify the feature vector. The first method uses non-monotonic reasoning with incomplete domain knowledge and the features to assign a class label and explain this decision. If the first method cannot classify the image, the second method

trains and uses a decision tree to map the feature vector to a class label and explain the classification.

3. A component that answers explanatory questions. If non-monotonic logical reasoning is used for classification, it is also used to provide answers to these questions. If a decision tree is instead used for classification, an RNN is trained to map the decision tree's output, the image features, and the question, to the corresponding answer.
4. A component that uses the learned decision tree and the existing knowledge base to incrementally construct and validate constraints on the state of the domain. These constraints revise the existing knowledge that is used for subsequent reasoning.

This architecture exploits the complementary strengths of deep learning, non-monotonic logical reasoning, and incremental inductive learning with decision trees. Reasoning with commonsense knowledge guides learning, e.g., the RNN is trained on (and processes) input data that cannot be processed using existing knowledge. The CNNs and RNN can be replaced by other methods for extracting image features and answering explanatory questions (respectively). Also, although the CNNs and RNN are trained in an initial phase in this paper, these models can be revised over time if needed. We hypothesize that embedding non-monotonic logical reasoning with commonsense knowledge and the incremental updates of the decision tree, between the CNNs and the RNN, makes the decisions more transparent, and makes learning more time and sample efficient. Furthermore, the overall architecture and



methodology can be adapted to different domains. In this paper, we will use the following two domains to illustrate and evaluate the architecture's components and the methodology.

1. **Structure Stability (SS):** this domain has different structures, i.e., different arrangements of simulated blocks of different colors and sizes, on a tabletop—see **Figure 2** for some examples. We generated 2,500 such images using a physics-based simulator. The relevant features of the domain include the number of blocks, whether the structure is on a lean, whether the structure has a narrow base, and whether any block is placed such that it is not well balanced on top of the block below. The objective in this domain is to classify structures as being stable or unstable, and to answer explanatory questions such as “why is this structure unstable?” and “what should be done to make this structure stable?”
2. **Traffic Sign (TS):** this domain focuses on recognizing traffic signs from images—see **Figure 3** for some examples. We used the BelgiumTS benchmark dataset (Timofte et al., 2013) with ≈ 7000 real-world images (total) of 62 different traffic signs. This domain's features include the primary symbol of the traffic sign, the secondary symbol, the shape of the sign, the main color in the middle, the border color, the sign's background image, and the presence or absence of a cross (e.g., some signs have a red or black cross across them to indicate the end of a zone, with the absence of the cross indicating the zone's beginning). The objective is to classify the traffic signs and answer explanatory questions such as “what is the sign's message?” and “how should the driver respond to this sign?”

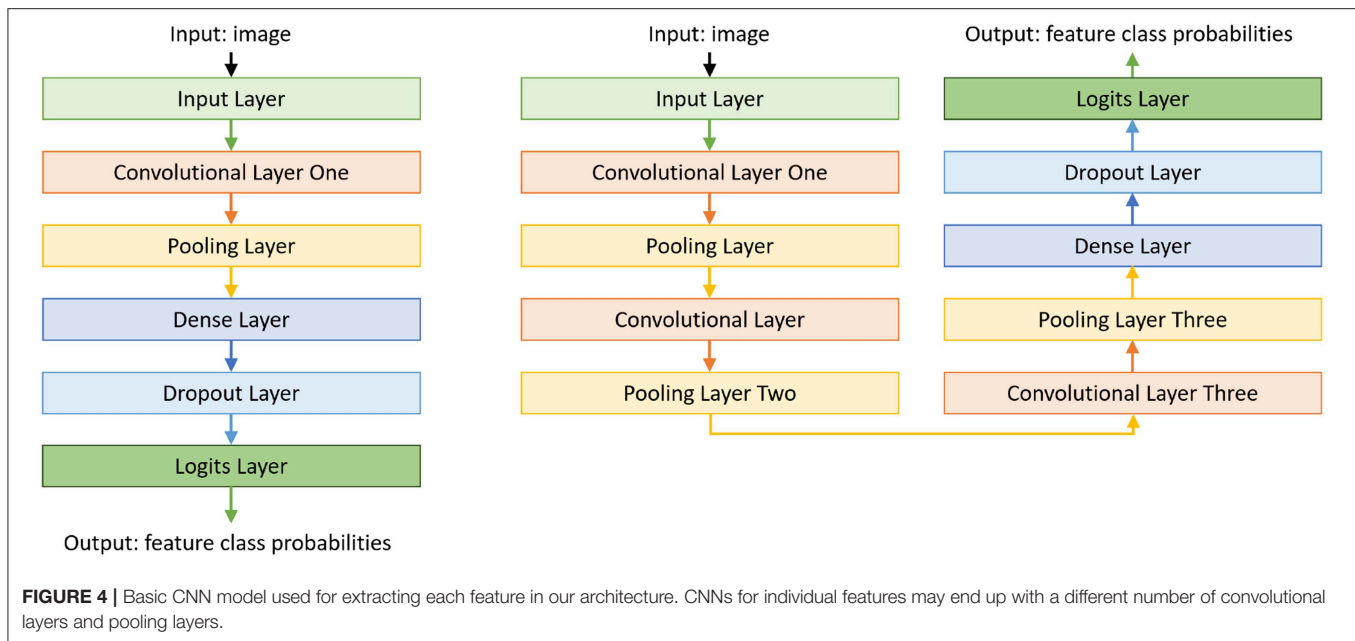
In addition to these two domains, section 3.5 will introduce the **Robot Assistant (RA)** domain, a simulated domain to demonstrate the use of our architecture for computing and executing plans to achieve assigned goals. In the RA domain, a simulated robot reasons with existing knowledge to deliver messages to target people in target locations, and to answer explanatory questions about the plans and observed scenes.

The focus of our work is on understanding and using the interplay between deep learning, commonsense reasoning, and incremental learning, in the context of *reliable and efficient scene understanding in any given dynamic domain*. The benchmark VQA datasets and the algorithms, on the other hand, focus on generalizing across images from different scenarios in different domains, making it difficult to support the reasoning and learning capabilities of our architecture. We thus do not use these datasets or algorithms in our evaluation.

3.1. Feature Extraction Using CNNs

The first component of the architecture trains CNNs to map input images to concise features representing the objects of interest in the images. For the SS domain and TS domain, semi-automated annotation was used to label the relevant features in images for training and testing. The selection of these features for each domain was based on domain expertise. In the SS domain, the features of interest are:

- Number of blocks in structure (number $\in [1, 5]$);
- Whether the structure is on a lean (true, false);
- Width of the base block (wide, narrow); and



- Whether any block is displaced, i.e., not well balanced on top of the block below (true, false).

In the TS domain, the features of interest are:

- Primary symbol in the middle of the traffic sign; 39 primary symbols such as *bumpy_road*, *slippery_road*, *stop*, *left_turn*, and *speed_limit*;
- Secondary symbol in the traffic sign; 10 secondary symbols such as *disabled*, *car* and *fence*;
- Shape of the sign; *circle*, *triangle*, *square*, *hexagon*, *rectangle*, *wide rectangle*, *diamond*, or *inverted triangle*;
- Main color in the middle of the sign; *red*, *white*, or *blue*;
- Border color at the edge of the sign; *red*, *white*, or *blue*;
- Background image, e.g., some symbols are placed over a square or a triangle; and
- Presence of a red or black cross across a sign to indicate a zone's end or invalidity; the sign without the cross indicates the zone's beginning or validity, e.g., a parking sign with a cross implies no parking.

To reduce the training data requirements and simplify the training of CNNs, we (i) train a separate CNN for each feature to be extracted from an image; and (ii) start with a basic model for each CNN and incrementally make it more complex as needed. The number of CNNs is thus equal to the number of features to be extracted from each image for any given domain, and the CNN trained for each feature may be different even within a particular domain. The basic CNN model we begin with has an input layer, a convolutional layer, a pooling layer, a dense layer, a dropout layer, and a logit layer, as seen on the left of **Figure 4**. Additional convolutional and pooling layers are added until the feature extraction accuracy converges or exceeds a threshold (e.g., $\geq 90\%$). Our architecture also includes the option of fine-tuning previously trained CNN models instead of starting from

scratch. The right side of **Figure 4** shows a CNN model learned in our example domains, which has three convolutional layers and pooling layers. We trained and validated these CNNs in an initial phase, and used them for evaluation. Our code for constructing these CNNs for features (in our example domains) is in our repository (Riley and Sridharan, 2018b).

3.2. Classification Using Non-monotonic Logical Reasoning or Decision Trees

The feature vector extracted from an image is used for decision making. In the SS domain and TS domain, decisions take the form of assigning a class label to each feature vector¹. The second component of our architecture performs this task using one of two methods: (i) non-monotonic logical inference using ASP; or (ii) a classifier based on a learned decision tree. We describe these two methods below.

3.2.1. ASP-Based Inference With Commonsense Knowledge

The first step in reasoning with incomplete commonsense domain knowledge is the representation of this knowledge. In our architecture, an *action language* is used to describe the dynamics of any domain under consideration. Action languages are formal models of parts of natural language used for describing transition diagrams of dynamic systems. Our architecture uses action language $\mathcal{AL}_d$ (Gelfond and Inlezan, 2013), with a *sorted signature* Σ that can be viewed as the *vocabulary* used to describe the domain's transition diagram. The signature Σ comprises *basic sorts*, which are similar to *types* in a programming language, *statics*, i.e., domain attributes whose values do not change over time, *fluents*, i.e., domain attributes whose values can change over

¹In the RA domain discussed in section 3.5, decision making also includes planning and diagnostics.

time, and *actions*. The domain's fluents can be *basic*, i.e., those that obey the laws of inertia and are changed directly by actions, or *defined*, i.e., those that do not obey the laws of inertia and are defined by other attributes. A domain attribute or its negation is a *literal*; if all its variables are ground, it is a *ground literal*. $\mathcal{AL}_d$ allows three types of statements: *causal law*, *state constraint* and *executability condition*.

a **causes** l_b **if** $p_0, \dots, p_m$ (Causal law)

l **if** $p_0, \dots, p_m$ (State constraint)

impossible $a_0, \dots, a_k$ **if** $p_0, \dots, p_m$ (Executability condition)

where a is an action, l is a literal, l_b is a basic literal, and $p_0, \dots, p_m$ are domain literals.

The domain representation (i.e., the knowledge base) comprises a *system description* $\mathcal{D}$, which is a set of statements of $\mathcal{AL}_d$, and a *history* $\mathcal{H}$. $\mathcal{D}$ comprises a sorted signature Σ and axioms describing the domain dynamics. For instance, in the SS domain, Σ includes basic sorts such as *structure*, *color*, *size*, and *attribute*; the basic sorts of the TS domain include *main_color*, *other_color*, *main_symbol*, *other_symbol*, *shape*, *cross* etc. The sort *step* is also in Σ to support temporal reasoning over time steps. The statics and fluents in the SS domain include:

$num_blocks(structure, num)$, $block_color(block, color)$, (1)
 $block_size(block, size)$, $block_displaced(structure)$, $stable(structure)$

which correspond to the image features extracted in the domain, and are described in terms of their arguments' sorts. In a similar manner, statics and fluents of the TS domain include:

$primary_symbol(sign, main_symbol)$,
 $primary_color(sign, main_color)$
 $secondary_symbol(sign, other_symbol)$,
 $secondary_color(sign, other_color)$
 $sign_shape(shape)$, $background_image(image)$ (2)

In both domains, signature Σ includes a predicate *holds*(*fluent*, *step*), which implies that a particular fluent holds true at a particular time step. As stated above, Σ for a dynamic domain typically includes actions that cause state transitions, but this capability is not needed to answer explanatory questions about specific scenes and the underlying classification problem in our (SS, TS) domains. For ease of explanation, we thus temporarily disregard the modeling of actions, and their preconditions and effects. We will revisit actions in section 3.5 when we consider planning tasks in the RA domain.

Given a signature Σ for a domain, a *state* of the domain is a collection of ground literals, i.e., statics, fluents, actions and relations with values assigned to their arguments—for more details, please see Gelfond and Kahl (2014) and Sridharan et al. (2019). The axioms of $\mathcal{D}$ are defined in terms of the signature and govern domain dynamics; this typically includes a distributed representation of the constraints related to domain actions, i.e., causal laws and executability conditions that define the

preconditions and effects of actions, and constraints related to states, i.e., state constraints. In the SS domain and TS domain, axioms govern the belief about domain states; we will discuss axioms related to actions in section 3.5 when we discuss the RA domain. Specifically, the axioms of the SS domain include state constraints such as:

$\neg stable(S)$ **if** $block_displaced(S)$ (3a)

$stable(S)$ **if** $num_blocks(S, 2)$, $\neg structure_type(S, lean)$ (3b)

where Statement 3(a) says that any structure with a block that is displaced significantly is unstable, and Statement 3(b) says that any pair of blocks without a significant lean is stable.

Axioms of the TS domain include statements such as:

$sign_type(TS, no_parking)$ **if** $primary_color(TS, blue)$,
 $primary_symbol(TS, blank)$,
 $cross(TS)$, $shape(TS, circle)$ (4a)

$sign_type(TS, stop)$ **if** $primary_color(TS, red)$,
 $primary_symbol(TS, stoptext)$,
 $shape(TS, octagon)$ (4b)

where Statement 4(a) implies that a blue, blank, circular traffic sign with a cross across it is a no parking sign. Statement 4(b) implies that a red, octagon-shaped traffic sign with the text “stop” is a stop sign.

The history $\mathcal{H}$ of a dynamic domain is usually a record of fluents observed to be true or false at a particular time step, i.e., *obs*(*fluent*, *boolean*, *step*), and the successful execution of an action at a particular time step, i.e., *hpd*(*action*, *step*); for more details, see Gelfond and Kahl (2014). The domain knowledge in many domains often includes default statements that are true in all but a few exceptional circumstances. For example, we may know in the SS domain that “structures with two blocks of the same size are usually stable.” To encode such knowledge, we use our recent work that expanded the notion of history to represent and reason with defaults describing the values of fluents in the initial state (Sridharan et al., 2019).

Key tasks of an agent equipped with a system description $\mathcal{D}$ and history $\mathcal{H}$ include reasoning with this knowledge for inference, planning and diagnostics. In our architecture, these tasks are accomplished by translating the domain representation to a program $\Pi(\mathcal{D}, \mathcal{H})$ in CR-Prolog, a variant of ASP that incorporates consistency restoring (CR) rules (Balduccini and Gelfond, 2003). In this paper, we use the terms “ASP” and “CR-Prolog” interchangeably. ASP is a declarative programming paradigm designed to represent and reason with incomplete commonsense domain knowledge. It is based on stable model semantics, and supports *default negation* and *epistemic disjunction*. For instance, unlike “ $\neg a$ ”, which implies that *a* is believed to be false, “*not a*” only implies *a* is not believed to be true. Also, unlike “ $p \vee \neg p$ ” in propositional logic, “*p or not p*” is not tautological. Each literal can thus be true, false or unknown, and the agent reasoning with domain knowledge does not believe anything that it is not forced to believe. ASP can represent recursive definitions, defaults, causal

relations, special forms of self-reference, and language constructs that occur frequently in non-mathematical domains, and are difficult to express in classical logic formalisms (Baral, 2003; Gelfond and Kahl, 2014). Unlike classical first-order logic, ASP supports non-monotonic logical reasoning, i.e., it can revise previously held conclusions or equivalently reduce the set of inferred consequences, based on new evidence—this ability helps the agent recover from any errors made by reasoning with incomplete knowledge. ASP and other paradigms that reason with domain knowledge are often criticized for requiring considerable (if not complete) prior knowledge and manual supervision, and for being unwieldy in large, complex domains. However, modern ASP solvers support efficient reasoning in large knowledge bases with incomplete knowledge, and are used by an international research community for cognitive robotics (Erdem and Patoglu, 2012; Zhang et al., 2015) and other applications (Erdem et al., 2016). For instance, recent work has demonstrated that ASP-based non-monotonic logical reasoning can be combined with: (i) probabilistic reasoning for reliable and efficient planning and diagnostics (Sridharan et al., 2019); and (ii) relational reinforcement learning and active learning methods for interactively learning or revising commonsense domain knowledge based on input from sensors and humans (Sridharan and Meadows, 2018).

In our architecture, the automatic translation from statements in $\mathcal{AL}_d$ to the program Π is based on a custom-designed script². The resultant program Π includes the signature and axioms of $\mathcal{D}$, inertia axioms, reality checks, closed world assumptions for defined fluents and actions, and observations, actions, and defaults from $\mathcal{H}$. For instance, Statements 3(a-b) are translated to:

$$\neg \text{stable}(S) \leftarrow \text{block_displaced}(S) \quad (5a)$$

$$\text{stable}(S) \leftarrow \text{num_blocks}(S, 2), \neg \text{structure_type}(S, \text{lean}) \quad (5b)$$

In addition, features extracted from an input image (to be processed) are encoded as the initial state of the domain in Π . Each *answer set* of $\Pi(\mathcal{D}, \mathcal{H})$ then represents the set of beliefs of an agent associated with this program. Algorithms for computing entailment, and for planning and diagnostics, reduce these tasks to computing answer sets of CR-Prolog programs. We compute answer sets of CR-Prolog programs using the SPARC system (Balai et al., 2013). The CR-Prolog programs for our example domains are in our open-source software repository (Riley and Sridharan, 2018b). For the classification task in our example domains, the relevant literals in the answer set provide the class label and an explanatory description of the assigned label (see section 3.3); we will consider the planning task in section 3.5. The accuracy of the inferences drawn from the encoded knowledge depends on the accuracy and extent of the knowledge encoded, but encoding comprehensive domain knowledge is difficult. The decision of what and how much knowledge to encode is made by the designer.

² An independent group of researchers has developed a general-purposed software to automatically translate any description in $\mathcal{AL}_d$ to the corresponding CR-Prolog program; we expect this software to be made publicly available soon.

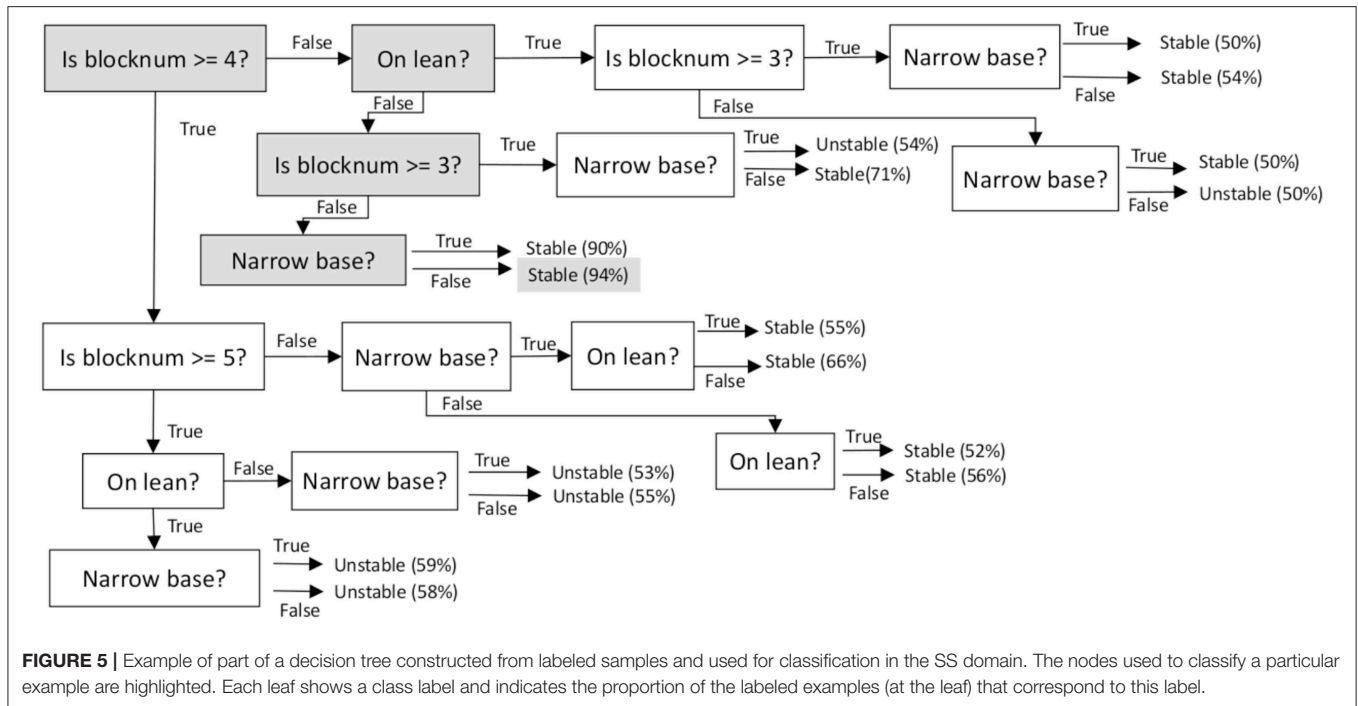
3.2.2. Decision Tree Classifier

If ASP-based inference cannot classify the feature vector extracted from an image, the feature vector is mapped to a class label using a decision tree classifier learned from labeled training examples. In a decision tree classifier, each node is associated with a question about the value of a particular feature, with the child nodes representing the different answers to the question, i.e., the possible values of the feature. Each node is also associated with samples that satisfy the corresponding values of the features along the path from the root node to this node. We use a standard implementation of a decision tree classifier (Duda et al., 2000). This implementation uses the Gini measure to compute information gain (equivalently, the reduction in entropy) that would be achieved by splitting an existing node based on each feature that has not already been used to create a split in the tree. Among the features that provide a significant information gain, the feature that provides the maximum information gain is selected to split the node. If none of the features would result in any significant information gain, this node becomes a leaf node with a class label that matches a majority of the samples at the node.

The decision tree's search space is quite specific since it only considers samples that could not be classified by ASP-based reasoning. The decision tree does not need to generalize as much as it would have to if it had to process every training (or test) sample in the dataset. Also, although overfitting is much less likely, we still use pruning to minimize the effects of overfitting. **Figure 5** shows part of a learned decision tree classifier; specific nodes used to classify a particular example are highlighted to indicate that 94% of the observed examples of structures that have fewer than three blocks, do not have a significant lean, and do not have a narrow base, correspond to stable structures. These “active” nodes along any path in the decision tree that is used to classify an example can be used to explain the classification outcome in terms of the values of particular features that were used to arrive at the class label assigned to a specific image under consideration.

3.3. Answering Explanatory Questions

The third component of the architecture provides two methods for answering explanatory questions. The available inputs are the (i) question; (ii) vector of features extracted from the image under consideration; and (iii) classification output. The human designer also provides pre-determined templates for questions and their answers. In our case, we use a controlled vocabulary, templates based on language models and parts of speech for sentences, and existing software for natural language processing. Any given question is transcribed using the controlled vocabulary, parsed (e.g., to obtain parts of speech), and matched with the templates to obtain a relational representation. Recall that questions in the SS domain are of the form: “is this structure stable/unstable?” and “what is making this structure stable/unstable?” These questions can be translated into relational statements such as *stable*(S) or $\neg \text{stable}(S)$ and used as a question, or as the desired consequence, during inference or in a search process. In a similar manner, questions in the TS domain such as: “what sign is this?” and “what



is the sign's message?" can be translated into $sign_type(S, sign)$ and used for subsequent processing.

The first method for answering explanatory questions is based on the understanding that if the feature vector extracted from the image is processed successfully using ASP-based reasoning, it is also possible to reason with the existing knowledge to answer explanatory questions about the scene. To support such question answering, we need to revise the signature Σ in the system description $\mathcal{D}$ of the domain. For instance, we add sorts such as $query_type$, $answer_type$, and $query$ to encode different types of queries and answers. We also introduce suitable relations to represent questions, answers to these questions, and more abstract attributes, e.g., of structures of blocks, traffic signs etc.

In addition to the signature, we also augment the axioms in $\mathcal{D}$ to support reasoning with more abstract attributes, and to help construct answers to questions. For instance, we can include an axiom such as:

$$\begin{aligned} many_blocks(S) \leftarrow & \text{unstable}(S), \neg base(S, narrow), \\ & \neg struc_type(S, lean), \neg block_displaced(S) \end{aligned} \quad (6)$$

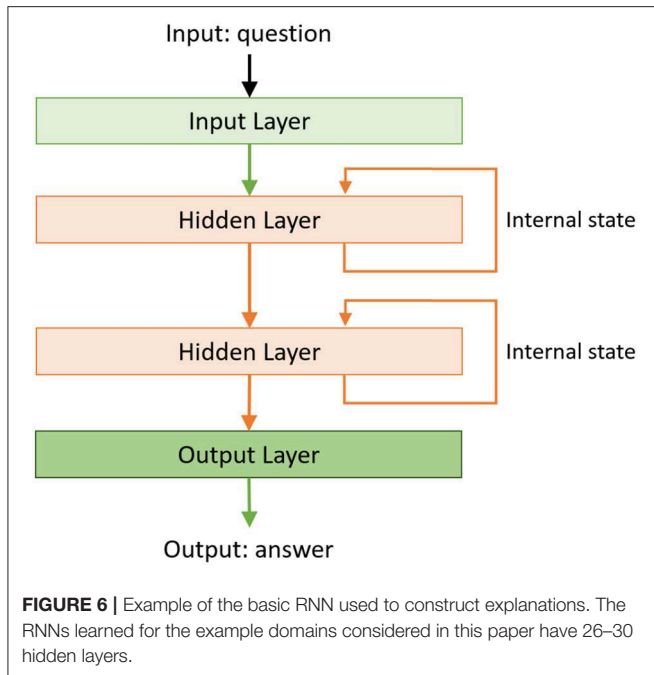
which implies that if a structure (of blocks) is not on a narrow base, does not have a significant lean, and does not have blocks significantly displaced, any instability in the structure implies (and is potentially because) there are too many blocks in the structure. Once the ASP program $\Pi(\mathcal{D}, \mathcal{H})$ has been revised as described above, we can compute answer set(s) of this program to obtain the beliefs of the agent associated with this program. For any given question, the answer set(s) are parsed based on the known controlled vocabulary and templates (for questions and answers) to extract relevant literals—these literals are included in

the corresponding templates to construct answers to explanatory questions. These answers can also be converted to speech using existing software.

The second method for answering explanatory questions is invoked if the decision tree is used to process (i.e., classify in the context of the SS domain and TS domain) the vector of image features. The inability to classify the feature vector through ASP-based reasoning is taken to imply that the encoded domain knowledge is insufficient to answer explanatory questions about the scene. In this case, an LSTM network-based RNN is trained and used to answer the explanatory questions. The inputs are the feature vector, classification output, and a vector representing the transcribed and parsed query. The output (provided during training) is in the form of answers in the predetermined templates. Similar to the approach used in section 3.2, the RNN is built incrementally during training. We begin with one or two hidden layer(s), as shown in **Figure 6**, and add layers as long as it results in a significant increase in the accuracy. We also include the option of adding a stack of LSTMs if adding individual layers does not improve accuracy significantly. In our example domains, the RNN constructed to answer explanatory questions had as many as 26–30 hidden layers and used a softmax function to provide one of about 50 potential answer types. An example of the code used to train the RNN is available in our repository (Riley and Sridharan, 2018b).

3.4. Learning State Constraints

The components of the architecture described so far support reasoning with commonsense knowledge, learned decision trees, and deep networks, to answer explanatory questions about the scene and an underlying classification problem. In many practical domains, the available knowledge is incomplete, the number of



labeled examples is small, or the encoded knowledge changes over time. The decisions made by the architecture can thus be incorrect or sub-optimal, e.g., a traffic sign can be misclassified or an ambiguous answer may be provided to an explanatory question. The fourth component of our architecture seeks to address this problem by supporting incremental learning of domain knowledge. Our approach is inspired by the inductive learning methods mentioned in section 2, e.g., Sridharan and Meadows (2018) use relational reinforcement learning and decision tree induction to learn domain axioms. The work described in this paper uses decision tree induction to learn constraints governing domain states. The methodology used in this component, in the context of VQA, is as follows:

1. Identify training examples that are not classified, or are classified incorrectly, using the existing knowledge. Recall that this step is accomplished by the component described in section 3.2, which processes each training example using the existing knowledge encoded in the CR-Prolog program, in an attempt to assign a class label to the example.
2. Train a decision tree using the examples identified in Step-1 above. Recall that this step is also accomplished by the component described in section 3.2.
3. Identify paths in the decision tree (from root to leaf) such that (i) there are a sufficient number of examples at the leaf, e.g., 10% of the training examples; and (ii) all the examples at the leaf have the same class label. Since the nodes correspond to checks on the values of domain features, the paths will correspond to combinations of partial state descriptions and class labels that have good support among the labeled training examples. Each such path is translated into a candidate axiom. For instance, the following are two axioms identified by this approach in the SS domain:

$$\neg \text{stable}(S) \leftarrow \text{num_blocks}(S, 3), \text{base}(S, \text{wide}), \text{struc_type}(S, \text{lean}) \quad (7a)$$

$$\neg \text{stable}(S) \leftarrow \text{num_blocks}(S, 3), \text{base}(S, \text{narrow}), \text{struc_type}(S, \text{lean}) \quad (7b)$$

4. Generalize candidate axioms if possible. For instance, if one candidate axiom is a over-specification of another existing axiom, the over-specified version is removed. In the context of the axioms in Statement 7(a-b), the second literal represents redundant information, i.e., if a structure with three blocks has a significant lean, it is unstable irrespective of whether the base of the structure is narrow or wide. Generalizing over these two axioms results in the following candidate axiom:

$$\neg \text{stable}(S) \leftarrow \text{num_blocks}(S, 3), \text{struc_type}(S, \text{lean}) \quad (8)$$

which only includes the literals that encode the essential information.

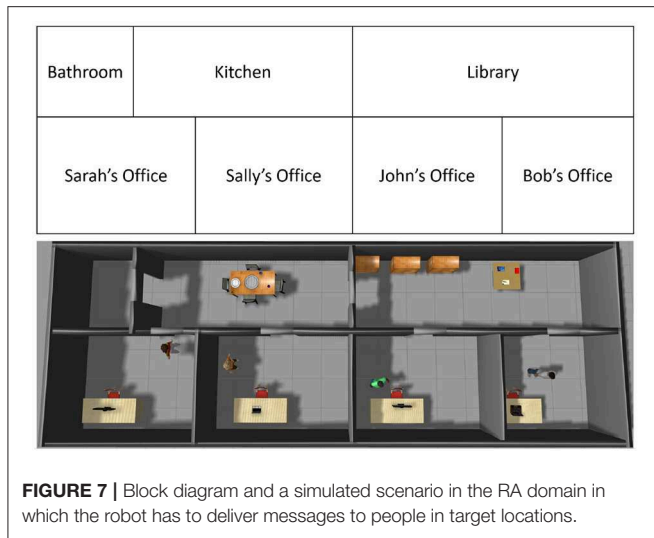
5. Validate candidate axioms one at a time. To do so, the candidate axiom is added to the CR-Prolog program encoding the domain knowledge. A sufficient number of training examples (e.g., 10% of the dataset, as before) relevant to this axiom, i.e., the domain features encoded by the examples should satisfy the body of the axiom, are drawn randomly from the training dataset. If processing these selected examples with the updated CR-Prolog program results in misclassifications, the candidate axiom is removed from further consideration.
6. Apply sanity checks to the validated axioms. The validated axioms and existing axioms are checked to remove over-specifications and retain the most generic version of any axiom. Axioms that pass these sanity checks are added to the CR-Prolog program and used for subsequent reasoning.

Section 4.3 examines the effect of such learned constraints on classification and VQA performance.

3.5. Planning With Domain Knowledge

The description of the architecture's components has so far focused on classification and VQA, and reasoning has been limited to inference with knowledge. However, the architecture is also applicable to planning (and diagnostics) problems. Consider the **RA domain** in which a simulated robot has to navigate and deliver messages to particular people in particular places, and to answer explanatory questions, i.e., the domain includes aspects of planning and VQA. **Figure 7** depicts this domain and a simulated scenario in it; semantic labels of the offices and rooms are shown in the upper half.

A robot planning and executing actions in the real world has to account for the uncertainty in sensing and actuation. In other work, we addressed this issue by coupling ASP-based coarse-resolution planning with probabilistic fine-resolution planning and execution (Sridharan et al., 2019). In this paper, we temporarily abstract away such probabilistic models of uncertainty to thoroughly explore the interplay between reasoning and learning, including the effect of added noise in sensing and actuation (in simulation).



To support planning, the signature Σ of system description $\mathcal{D}$ has basic sorts such as: *place*, *robot*, *person*, *object*, *entity*, *status*, and *step*, which are arranged hierarchically, e.g., *robot* and *person* are subsorts of *agent*, and *agent* and *object* are subsorts of *entity*. Σ also includes ground instances of sorts, e.g., *office*, *workshop*, *kitchen*, and *library* are instances of *place*, and *Sarah*, *Bob*, *John*, and *Sally* are instances of *person*. As before, domain attributes and actions are described in terms of the sorts of their arguments. The fluents include *loc(agent, place)*, which describes the location of the robot and people in the domain, and *message_status(message_id, person, status)*, which denotes whether a particular message has been delivered (or remains undelivered) to a particular person. Static attributes include relations such as *next_to(place, place)* and *work_place(person, place)* to encode the arrangement of places and the work location of people (respectively) in the domain. Actions of the domain include:

$$\begin{aligned} \text{move}(\text{robot}, \text{place}) \\ \text{deliver}(\text{robot}, \text{message_id}, \text{person}) \end{aligned} \quad (9)$$

which move the robot to a particular place, and cause a robot to deliver a particular message to a particular person (respectively). For ease of explanation, we assume that the locations of people are defined fluents whose values are determined by external sensors, and that the locations of objects are static attributes; as a result, we do not consider actions that change the value of these attributes. The signature Σ also includes (as before) the relation *holds(fluent, step)* to imply that a particular fluent is true at a particular time step.

Axioms of the RA domain capture the domain's dynamics. These axioms include causal laws, state constraints and executability conditions encoded as statements in $\mathcal{AL}_d$ such as:

$$\text{move}(\text{rob}_1, L) \text{ causes } \text{loc}(\text{rob}_1, L) \quad (10a)$$

$$\text{deliver}(\text{rob}_1, ID, P) \text{ causes } \text{message_status}(ID, P, \text{delivered}) \quad (10b)$$

$$\text{loc}(P, L) \text{ if } \text{work_place}(P, L), \text{ not } \neg \text{loc}(P, L) \quad (10c)$$

$$\neg \text{loc}(T, L_2) \text{ if } \text{loc}(T, L_1), L_1 \neq L_2 \quad (10d)$$

$$\text{impossible } \text{deliver}(\text{rob}_1, ID, P) \text{ if } \text{loc}(\text{rob}_1, L_1), \text{ loc}(P, L_2), \\ L_1 \neq L_2 \quad (10e)$$

$$\text{impossible } \text{move}(\text{rob}_1, L) \text{ if } \text{loc}(\text{rob}_1, L) \quad (10f)$$

where Statement 10(a) states that executing a move action causes the robot's location to be the target place; Statement 10(b) states that executing a deliver action causes the message to be delivered to the desired person; Statement 10(c) is a constraint stating that unless told otherwise the robot expects (by default) a person to be in her/his place of work; Statement 10(d) is a constraint stating that any thing can be in one place at time; Statement 10(e) implies that a robot cannot deliver a message to an intended recipient if the robot and the person are not in the same place; and Statement 10(f) states that a robot cannot move to a location if it is already there.

As described in section 3.2, the domain history is a record of observations (of fluents), the execution of actions, and the values of fluents in the initial state. Also, planning (similar to inference) is reduced to computing answer set(s) of the program $\Pi(\mathcal{D}, \mathcal{H})$ after including some helper axioms for computing a minimal sequence of actions; for examples, please see Gelfond and Kahl (2014) and Sridharan et al. (2019). If the robot's knowledge of the domain is incomplete or incorrect, the computed plans may be suboptimal or incorrect. The approach described in section 3.4 can then be used to learn the missing constraints; we will explore the interplay between learning and planning in section 4.4.

4. EXPERIMENTAL SETUP AND RESULTS

In this section, we describe the results of experimentally evaluating the following hypotheses about the capabilities of our architecture:

- **H1:** for the underlying classification problem, our architecture outperforms an architecture based on just deep networks for small training datasets, and provides comparable performance as the size of the dataset increases;
- **H2:** in the context of answering explanatory questions, our architecture provides significantly better performance in comparison with an architecture based on deep networks;
- **H3:** our architecture supports reliable and incremental learning of state constraints, which improves the ability to answer explanatory questions; and
- **H4:** our architecture can be adapted to planning tasks, with the incremental learning capability improving the ability to compute minimal plans.

These hypotheses were evaluated in the context of the domains (SS, TS, and RA) introduced in section 3. Specifically, hypotheses H1, H2, and H3 are evaluated in the SS domain and TS domain in the context of VQA. As stated in section 1, VQA is used in this paper only as an instance of a complex task that requires explainable reasoning and learning. We are primarily interested in exploring the interplay between reasoning with commonsense domain knowledge, incremental learning, and deep learning,

in any given domain in which large labeled datasets are not readily available. State of the art VQA algorithms, on the other hand, focus instead on generalizing across different domains, using benchmark datasets of several thousand images. Given the difference in objectives between over work and the existing work on VQA, we thus do not compare with state of the art algorithms, and do not use the benchmark VQA datasets. Furthermore, we evaluated hypothesis *H4* in the RA domain in which the robot's goal was to deliver messages to appropriate people and answer explanatory questions about this process.

We begin by describing some execution traces in section 4.1 to illustrate the working of our architecture. This is followed by sections 4.2–4.4, which describe the results of experimentally evaluating the classification, VQA, axiom learning, and planning capabilities, i.e., hypotheses *H1*–*H4*. We use accuracy (precision) as the primary performance measure. Classification accuracy was measured by comparing the assigned labels with the ground truth values, and question answering accuracy was evaluated heuristically by computing whether the answer mentions all image attributes relevant to the question posed. This relevance was established by a human expert, the lead author of this paper. Unless stated otherwise, we used two-thirds of the available data to train the deep networks and other computational models, using the remaining one-third of the data for testing. For each image, we randomly chose from the set of suitable questions for training the computational models. We repeated this process multiple times and report the average of the results obtained in these trials. For planning, accuracy was measured as the ability to compute minimal and correct plans for the assigned goals. Finally, section 4.5 discusses the reduction in computational effort achieved by our architecture in comparison with the baselines.

4.1. Execution Traces

The following execution traces illustrate our architecture's ability to reason with commonsense knowledge and learned models to provide intuitive answers for explanatory questions.

Execution Example 1. [*Question Answering, SS domain*] Consider a scenario in the SS Domain in which the input (test) image is the one on the extreme right in **Figure 2**.

- First **classification-related question** posed: “*is this structure unstable?*”
The architecture's **answer**: “*no*”.
- The **explanatory question** posed: “*what is making this structure stable?*”
The architecture's **answer**: “*the structure has five blocks and a narrow base, it is standing straight, and there is no significant lean*”.
- This answer was based on the following features extracted by CNNs from the image: (i) five blocks; (ii) narrow base; (iii) standing straight; and (iv) no significant lean, i.e., all blocks in place.
- The extracted features were converted to literals. ASP-based inference provided an answer about the stability of the arrangement of objects in the scenario. Relevant literals in the

corresponding answer set were then inserted into a suitable template to provide the answers described above.

- Since the example was processed successfully using ASP-based inference, it was not processed using the decision tree (for classification) or the RNN (for answering the explanatory question).

Execution Example 2. [*Question Answering, TS domain*] Consider a scenario in the TS Domain with the input (test) image is the one on the extreme right in **Figure 3**.

- The **classification question** posed was: “*what is the sign's message?*”
The architecture's **answer**: “*uneven surfaces ahead*”.
- When asked to explain this answer (“*Please explain this answer*”), the architecture identified that the CNNs extracted the following features of the sign in the image: (i) it is triangle-shaped; (ii) main color is white and other (i.e., border) color is red; (iii) it has no background image; (iv) it has a bumpy-road symbol and no secondary symbol; and (v) it has no cross.
- These features were converted to literals and used in ASP-based inference based on existing knowledge in the TS domain. ASP-based inference is unable to provide an answer, i.e., unable to classify the sign.
- The extracted features were processed using the trained decision tree, which only used the colors in the sign to assign the class label. The main (or border) color is normally insufficient to accurately classify signs. However, recall that the decision tree is trained to classify signs that cannot be classified by reasoning with existing knowledge.
- The decision tree output, image feature vector, and input question, were processed by the previously trained RNN to provide the answer type and the particular answer described above.

These (and other such) execution traces illustrate the working of our architecture, especially that:

- The architecture takes advantage of (and perform non-monotonic logical inference with) the existing commonsense domain knowledge to reliably and efficiently address the decision-making problem (classification in the examples above) when possible. In such cases, it is also able to answer explanatory questions about the classification decision and the underlying scene.
- When the desired decision cannot be made using non-monotonic logical inference with domain knowledge, the architecture smoothly transitions to training and using a decision-tree to make and explain the classification decision. In such cases, the architecture also learns and uses an RNN to answer explanatory questions about the scene.

4.2. Experimental Results: Classification + VQA

To quantitatively evaluate hypotheses *H1* and *H2*, we ran experimental trials in which we varied the size of the training dataset. In these trials, the baseline performance was provided by a CNN-RNN architecture, with the CNNs processing images

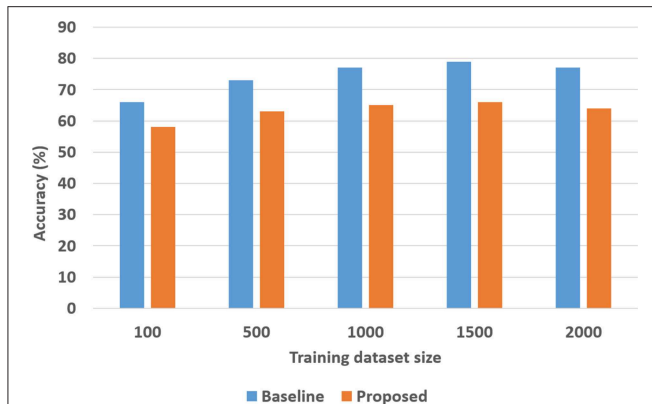


FIGURE 8 | Classification accuracy as a function of the number of training samples in the SS domain.

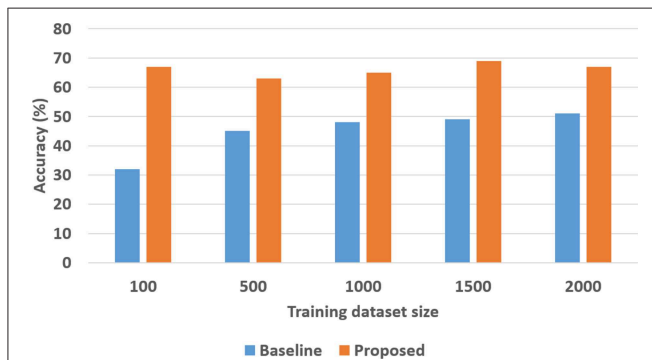


FIGURE 9 | VQA accuracy as a function of the number of training samples in the SS domain.

to extract and classify features, and the RNN providing answers to explanatory questions. The number of questions considered depends on the complexity of the domain, e.g., we included eight different types of questions in the SS domain and 248 different types of questions in the TS domain. We repeated the trials 50 times (choosing the training set randomly each time) and the corresponding average results are summarized in **Figures 8, 9** for the SS domain, and in **Figures 10, 11** for the TS domain. We make some observations based on these figures:

1. The classification performance of our architecture depends on the domain. In the relatively simpler SS domain, the baseline deep network architecture is at least as accurate as our architecture, even with a small training set—see **Figure 8**. This is because small differences in the position and arrangement of blocks (which could almost be considered as noise) influence the decision about stability. For instance, two arrangements of blocks that are almost identical end up receiving different ground truth labels for stability, and it is not possible to draft rules based on abstract image features to distinguish between these cases. The baseline deep network architecture, which generalizes from data, is observed to be more sensitive to these small changes than our architecture. Exploring the

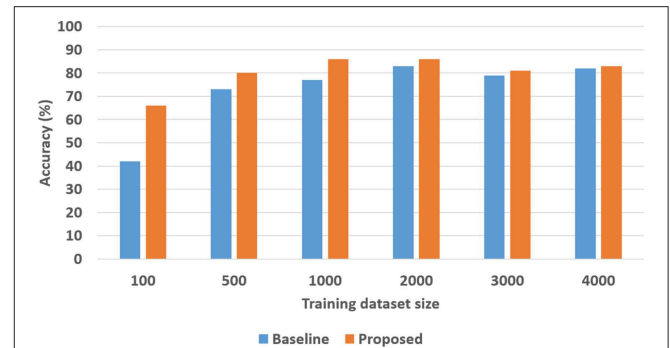


FIGURE 10 | Classification accuracy as a function of number of training samples in TS domain.

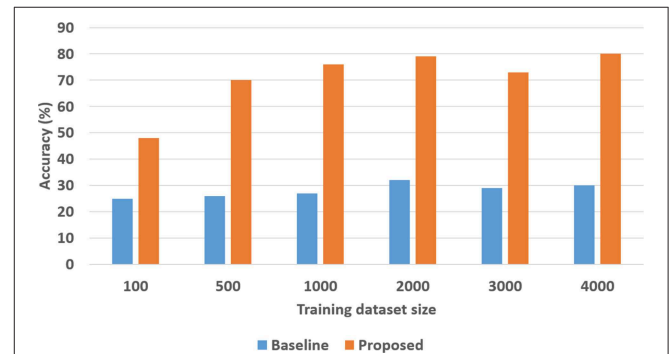
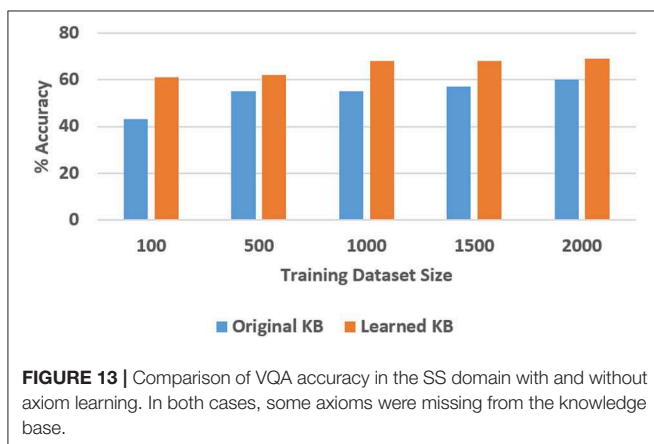
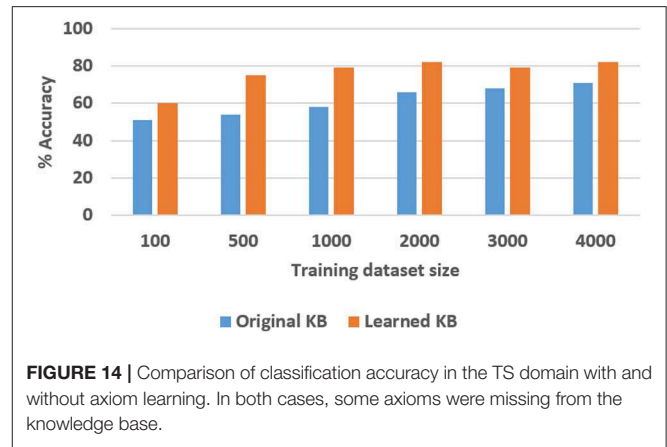
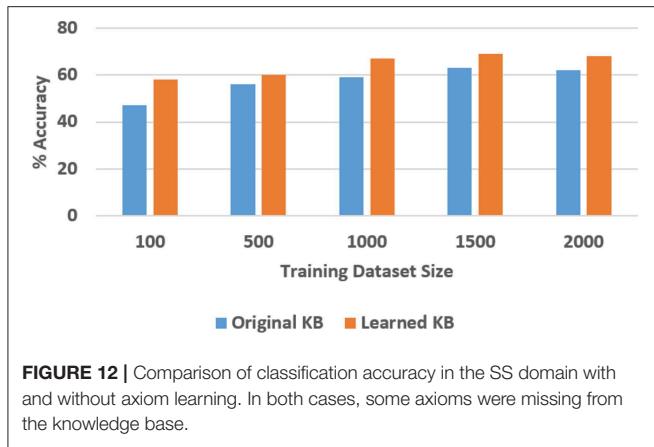


FIGURE 11 | VQA accuracy as a function of number of training samples in TS domain.

reason for this performance is an interesting direction for further research.

2. In the more complex TS domain, our architecture provides better classification accuracy than the baseline architecture based on just deep networks, especially when the size of the training set is small—see **Figure 10**. The classification accuracy increases with the size of the training set³, but our architecture is always at least as accurate as the baseline architecture.
3. Our architecture is much more capable of answering explanatory questions about the classification decisions than the baseline architecture. When the answer provided by our architecture does not match the ground truth, we are able to examine why that decision was made. We were thus able to understand and explain the lower classification accuracy of our architecture in the SS domain. The baseline architecture does not provide this capability.
4. Unlike classification, the VQA performance of our architecture is much better than that of the baseline architecture in both domains. Also performance does not improve just by increasing the size of training set, even in simpler domains, e.g., see **Figure 9**. This is because

³We limit ourselves to training sets that are not too large in order to match the focus of our paper.



VQA performance also depends on the complexity of the explanatory questions. For more complex domains, the improvement in VQA accuracy provided by our architecture is much more pronounced, e.g., see **Figure 11**.

We explored the statistical significance of the observed performance by running paired t-tests. We observed that the VQA performance of the proposed architecture was significantly better than that of the baseline architecture; this is more pronounced in the TS domain that is more complex than the SS domain. Also, although the baseline architecture provides better classification performance in the SS domain, the difference is not always statistically significant.

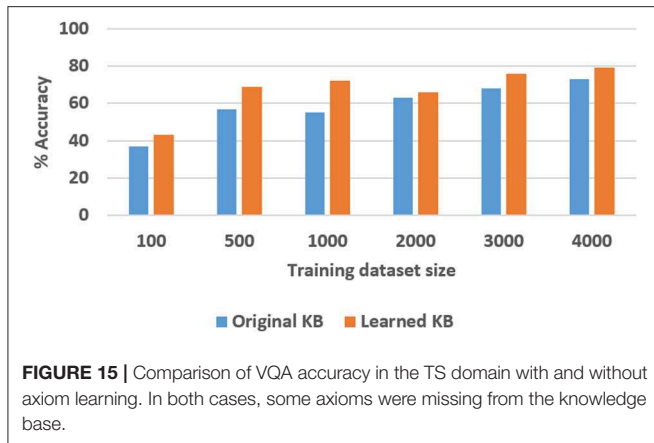
To further explore the observed results, we obtained a “confidence value” from the logits layer of each CNN used to extract a feature from the input image. For each CNN, the confidence value is the largest probability assigned to any of the possible values of the corresponding feature, i.e., it is the probability assigned to the most likely value of the feature. These confidence values are considered to be a measure of the network’s confidence in the corresponding features being a good representation of the image. We trained a version of our architecture in which if the confidence value for any feature was low, the image features were only used to revise

the decision tree (during training), or were processed using the decision tree (during testing). In other words, features that do not strongly capture the essence of the image are not used for non-monotonic logical reasoning; the deep network architectures provide much better generalization to noise. We hypothesized that this approach would improve the accuracy of classification and question answering, but it did not make any significant difference in our experimental trials. We believe this is because the extracted features were mostly good representations of the objects of interest in the images. We thus did not use such networks (that compute the confidence value) in any other experiments.

4.3. Experimental Results: Learn Axiom + VQA

Next, we experimentally evaluated the ability to learn axioms, and the effect of such learning on the classification and VQA performance. For the SS domain, we designed a version of the knowledge base with eight axioms related to stability or instability of the structures. Out of these, four were chosen (randomly) to be removed and we examined the ability to learn these axioms, and the corresponding accuracy of classification and VQA, as a function of the number of labeled training examples (ranging from 100 to 2,000). We repeated these experiments 30 times and the results (averaged over the 30 trials) are summarized in **Figures 12, 13**. In the TS domain, the methodology for experimental evaluation was the same. However, since the domain was more complex, there were many more axioms in the domain description (for classification and VQA); we also had access to more labeled training examples. In each experimental trial, a quarter of the available axioms were thus selected and commented out, and the accuracy of classification and VQA were evaluated with the number of labeled training examples varying from 100 to 4000. The results averaged over 30 such trials are summarized in **Figures 14, 15**.

In these figures, “Original KB” (depicted in blue) represents the baseline with some axioms missing from the system description, e.g., four in the SS domain and one quarter of the axioms in the TS domain. The results obtained by using the available labeled examples to learn the axioms that are then used



for classification and answering explanatory questions about the scene, are shown as “Learned KB” in orange. We observe that our approach supports incremental learning of the domain axioms, and that using the learned axioms improves the classification accuracy and the accuracy of answering explanatory questions, in comparison with the baseline. This improvement was found to be statistically significant using paired tests at 95% level of significance. These results support hypothesis *H3*.

4.4. Experimental Results: Learn Axiom + Plan

Next, we experimentally evaluated the ability to learn axioms and the effect of the learned axioms on planning, in the RA domain. The simulated robot was equipped with domain knowledge for planning, classification, and question answering. It uses this knowledge to navigate through an office building, locate the intended recipient of a message, deliver the message, detect and reason about objects in its surroundings, and answer questions about the rooms it has visited. We considered 24 different types of questions in this domain. As stated in section 3.5, we limit uncertainty in sensing and actuation on robots to noise added in simulation. Average results from 100 trials indicates a VQA accuracy of $\approx 85\%$ after training the architecture’s components with just 500 labeled images. The domain knowledge includes learned axioms—the corresponding experimental results and the planning performance are discussed later in this section. We begin with an execution trace in this domain.

Execution Example 3. [*Question Answering, RA Domain*] Consider the scenario in the RA domain (Figure 7) in which the robot’s goal was to deliver a message from John to Sally, and return to John to answer questions.

- The robot was initially in John’s office. It computed a plan that comprises moving to Sally’s office through the library and the kitchen, delivering the message to Sally, and returning to John’s office through the same route to answer questions.
- During plan execution, the robot periodically takes images of the scenes in the domain, which are used for planning, classification and question answering.

- After returning to John’s office, the robot and the human had an exchange about the plan constructed and executed, and the observations received. The exchange includes instances such as:

John’s question: “is Sally’s location cluttered?”

Robot’s answer: “Yes.”

When asked, the robot provides an **explanation** for this decision: “Sally is in her office. Objects detected are Sally’s chair, desk, and computer, and a cup, a large box, and a sofa. The room is cluttered because the cup, large box, and sofa are not usually in that room.”

The RA domain was also used to evaluate the effects of axiom learning. There were four employees in offices in the simulated scenario, as shown in Figure 7, and the robot was asked to find particular individuals and deliver particular messages to them. Employees are initially expected to be in their assigned workplace (i.e., their office), and spend most of their time in these offices, unless this default knowledge has been negated by other knowledge or observations. This information is encoded as follows:

$$\text{holds}(\text{loc}(P, L), 0) \leftarrow \text{not default_negated}(P, L), \text{work_place}(P, L)$$

where $\text{work_place}(P, L)$ specifies the default location of each person, and $\text{default_negated}(P, L)$ is used to encode that a particular person may not be in their default location. These exceptions to the defaults can be encoded as follows:

$$\text{default_negated}(P, L) \leftarrow \text{obs}(\text{loc}(P, L1), \text{true}, I), L \neq L1 \quad (11a)$$

$$\text{default_negated}(P, L) \leftarrow \text{obs}(\text{loc}(P, L), \text{false}, I) \quad (11b)$$

Statement 11(a) implies that the default assumption should be ignored if the person in question is observed to be in a location other than their workplace, and Statement 11(b) implies that a default assumption should be ignored if the corresponding person is not observed in their workplace. Including such default knowledge (and exceptions) in the reasoning process allows the robot to compute better plans and execute the plans more efficiently, e.g., when trying to deliver a message to a particular person. However, this knowledge may not be known in advance, the existing knowledge may be inaccurate or change with time (e.g., humans can move between the different places), or the observations may be incorrect. Our axiom learning approach was used in this domain to acquire previously unknown information about the default location of people and exceptions to these defaults. In all the trials, the simulated robot was able to learn the previously unknown axioms.

We then conducted 100 paired trials to explore the effects of the learned axioms on planning, with the corresponding results summarized in Table 1. In each trial, we randomly chose a particular goal and initial conditions, and measured planning performance before and after the previously unknown axioms had been learned and used for reasoning. Since the initial conditions are chosen randomly, the object locations, the initial location of the robot, and the goal, may vary significantly between trials. Under these circumstances, it is not meaningful to average

TABLE 1 | Planning performance in a scenario in the RA domain (see **Figure 7**) before and after axiom learning.

Axiom learning	Plans (per trial)	Actions (per trial)	Execution time (per trial)	Planning time (per trial)	Planning time (per plan)
Before	4	2.3	1.6	6.0	1.6
After	1	1	1	1	1

Results averaged over 100 paired trials indicate that reasoning with previously unknown axioms results in fewer plans with fewer actions in each trial, and significantly reduces the time taken to compute and execute the plans.

the results obtained in the individual trials for performance measures such as planning time and execution time. Instead, the results obtained without including the learned axioms were computed as a ratio of the results obtained after including the learned axioms; the numbers reported in **Table 1** are the average of these computed ratios. Before axiom learning, the robot often explored an incorrect location (for a person) based on other considerations (e.g., distance to the room) and ended up having to replan. After the previously unknown axioms were included in the reasoning process, the robot went straight to the message recipient's most likely location, which also happened to be the actual location of the recipient in many trials. As a result, we observe a (statistically) significant improvement in planning performance after the learned axioms are used for reasoning. Note that in the absence of the learned axioms, the robot computes four times as many plans taking six times as much time in any given trial (on average) as when the learned axioms are included in reasoning. Even the time taken to compute each plan (with potentially multiple such plans computed in each trial) is significantly higher in the absence of the learned axioms. This is because the learned axioms enable the robot to eliminate irrelevant paths in the transition diagram from further consideration. In a similar manner, reasoning with intentional actions enables the robot to significantly reduce the plan execution time by terminating or revising existing plans when appropriate, especially in the context of unexpected successes and failures. These results provide evidence in support of hypothesis *H4*.

Finally, we conducted some initial proof of concept studies exploring the use of our architecture on physical robots. We considered a robot collaborating with a human to jointly describe structures of blocks on a tabletop (similar to the SS domain described in this paper). We also considered a mobile robot finding and moving objects to desired locations in an indoor domain (similar to the RA domain). These initial experiments provided some promising outcomes. The robot was able to provide answers to explanatory questions, compute and execute plans to achieve goals, and learn previously unknown constraints. In the future, we will conduct a detailed experimental analysis on robots in different domains.

4.5. Computational Effort

In addition to the improvement in accuracy of classification and VQA, we also explored the reduction in computational effort provided by our architecture in comparison with the baselines. Measuring this time quantitatively is challenging because it

depends on various factors such as the task being performed (e.g., classification, VQA), the knowledge encoded in the knowledge base, the size and order of samples in the training set, and the parameters of the deep networks. However, we were able to gain the following insights.

The computation time includes the training time and the testing (i.e., execution) time, and we first considered the training time. Depending on the task being performed (e.g., classification, VQA, and/or planning), this time includes the time taken to encode and draw inferences from the knowledge base, process queries and construct answers, and train the deep network models. Encoding the incomplete domain knowledge is a one-time exercise for any given domain. The time taken to reason with this knowledge, and the time taken to process queries and construct answers, are negligible in comparison with the time taken to learn the deep network models. Also, the use of CNNs to extract features from images is common to both our architecture and the baselines, and these networks (for the most part) do not need to be retrained multiple times for any given domain. The key difference between our architecture and the baselines is observed in the context of answering explanatory questions about the scenes and the underlying classification problem. Recall that with our architecture, only examples that cannot be processed by ASP-based reasoning are processed by decision-trees and the RNNs for VQA. In our experimental trials, $\approx 10 - 20\%$ of a training set is used (on average) to train the RNNs with our architecture, whereas the entire training set is used for training the RNNs with the baseline architectures. This difference often translates to an order of magnitude difference in the training time, e.g., a few minutes for each training set (in a particular domain) with our architecture compared with hours or days with the baseline architectures. Note that accuracy of our architecture is still much better than that of the baselines, e.g., see **Figures 9, 11**, i.e., any given accuracy is achieved using a much smaller number of training samples.

The execution time of our architecture is comparable with that of the baselines and is often less. Once the deep network models have been learned, using them for the different tasks does not take much time, e.g., a few seconds to process the input and provide a decision and/or the answer to a query. However, similar to the situation during training, only test samples that cannot be processed by ASP-based reasoning are processed by the decision trees and RNNs with our architecture. Also, since the deep networks in our architecture only need to disambiguate between a small(er) number of training examples, they often have a much simpler structure than the deep networks in the baseline architectures.

Note that in addition to classification and VQA, our architecture also supports explainable reasoning for planning and incremental learning of previously unknown constraints. Providing similar capabilities using just deep network architectures will (at the very least) require a large number of training examples of planning under different conditions; it is often not possible to provide such training examples in dynamic domains. We thus conclude that our architecture significantly reduces the computational effort while supporting a range of capabilities in comparison with the baseline architectures comprising deep networks.

5. DISCUSSION AND FUTURE WORK

Visual question answering (VQA) combines challenges in computer vision, natural language processing, and explainability in reasoning and learning. Explanatory descriptions of decisions help identify errors, and to design better algorithms and frameworks. In addition, it helps improve trust in the use of reasoning and learning systems in critical application domains. State of the art algorithms for VQA are based on deep networks and the corresponding learning algorithms. Given their focus on generalizing across different domains, these approaches are computationally expensive, require large training datasets, and make it difficult to provide explanatory descriptions of decisions. We instead focus on enabling reliable and efficient operation in any given domain in which a large number of labeled training examples may not be available. Inspired by research in cognitive systems, our architecture tightly couples representation, reasoning, and interactive learning, and exploits the complementary strengths of deep learning, non-monotonic logical reasoning with commonsense knowledge, and decision tree induction. Experimental results on datasets of real world and simulated images indicate that our architecture provides the following benefits in comparison with a baseline architecture for VQA based on deep networks:

1. Better accuracy, improved sample efficiency, and reduced computational effort on classification problems when the training dataset is small, and comparable accuracy with larger datasets while still using only a subset of these samples for training;
2. Ability to provide answers to explanatory questions about the scenes and the underlying decision making problems (e.g., classification, planning);
3. Incremental learning of previously unknown domain constraints, whose use in reasoning improves the ability to answer explanatory questions; and
4. Ability to adapt the complementary strengths of non-monotonic logical reasoning with commonsense domain knowledge, inductive learning, and deep learning, to address decision-making (e.g., planning) problems on a robot.

Our architecture opens up multiple directions of future work, which will address the limitations of existing work and significantly extend the architecture's capabilities. We discuss some of these extensions below:

1. The results reported in this paper are based on image datasets (simulated, real-world) chosen or constructed to mimic domains in which a large, labeled dataset is not readily available. One direction of future work is to explore the use of our architecture in other domains that provide datasets of increasing complexity, i.e., with a greater number of features and more complex explanatory questions. This exploration may require us to consider larger datasets, and to examine the trade-off between the size of the training dataset, the computational effort involved in processing such a dataset with many labeled examples, and the effort involved in encoding and reasoning with the relevant domain knowledge.

2. In our architecture, we have so far used variants of existing network structures as the deep network components (i.e., CNN, RNN). In the future, we will explore different deep network structures in our architecture, using the explanatory answers to further understand the internal representation of these network structures. Toward this objective, it would be particularly instructive to construct and explore deep networks and logic-based domain representations that provide similar behavior on a set of tasks, or provide different behavior when operating on the same dataset. As stated in the discussion in section 4.2, such an exploration may help us better understand (and improve) the design and use of deep network models for different applications.
3. This paper used VQA as a motivating problem to address key challenges in using deep networks in dynamic domains with limited labeled training examples. We also described the use of our architecture (with tightly-coupled reasoning and learning components) for planning on a simulated robot. In the future, we will combine this architecture with other architectures we have developed for knowledge representation, reasoning, and interactive learning in robotics (Sridharan and Meadows, 2018; Sridharan et al., 2019). The long-term goal will be to support explainable reasoning and learning on a physical robot collaborating with humans in complex domains.

DATA AVAILABILITY STATEMENT

The datasets generated or analyzed for this study, and the software implementation of the architecture and algorithms, can be found in the following online repository: https://github.com/hril230/masters_code.

AUTHOR CONTRIBUTIONS

HR and MS designed the algorithms and architecture, experimental setup, and analyzed the results. HR implemented the algorithms and architecture with feedback from MS. HR conducted the evaluation. MS wrote the paper with contributions from HR.

FUNDING

This work was supported in part by the US Office of Naval Research Science of Autonomy award N00014-17-1-2434, and the Asian Office of Aerospace Research and Development award FA2386-16-1-4071.

ACKNOWLEDGMENTS

The authors thank Ales Leonardis for feedback on the architecture described in this paper. An earlier version of this architecture appeared in a conference paper (Riley and Sridharan, 2018a). In this paper, we have introduced a new component for incrementally learning constraints governing domain states, expanded reasoning to support planning and diagnostics, and discussed more detailed experimental results.

REFERENCES

- Agrawal, A., Batra, D., Parikh, D., and Kembhavi, A. (2018). "Don't just assume; look and answer: overcoming priors for visual question answering," in *International Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT).
- Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., et al. (2018). "Bottom-up and top-down attention for image captioning and visual question answering," in *International Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT).
- Bai, Y., Fu, J., Zhao, T., and Mei, T. (2018). "Deep attention neural tensor network for visual question answering," in *European Conference on Computer Vision (ECCV)* (Munich).
- Balai, E., Gelfond, M., and Zhang, Y. (2013). "Towards answer set programming with sorts," in *International Conference on Logic Programming and Nonmonotonic Reasoning* (Corunna).
- Balduccini, M. (2007). "Learning action descriptions with a-prolog: action language C," in *AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning* (Stanford, CA).
- Balduccini, M., and Gelfond, M. (2003). "Logic programs with consistency-restoring rules," in *AAAI Spring Symposium on Logical Formalization of Commonsense Reasoning* (Stanford, CA), 9–18.
- Baral, C. (2003). *Knowledge Representation, Reasoning and Declarative Problem Solving*. New York, NY: Cambridge University Press.
- Chai, J. Y., Gao, Q., She, L., Yang, S., Saba-Sadiya, S., and Xu, G. (2018). "Language to action: towards interactive task learning with physical agents," in *International Joint Conference on Artificial Intelligence (IJCAI)* (Stockholm).
- Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification, 2nd Edn*. Wiley-Blackwell.
- Erdem, E., Gelfond, M., and Leone, N. (2016). Applications of answer set programming. *AI Mag.* 37, 53–68. doi: 10.1609/aimag.v37i3.2678
- Erdem, E., and Patoglu, V. (2012). "Applications of action languages to cognitive robotics," in *Correct Reasoning*, eds E. Erdem, J. Lee, Y. Lierler, and D. Pierce (Berlin: Springer-Verlag), 229–246.
- Furbach, U., Glöckner, I., Helbig, H., and Pelzer, B. (2010). Logic-based question answering. *KI - Künstliche Intelligenz* 24, 51–55. doi: 10.1007/s13218-010-0010-x
- Gelfond, M., and Inlezan, D. (2013). Some properties of system descriptions of AL_q . *J. Appl. Non-Class. Logics Special Issue Equilibrium Logic Answer Set Progr.* 23, 105–120. doi: 10.1080/11663081.2013.798954
- Gelfond, M., and Kahl, Y. (2014). *Knowledge Representation, Reasoning and the Design of Intelligent Agents*. Cambridge: Cambridge University Press.
- Gil, Y. (1994). "Learning by experimentation: incremental refinement of incomplete planning domains," in *International Conference on Machine Learning* (New Brunswick, NJ), 87–95.
- Goyal, Y., Khot, T., Stay, D. S., Batra, D., and Parikh, D. (2017). "Making the V in VQA matter: elevating the role of image understanding in visual question answering," in *International Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 6325–6334.
- Jabri, A., Joulin, A., and van der Maaten, L. (2016). "Revisiting visual question answering baselines," in *European Conference on Computer Vision* (Amsterdam).
- Jiang, A., Wang, F., Porikli, F., and Li, Y. (2015). *Compositional Memory for Visual Question Answering*. Technical report. Available online at: <https://arxiv.org/abs/1511.05676>
- Kim, J.-H., Jun, J., and Zhang, B.-T. (2018). "Bilinear attention networks," in *Neural Information Processing Systems* (Montreal, QC).
- Koh, P. W., and Liang, P. (2017). "Understanding black-box predictions via influence functions," in *International Conference on Machine Learning (ICML)* (Sydney, NSW), 1885–1894.
- Laird, J. E. (2012). *The Soar Cognitive Architecture*. Cambridge, MA: The MIT Press.
- Laird, J. E., Gluck, K., Anderson, J., Forbus, K. D., Jenkins, O. C., Lebiere, C., et al. (2017). Interactive task learning. *IEEE Intell. Syst.* 32, 6–21. doi: 10.1109/MIS.2017.3121552
- Law, M., Russo, A., and Broda, K. (2018). The complexity and generality of learning answer set programs. *Artif. Intell.* 259, 110–146. doi: 10.1016/j.artint.2018.03.005
- Li, Q., Fu, J., Yu, D., Mei, T., and Luo, J. (2018). *Tell-and-Answer: Towards Explainable Visual Question Answering using Attributes and Captions*. Technical report. Available online at: <https://arxiv.org/abs/1801.09041>
- Lin, T., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., et al. (2014). "Microsoft COCO: common objects in context," in *European Conference on Computer Vision* (Zurich), 740–755.
- Lu, J., Yang, J., Batra, D., and Parikh, D. (2016). "Hierarchical question-image co-attention for visual question answering," in *Advances in Neural Information Processing Systems* (Barcelona).
- Malinowski, M., Rohrbach, M., and Fritz, M. (2017). Ask your neurons: a deep learning approach to visual question answering. *Int. J. Comput. Vis.* 125, 110–135. doi: 10.1007/s11263-017-1038-2
- Mascharka, D., Tran, P., Soklaski, R., and Majumdar, A. (2018). "Transparency by design: closing the gap between performance and interpretability in visual reasoning," in *International Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT).
- Masuda, I., de la Puente, S. P., and Nieto, X. G. (2016). "Open-ended visual question-answering," in *International Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV).
- Mota, T., and Sridharan, M. (2019). "Commonsense reasoning and knowledge acquisition to guide deep learning on robots," in *Robotics Science and Systems* (Freiburg).
- Norcliffe-Brown, W., Vafeais, E., and Parisot, S. (2018). "Learning conditioned graph structures for interpretable visual question answering," in *Neural Information Processing Systems* (Montreal, QC).
- Otero, R. P. (2003). "Induction of the effects of actions by monotonic methods," in *International Conference on Inductive Logic Programming* (Szeged), 299–310.
- Pandhre, S., and Sodhani, S. (2017). *Survey of Recent Advances in Visual Question Answering*. Technical report. Available online at: <https://arxiv.org/abs/1709.08203>
- Rajani, N. F., and Mooney, R. J. (2018). "Stacking with auxiliary features for visual question answering," in *16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (New Orleans, LA), 2217–2226.
- Ribeiro, M., Singh, S., and Guestrin, C. (2016). "Why should I trust you? Explaining the predictions of any classifier," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (San Francisco, CA), 1135–1144.
- Riley, H., and Sridharan, M. (2018a). "Non-monotonic logical reasoning and deep learning for explainable visual question answering," in *International Conference on Human-Agent Interaction* (Southampton).
- Riley, H., and Sridharan, M. (2018b). *Software for Architecture combining Non-monotonic Logical Reasoning, Inductive Learning and Deep Learning for VQA*. Available online at: https://github.com/hril230/masters_code (accessed January 2019).
- Sarathy, V., and Scheutz, M. (2018). A logic-based computational framework for inferring cognitive affordances. *IEEE Trans. Cogn. Dev. Syst.* 10, 26–43. doi: 10.1109/TCDS.2016.2615326
- Schwartz, I., Schwing, A., and Hazan, T. (2017). "High-order attention models for visual question answering," in *Advances in Neural Information Processing Systems (NIPS)* (Long Beach, CA), 3664–3674.
- Selvaraju, R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). "Grad-CAM: visual explanations from deep networks via gradient-based localization," in *International Conference on Computer Vision* (Venice), 618–626.
- Shrestha, R., Kafle, K., and Kanan, C. (2019). "Answer Them All! Toward universal visual question answering models," in *International Conference on Computer Vision and Pattern Recognition* (Long Beach, CA).
- Sridharan, M., Gelfond, M., Zhang, S., and Wyatt, J. (2019). REBA: a refinement-based architecture for knowledge representation and reasoning in robotics. *J. Artif. Intell. Res.* 65, 87–180. doi: 10.1613/jair.1.11524
- Sridharan, M., and Meadows, B. (2018). Knowledge representation and interactive learning of domain knowledge for human-robot collaboration. *Adv. Cogn. Syst.* 7, 69–88.
- Teney, D., and van den Hengel, A. (2016). *Zero-Shot Visual Question Answering*. Technical report. Available online at: <https://arxiv.org/abs/1611.05546>
- Teney, D., and van den Hengel, A. (2018). "Visual question answering as a meta learning task," in *European Conference on Computer Vision (ECCV)* (Munich).

- Timofte, R., Mathias, M., Benenson, R., and Gool, L. V. (2013). "Traffic sign recognition - how far are we from the solution?," in *International Joint Conference on Neural Networks (IJCNN)* (Dallas, TX), 1–8.
- Wagner, M., Basevi, H., Shetty, R., Li, W., Malinowski, M., Fritz, M., et al. (2018). "Answering visual *What-If* questions: from actions to predicted scene descriptions," in *Visual Learning and Embodied Agents in Simulation Environments (VLEASE) Workshop at ECCV* (Munich). Available online at: <https://arxiv.org/abs/1809.03707>
- Wang, P., Wu, Q., Shen, C., van den Hengel, A., and Dick, A. R. (2017). "Explicit knowledge-based reasoning for visual question answering," in *International Joint Conference on Artificial Intelligence* (Melbourne, VIC).
- Wu, C., Liu, J., Wang, X., and Dong, X. (2018). "Chain of reasoning for visual question answering," in *Advances in Neural Information Processing Systems (NeurIPS)* (Montreal, QC), 273–283.
- Yang, Z., He, X., Gao, J., Deng, L., and Smola, A. J. (2016). "Stacked attention networks for image question answering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV), 21–29.
- Yi, K., Wu, J., Gan, C., Torralba, A., Kohli, P., and Tenenbaum, J. B. (2018). "Neural-symbolic VQA: disentangling reasoning from vision and language understanding," in *Neural Information Processing Systems* (Montreal, QC).
- Zhang, S., Sridharan, M., and Wyatt, J. (2015). Mixed logical inference and probabilistic planning for robots in unreliable worlds. *IEEE Trans. Robot.* 31, 699–713. doi: 10.1109/TRO.2015.2422531
- Zhang, T., Dai, D., Tuytelaars, T., Moens, M.-F., and Gool, L. V. (2017). *Speech-Based Visual Question Answering*. Technical report. Available online at: <https://arxiv.org/abs/1705.00464>

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2019 Riley and Sridharan. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Symbolic Representation and Learning With Hyperdimensional Computing

Anton Mitrokhin^{1†}, Peter Sutor^{1*†}, Douglas Summers-Stay², Cornelia Fermüller¹ and Yiannis Aloimonos¹

¹ Computer Vision Laboratory, Department of Computer Science, University of Maryland Institute for Advanced Computer Studies, University of Maryland, College Park, MD, United States, ² Computational and Information Sciences Directorate, Army Research Laboratory, Adelphi, MD, United States

OPEN ACCESS

Edited by:

Amy Loutfi,
Örebro University, Sweden

Reviewed by:

Elena Bellodi,
University of Ferrara, Italy
Hector Zenil,
Karolinska Institutet (KI), Sweden

*Correspondence:

Peter Sutor
psutor@umd.edu

[†]These authors have contributed
equally to this work

Specialty section:

This article was submitted to
Computational Intelligence in
Robotics,
a section of the journal
Frontiers in Robotics and AI

Received: 15 February 2020

Accepted: 15 April 2020

Published: 09 June 2020

Citation:

Mitrokhin A, Sutor P,
Summers-Stay D, Fermüller C and
Aloimonos Y (2020) Symbolic
Representation and Learning With
Hyperdimensional Computing.
Front. Robot. AI 7:63.
doi: 10.3389/frobt.2020.00063

It has been proposed that machine learning techniques can benefit from symbolic representations and reasoning systems. We describe a method in which the two can be combined in a natural and direct way by use of hyperdimensional vectors and hyperdimensional computing. By using hashing neural networks to produce binary vector representations of images, we show how hyperdimensional vectors can be constructed such that vector-symbolic inference arises naturally out of their output. We design the Hyperdimensional Inference Layer (HIL) to facilitate this process and evaluate its performance compared to baseline hashing networks. In addition to this, we show that separate network outputs can directly be fused at the vector symbolic level within HILs to improve performance and robustness of the overall model. Furthermore, to the best of our knowledge, this is the first instance in which meaningful hyperdimensional representations of images are created on real data, while still maintaining hyperdimensionality.

Keywords: hyperdimensional computing, semantic vectors, hashing, machine learning, image processing

1. INTRODUCTION

Over the past decade, Machine Learning (ML) has made great strides in its capabilities to the point that many today cannot imagine solving complex, data-hungry tasks without its use. Indeed, as learning by example is a very necessary skill for an artificial general intelligence, it seems that ML's success bodes its necessity - in some form or other - in future AI systems. At the same time, end-to-end ML solutions suffer from several disadvantages; results are generally not interpretable or explainable from a human perspective, new data is difficult to absorb without significant retraining, and the amount of data/internalized knowledge required to train can be untenable for tasks that are easy for humans to solve. Symbolic reasoning solutions, on the other hand, can offer a solution to these problems.

One issue with symbolic reasoning is that symbols preferred by humans may not be easy to teach an AI to understand in human-like terms. Problems like these have led to the interesting solution of representing symbolic information as vectors embedded into high dimensional spaces, such as systems like word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014). These are often used to inform other symbolic or ML systems to give semantic context to information represented textually. In some systems, symbolic concepts themselves are represented entirely as high dimensional vectors that coexist in a common space-these are often referred to as Vector

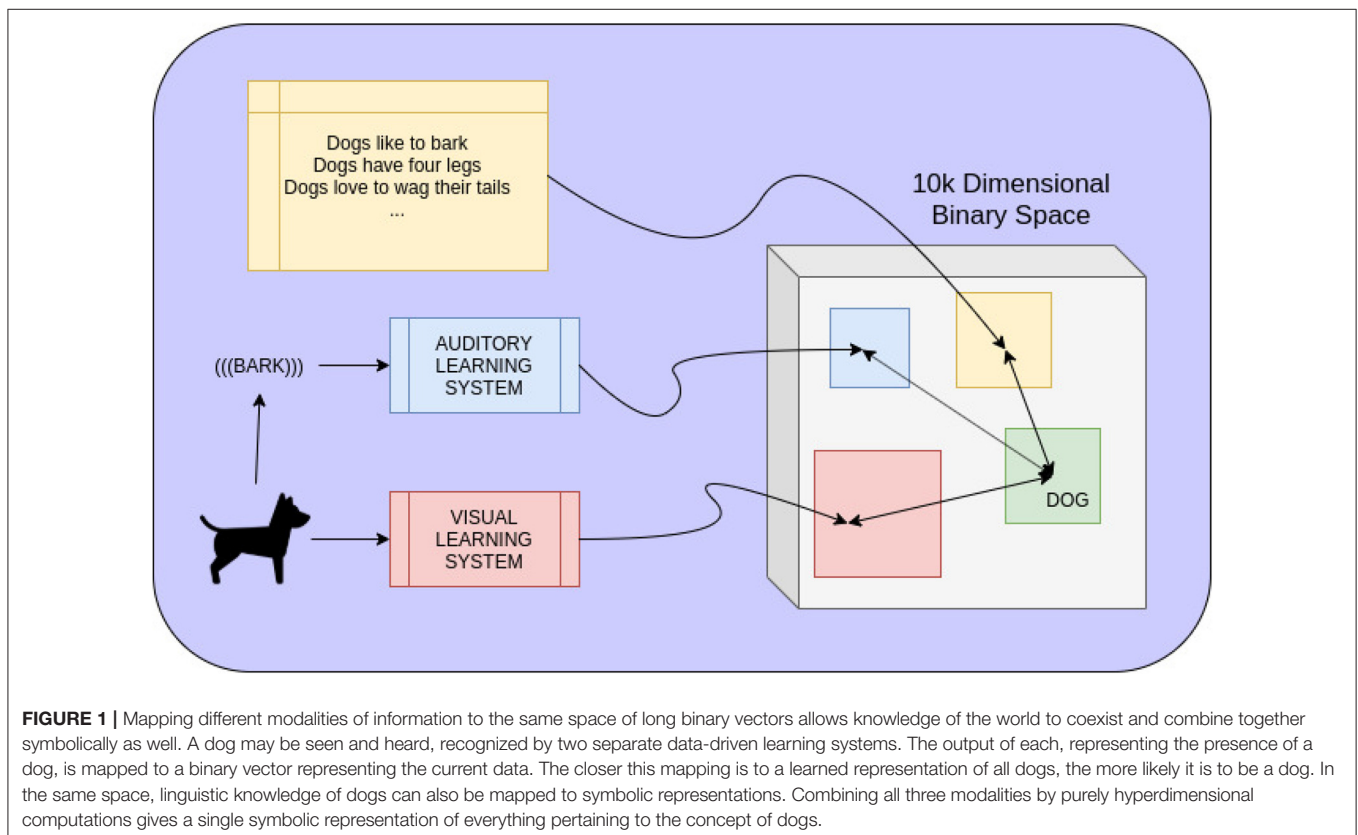
Symbolic Architectures (VSA). This notion is of particular interest, as many ML techniques produce such high dimensional vectors as a byproduct of their learning process or their operation.

In this article, we have focused on the notion of combining ML systems and VSA using high dimensional vectors directly. Specifically, we focused on the use of hyperdimensional vectors and Hyperdimensional Computing to achieve this (Kanerva, 2009). The properties of hyperdimensionality give rise to interesting ways to manipulate symbolic information so long as that information can be represented with long binary vectors. Moreover, this combination is achieved naturally, and is highly modifiable. Hyperdimensional computing can even improve the results of ML methods. Since hyperdimensional computing requires a method to convert data into long binary vectors, we focused mostly on hashing techniques for images, though the results are applicable for any ML approach that produces long binary vectors, either by directly producing them or by a special encoding. This allows a convenient method for converting images into hyperdimensional representations that naturally work with symbolic reasoning systems, such as fuzzy logic systems.

Consider **Figure 1**, which demonstrates how hyperdimensional vectors could be used to convert data driven systems of different modalities to a shared space of long binary vectors. Once mapped to such a space, where distance between mappings is meaningful, it is clear that the

binary space is a purely symbolic representation of both the input to each data-driven system and their respective output. Once symbolically represented, operations performed in the hyperdimensional space can further map vectors to more complicated representations: in this particular instance, the symbolic concept of a dog. In a more complicated system, the entire hyperdimensional space can be overlaid by a knowledge graph, fuzzy logic system, VSA, or any other symbolic reasoning system. We largely focused on how to achieve this mapping from an external learning system to a binary space and, consequently, how to symbolically “fuse” different modalities together to get a better symbolic representation of a real-world, data-driven concept. Our experimental results indicate impressive improvement in terms of performance, when fusing the outputs of multiple data-driven models, at little to no computational cost. The structure of this fusing allows for more models to be added or removed as desired without requiring expensive computation or retraining.

The remainder of this article is structured as follows. First, in section 2, we have provided necessary background information on hyperdimensional computation. Next, in section 3, we have discussed related work and results that are pertinent to this article. Then, in section 4, we have presented the architecture of a system that could achieve the desired functionality shown in **Figure 1** and shown how it can be trained and used at testing time. Of particular importance is the notion of the Hyperdimensional Inference Layer, which can effectively fuse



symbolic representations in the hyperdimensional space. In section 5, we have outlined an experiment to test how well such an architecture would work in practice. Notably, we have constrained ourselves to using image hashing networks and have shown that not only can our architecture effectively fuse the outputs of different networks together in the hyperdimensional space but also that the mere usage of hyperdimensional vectors as a memory mechanism can improve performance as well. Naturally, in section 6, we have shown the results of these experiments. Finally, in section 7, we have discussed our results and outlined the pros/cons of using hyperdimensional vectors to fuse learning systems together at a symbolic level as well as what future work is necessary.

2. BACKGROUND INFORMATION

We first covered some of the relevant properties of hyperdimensional vectors for comprehension, as discussed in Kanerva (2009). Hyperdimensionality arises in binary vectors of sufficiently long length, usually on the order of 10,000 bits. Given two random vectors a and b from $\mathbb{B}^n = \{0, 1\}^n$ for large n , their overlap in bits has a high probability of being close to the expected value of $n/2$ with a standard deviation of $\sqrt{n/4}$. Therefore, two randomly selected vectors will overwhelmingly have a Hamming Distance of $n/2$; in this case, we can say the vectors are uncorrelated.

Two vectors a and b can be bound together by using the *exclusive-or* (XOR) operation, which we will represent with $*$ symbol:

$$c = a * b \quad (1)$$

Trivially, given one of the vectors, say a , we can unbind c to get b :

$$a * c = a * (a * b) = (a * a) * b = b \quad (2)$$

Suppose that a and b represent symbolic concepts; binding them with Equation (1) and unbinding with Equation (2) produces a new symbolic concept c that is deconstructed to its atomic symbols. Additionally, it is trivial that the Hamming Distance between two vectors is preserved when both are mapped by either another XOR with m or by a common permutation Π of bits:

$$|a_m * b_m| = |(a * m) * (b * m)| = |a * b| \quad (3)$$

$$|\Pi a * \Pi b| = |\Pi(a * b)| = |a * b| \quad (4)$$

For our purposes, permutation and XOR are used interchangeably as “multiplication” operations. In order to create a more sophisticated and structured vector, we required an “addition” operation. We primarily concerned ourselves with the “consensus sum,” where each bit of the resultant vector is set to be the bit value that appears more often in that component across the terms:

$$+_c(\{a_1, a_2, \dots, a_l\}) = a_1 +_c a_2 \dots +_c a_l = a_c \quad (5)$$

where for a count z of 0's across the l terms:

$$a_c^i = \begin{cases} 0 & z > l/2 \\ 1 & z < l/2 \\ \text{random} & z = l \end{cases} \quad (6)$$

However, if permutation is used for multiplication, it is valid to use XOR for addition. For either $*$ or $+_c$ as the $+$ operator, a sequence of symbolic information $A = A_1 A_2 \dots A_l$ can be represented as

$$\begin{aligned} a &= \Pi(\dots(\Pi(\Pi a_1 + a_2) + a_3) + \dots) + a_l \\ &= \Pi^{l-1} a_1 + \Pi^{l-2} a_2 + \dots + \Pi a_{l-1} + a_l \end{aligned} \quad (7)$$

where a_i are vector representations of corresponding A_i and Π is a permutation that represents the sequence. When using XOR, subsequences can be removed, replaced, or extended by constructing them and XOR-ing with a .

Finally, a *record* r of fields $f = [f_1, f_2, \dots, f_l]$, and their values $v = [v_1, v_2, \dots, v_l]$ can be constructed symbolically by binding each f_i with its corresponding v_i using Equation (1) and summing the result with Equation (5):

$$r = v \cdot f = v_1 * f_1 +_c \dots +_c v_i * f_i +_c v_l * f_l \quad (8)$$

Given a value v_x , a record r can be probed by performing an XOR and finding the f_j with the smallest Hamming Distance, thus checking the existence of a field:

$$\min_j |(v_x * r) * f_j| \quad (9)$$

A similar approach can be done to approximately recover the value of a field:

$$\min_j |(f_x * r) * v_j| \quad (10)$$

When the bits of a probe do not correlate with a term in r , the term collapses into a noise vector, whereas the term that does correlate will produce a signal that approximately undoes the XOR binding.

3. RELATED WORK

Our work is primarily an extension of HAP (Mitrokhin et al., 2019) in which drones are trained to predict their egomotion in 3-D space. A neuromorphic camera's events are converted into a time image slice of motion, represented as a sparse RGB image, whose pixel data is symbolically represented with a structured hyperdimensional vector. Raw RGB values are semantically embedded into the hyperdimensional space such that each color component is closer to its nearest values than further values. Possible velocity values are finely binned and likewise semantically embedded. A structured record m is constructed to associate the egomotion in one component velocity to another record containing all time image slices that fall into the same velocity bin, with Equation (8), at training time. Egomotion prediction is achieved purely by these memory units m via

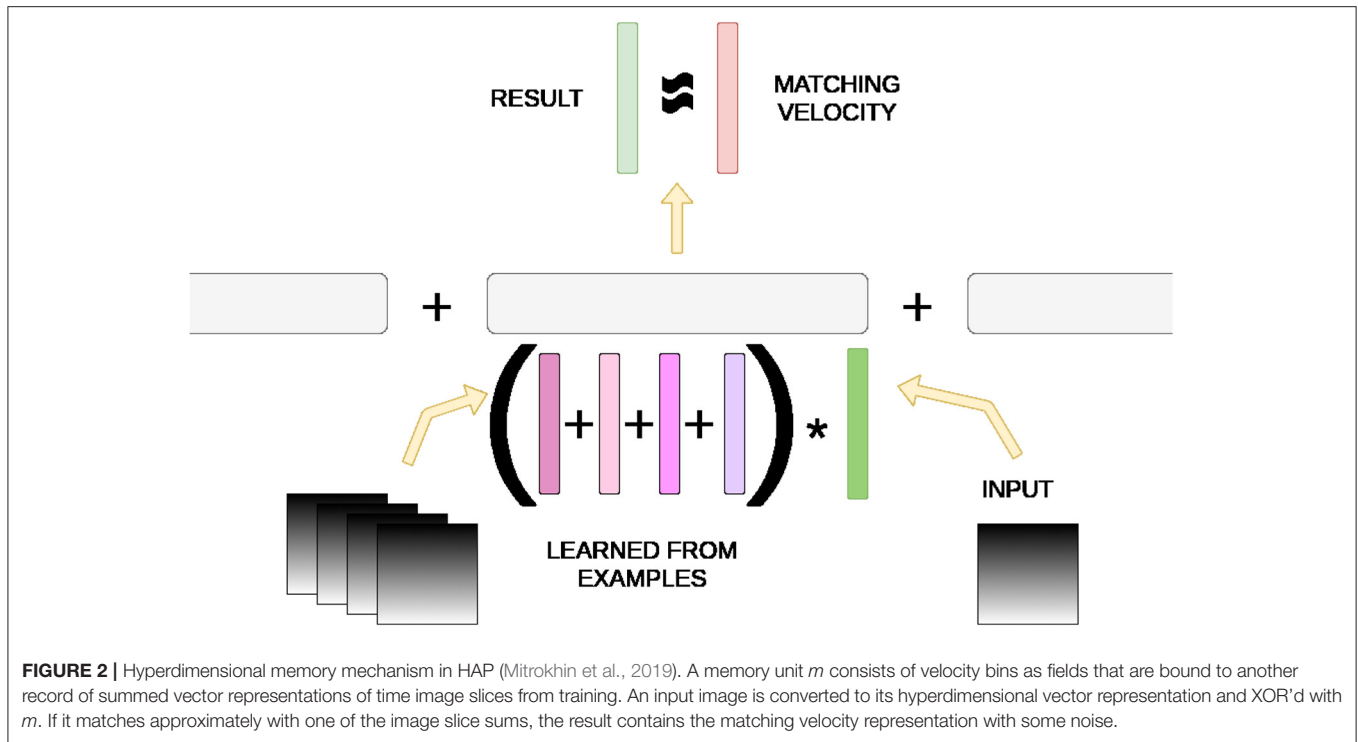


FIGURE 2 | Hyperdimensional memory mechanism in HAP (Mitrokhin et al., 2019). A memory unit m consists of velocity bins as fields that are bound to another record of summed vector representations of time image slices from training. An input image is converted to its hyperdimensional vector representation and XOR'd with m . If it matches approximately with one of the image slice sums, the result contains the matching velocity representation with some noise.

XORing a novel time slice image with m and probing the possible velocities to find the closest match. **Figure 2** demonstrates this process.

This method works because of the sparseness of pixel data in time image slices. The collection of time slices that are associated to a velocity bin average out to be representative of the motion changes the neuromorphic camera experienced. Surprisingly, this is sufficient to achieve neural-network-like performance, with a tiny fraction of memory, training samples, computation power, and training time of a neural approach. However, it is completely interpretable, can be trained online, and is effectively a symbolic reasoning system. Unfortunately, regular image data is too dense in information for this approach to work as implemented in HAP (Mitrokhin et al., 2019).

There exist other methods that have used hyperdimensional techniques to perform recognition (Imani et al., 2017) and classification (Moon et al., 2013; Rahimi et al., 2016; Imani et al., 2018; Kleyko et al., 2018). As with HAP (Mitrokhin et al., 2019), there have been other attempts to perform feature and decision fusion (Jimenez et al., 1999) or paradigms that can operate with minuscule amounts of resources (Rahimi et al., 2017). We differ from these approaches in that we try to assume as little about the model as possible except that it would be used in some form of classification for information that can be represented symbolically and modified with additional classifiers. Our results are a benchmark to see how much a hyperdimensional approach could facilitate a direct connection between ML systems and symbolic reasoning. On the solely symbolic representation and reasoning side, there exists relevant work on using cellular automata based hyperdimensional

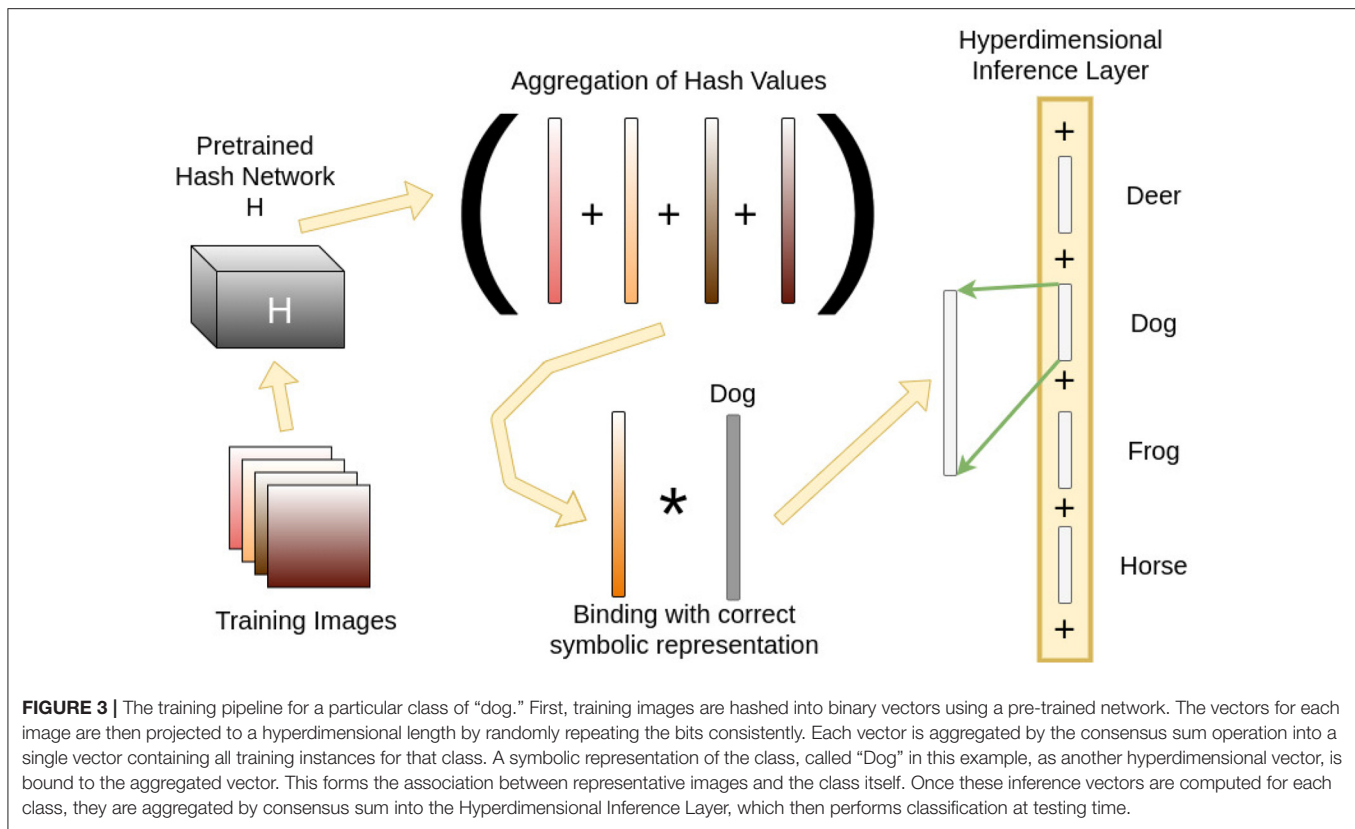
computing (Yilmaz, 2015). Some formulations based on real-valued vectors can also exhibit similar properties to long binary vectors so far as compositionality and decomposability is concerned (Summers-Stay et al., 2018).

4. ARCHITECTURE

Extending the model from HAP (Mitrokhin et al., 2019), the input vector is treated as any output from an ML system and the output velocity bins are now a symbolic representation of the output classes of the network. These would then feed in to a larger VSA system, that could feasibly be composed of other ML systems. Suppose that we have a pre-trained ML system, such as a Hashing Network, which can produce binary vectors as output to represent images.

4.1. Training the Hyperdimensional Inference Layer

For a classification task, during training time, training images are hashed into binary vector representations. These are aggregated with the consensus sum operation in Equation (5) across their corresponding gold-standard classes, and a random basis vector meant to symbolically represent the correct class is bound to the aggregate with Equation (1). The resultant vector now represents a *memory* containing all training instances observed but that are represented symbolically with appropriated hashed binary vectors that are projected into a hyperdimensional binary space by randomly permuting and assembling the hash vector. **Figure 3** shows this process when training to classify a “dog” in an image. This dog class is aggregated into a larger vector, once



again with the consensus sum operation in Equation (5), to produce a hyperdimensional vector containing similar memory vectors across the other classes. This is referred to as the Hyperdimensional Inference Layer (HIL), which then infers the correct class at testing time for a novel image.

4.2. Testing the Hyperdimensional Inference Layer

Once training is complete, classification of a novel image is relatively straightforward. An image is converted to a binary vector by the pre-trained hashing network. This vector is then projected into a hyperdimensional vector in the same manner as during training. Finally, the XOR between this vector and the HIL is computed. The Hamming Distance between the resultant vector and each of the class representations is measured. The class vector with the smallest Hamming Distance is selected as the correct classification. **Figure 4** shows this process in action.

4.3. Consensus With Multiple Models

One advantage of the hyperdimensional architecture for inference is how it can be easily manipulated. Of particular interest is when there are multiple models that can produce features in the form of hyperdimensional vectors for an input. Suppose we had several models, each with their own advantages. We can fuse their output together to form a consensus system that will consider each network's feature output before classification. We simply repeat the same method as we did for our classes but with symbolic identifiers for

which model aggregated which data. Prediction is done as before, probing each model's output with XOR and finding the closest matching network vector. **Figure 5** demonstrates how this pipeline would work.

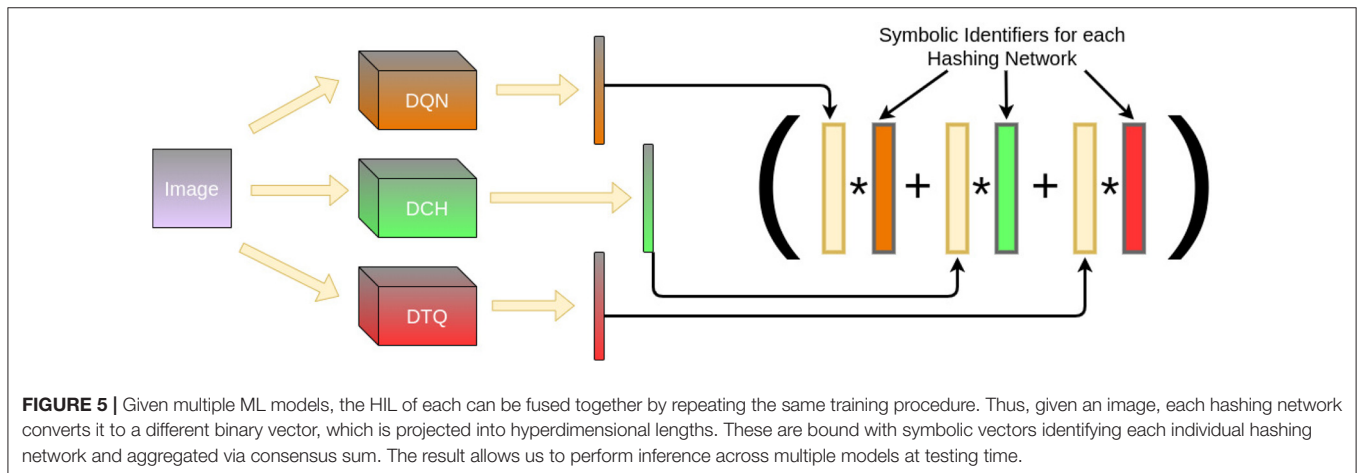
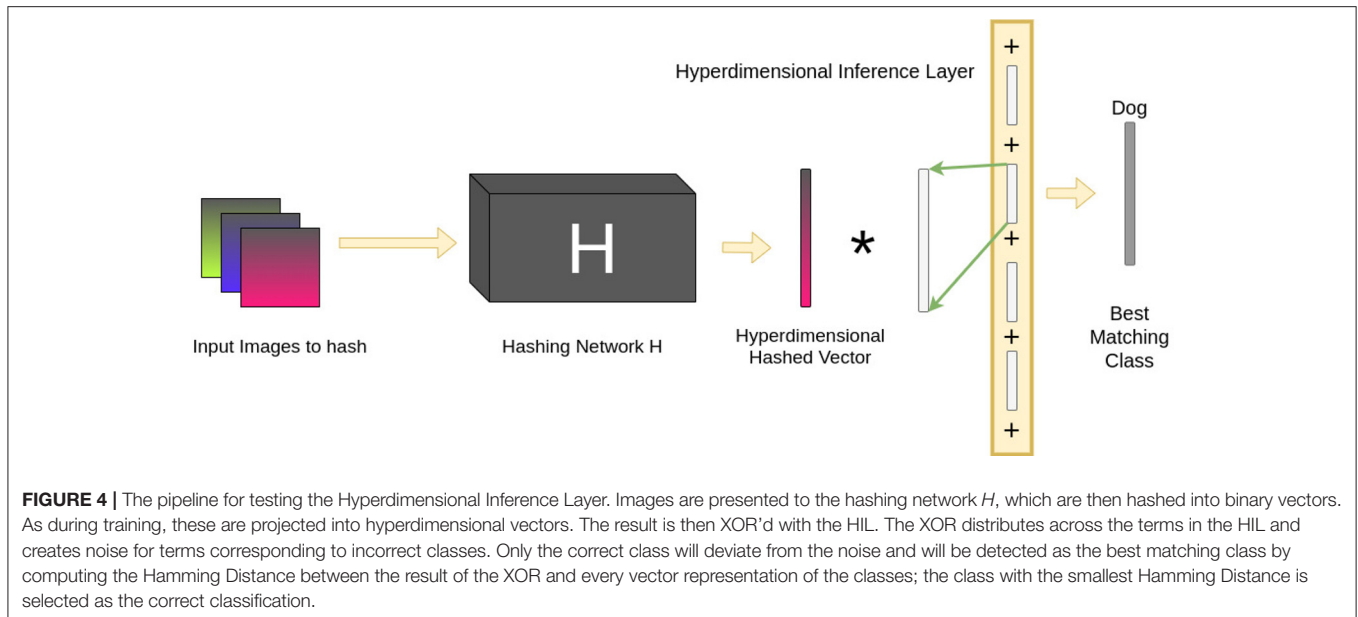
5. MATERIALS AND METHODS

The methodology, external systems, and datasets used for testing were as follows.

5.1. Methodology

To test how well hyperdimensional vectors can facilitate the mapping from the input/output of an ML system to a symbolic system, we required a model problem where it was possible to convert an ML result into hyperdimensional vectors. We studied the typical image classification problem but with hashing networks, as they directly convert raw images into binary vectors of variable length, which are used for classification and ranking based on Hamming Distance. This is simply done for convenience, as most neural methods do not produce binary vectors of such large length that are also rankable, and we did not want other methods for embedding real numbered vectors into binary spaces to affect the results. We utilized the *DeepHash*¹ library, which incorporates recent deep hashing techniques for image classification and ranking (Cao et al., 2016, 2017, 2018; Zhu et al., 2016; Liu et al., 2018). Our goal was to show that

¹Code repository available on GitHub at <https://github.com/thulab/DeepHash>.



an added layer of inference to the outputs of these methods with hyperdimensional computing allows us to convert their results into common length, hyperdimensional vectors, without losing performance. In fact, as we have seen, performance can even increase.

Two separate experiments were performed to evaluate how well a structure like the one shown in **Figure 1** would work in practice. Again, we limited ourselves to visual learning systems for simplicity, though there is no reason for such a limitation in practice.

1. We first tested how well a hyperdimensional representation of a given hashing network's output can work with a HIL. That is, does the inclusion of a HIL (and by extension, hyperdimensional representations) obfuscate the classification, thereby worsening performance, or does it perhaps improve the performance? In theory, the system should not do worse. However, the nature of HIL's structure may enable a better *memorization* of training examples.

We trained individual Hash Networks to perform image classification and then compared their performance with and without a HIL. Performance is tracked by comparing the F1 score for classification to the number of training iterations for the Hash network or epochs. We were also interested in how the HIL affects the F1 score as the Hamming Distance threshold for similarity increases.

2. Additionally, we studied whether the HIL could improve the overall performance of our Hash Networks if we fused them at the symbolic level of their outputs, using a HIL, as shown in **Figure 5**. We designed an experiment where all networks used in our first experiment are combined together by fusing their individual HIL into a new HIL. The idea asisw that, individually, these Hash Networks have different strengths and weaknesses based on their formulation. When fused into a HIL, each contributes toward the overall classification result, allowing the best matching classification across all models simultaneously.

5.2. External Systems

We used three of the image hashing networks from DeepHash in our experiments. In the following sections, we have described and outlined each one individually. In general, these networks use features provided by another system and compute hashes based on features extracted from the images into compact codes for image retrieval and classification. Additionally, we built our hyperdimensional inference layer by using the open source framework pyhdc² library, as used in HAP (Mitrokhin et al., 2019), which contains basic, but very efficiently implemented, operations for hyperdimensional computing and representing hyperdimensional vectors. Finally, AlexNet (Krizhevsky et al., 2012) features pre-trained on ImageNet (Deng et al., 2009) are used in the DeepHash pipeline and are available for download from the GitHub repository.

5.2.1. Deep Quantization Network (DQN)

The *Deep Quantization Network* (DQN) is a hashing-by-quantization network used for efficient image retrieval (Cao et al., 2016). The system supervises its hashing and allows statistical minimization of quantization errors from hand-crafted or machine learned features in a step separate from what traditional quantized hashing networks did prior. DQN formally controls this quantization error. The system is composed of four main subsystems:

1. Multiple convolution-pooling layers that capture deep image representations.
2. A fully connected layer that bottlenecks deep representations and projects them into an optimal lower dimensional representation for hashing.
3. A pairwise cosine layer for learning similarity preservation.
4. The quantization loss product that controls the quality of the hash and quantizes the bottleneck representations.

5.2.2. Deep Cauchy Hashing Network (DCH)

The *Deep Cauchy Hashing Network* (DCH) seeks to improve hash quality by penalizing similar image pairs having a Hamming Distance bigger than the radius specified by the hashing network (Cao et al., 2018). The authors argue that hashing networks tend to concentrate related images within a specified Hamming ball due to mis-specified loss function. By penalizing the network for when this happens with a pairwise cross-entropy loss based on a Cauchy distribution, the rankings become stronger.

5.2.3. Deep Triplet Quantization Network (DTQ)

The *Deep Triplet Quantization Network* (DTQ) further improves hashing quality by incorporating similarity triplets into the learning pipeline. By a new triplet selection approach, Group Hard, triplets are selected randomly from each image group that are deemed to be “hard.” Binary codes are further compacted by use of triplet quantization with weak orthogonality at training time.

5.3. Datasets

Evaluations of the hashing networks by themselves and with the hyperdimensional inference layer are performed on the CIFAR-10 standard dataset (Krizhevsky and Hinton, 2009) and the NUSWIDE_81 dataset (Chua et al., 2009), which contains tagged Flickr images with 81 concepts for classification.

6. RESULTS

In the following sections, we present the results of our evaluation of the hyperdimensional inference layers in both experiments.

6.1. Hyperdimensional Inference Layer Results

To test the capabilities of the hyperdimensional inference layers in preserving the output of ML models when transformed into hyperdimensional vectors, we compared the performance of each hashing network individually vs. the performance when the hyperdimensional inference layer is added to the hashing network, as shown in Figures 3, 4.

6.1.1. Results for CIFAR-10

Figure 6 compares the F1 scores of each hashing network with and without the HIL on CIFAR-10. The left column shows performance across iterations of network training (DTQ shows epochs instead). The threshold for Hamming Distance to search in is set to 2 (out of 128 bit vectors) for the baseline networks. For HIL results, as the vectors are hyperdimensional, the threshold is set to be proportionally that many bits out of 8,000. In all cases, the HIL improves performance greatly and with less iterations/epochs. In the right column, the F1 score is shown for successively more lax Hamming Distances in both methods, taking the best matching vector in a Hamming ball of that size. In the case of hyperdimensional vectors for the HIL, the size is once again proportional to 8,000 bit long vectors. For each baseline hashing network, there is clearly an optimal Hamming Distance to use. This is not the case for HIL, where it plateaus in each case for any distance smaller than the peak. As the number of bits increase, the performance quickly degrades to be more in line with the hashing network.

6.1.2. Results for NUSWIDE-81

Figure 7 compares the F1 scores of each hashing network with and without the HIL on NUSWIDE-81. As with CIFAR-10, the left column shows the performance of each hashing network across iterations of network training. The threshold for Hamming Distance is once again set to 2 bits out of 128 for the baseline networks. For HIL, the distance is proportionally scaled to hyperdimensional lengths in 8,000. Once again, in all cases the HIL greatly improves the F1 score. In the right column, the F1 score is shown for successively more lax Hamming Distances in both methods, retrieving the best match in the Hamming ball of that size. In the case of hyperdimensional vectors, the distances are scaled up to the appropriate values. For each baseline hashing network, there is

²Code repository available on GitHub at <https://github.com/ncos/pyhdc>.

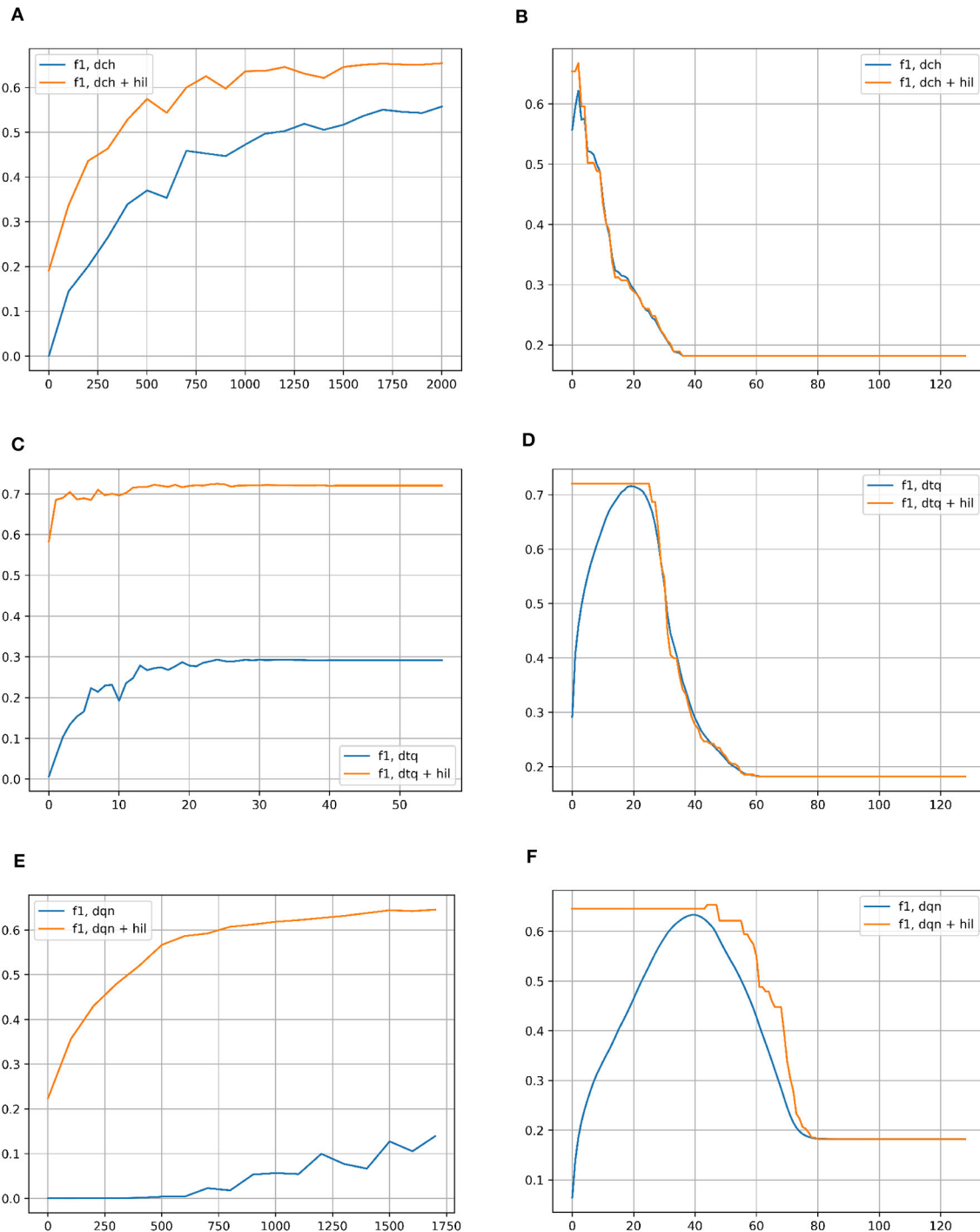


FIGURE 6 | (A) F1 score for classification on the CIFAR-10 dataset with DCH with and without the HIL, as a function of the number of iterations of training of the DCH network. **(B)** F1 score for classification on the CIFAR-10 dataset with DCH with and without the HIL, as a function of the Hamming Distance for classification. The networks are fully trained to the end point shown in subplot **(A)**. **(C)** F1 score for classification on the CIFAR-10 dataset with DTQ with and without the HIL, as a function of the number of iterations of training of the DTQ network. **(D)** F1 score for classification on the CIFAR-10 dataset with DTQ with and without the HIL, as a function of the Hamming Distance for classification. The networks are fully trained to the end point shown in subplot **(C)**. **(E)** F1 score for classification on the CIFAR-10 dataset with DQN with and without the HIL, as a function of the number of iterations of training of the DQN network. **(F)** F1 score for classification on the CIFAR-10 dataset with DQN with and without the HIL, as a function of the Hamming Distance for classification. The networks are fully trained to the end point shown in subplot **(E)**. Baseline networks are shown in blue, while the same network with a HIL appended to the end is shown in yellow. Note that in the left column of subplots, the Hamming Distance for classification is set to 2 for inlier/outlier count. The left column of results show that the HIL boosts the speed at which the network trains,

(Continued)

FIGURE 6 | achieving a higher performance in far fewer iterations of expensive network training. As the HIL adds negligible overhead in memory/computation time, there is no downside to using a HIL. The right column of results show that the HIL prevents the need for searching for an optimal Hamming Distance threshold to classify with, as it supercedes peak performance of the network right away for the lowest possible distance thresholds. After peak performance of the baseline network, larger Hamming Distance thresholds eventually decay to the performance of the baseline.

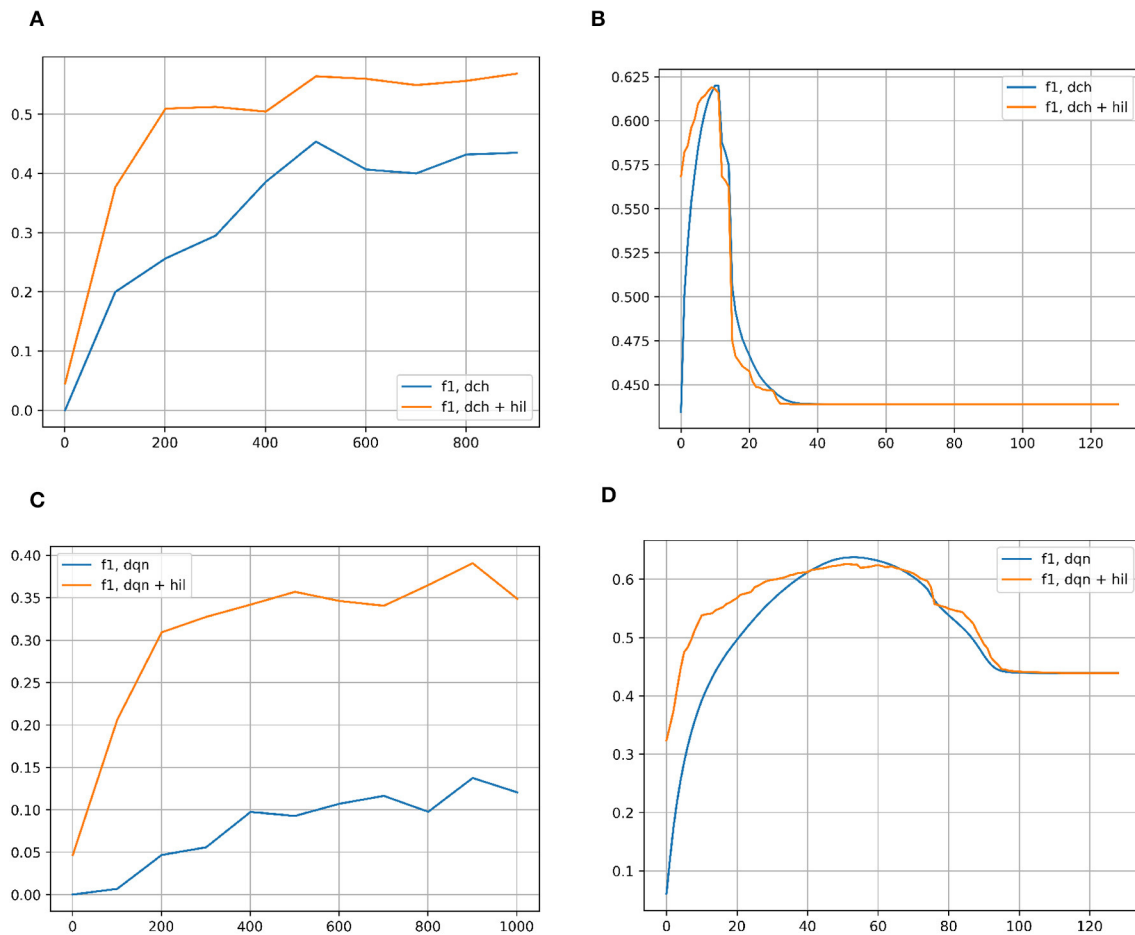


FIGURE 7 | (A) F1 score for classification on the NUSWIDE-81 dataset with DCH with and without the HIL, as a function of the number of iterations of training of the DCH network. **(B)** F1 score for classification on the NUSWIDE-81 dataset with DCH with and without the HIL, as a function of the Hamming Distance for classification. The networks are fully trained to the end point shown in subplot **(A)**. **(C)** F1 score for classification on the NUSWIDE-81 dataset with DQN with and without the HIL, as a function of the number of iterations of training of the DQN network. **(D)** F1 score for classification on the NUSWIDE-81 dataset with DQN with and without the HIL, as a function of the Hamming Distance for classification. The networks are fully trained to the end point shown in subplot **(C)**. Baseline networks are shown in blue, while the same network with a HIL appended to the end is shown in yellow. Note that in the left column of subplots, the Hamming Distance for classification is set to 2 for inlier/outlier count. Results for DTQ are omitted for incompatibility with NUSWIDE-81. We largely get the same results in the left column as with CIFAR-10, showing an improvement in performance versus training iterations when an HIL is appended to the end of the baseline network, which adds negligible memory/computation costs. In the right column of results, the HIL differs from CIFAR-10's results in that there is a peak to the performance of the HIL enhanced network. This is likely due to NUSWIDE-81 being designed for the task of web image annotation and retrieval.

clearly an optimal Hamming Distance to use, though it is much less pronounced with HIL. In all cases, it is safer to use a smaller Hamming Distance rather than a larger one, except near the optimal values.

6.2. Results for the Consensus Architecture

We tested the capability of hyperdimensional computing to fuse the results of different models at the vector-symbolic level. This setup allows to compensate for the shortcomings

of the individual models and give a more robust result - a desirable property of hyperdimensional representations. We tested the consensus pipeline on all three hashing networks and on CIFAR-10's full dataset, with fully trained hashing networks and HILs for each. The F1 scored increased to 0.79, ~10% more than the scores any of the models achieved individually with HIL, as seen in **Figure 6**. This confirms our suspicion that direct fusion at the symbolic level gives far more robust results.

6.3. Summary of Experimental Results

Our experiments indicate no performance downside to adding an HIL to an existing, Deep Hash Network. Indeed, it seems that the HIL enables better results with fewer epochs and even improves the F1 score. Furthermore, fusion of multiple networks into a single HIL increased the F1 score greatly above any of the individual networks, even with an HIL. Since each Hash Network formulation differs significantly from each other, one network might be better suited at hashing particular information. We surmise the improvement of performance is because the robustness of the HIL allows each network to naturally contribute its classification to the overall classification decision in a consensus-like fashion.

It should be noted that hyperdimensional computations are very fast. The `pyhdc` package is designed to perform these computations very efficiently. As a result, the addition of the HIL, in either experiment, is negligible in terms of extra computations and execution time. This is in line with previous results shown in HAP (Mitrokhin et al., 2019), where in a matter of milliseconds the HIL can be trained, retrained from scratch, and even perform classification, on a standard CPU processor. In our results, the HIL also incurred milliseconds of additional runtime. This further indicates that there is virtually no downside to adopting the hyperdimensional approach presented in our architecture.

7. DISCUSSION

Hyperdimensional computing has many attractive properties. Our results confirm the notion that hyperdimensional representations can be useful in VSA and symbolic reasoning systems. It is also important to note that hyperdimensional vectors have not yet been effectively used to represent dense RGB images in prior work. This potentially opens up new avenues for combining symbolic reasoning and ML methods. Hyperdimensional representations produced by converting the output of deep hashing networks into symbolic inference structures allows the use of fuzzy logic systems, of which the use of HILs in our experiments are a simple example of. Since HIL structures can be fused across different modalities, this increases the robustness and interpretability of the inference process. We have shown the potential advantages of multi-modal fusion in the HIL by combining three separately trained, differently constructed deep hashing networks without the need of any additional training or oversight, improving the overall result. This is despite the fact that each model is successively more state-of-the-art, meaning that there is no catastrophic loss in integrating newer models into the inference system as more are developed.

Although the results so far are quite interesting and point to a potential future of hyperdimensional computing in the marriage of ML and symbolic reasoning systems, there are still many drawbacks to the approach we have presented. First of all, it would be preferable to use non-hashing (or perhaps even non-supervised) networks to bootstrap our system, as these tend to perform much better than hashing methods. However, this would require the ability to convert embeddings in a more

sophisticated neural system into corresponding binary vectors. Special quantization methods may need to be developed to facilitate this in future work in order to fully take advantage of hyperdimensional representations.

It is clear that more work is required to fully integrate hyperdimensional representations into ML systems. Specifically, these need to be more compliant to deep representations of features. There are many avenues of future research that can improve upon these limitations, especially in regard to special conversion between deep features in different modalities, such as text, and images. On the symbolic reasoning side, our results do not produce a full-scale, fully realized symbolic system. For example, **Figure 1** would indicate that, given the high likelihood of detection of a dog, the system could reason that there is a high likelihood that what is currently observed likes to bark, has four legs, and loves to wag its tail. However, it is not clear how this linguistic knowledge would be incorporated into the associated hyperdimensional space. One can imagine knowledge graph like structures overlaying the hyperdimensional space, or perhaps more sophisticated structures, but it is not readily clear what the best formulation is.

Furthermore, we must point out some of the drawbacks of using hyperdimensional representations to facilitate a connection between data-driven systems and symbolic reasoning systems:

- We have the necessary requirement that data-driven systems can be readily converted into long binary vectors. This is a severe restriction, as most state-of-the-art methods naturally use real-valued computations. Most neural methods produce samples on complex manifolds that may be difficult to effectively map to hyperdimensional vectors. Thus, there is a need for a general technique to project real-valued embeddings from data-driven systems to binary spaces. As a result, real-value hyperdimensional vectors may be better suited to certain tasks (Summers-Stay et al., 2018; Sutor et al., 2019).
- Along the same lines, many modern-day symbolic reasoning systems also rely on real-value computations or representations, especially when data driven. New methods would have to be developed to work with more sophisticated systems.
- While hyperdimensional vector representations of different modalities can be embedded effectively into a common space, they may also require a nearest neighbor lookup when looking for similar, known concepts. This may become expensive when the hyperdimensional space contain many concepts. In order to maintain that data of a particular modality is closer to other examples of that modality, it may be necessary to adopt an approach that facilitates this, such as in Sutor et al. (2018).

DATA AVAILABILITY STATEMENT

The datasets analyzed for this study can be found on the DeepHash project page (<https://github.com/thulab/DeepHash>). The `pyhdc` library code used in this work can be found in <https://github.com/ncos/pyhdc>.

AUTHOR CONTRIBUTIONS

AM contributed to the experiments, evaluations, plots, and text of the manuscript. PS contributed to experiments, the text of the manuscript, and illustrations. DS-S, CF, and YA contributed to the text of the manuscript. All authors contributed to the conceptual ideas at the heart of this research.

REFERENCES

- Cao, Y., Long, M., Liu, B., and Wang, J. (2018). "Deep cauchy hashing for hamming space retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT), 1229–1237. doi: 10.1109/CVPR.2018.00134
- Cao, Y., Long, M., Wang, J., and Liu, S. (2017). "Deep visual-semantic quantization for efficient image retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 1328–1337. doi: 10.1109/CVPR.2017.104
- Cao, Y., Long, M., Wang, J., Zhu, H., and Wen, Q. (2016). "Deep quantization network for efficient image retrieval," in *Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, AZ).
- Chua, T.-S., Tang, J., Hong, R., Li, H., Luo, Z., and Zheng, Y. (2009). "NUS-wide: a real-world web image database from National University of Singapore," in *Proceedings of the ACM International Conference on Image and Video Retrieval* (Santorini), 1–9. doi: 10.1145/1646396.1646452
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). "Imagenet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (Miami, FL: IEEE), 248–255. doi: 10.1109/CVPR.2009.5206848
- Imani, M., Huang, C., Kong, D., and Rosing, T. (2018). "Hierarchical hyperdimensional computing for energy efficient classification," in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)* (San Francisco, CA: IEEE), 1–6. doi: 10.1109/DAC.2018.8465708
- Imani, M., Kong, D., Rahimi, A., and Rosing, T. (2017). "Voicehd: hyperdimensional computing for efficient speech recognition," in *2017 IEEE International Conference on Rebooting Computing (ICRC)* (McLean, VA: IEEE), 1–8. doi: 10.1109/ICRC.2017.8123650
- Jimenez, L. O., Morales-Morell, A., and Creus, A. (1999). Classification of hyperdimensional data based on feature and decision fusion approaches using projection pursuit, majority voting, and neural networks. *IEEE Trans. Geosci. Rem. Sens.* 37, 1360–1366. doi: 10.1109/36.763300
- Kanerva, P. (2009). Hyperdimensional computing: an introduction to computing in distributed representation with high-dimensional random vectors. *Cogn. Comput.* 1, 139–159. doi: 10.1007/s12559-009-9009-8
- Kleyko, D., Rahimi, A., Rachkovskij, D. A., Osipov, E., and Rabaey, J. M. (2018). Classification and recall with binary hyperdimensional computing: tradeoffs in choice of density and mapping characteristics. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 5880–5898. doi: 10.1109/TNNLS.2018.2814400
- Krizhevsky, A., and Hinton, G. (2009). *Learning Multiple Layers of Features From Tiny Images*. The University of Toronto.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (Lake Tahoe), 1097–1105.
- Liu, B., Cao, Y., Long, M., Wang, J., and Wang, J. (2018). "Deep triplet quantization," in *Proceedings of the 26th ACM International Conference on Multimedia* (Seoul), 755–763. doi: 10.1145/3240508.3240516

ACKNOWLEDGMENTS

The support of Northrop Grumman Mission Systems University Research Program, of ONR under grant award N00014-17-1-2622, of the Brin Family Foundation and the support of the National Science Foundation under grants BCS 1824198 and CNS 1544787 were gratefully acknowledged.

- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. Scottsdale, AZ: International Conference on Learning Representations (ICLR).
- Mitrokhin, A., Sutor, P., Fermüller, C., and Aloimonos, Y. (2019). Learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception. *Sci. Robot.* 4:eaaw6736. doi: 10.1126/scirobotics.aaw6736
- Moon, S., Berster, B.-T., Xu, H., and Cohen, T. (2013). "Word sense disambiguation of clinical abbreviations with hyperdimensional computing," in *AMIA Annual Symposium Proceedings*, Vol. 2013 (Washington, DC: American Medical Informatics Association), 1007.
- Pennington, J., Socher, R., and Manning, C. D. (2014). "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha), 1532–1543. doi: 10.3115/v1/D14-1162
- Rahimi, A., Datta, S., Kleyko, D., Frady, E. P., Olshausen, B., Kanerva, P., et al. (2017). High-dimensional computing as a nanoscale paradigm. *IEEE Trans. Circuits Syst. I Reg. Pap.* 64, 2508–2521. doi: 10.1109/TCSI.2017.2705051
- Rahimi, A., Kanerva, P., and Rabaey, J. M. (2016). "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design* (San Francisco, CA), 64–69. doi: 10.1145/2934583.2934624
- Summers-Stay, D., Sutor, P., and Li, D. (2018). Representing sets as summed semantic vectors. *Biol. Inspired Cogn. Archit.* 25, 113–118. doi: 10.1016/j.bica.2018.07.002
- Sutor, P., Aloimonos, Y., Fermüller, C., and Summers-Stay, D. (2019). "Metaconcepts: isolating context in word embeddings," in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* (San Jose, CA: IEEE), 544–549. doi: 10.1109/MIPR.2019.00110
- Sutor, P., Summers-Stay, D., and Aloimonos, Y. (2018). "A computational theory for life-long learning of semantics," in *International Conference on Artificial General Intelligence* (Prague: Springer), 217–226. doi: 10.1007/978-3-319-97676-1_21
- Yilmaz, O. (2015). Symbolic computation using cellular automata-based hyperdimensional computing. *Neural Comput.* 27, 2661–2692. doi: 10.1162/NECO_a_00787
- Zhu, H., Long, M., Wang, J., and Cao, Y. (2016). "Deep hashing network for efficient similarity retrieval," in *Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, AZ).

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Mitrokhin, Sutor, Summers-Stay, Fermüller and Aloimonos. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



In Search of Trustworthy and Transparent Intelligent Systems With Human-Like Cognitive and Reasoning Capabilities

Nikhil R. Pal*

Indian Statistical Institute, Electronics and Communication Sciences Unit, The Centre for Artificial Intelligence and Machine Learning, Calcutta, India

OPEN ACCESS

Edited by:

Georg Martius,
Max Planck Institute for Intelligent
Systems, Germany

Reviewed by:

Andreas Holzinger,
Medical University of Graz, Austria
Yoichi Hayashi,
Meiji University, Japan

*Correspondence:

Nikhil R. Pal
nrpal59@gmail.com

Specialty section:

This article was submitted to
Computational Intelligence in
Robotics,

a section of the journal
Frontiers in Robotics and AI

Received: 30 November 2018

Accepted: 07 May 2020

Published: 19 June 2020

Citation:

Pal NR (2020) In Search of
Trustworthy and Transparent
Intelligent Systems With Human-Like
Cognitive and Reasoning Capabilities.
Front. Robot. AI 7:76.
doi: 10.3389/frobt.2020.00076

At present we are witnessing a tremendous interest in Artificial Intelligence (AI), particularly in Deep Learning (DL)/Deep Neural Networks (DNNs). One of the reasons appears to be the unmatched performance achieved by such systems. This has resulted in an enormous hope on such techniques and often these are viewed as all—cure solutions. But most of these systems cannot explain why a particular decision is made (black box) and sometimes miserably fail in cases where other systems would not. Consequently, in critical applications such as healthcare and defense practitioners do not like to trust such systems. Although an AI system is often designed taking inspiration from the brain, there is not much attempt to exploit cues from the brain in true sense. In our opinion, to realize intelligent systems with human like reasoning ability, we need to exploit knowledge from the brain science. Here we discuss a few findings in brain science that may help designing intelligent systems. We explain the relevance of transparency, explainability, learning from a few examples, and the trustworthiness of an AI system. We also discuss a few ways that may help to achieve these attributes in a learning system.

Keywords: Artificial Intelligence, Deep Neural Networks, explainable AI, trustworthy AI, machine learning, sustainable AI, deep learning

1. INTRODUCTION

In this article, we focus on a few issues that need attention to realize AI systems with human-like cognitive and reasoning abilities. For some of these issues we are not in a position to suggest precise solutions, but in some cases we shall provide cues or pointers to areas where we may look for to find possible solutions.

1.1. Learning and Intelligence

We begin with a definition of learning. Oxford dictionary¹ defines *learning* as “The acquisition of knowledge or skills through study, experience, or being taught.” It defines *intelligence* as “The ability to acquire and apply knowledge and skills.” The Cambridge dictionary², on the other hand, defines intelligence as “the ability to learn, understand, and make judgments or have opinions that are based on reason.”

Thus, learning and intelligence are intimately related. In our view, judicious applications of the knowledge learnt lead to intelligence. Learning, in the context of machine, raises several

¹<https://en.oxforddictionaries.com/definition/learning>, June 2018.

²<https://dictionary.cambridge.org/dictionary/english/intelligence>, June 2018.

fundamental questions: What to learn (extraction of knowledge), how to learn (representation), and how to use the same. Today to an AI researcher, learning usually means extracting knowledge from data (explicitly or implicitly) and then applying that knowledge to make decisions. Here the word “data” has been used in a broader sense, where data can come from observations/measurements or from human beings. Some desirable attributes of an intelligent system are: ability to learn (from data/experience), understand, make judgment, reason, and apply what has been learnt to unknown situations. It should also know when to refrain from making a judgement.

1.2. Deep Learning and Artificial Intelligence

At present Deep Learning (DL) (LeCun et al., 1998; Krizhevsky et al., 2012) is probably the most successful vehicle for designing AI systems. It is fantastic in learning from data and making decisions but it is almost like a black box - no transparency. We shall see that these systems may fail miserably in cases where other systems possibly would not.

But what is deep learning? Before defining DL, it is probably natural to introduce what Machine Learning is. Arthur Samuel in 1959 coined the term “Machine Learning” (Samuel, 1959) in the context of a machine playing the game of Checkers. He wrote a checker program that could play against a human player as well as it could play against itself. Consequently, the program could play many games in a short time and by this process could learn the game better than its opposition. This gives us some idea of what machine learning is. More recently, Mitchell in his book “Machine Learning” (Mitchell, 1997) gave a formal definition of (Machine) Learning: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .” In the checker program of Samuel (1959), T is playing of checkers, P is percentage of times the ML system wins, E is either playing with itself or with a human, which generates a sequence of moves (data or experience). Similarly, for a machine learning system to classify images we can have: T as the task of classification, P as the misclassification error or square loss or cross-entropic loss, and E as a set of labeled image data. Equivalently, we can say that an ML system finds a function in a given class of functions (defined by the associated learning model such as neural network, a decision tree, a polynomial regression function) to fulfill a given purpose (say the task of classification) as best as it can (typically optimizing some criterion) based on experience (typically using a given set of training data). With this introduction to ML, we get back to Deep Learning. The basic idea of DL is to learn complex concepts in terms of a hierarchy of concepts, where each concept in the hierarchy is defined in terms of its immediate lower level simpler concepts. The hierarchical organization allows representation of complicated concepts in terms of simpler concepts. If this hierarchical architecture is represented using a graph, the graph would be deep and hence it is called deep learning (Goodfellow et al., 2016). At present most DL systems are primarily based on multilayered neural

network architectures. The way a Deep Neural Network (DNN)/Convolutional Neural Network (CNN) works does not seem to have a strong relation to the way a human brain works. We shall discuss this issue later. However, DNN seems to enjoy a few properties: a large neural network with many layers, uses hierarchies of representation/abstraction, and gets better results with bigger models and bigger training data. Often such models have more free parameters than would be required to solve a given problem. This may lead to overfitting/memorization resulting in poor generalization. In order to increase the robustness of the system and to achieve better generalization to unseen data, various regularization techniques such as norm (L_1 or L_2 or both) of the weight parameters, dataset augmentation adding noise, drop outs, early stopping, and parameter sharing are used (Goodfellow et al., 2016).

1.3. Are We on the Right Track?

After the success of deep learning in many areas like image recognition and games (defeating Se-dol Lee, the best human GO player, by Google DeepMind's AlphaGo), often deep learning is viewed as an all-cure solution. Sometimes we get an impression that today AI is almost synonymous to DL. A Google search with ‘Neural Networks’ retrieves about 30,800,000 results (April 17, 2020), ‘Artificial Intelligence’ brings About 127,000,000 results, while ‘Deep Learning’ brings About 39,800,000 results. It clearly reveals the rapid growth of interest in deep learning. It is worth noting that all pages with “neural networks” may not be related to artificial neural networks. It cannot be questioned that DL is one of the most successful tools to realize AI systems in some specific areas. Although DNNs have demonstrated unbeatable performance in some applications, DNNs are found to recognize images completely “unrecognizable to humans” as human recognizable objects with a very high confidence (Nguyen et al., 2015). At the other extreme, with very minor changes in an image that are imperceptible to human eyes, DNNs are found to mislabel the image (Papernot et al., 2017). It raises a big question. Why? A DNN or any other intelligent system usually optimizes some objective functions with respect to a set of learnable parameters. Such a system usually can make correct decisions. But for DNNs (and many others) neither the architecture nor the decision making rule can help to explain the rationale for such a decision in terms of human understandable knowledge. Although we know precisely the computation done at different nodes, the system is not transparent because we do not know the semantics associated with each node or a group of nodes or the presence of causal relations, if any, between the nodes. It behaves like a black box. Moreover, while designing a DNN often the principle of Minimum Description Length (MDL) is not given its due importance, as a result a system may have more degrees of freedom than what is required, which may lead to inappropriate generalization. Given a finite data set, the MDL principle suggests to pick the most compact description (parameterization) of the model as well as the description of the data under that model. However, the current deep networks are far from that. For example, the VGG 19 network has about 150 M parameters (Canziani et al., 2016). One of the possible reasons is that small networks cannot be trained with

the present available techniques to achieve similar performance for large scale classification problems. However, large number of free parameters increases the expressibility of such networks and they can easily memorize data. For example, Zhang et al. (2017) demonstrated that networks like Inception and Alexnet can learn/memorize almost perfectly the CIFAR10 dataset with random permutation of labels. Of course, the test accuracy is no better than random chance. Another related important issue is that these systems are not equipped to deal with the open world nature of most decision making problems. In order to realize a transparent, accurate, and trustworthy system with human like reasoning abilities, in our view, there are few areas that we may need to focus on. Next we discuss a few such issues/areas.

2. WHAT'S NEXT?

2.1. Look Into the Brain

The human brain is one of the most complex systems in the known universe. In spite of tremendous efforts, our understanding about how brain learns is quite limited, yet a good amount is known. In order to have a learning system with reasoning abilities like a human being, in our view, we should try to exploit our knowledge from the brain science. The more we can mimic the brain, the higher is the chance that we shall be able to realize human-like attributes in an AI system. For example, functional imaging studies with cellular resolution (*in vivo* two-photon calcium imaging) revealed that in area 18 of cat visual cortex there are extremely ordered groups of neurons that respond to visual stimuli of a particular orientation. Neurons having opposite directional preferences for visual stimuli were separated with a high spatial precision in three dimension (Ohki et al., 2005). If we could explicitly incorporate such features in our computational NN, it would behave a little more like a biological system. To design a learning system, we can generalize this idea assuming that for different types of visual objects different clusters of neurons will respond. In this case, the network will be more transparent than the a conventional MLP. However, to the best of our knowledge, computational neural network models including DNNs do not exploit this, although, the response of some neuron or a group of neurons may be directionally oriented due to the training. In fact, it is possible to integrate the principle of self-organizing map along with a multilayered neural network so that different spatial clusters of neurons get activated by instances from different classes (Bandyopadhyay, 2018).

Usually the success of a class of DNNs is attributed to the unsupervised feature extraction, i.e., on the representation and abstraction of raw data. In a convolutional neural network (CNN), actually we compute cross-correlation and use max-pooling to achieve reduction of information with a hope to realize useful abstractions. There is no explicit cause-effect relation. But does it have anything to do with information processing in the brain of living beings? There are evidences supporting a complex and hierarchical feature representation along the ventral visual stream (DiCarlo et al., 2012; Kuzovkin et al., 2018). A CNN makes a hierarchy of complex feature representation for image recognition and from this point of view, a CNN has some similarity with the feature representation

in a biological vision system. We recall here a remark in DiCarlo et al. (2012) “*We do not yet fully know how the brain solves object recognition.*” Recently, analyzing the intracranial depth recordings from human subjects researchers suggested that the gamma activity matches the increasingly complex feature representation/abstraction in a deep convolutional neural network. In reality, the information abstraction in a biological system is much more complex. For example, in a natural scene there is a high degree of spatial and temporal correlation, and hence, representation of visual information at the level of photoreceptors would be highly inefficient because of tremendous redundancy. Dan et al. (1996) conducted some experiments with cats. They used movies of natural scenes as inputs and the responses of single lateral geniculate nucleus (LGN) neurons were recorded using a single tungsten or a multi-electrode array. They analyzed the temporal correlation and power spectra of the responses, which demonstrated that the natural visual information is temporally de-correlated at the lateral geniculate nucleus. Each LGN has six layers of neurons; it receives input from the retina and sends information to the primary visual cortex. Consequently, if an image recognition system exploits these concepts at the feature extraction stage, it would resemble more a biological system and is likely to yield better results. There have been several attempts to develop computational models of LGN neurons (Einevoll and Hanes, 2015; Sen-Bhattacharya et al., 2017). We need to investigate how these models can be adapted to develop machine learning systems for computer vision.

The primary visual cortex, the visual area 1 (V1), uses sparse code with minimal redundancy to efficiently represent and transmit information about the visual world and the non-classical receptive field plays an important role in this process (Vinje and Gallant, 2000). Sparse modeling is also used to identify “Sparse Connectivity Patterns” (SPCs) which make a parsimonious representation of brain functions (Eavani et al., 2015). At the level of neurons, we have sparse codes and at a higher level, we have these SPCs, which represent different system-level functions and relate to a set of spatially distributed, functionally synchronous brain regions. It is also known that the processing in the brain is distributed. In fact, an important characteristic of the brain is believed to be sparse distributed representation (SDR) (Ahmad and Hawkins, 2016; Hawkins, 2017; Pal, 2018a). In an SDR, at a given instant of time only some of the neurons may be active (producing an output of 1). If two SDRs have some overlap (have some common active neurons) then the two SDRs share some common attributes of the two concepts. The SDR characteristic is considered very important for biological intelligence. Since SDR makes an efficient representation of information and plays a key role in biological intelligence, incorporation of such ideas in designing learning systems is expected yield better AI systems.

2.2. Learning From a Small Sample

One of the claimed advantages of DNN is “more is better”—if we can train a big network with more data we can get a better performance. But does a human being need thousands of images to distinguish between various objects? Even a baby can learn to distinguish between a large number of animals with

just a few presentation of the animal images. More importantly, once a human learns the concept of “animals”, given a picture of a completely strange animal that has never been seen by the person, he/she can easily detect that it is an animal and will never mistake it to be a car or a human. This is an extension of the learned information by “common sense.” So human learning must be doing something different or at least something additional than what DNNs or other ML systems do. A child can learn many objects within a short time with few examples. We need to develop systems that can learn from a limited number/variety of examples as living beings do. Such a learning system should not need millions of examples with wide variations and a large number of cycling through the data. There have been a few attempts to learn a class from just one or a few examples (Fei-Fei et al., 2006; Maas and Kemp, 2009; Lake et al., 2011, 2013, 2015; Wang et al., 2019). This is known as Few-Shot Learning (FSL) (Wang et al., 2019). In FSL the learning is accomplished with a few instances with supervisory information for the target and it often exploits prior knowledge. In Lake et al. (2013), an interesting hierarchical Bayesian model based on compositionality and causality for one-shot learning has been proposed. Lake et al. (2013, 2015) combined principles of compositionality, causality, and learning to realize a Bayesian Program learning framework to learn visual concepts using just one example. This method can achieve human-level performance and has defeated two deep network models. Although such a method takes inspiration from cognitive science, it does not take into account how human brains learn concepts from just one or two examples. There have been a few more attempts to one-shot learning. Learning from limited examples is very important in areas like medical science where usually only a handful instances from the positive class are available. In such a case, the interactive machine learning (iML) with “human-in-the loop” could be very useful (Holzinger, 2016; Holzinger et al., 2016). There are three main advantages of iML: First, human intervention can reduce the search space drastically. Second, it can facilitate learning using a limited number of instances. Third, it can help to open the “black Box” to some extent. However, it has a few limitations also. For example, human knowledge often suffers from subjectivity and hence the resultant system may have subjective bias. Thus, an objective evaluation of such a system is difficult. Because of the incorporation of subjective knowledge, replicability of the system is also difficult. Interactive machine learning with human-in-the-loop appears to have good potential for learning with limited data but it needs more investigation. Wang et al. (2019) provides an excellent exposition of FSL explaining the advantages of FSL, the challenges associated with it, and how some of the challenges can be addressed.

There are theories suggesting that human beings recognize/learn images by segmenting them at deep concavity and then viewing the object as a set of simple geometric components characterized by attributes such as curvature, collinearity, and symmetry (Biederman, 1987). It is worth noting that this is consistent with the idea of SDR discussed earlier. This theory of recognition by components may be useful in learning from one or a few images. To a living being every image is not equally memorable and it is found that color or simple image

statistics, or object statistics do not make an image memorable (Isola et al., 2011). However, semantic information of objects and scenes is found to make an image memorable - a human being is able to remember such an image by just seeing once. This could be an important clue to design a learning system that can learn concepts with one or a few examples.

2.3. Explainable/Comprehensible/Transparent AI

Most computational learning algorithms including DL are “blind” in learning. They are good in decision making, but cannot explain why a decision is made. Time demands more emphasis on this aspect of learning. Using regularizers one can simplify (reduce the complexity) of a decision making system, which is good but it fails to bring the level of transparency that we would be happy with. We note here that transparency of an AI system is different from its ability to explain the rational behind a decision that it makes (explainability). Transparency refers to understanding of the semantics associated with the computation that goes on in the system. Comprehensibility/transparency is a fuzzy concept with grades of membership in [0,1]. For a black box system like an MLP the membership is zero while for a completely transparent system it is one. For example, a decision tree is highly transparent as long as the attributes are understandable properties and the depth of the tree is small. However, as the depth of the tree increases or if we use extracted features like the principle components, we start losing its comprehensibility. Yet, it will remain more comprehensible than, for example, a multilayer perceptron. Similarly, a fuzzy rule-based system is also transparent. A fuzzy rule based system uses rules of the form (Chen et al., 2012) : If the expression level of Gene X is HIGH and the expression level of Gene Y is MODERATE and the expression level of Gene Z is LOW then the patient is suffering from Neuroblastoma. Here HIGH, LOW and MODERATE are linguistic values that are modeled by fuzzy membership functions. As long as the number of antecedent clauses is small, such rules are human understandable. Even if such a rule involves many antecedent clauses, it is still more transparent than, for example, a DNN. This is so because a fuzzy rule models a small hyperellipsoidal volume in the input space and assigns data points falling in that volume to a particular class with different degrees. Because of this very nature of fuzzy reasoning, we can easily understand how fuzzy rules work and why these are not likely to make an unexpected/unrealistic decision. However, machine learning tools like fuzzy systems or decision trees are poor performer compared to MLPs or DNNs. In particular, designing transparent decision trees or fuzzy rule based classifiers would be quite challenging when it comes to, for example, image recognition. One possibility may be to integrate a DNN and a fuzzy rule based system. We can use a DNN for feature extraction and abstraction. Then at the highest level we can use a fuzzy system for prediction or classification. This may add some level of transparency using the extracted features. However, we certainly need more. Another alternative may be to integrate experts’ domain knowledge into the learning process or at the level of designing the system architecture.

The explainability problem can be approached at least in two different ways. First, by looking deeper into the trained AI systems to get some reasoning behind the decisions. Second, by using an additional layer or system to generate the explanations. For a decision tree or for a rule-based (fuzzy or crisp) system, generation of some explanation is relatively easy, but for a DNN, even for an MLP, it is quite difficult. A fuzzy rule-based system is naturally interpretable as it makes decisions based on fuzzy if-then rules (Hagras, 2018). These rules are easy to understand, for example, *If the body-temperature is HIGH and body-ache is SEVERE then the subject is suffering from flu*. Recently, there have been a few studies to explain why a DNN works. For example, researchers tried to discover which part of an image is primarily responsible to arrive at the final decision by the network (Simonyan et al., 2014; Zeiler and Fergus, 2014; Choo and Liu, 2018). There are other methods which use visual analytics to understand the learned representation and how it influences the output (Liu et al., 2017; Rauber et al., 2017; Choo and Liu, 2018). This kind of visual analytics are useful and help to understand a little better, but cannot make the system transparent or adequately explainable. Such tools/analytics cannot explain the reason behind a decision in a manner that a human would like to have. There have been other approaches to generate explanations for decisions made by a machine learning system (Hendricks et al., 2016; Ribeiro et al., 2016; Samek et al., 2017). For example, authors in Ribeiro et al. (2016) proposed a method to explain predictions made by any classifier by learning a local interpretable model around the prediction. On the other hand, Hendricks et al. (2016) proposed a method for generation of visual explanation for images classified by a deep network where the explanations provide some justifications behind the classification and hence it is different from caption generation. The authors use a Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997) along with the classifier. This is an interesting work but the explanation is generated by the LSTM. The big question of generating the explanation directly from the discriminator still remains. Very recent and useful expositions to the problem of explainable AI, its need and relevance, can be found in Hagras (2018), Goebel et al. (2018), and Holzinger (2018). Holzinger (2018) very nicely explains the advantages and limitations of automatic Machine learning (aML) and the advantages of iML. We have discussed earlier that a child needs only a few examples to learn different animals because humans can exploit the contextual information. Thus, we emphasize again that having a human-in-the-loop appears a very promising way of efficient learning. Holzinger (2018) also discussed a few promising approaches to realize explainable machine learning systems.

2.4. Recognize and React to the Open World Problems

Another consequence of blind learning is its failure to deal with the open world nature of recognition (not necessarily of visual information) problem. We have already mentioned that, DNNs sometime recognize images that are completely “unrecognizable to humans” as human recognizable objects with a very high

confidence (Nguyen et al., 2015). Majority of the decision making problems are some kind of classification problems. Any decision making system should make recommendations based on only what it is taught in a broader sense. This does not mean that a learning system should not generalize, it must but to a “plausible/reasonable” extent. Let us clarify this. Suppose an AI system is trained to distinguish between tigers, lions and cows, where every animal in the training data set has four legs. Now if a cow, which has lost one leg in an accident is presented to the system, we expect the system to declare it as a cow with a reasonable level of confidence. But if the system is confronted with a dog or a goat it should simply say “I do not know.” Note that, here we are referring to systems for which we know why and when they should say “Don’t know.” Most decision making systems including DNNs fail to respond properly when faced with known/unknown unknowns. Recognizing unknowns is very important for many applications such as medicine, healthcare, and defense. There are at least four problems related to this issue (Karmakar and Pal, 2018):

1. In an open-world situation, there are unknown classes (beyond the classes that a classifier is trained to classify). In this case, if a test data point comes from an unknown class, it will get wrongly classified.
2. In a closed-world situation, a test data point may come from one of the classes that the classifier is trained to classify, but it comes from outside the “sampling window” of the training data. In this case, the classifier will assign one of the trained classes and the assigned class may even be correct, but here the classifier should refuse to make any judgement.
3. In a closed-world scenario, we may get a test data point from outside the “sampling window.” In this case, the classifier may assign an unrealistic class. For example, the test data point may be located close to the training data of class i , but it is assigned to class j .
4. Even for a closed-world situation, there may be concept drift—the statistical characteristics of one or more classes may change with time. In such a situation, the classifier should not make any decision on the drifted data.

The above four problems are connected by a common thread; they arise when a test data point does not come from the sampling window of the training data. So, we can address all four problems if the machine learning system can detect and reject a point saying “Don’t Know” when it does not come from the sampling window of the training data.

Many researchers have tried to address some of these problems (Chow, 1957, 1970; Dubuisson and Masson, 1993; Chakraborty and Pal, 2002, 2003; Scheirer et al., 2013, 2014; Jain et al., 2014; Karmakar and Pal, 2018). There have been quite a few attempts to deal with this problem, for example, using the Extreme Value Theorem (Scheirer et al., 2013, 2014; Jain et al., 2014). But such an approach suffers from a conceptual problem because known unknowns and unknown unknowns are not necessarily extreme values of the training data. In fact, these samples may be (usually will be) generated from a completely different distribution than that of the training data. Recently in Karmakar and Pal (2018) authors proposed a scheme to equip a multilayer perceptron

network with the ability to say “do not know” when the test data come from outside the sampling window of the training data. In Karmakar and Pal (2018), a theoretically sound method for estimating the sampling window from a given training data set has been proposed.

Given the estimate of the sampling window, we can draw random samples from the complement of the estimated sampling window (of course over an slightly inflated hyper box containing the training data - any test data coming from outside this box can always be rejected) and use that to represent the “unknown world.” Now, for a c -class problem, we train a $(c + 1)$ class system, where the $(c + 1)^{st}$ class represents the “Don’t Know” class. Further details can be found in Karmakar and Pal (2018). While training the system we may use suitable regularizer to control the sensitivity of the output for the $(c + 1)^{st}$ class with respect to inputs. This will help to train the system with limited samples from the “Don’t know” class.

In principle, such a concept can be applied to DNNs also. But for very high dimensional data such methods would be computationally demanding. However, it may be possible to use appropriate regularizers to minimize the number of samples needed from the complement world for its faithful representation. In order to avoid occasional but catastrophic failure of an AI system, it must recognize its domain of operation beyond which it should not make any decision; otherwise, it may lead to situations giving a false perception that an AI system has “taken over the human.”

2.5. Trustworthiness of a Machine Learning System

When we use a so-called intelligent system either for medical diagnosis or for driving an unmanned vehicle, a natural question comes: how trustworthy the system is! To achieve trustworthiness we need to ensure two things: first, when test samples come from the sampling window, the system should not make

wrong decisions and second, when the test data come from outside the sampling window, the system should refuse to make any recommendation (reject that point). To check the trustworthiness of such a trained system we may proceed as follows Karmakar and Pal (2018). We find the smallest hypercube containing the training data and then expand its each side by a small percentage, say 5%. We generate a large set of points uniformly distributed over the extended box. For each such point, we compute its shortest distance from the points in the training set. Let D be the largest value of all such distances. We divide the interval $[0, D]$ into $k > 1$ intervals. Let $(d_i, d_i + \Delta d)$; $\Delta d > 0$; $0 \leq d_i \leq D$; $i = 0, 1, \dots, k$; $d_0 = 0, d_k = D, d_i = d_0 + i \cdot \Delta d$ be one such interval. Let N_i be the number of points for which the shortest distance lies in $(d_i, d_i + \Delta d)$ and of these N_i points, n_i points are rejected by the network. Then, $f_i = \frac{n_i}{N_i}$; $i = 0, 1, \dots, (k - 1)$ is the percentage of points with distance in $(d_i, d_i + \Delta d)$ that are rejected. For a trustworthy system, with higher values of i , i.e., of d_i , this percentage should increase. Thus, if we plot f_i vs. d_i , for a trustworthy classifier, we expect to see a curve like the one in **Figure 1**. If f_i quickly goes to one with d_i , this will suggest a very conservative and trustworthy system. Apart from the pictorial representation, it may be possible to come up with some index based on this curve to measure the trustworthiness of the trained system’. We just gave some idea of how to deal with this problem, but it certainly needs more focussed study.

2.6. Plausibility of Backpropagation

For majority of the learning algorithms, we use error backpropagation type approaches to optimize a learning objective. For the multilayer perceptron networks, the error backpropagation learning requires that the downstream errors are fed back to upstream neurons via an exact, symmetric copy of the downstream synaptic weight matrix. Thus, each neuron in the hidden layers requires the precise knowledge of all of its downstream connections (synapses). But it is believed to

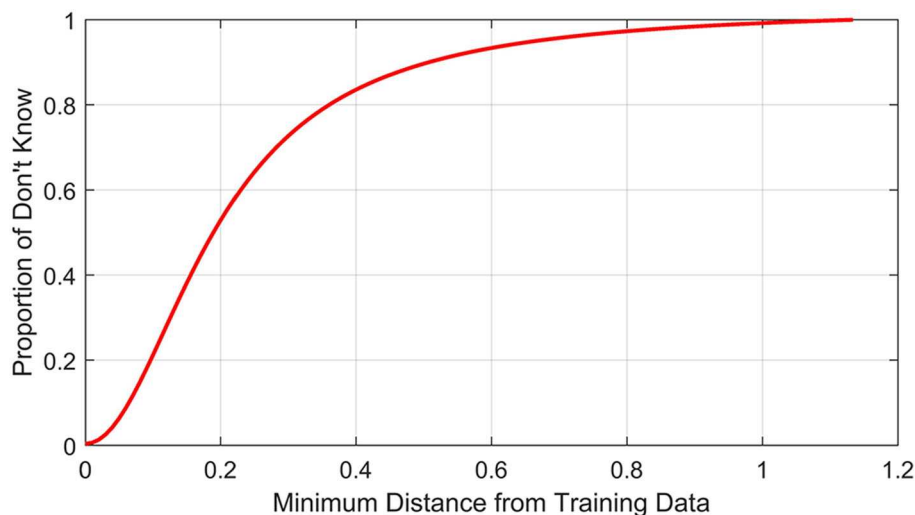


FIGURE 1 | Illustration of how trustworthiness can be assessed.

be almost impossible to happen in the brain (Lillicrap et al., 2016). Lillicrap et al. (2016) demonstrate that we can transmit teaching signal over different layers by multiplying errors with random synaptic weights and it becomes as effective as the backpropagation algorithm for training deep networks. This eliminates the strong structural constraint demanded by backpropagation making it a biologically more plausible architecture. Incorporation of such attributes in our NN would make the architecture of the system closer to that of a biological system. However, as clarified by Lillicrap et al. (2016) this does not answer more fundamental questions as to how exactly the brain computes and represents errors and how the feedforward and feedback paths may interact with each other.

2.7. Structure of Neurons

The neurons used in DNNs/CNNs are primarily of uniform structure and very simple in architecture. In biological vision, visual object recognition is realized by a hierarchical representation of increasingly complex features along the ventral visual stream (DiCarlo et al., 2012; Kuzovkin et al., 2018). This hierarchical representation (abstractions) are typical of deep learning. There have been several studies on establishing correspondence between the hierarchical representation of features with increasing complexity in the brain and that in CNN (Kriegeskorte, 2015; Yamins and DiCarlo, 2016). All these studies motivated researchers to investigate whether the neurons in our brain, which are responsible for thinking/reasoning and have much more complex tree-like structures with roots going deep in the brain and branches going close to the surface, can help to model computational neurons for realizing deep networks like CNN (Guerguiev et al., 2017). In the mammalian neocortex, feedforward and feedback signals are received at electronically segregated dendrites (Guerguiev et al., 2017). In order to realize a biologically more plausible neuron architecture for deep learning, Guerguiev et al. (2017) designed artificial neurons with two compartments, similar to the “roots” and “branches.” The hidden layer neurons are modeled to have segregated “basal” and “apical” dendritic compartments, which are used to separately integrate feedforward and feedback signals and the system did not require separate pathways for feedback signals. They demonstrated that such a network can easily learn to classify the MNIST data (LeCun et al., 1998) and it can also learn hierarchical abstract representation of the objects. This emphasizes as well as demonstrates that it is possible to design computational neurons which more closely model biological neurons as well as can make abstract representation of features like deep networks, and perform the task of classification. However, this does not necessarily mean that the brain replicates exactly this type of processing. Moreover, further investigations are needed to assess whether such networks are more robust or can perform reasoning more close to that by humans.

2.8. Artificial General Intelligence

We have indicated a few very minor cues from biology that may be useful, but there may be (actually there are) plenty of discoveries related to how brain stores, processes, and uses data/information to infer. We need to look at these areas if

we want truly intelligent systems. In our view, to design AI systems with human-type reasoning, such concepts could be very useful for representation. For example, almost all of the successful AI systems of today primarily focus on only single task, say image recognition, and that too specific to a domain. If an AI system is trained to recognize natural scenes or animals, it usually cannot understand an X-ray image, summarize text information, or make medical diagnosis. If we want to realize a generalized AI system capable of doing multiple tasks, we may need to partition the network according to functionality of different lobes of the brain. Creating such an architecture and its training are certainly going to be challenging tasks. This will demand a better understanding of the brain and integration of various discoveries about the brain that we already know. In our view, a purely data-based design of AI systems, certainly is useful and will lead to many unexpected and successful applications, but it may not be adequate to realize true human-type cognitive and reasoning abilities.

3. CONCLUSION AND DISCUSSION

Any AI system, in fact any decision making system, should have a few desirable attributes. It should be accurate, transparent, trustworthy, simple, and be able to explain the decisions it makes. In addition, it would be good if the system can be trained with limited data with limited computation time. In our opinion, if we borrow ideas from the brain science to design decision making systems, we are *more likely* to realize human-like cognitive and reasoning abilities. We say “more likely” because we have mentioned earlier that our knowledge of how the brain learns is limited and the use of the partial knowledge to build an AI system, may not replicate brain style reasoning. Moreover, it may not always be an easy task to incorporate neuroscience discoveries into a computational AI system to realize the desired benefits. In this context we have discussed a few findings in brain science which can be exploited to design AI systems. We have also alluded how one can make a system trustworthy so that the system does not make a decision when it should not. We have provided some ideas on how we can quantify the trustworthiness of a system.

There are other important issues related to design of AI systems that have not been discussed here. For example, time has come to focus on sustainable AI (Pal, 2018b). Here we like to refer to two issues: The first issue is that the development (training) of the AI system should have the minimum carbon footprint. To achieve human-like performance often this important issue is ignored. To illustrate the severity of this issue we consider a recent study which used an evolution-based search to find a better architecture for machine translation and language modeling than the Transformer model (So et al., 2019). The architecture search ran for 979M training steps requiring about 32,623 h on TPUv2 equivalently 274,120 h on 8 P100 GPUs. This may result in 626,155 lbs of CO₂ emission—this is about 5 times the lifetime average emission by an American Car (Strubell et al., 2019). The second point is that the solutions provided by an AI system should be sustainable with the minimum impact on the

environment. For example, an AI system to assist farmers should not just try to maximize the yield, but should also keep in mind the impact of high use of nitrogen fertilizer on the environment. The system should prescribe the use of the Right nutrient source at the Right rate in the Right place and at the Right time (R^4).

In near future, we shall see many remarkable advances in AI with many useful and innovative applications. In fact, time may come when just the accessibility of the pages of a medical book by a computer would enable the system to scan the pages, understand them, extract the rules, and behave like a real doctor! Robots may interact with each other to redistribute work loads among themselves or repair each other's problems. AI applications will be almost everywhere and very intimately related to our daily life. Mostly there will be good usage but there may be some bad ones too. AI will lead to many legal issues also. We certainly will need global policies to monitor the use and abuse of AI.

The world renowned physicist, Stephen Hawking, commented during a talk at the Web Summit technology conference in Lisbon, Portugal, "Success in creating effective AI, could be the

biggest event in the history of our civilization. Or the worst. We just don't know. So we cannot know if we will be infinitely helped by AI, or ignored by it and side-lined, or conceivably destroyed by it" (Kharpal, 2017). He also admitted that the future was uncertain.

It is not an easy task to equip any system (say a robot) with rules (based on data or otherwise) to deal with all possible scenarios for any non-trivial application. If such a robot is not explicitly trained to prevent itself from making decisions in unfamiliar situations, it may behave in an erratic manner and that may be viewed as if the robot has taken over the human. We believe, in near future AI systems will be extensively used almost everywhere and in some application areas (intentionally/unintentionally) uncontrolled behavior of robots may become a reality.

AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and has approved it for publication.

REFERENCES

- Ahmad, S., and Hawkins, J. (2016). How do neurons operate on sparse distributed representations? A mathematical theory of sparsity, neurons and active dendrites. *arXiv preprint arXiv:1601.00720*.
- Bandyopadhyay, A. (2018). *Neural Learning: Can We Make It A Little More Bio-inspired!*. Master's Thesis, Indian Statistical Institute, Calcutta, Available online at: library.isical.ac.in:8080/jspui/bitstream/123456789/6961/1/Diss-395.pdf
- Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychol. Rev.* 94, 115–147.
- Canziani, A., Paszke, A., and Culurciello, E. (2016). An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*.
- Chakraborty, D., and Pal, N. R. (2002). "Making a multilayer perceptron say 'Don't Know' when it should," in *Proceedings of the International Conference on Neural Information Processing, ICONIP 2002* (Singapore), 45–49.
- Chakraborty, D., and Pal, N. R. (2003). A novel training scheme for multilayered perceptrons to realize proper generalization and incremental learning. *IEEE Trans. Neural Netw.* 14, 1–14. doi: 10.1109/TNN.2002.806953
- Chen, Y.-C., Pal, N. R., and Chung, I.-F. (2012). Integrated mechanism for feature selection and fuzzy rule extraction for classification. *IEEE Trans. Fuzzy Syst.* 20, 683–698. doi: 10.1109/TFUZZ.2011.2181852
- Choo, J., and Liu, S. (2018). Visual analytics for explainable deep learning. *IEEE comput. graph.* 38, 84–92. doi: 10.1109/MCG.2018.042731661
- Chow, C. (1970). On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theory* 16, 41–46.
- Chow, C. K. (1957). An optimum character recognition system using decision functions. *IRE Trans. Electron. Comput.* 4, 247–254.
- Dan, Y., Atick, J. J., Reid, R. C. (1996). Efficient coding of natural scenes in the lateral geniculate nucleus: experimental test of a computational theory. *J. Neurosci.* 16, 3351–3362.
- DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron* 73, 415–434. doi: 10.1016/j.neuron.2012.01.010
- Dubuisson, B., and Masson, M. (1993). A statistical decision rule with incomplete knowledge about classes. *Pattern Recogn.* 26, 155–165.
- Eavani, H., Satterthwaite, T. D., Filipovich, R., Gur, R. E., Gur, R. C., and Davatzikos, C. (2015). Identifying sparse connectivity patterns in the brain using resting-state fMRI. *Neuroimage* 105, 286–299. doi: 10.1016/j.neuroimage.2014.09.058
- Einevoll, G. T., and Haldnes, G. (2015). "Lateral Geniculate Nucleus (LGN) models," in *Encyclopedia of Computational Neuroscience*, eds D. Jaeger, and R. Jung (New York, NY: Springer). doi: 10.1007/978-1-4614-6675-8_556
- Fei-Fei, L., Fergus, R., and Perona, P. (2006). One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 594–611. doi: 10.1109/TPAMI.2006.79
- Goebel, R., Chander, A., Holzinger, K., Lecue, F., Akata, Z., Stumpf, S., et al. (2018). "Explainable AI: the new 42?" in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction* (Cham: Springer), 295–303.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT Press.
- Guerguiev, J., Lillicrap, T. P., and Richards, B. A. (2017). Towards deep learning with segregated dendrites. *eLife* 6:e22901 doi: 10.7554/eLife.22901
- Hagras, H. (2018). Toward human-understandable, explainable AI. *Computer* 51, 28–36. doi: 10.1109/MC.2018.3620965
- Hawkins, J. (2017). What intelligent machines need to learn from the neocortex. *IEEE Spectrum* 54, 34–71. doi: 10.1109/MSPEC.2017.7934229
- Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., and Darrell, T. (2016). "Generating visual explanations," in *European Conference on Computer Vision* (Cham: Springer), 3–19.
- Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780.
- Holzinger, A. (2016). Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informat.* 3, 119–131. doi: 10.1007/s40708-016-0042-6
- Holzinger, A. (2018). "From machine learning to explainable AI," in *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)* (Kosice: IEEE), 55–66.
- Holzinger, A., Plass, M., Holzinger, K., Crisan, G. C., Pintea, C.-M., and Palade, V. (2016). "Towards interactive machine learning (iML): applying ant colony algorithms to solve the traveling salesman problem with the human-in-the-loop approach," in *Availability, Reliability, and Security in Information Systems. CD-ARES 2016*, eds F. Buccafurri, A. Holzinger, P. Kieseberg, A. Tjoa, and E. Weippl (Cham: Springer), 81–95. doi: 10.1007/978-3-319-45507-5_6
- Isola, P., Xiao, J., Torralba, A., and Oliva, A. (2011). "What makes an image memorable?" in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Colorado Springs, CO: IEEE), 145–152.
- Jain, L. P., Scheirer, W. J., and Boulton, T. E. (2014). "Multi-class open set recognition using probability of inclusion," in *Computer Vision-ECCV 2014. Lecture Notes in Computer Science*, eds D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars (Cham: Springer), 393–409.
- Karmakar, B., and Pal, N. R. (2018). How to make a neural network say "Don't know". *Inform. Sci.* 430–431, 444–466. doi: 10.1016/j.ins.2017.11.061

- Kharpal, A. (2017). *Stephen Hawking Says A.I. Could Be 'Worst Event in the History of our Civilization'*. Available online at: <https://www.cnbc.com/2017/11/06/stephen-hawking-ai-could-be-worst-event-in-civilization.html> (accessed May 2, 2018).
- Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annu. Rev. Vis. Sci.* 1, 417–446. doi: 10.1146/annurev-vision-082114-035447
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, eds F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. 1097–1105.
- Kuzovkin, I., Vicente, R., Petton, M., Lachaux, J.-P., Baciú, M., Kahane, P., et al. (2018). Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex. *Commun. Biol.* 1:107. doi: 10.1038/s42003-018-0110-y
- Lake, B. M., Salakhutdinov, R., Gross, J., and Tenenbaum, J. B. (2011). "One shot learning of simple visual concepts," in *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society* (Boston).
- Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*. 350, 1332–1338. doi: 10.1126/science.aab3050
- Lake, B. M., Salakhutdinov, R. R., and Tenenbaum, J. (2013). "One-shot learning by inverting a compositional causal process," in *Advances in Neural Information Processing Systems*, eds C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Curran Associates, Inc.), 2526–2534.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324.
- Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nat. Commun.* 7:13276. doi: 10.1038/ncomms13276
- Liu, M., Shi, J., Li, Z., Li, C., Zhu, J., and Liu, S. (2017). Towards better analysis of deep convolutional neural networks. *IEEE Trans. Visual. Comput. Graph.* 23, 91–100. doi: 10.1109/TVCG.2016.2598831
- Maas, A., and Kemp, C. (2009). "One-shot learning with Bayesian networks," in *Proceedings of the 31st Annual Meeting of the Cognitive Science Society*. 2009 (Amsterdam).
- Mitchell, T. (1997). *Machine Learning*. McGraw-Hill Science/Engineering/Math.
- Nguyen, A., Yosinski, J., and Clune, J. (2015). "Deep neural networks are easily fooled: high confidence predictions for unrecognizable images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 427–436.
- Ohki, K., Chung, S., Ch'ng, Y. H., Kara, P., and Reid, R. C. (2005). Functional imaging with cellular resolution reveals precise micro-architecture in visual cortex. *Nature* 433, 597–603. doi: 10.1038/nature03274
- Pal, N. R. (2018a). Some haphazard thoughts [President's Message]. *IEEE Comput. Intell. Mag.* 13, 4–6. doi: 10.1109/MCI.2017.2773879
- Pal, N. R. (2018b). Random thoughts: "comprehensible & sustainable" computational intelligence [President's Message]. *IEEE Comput. Intell. Mag.* 13, 3–4. doi: 10.1109/MCI.2018.2806982
- Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. (2017). "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (Abu Dhabi: ACM), 506–519.
- Rauber, P. E., Fadel, S. G., Falcao, A. X., and Telea, A. C. (2017). Visualizing the hidden activity of artificial neural networks. *IEEE Trans. Visual. Comput. Graph.* 23, 101–110. doi: 10.1109/TVCG.2016.2598838
- Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should i trust you?: explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, CA: ACM), 1135–1144.
- Samek, W., Wiegand, T., and Müller, K. R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv:1708.08296*.
- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM J.* 3, 210–229.
- Scheirer, W. J., Jain, L. P., and Boulton, T. E. (2014). Probability models for open set recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 2317–2324. doi: 10.1109/TPAMI.2014.2321392
- Scheirer, W. J., Rocha, A., Sapkota, A., and Boulton, T. E. (2013). Towards open set recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1757–1772. doi: 10.1109/TPAMI.2012.256
- Sen-Bhattacharya, B., Serrano-Gotarredona, T., Balassa, L., Bhattacharya, A., Stokes, A. B., Rowley, A., et al. (2017). A spiking neural network model of the lateral geniculate nucleus on the SpiNNaker machine. *Front. Neurosci.* 11:454. doi: 10.3389/fnins.2017.00454
- Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep inside convolutional networks: visualising image classification models and saliency maps, *arXiv preprint arXiv:1312.6034v2*
- So, D. R., Liang, C., and Le, Q. V. (2019). "The evolved transformer," in *Proceedings of the 36th International Conference on Machine Learning (ICML)* (Long Beach, CA).
- Strubell, E., Ganesh, A., and McCallum, A. (2019). "Energy and policy considerations for deep learning in NLP," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Florence), 3645–3650. doi: 10.18653/v1/P19-1355
- Vinje, W. E., and Gallant, J. L. (2000). Sparse coding and decorrelation in primary visual cortex during natural vision. *Science*. 287, 1273–1276. doi: 10.1126/science.287.5456.1273
- Wang, Y., Yao, Q. U., Kwok, J., and Ni, L. M. (2019). Generalizing from a few examples: a survey on few-shot learning. *arXiv:1904.05046*.
- Yamins, D. L., and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* 19, 356–365. doi: 10.1038/nn.4244
- Zeiler, M. D., and Fergus, R. (2014). "Visualizing and understanding convolutional networks," in *Proceedings of the European Conference on Computer Vision* (Cham: Springer), 818–833.
- Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). "Understanding deep learning requires rethinking generalization," in *International Conference on Learning Representations, ICLR 2017* (Toulon). Available online at: <https://openreview.net/pdf?id=Sy8gdB9xx>

Conflict of Interest: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Pal. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



From Continuous Observations to Symbolic Concepts: A Discrimination-Based Strategy for Grounded Concept Learning

Jens Nevens*, Paul Van Eecke and Katrien Beuls

Artificial Intelligence Laboratory, Vrije Universiteit Brussel, Brussels, Belgium

OPEN ACCESS

Edited by:

Georg Martius,
Max Planck Institute for Intelligent
Systems, Germany

Reviewed by:

Emre Ugur,
Boğaziçi University, Turkey
Remi van Trijp,
Sony Computer Science Laboratories,
France

*Correspondence:

Jens Nevens
jens@ai.vub.ac.be

Specialty section:

This article was submitted to
Computational Intelligence in
Robotics,

a section of the journal
Frontiers in Robotics and AI

Received: 11 March 2020

Accepted: 26 May 2020

Published: 26 June 2020

Citation:

Nevens J, Van Eecke P and Beuls K
(2020) From Continuous Observations
to Symbolic Concepts: A
Discrimination-Based Strategy for
Grounded Concept Learning.
Front. Robot. AI 7:84.
doi: 10.3389/frobt.2020.00084

Autonomous agents perceive the world through streams of continuous sensori-motor data. Yet, in order to reason and communicate about their environment, agents need to be able to distill meaningful concepts from their raw observations. Most current approaches that bridge between the continuous and symbolic domain are using deep learning techniques. While these approaches often achieve high levels of accuracy, they rely on large amounts of training data, and the resulting models lack transparency, generality, and adaptivity. In this paper, we introduce a novel methodology for grounded concept learning. In a tutor-learner scenario, the method allows an agent to construct a conceptual system in which meaningful concepts are formed by discriminative combinations of prototypical values on human-interpretable feature channels. We evaluate our approach on the CLEVR dataset, using features that are either simulated or extracted using computer vision techniques. Through a range of experiments, we show that our method allows for incremental learning, needs few data points, and that the resulting concepts are general enough to be applied to previously unseen objects and can be combined compositionally. These properties make the approach well-suited to be used in robotic agents as the module that maps from continuous sensory input to grounded, symbolic concepts that can then be used for higher-level reasoning tasks.

Keywords: grounded concept learning, language games, hybrid AI, CLEVR, emergent communication

1. INTRODUCTION

A concept can be described as a mapping between a symbolic label and a collection of attributes that can be used to distinguish exemplars from non-exemplars of various categories (Bruner et al., 1956). In the context of grounded, autonomous agents, these attributes correspond to streams of continuous-valued data, obtained through the agent's various sensors. In order to communicate and reason about the world, agents require a repertoire of concepts that abstracts away from the sensori-motor level. Without this layer of abstraction, communication would happen by directly transmitting numerical observations. Such a system easily leads to errors in communication, for example when the agents observe the world from different perspectives, or when calibration is difficult because of changing lighting conditions or other external factors. To obtain a repertoire of concepts, i.e., mappings from labels to attribute combinations, autonomous agents face two learning problems simultaneously. First, the agents need to find out which attributes are important for each concept. This requires a mechanism for identifying meaningful

combinations of attributes from their sensori-motor data streams and attaching a symbolic label to each of these combinations. Second, the agents must be able to recognize instances of particular concepts and distinguish concepts from each other. For representing concepts, we make use of prototype theory (Rosch, 1973), although also other approaches have been proposed in the psychological literature (McCarthy and Warrington, 1990; Squire and Knowlton, 1995; Patalano et al., 2001; Grossman et al., 2002).

A range of different approaches have been applied to concept learning, including version spaces and deep learning techniques. However, we identify a number of drawbacks in these approaches. In version space learning, a concept is represented as an area in a hypothesis space. This space can for example denote the possible ranges of values of various attributes of the concept. Each concept is bound by the most general and the most specific consistent hypothesis. Using positive and negative examples, these boundaries can be updated using the candidate elimination algorithm (Mitchell, 1982). A well-known caveat of this technique, however, is its inability to handle noisy data. In deep learning approaches, concepts are often represented through embeddings, i.e., high-dimensional numerical representations, which lack human-interpretability (see e.g., Mao et al., 2019; Shi et al., 2019). Additionally, as these embeddings are learned in a statistical way, they often fail to adapt to unseen scenarios and require huge amounts of training data. Neither of these approaches offer a learning mechanism that would be suitable for an autonomous agent, i.e., responsive to changes in the environment, able to support incremental learning, and able to dynamically expand the agent's repertoire of concepts.

In this paper, we propose a novel approach to grounded concept learning. Using the language game methodology (Steels, 2001), we set up series of scripted, task-oriented communicative interactions in a tutor-learner scenario. The environment in which these interactions take place is adopted from the CLEVR dataset (Johnson et al., 2017). This environment consists of scenes made up of geometrical objects, where the objects differ in color, size, shape, material, and spatial position. Through the communicative task, an agent must learn the concepts present in this dataset, such as SMALL, RED, or LEFT. Learning these concepts requires not only finding relevant attribute combinations (e.g., “r,” “g,” and “b” for color), but also their prototypical values (e.g., “r:44,” “g:76,” and “b:215” for BLUE). Both the tutor and the learner agent make use of the notion of discrimination, i.e., maximally separating one particular object from the other objects in the scene. Discrimination is an often-used mechanism in experiments on the emergence and evolution of language (Steels, 1997; Vogt, 2002; Pauw and Hilferty, 2012; Wellens, 2012; Bleys, 2016). In language production, the tutor looks for the concept that is maximally discriminating for a particular object, thereby helping the learner to solve the communicative task. The learner, on the other hand, uses the tutor's feedback and discrimination to update its repertoire of concepts after every interaction. This ensures that the concepts are optimally relevant for the communicative task and the environment in which they occur.

The main contribution of this paper is a novel method to represent and learn symbolic concepts that provide an abstraction layer over continuous-valued observations. This method builds on earlier work by Wellens (2012) and extends the discrimination-based learning of concepts represented by weighted combinations of attributes, so that they can be learned from continuous streams of data. Through various experiments, we demonstrate how the learner acquires a set of human-interpretable concepts in a way that is (i) general, (ii) adaptive to the environment, (iii) requires few interactions, and (iv) allows for compositionality.

The remainder of this paper is structured as follows. In section 2, we discuss existing approaches to concept learning. Section 3 introduces the environment in which the agents operate and the language game setup. In section 4, we introduce the experiments, each showcasing a desirable property of our approach. The experimental results are provided and discussed in section 5. Finally, in section 6, we summarize and conclude.

2. RELATED WORK

2.1. Version Space Learning

One method for representing and learning concepts is through version spaces (Mitchell, 1982). In this method, a concept is represented as an area in a space with dimensionality equal to the number of attributes. The concept area is bounded by both the most specific consistent hypothesis and the most general consistent hypothesis. A hypothesis consists of a combination of attribute values and it is considered consistent when it agrees with the observed examples. With this representation, the simplest way of learning concepts is through the candidate elimination algorithm. Provided with both positive and negative training examples, the algorithm works as follows. The most general and most specific hypotheses are being updated in such a way that the former covers all positive training examples, including as much as possible of the remaining attribute space but excluding any negative examples, and the latter covers all positive training examples with as little as possible of the remaining attribute space. These updates happen in an incremental manner, looking for the minimal specialization for the most general hypothesis and the minimal generalization for the most specific hypothesis.

A major drawback of the candidate elimination algorithm is its inability to handle noisy data. Noisy or wrongly labeled training examples can incorrectly update one or both of the boundaries and recovering from such errors is often difficult. On the positive side, because of the relatively simple representation and learning algorithm, concepts learned using version spaces are often human-explainable and transparent. Furthermore, when the boundaries are allowed to be updated after training, the concepts remain adaptive over time.

2.2. Neural Approaches

More recent approaches to concept learning are dominated by deep learning techniques. State-of-the-art results have been achieved by Higgins et al. (2016) and Shi et al. (2019). These two approaches vary strongly in the neural network architecture, the learning regime (e.g., binary or multi-class classification

or unsupervised learning), the concept representation (e.g., a label in a classifier or a group of latent variables) and the task or domain in which concepts are being learned (e.g., hand-written characters or generated graphics). However, the aforementioned papers are particularly interesting since both of them take inspiration from human concept learning and incorporate this in their models. For example, how humans require only one or a few examples to acquire a concept is incorporated through one-shot or few-shot learning or how known concepts can be used to recognize new exemplars is achieved through incremental learning and memory modules. Many more approaches to concept learning using deep learning techniques exist (e.g., Wang et al., 2015; Dolgikh, 2018; Xu et al., 2018; Rodriguez et al., 2019). In general, these approaches yield high levels of accuracy but require huge amounts of training data and/or training time. Additionally, the concepts are represented in a way that is often not human-interpretable and the set of concepts is often predefined and fixed over time. Some of the aforementioned approaches tackle one or two of these issues, but not all together.

In other approaches, concepts are learned as a “side effect” while tackling another, typically larger task. In the work by Mao et al. (2019) and Han et al. (2019), not only concepts but also words and semantic parses of sentences are learned in the context of a Visual Question Answering task. Specifically, a perception module learns visual concepts, represented as embeddings, based on the linguistic description of the object being referred to. As reported by Mao et al. (2019), the concepts are acquired with near perfect accuracy (99.9%) and a relatively small amount of training data (5K images), but the resulting concept representations are not human-interpretable. The proposed model does allow for incremental learning and generalizes well to unseen combinations of attributes. This generalization, however, requires fine-tuning the model on a held-out dataset.

2.3. The Omniglot Challenge and Bayesian Program Learning

One particular line of research that focusses exclusively on human-like concept learning is centered around the Omniglot dataset (Lake et al., 2015). This is a dataset of hand-written characters from 50 different alphabets. Each character is written by 20 people and stored as both an image and pen stroke data. The Omniglot challenge aims to push forward the state-of-the-art in human-like concept learning. The main challenge consists of a within-alphabet one-shot classification task: given a new character and an alphabet, identify the character in the alphabet that is the same character as the one presented. This task aims to replicate the ability of humans to acquire a new concept with only a single example. Next to this, there are three other tasks designed to test several concept learning-related abilities: parsing of exemplars into parts and relations, generating new exemplars of a given concept and generating new concepts of a particular type.

In his own work, Lake et al. (2015) introduces Bayesian Program Learning (BPL) to tackle the Omniglot challenge. Here,

concepts are represented as probabilistic generative models, trained using the pen stroke data and built in a compositional way such that complex concepts can be constructed from (parts of) simpler concepts. In this case, the model builds a library of pen strokes and characters can be generated by combining these pen strokes in many different ways. This approach has many advantages, including the ability to do one-shot learning and a powerful compositional representation of concepts that allows not only to classify concepts but also to generate them. While this model achieves impressive results, learning through pen stroke data offers a limited range of possibilities. Other researchers have tackled the Omniglot challenge, mostly using neural approaches as reported by Lake et al. (2019). Almost all of them have focussed on the one-shot classification task using the image data as input. As a result, the BPL approach remains the SOTA model for all tasks in the Omniglot challenge.

2.4. Reinforcement Learning

Concept learning has also been approached from a reinforcement learning perspective. In this context, a concept is regarded as an abstraction over an agent's states or actions. Abstraction over discrete states can be achieved through tile-coding (Sutton, 1996). Recently however, following advances in the domain of deep reinforcement learning, abstraction over continuous states is often performed through function approximation (Mnih et al., 2015). Abstraction over actions is commonly achieved through the use of options (Sutton et al., 1999).

One line of research that is particularly relevant to our approach is the work by Konidaris and colleagues. Initially, the authors mapped propositional symbols to a set of low-level states (Konidaris et al., 2014). These states were obtained from the continuous environment through a classifier. A planning problem is then solved using the propositional symbols as operators, which can be translated to sets of low-level states, executed in the environment. In later work, the set-based representation was replaced by a probability distribution, to better capture the uncertainty about the successful execution of each high-level step (Konidaris et al., 2015, 2018). Again, this approach was validated through a planning problem in a continuous state space, where policies for high-level planning problems in a game environment, such as “obtain key” or “obtain treasure,” could be computed efficiently.

The symbolic high-level steps can be represented in a human-interpretable way, as the pre- and postconditions can be easily visualized in the game environment. Additionally, the model proposed by Konidaris et al. (2015) can be learned efficiently with relatively few data points: 40 iterations of 100 randomly chosen actions were used to extract the high-level steps. As is typical in a Reinforcement Learning setting, the planning steps are learned through experience. Hence, new planning steps must be learned by collecting new experiences specific to this concept. Additionally, the resulting steps are relatively domain-specific. No experiments are reported that investigate generality, e.g., would JUMP-LEFT generalize to other game settings, or adaptivity, e.g., does the concept JUMP-LEFT change when the game physics change.

2.5. Robotics

A large body of work exists in the robotics community that considers various tasks very similar to what we refer to as concept learning. Common names for this task include symbol emergence, perceptual anchoring, affordance learning, and category learning.

As a first approach, we consider the task of perceptual anchoring. The goal of perceptual anchoring is to establish and maintain a link between symbols and sensor data that refer to the same physical object (Coradeschi and Saffiotti, 2003). This link should remain stable through time and space, e.g., when an object moves through a robot's view, when it is covered by another object, or when it disappears and later reappears. The symbol system can manipulate individual symbols, referring to objects as a whole, but also predicates reflecting properties of the objects. Different representations can be used by the sensor system, e.g., a set of continuous-valued features or a vector in some embedding space. An anchoring system can be bottom-up, starting from the perceptual level, and top-down, starting from the symbolic level. In the context of perceptual anchoring, the combination of a symbol, a set of predicates and sensor data can be considered a single concept.

In recent work, a bottom-up perceptual anchoring system was combined with a probabilistic symbolic reasoning system (Persson et al., 2019). This approach allowed to improve the overall anchoring process by predicting, on the symbolic level, the state of objects that are not directly perceived. There are multiple advantages to this approach. First, the authors achieve high accuracy (96.4%) on anchoring objects and maintaining these anchors in dynamic scenes with occlusions, using relatively little training data (5400 scenes, 70% used for training). Additionally, their system is completely open-ended and allows for incremental learning, since the anchor matching function will simply create new anchors when it encounters previously unseen objects. The anchor matching function, in some way a similarity measure, is closely related to the notion of discrimination. The difference being that discrimination also takes the other objects into account. Finally, the representation of a concept can be human-interpretable, depending on the representation of objects in the sensor system and the corresponding symbols and predicates.

For a second approach, we focus on affordance learning. With this approach, the focus lies on the interaction between the perceptual system and the motor system of an autonomous agent. Put differently, an affordance can be considered as a learned relation between an action in the environment, caused by the motor system, and the effect observed in the environment, captured by the perceptual system (Şahin et al., 2007). Building on this, the agent can learn concepts in terms of affordances. As proposed by Ugur et al. (2011) and further worked out in Ugur and Piater (2015a,b), affordances can be grouped together in effect categories. These are consequently mapped to clustered object properties to form a particular concept. For example, the concept BALL is an object with spherical properties that exhibits the roll-effect when pushed and the disappear-effect when lifted, as it rolls off the table when dropped. In these works, the authors use concepts learned through their affordances in

plan generation and execution, with an agent being capable of planning the necessary actions involving specific objects to reach a given goal state. This approach offers a more action-centric view on the agent's world, which is complementary to our approach. It not only allows an agent to recognize and describe objects in the world, but also correctly act on them. The concepts that are acquired, combining effect categories with object properties, offer a transparent view. The effect categories are expressed in terms of change in visibility, shape and position, and the object properties are stored in a numerical vector with explainable entries, such as features relating to position and shape (Ugur et al., 2011). Additionally, since the concepts are learned through unsupervised exploration, the proposed model is adaptive to the environment. New concepts can be added incrementally through additional exploration and learned concepts can be progressively updated (Ugur and Piater, 2015b). As is typical in robotics, the proposed approach combines learning in simulation and using physical robots. The concepts, specifically, could be acquired after only 4,000 simulated interactions (Ugur et al., 2011). The robot is used to validate these concepts in several planning problems. Finally, as the agent assesses the object features relevant for each effect category, the resulting mappings offer some generality, e.g., a ball exhibits the same effect categories regardless of its color.

Other approaches take a probabilistic perspective on concept learning, similar to Lake et al. (2015), but focussing on the domain of robotics. Concepts are learned through unsupervised online learning algorithms, combining multi-modal data streams (most often perceptual data and raw speech data) through statistical approaches such as Bayesian generative models or latent semantic analysis (Nakamura et al., 2007; Aoki et al., 2016; Taniguchi et al., 2016, 2017). Through this integration of data streams, the acquired concepts constitute mappings between words and objects, as studied by Nakamura et al. (2007) and Aoki et al. (2016), or between words and spatial locations, as studied by Taniguchi et al. (2016, 2017). The latter further used these concepts to aid a mobile robot in generating a map of the environment without any prior information. The statistical methods have the advantage of being able to infer a considerable amount of information from a limited number of observations, and are therefore suitable for use in robotics scenarios. Additionally, they offer model interpretability to a certain extent, through a graphical model representation such as a Bayesian network. Finally, the proposed models are adaptive to changes in the environment and offer incremental learning through the online learning algorithms.

Among the various approaches to concept learning discussed so far, our proposed approach is most closely related to the robotics literature, as many of these studies deal with similar issues such as grounding, adaptivity, generality, and fast learning. For a more comprehensive overview on symbol emergence from the viewpoint of cognitive systems/robotics, we refer to Taniguchi et al. (2018).

2.6. Discrimination-Based Learning

One particular experiment by Wellens (2012) has heavily inspired this work. Wellens makes use of the language game methodology to study multi-dimensionality and compositionality during the

emergence of a lexicon in a population of agents. In this language game, called the compositional guessing game, the speaker tries, using language, to draw the attention of the listener to a particular object in a shared scene. Each object in such a scene is observed by the agent as a collection of symbolic attributes, e.g., “a-1,” “a-2,” “a-3” and so on. The words used by the agents have one or multiple of these same symbols as their meaning (multi-dimensionality) and the agents can use multiple words to describe a particular object (compositionality). At the end of a game, the agents give each other feedback on the outcome of the game and the speaker points to the intended object in case of failure. This setup leads to a large amount of uncertainty for the agents, as they should find out what part of the meaning should be linked to which word in the multi-word utterance.

In his work, Wellens proposes two distinct types of strategies for reducing this uncertainty: competitive strategies and adaptive strategies. Both make use of the notion of discrimination, i.e., maximally separating one object from the others, for both language production (the speaker) and interpretation (the hearer). However, in the former type of strategies, the agents explicitly enumerate competing hypotheses (i.e., the same word with a different meaning) and mechanisms are in place to gradually reduce this enumeration. This soon becomes intractable, leading to scaling issues in environments with many objects or many attributes per object. The latter type of strategies, on the other hand, avoids enumerating competing hypotheses. Instead, only a single meaning, composed of a set of attributes, is kept for each word. Over the course of interactions, this meaning is gradually being shaped based on the feedback provided after each interaction. How this shaping is implemented depends on the particular strategy. Adaptive strategies focus on re-use, allowing agents to use words even when the associated meanings are not (yet) fully compatible with the topic object. **Figure 1** illustrates the difference between the two types of strategies.

Within the realm of adaptive strategies, a distinction is made between the baseline adaptive strategy and the weighted adaptive strategy. In the former strategy, the ideas underpinning adaptive strategies are implemented in a rather crude way. The agents gradually shape the meaning of words simply by adding or removing attributes from the set, based on the feedback after the game. The latter strategy offers a more gradual shaping of

the meaning. Here, the meaning is no longer a regular set of attributes but instead it is a weighted set. Each attribute receives a score, expressing the certainty that the attribute is important for the word it is linked to. Based on the received feedback, agents cannot only add or remove attributes, but also alter the score of attributes to reflect changes in certainty. Over time, the meanings are shaped to capture attribute combinations that are functionally relevant in the world, driven by the force to obtain communicative success and the notions of discrimination and alignment. For more details about the compositional guessing game and the various strategies, we refer to Wellens (2012).

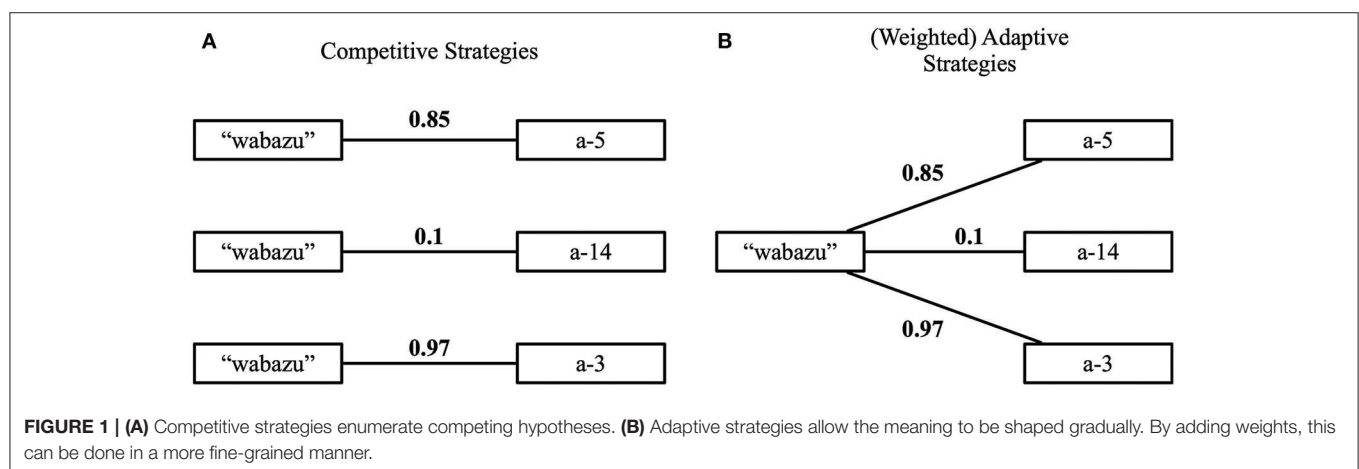
Our approach to concept learning is heavily inspired by the weighted adaptive strategy. As we will discuss later on, concepts in our approach are also represented by weighted attribute sets. However, where previous work only considers symbolic attributes, we extend this approach to continuous-valued attributes, introducing the need for more sophisticated representations and processing mechanisms.

3. METHODOLOGY

The goal of this work is for an agent to distill meaningful concepts from a stream of continuous sensory data through a number of communicative interactions called language games. These interactions are set in a tutor-learner scenario and take place in a shared environment consisting of scenes of geometric shapes. Driven by the communicative task and the notion of discrimination, the agent will gradually shape its repertoire of concepts to be functional in its environment. In this section, we elaborate on the language game methodology (section 3.1), the environment in which the agents operate (section 3.2), the concept representation and update mechanism as used by the learner (section 3.3) and the mechanisms used by the tutor (section 3.4).

3.1. Language Game

The language game methodology is commonly used to study how a population of agents can self-organize a communication system that is effective and efficient in their native environment. By playing language games, agents take part in a series of scripted and task-oriented communicative interactions. A language game



is typically played by two agents from the population, one being the speaker and another being the hearer. There is no central control and the agents have no mind-reading capabilities. The agents are only allowed to communicate through language. After a number of games, the population converges on a shared communication system through selection and self-organization. This methodology has been used to study the emergence of a wide range of linguistic phenomena, including grammatical agreement (Beuls and Steels, 2013), color lexicons (Bleys, 2016), argument marking (Lestrade, 2016), quantifiers (Pauw and Hilferty, 2012), spatial language (Spranger and Steels, 2012), case (van Trijp, 2016), etc.

The language game in this work is set up in a tutor-learner scenario. The tutor is an agent with an established repertoire of concepts, while the learner starts the experiment with an empty repertoire. The tutor is always the speaker and the learner is always the listener. Before each game, both agents observe a randomly sampled scene of geometric shapes. The environment itself will be explained in greater detail in section 3.2. For now, we note that the tutor has access to a high-level symbolic annotation of the scene, while the learner observes the scene through streams of continuous data. The symbolic annotation constitutes the ground-truth of the scene and the learning target for the learner agent. This avoids having to manually design a number of concepts in terms of the observed data stream for the tutor, which could bias the system.

The interaction script, which are the steps both agents go through during a single language game, goes as follows. The tutor starts the interaction by choosing one object from the scene as the topic. Using the symbolic annotation, the tutor looks for a concept that optimally discriminates the topic and utters it. By looking for the most discriminative concept, the tutor is actively trying to help the learner in solving the communicative task. If the topic cannot be discriminated using a single concept, the tutor picks another object or scene. This restriction will be lifted later on in one of the more advanced experiments. The learner receives this word and checks its repertoire of concepts. If the concept denoted by this word is unknown, the learner indicates failure to the tutor. Alternatively, if the learner does know the word, it will try to interpret the corresponding concept in the current scene. In other words, the learner will look for the object that best matches the concept. The learner points to this object and the tutor provides feedback on whether or not this is correct.

After each interaction, the tutor provides feedback by pointing to the intended topic. This is a learning opportunity for the learner. We call this phase of the game “alignment.” If the concept was unknown for the learner, it is now able to create a new concept. At this stage, the learner cannot yet know which attributes are important for the concept. It does know, however, that the tutor could discriminate the topic using this concept. Thus, the learner stores an exact copy of the topic object as the initial seed for the corresponding concept. Each attribute receives an initial score of 0.5, reflecting the uncertainty that the attribute is important for the newly created concept. Alternatively, if the learner did know the concept, it can refine its representation using the newly acquired example. This involves updating the prototypical values and the certainty scores of the

attributes. We elaborate on this mechanism in section 3.3. A schematic overview of the complete interaction script is shown in **Figure 2**.

Note that in our description of the interaction script in the previous paragraphs, we have used the words “concept” and “word” interchangeably. We will continue to do so in the remainder of this paper, as in the experiments that we describe, there is a one-to-one correspondence between words and concepts.

To evaluate the learner agent, we measure both communicative success and concept repertoire size. Communicative success indicates whether or not the interaction was successful. In other words, it tells us if the learner could successfully use the concept in interpretation and consequently points to the topic intended by the tutor. Also, we can monitor the number of interactions required to reach a particular level of communicative success, indicating the speed at which the agent is learning. By keeping track of the size of the learners concept repertoire over time, we can check how many interactions are required for the learner to acquire all concepts known by the tutor. In the experimental environment, there are 19 concepts to be learned in total. These are summarized in **Table 1**.

3.2. Environment

3.2.1. The CLEVR Dataset

The agent’s environment is based on the CLEVR dataset (Johnson et al., 2017). This dataset contains 100K rendered scenes of geometric objects. Each scene contains between 3 and 10 randomly placed objects. The objects have four basic properties: color, size, material, and shape. In total, there are 8 distinct colors, 2 sizes, 2 materials, and 3 shapes. Next to an image of the scene, there is also a ground-truth symbolic annotation, encoded in JSON format. An example scene and annotation are shown in **Figure 3**. The CLEVR dataset is split into a training set (70K images), a validation set (15K images) and a test set (15K images). In this work, we only make use of the images of the validation set, as no ground-truth annotations are available for the test set. Additionally, since the language game paradigm features online interactive learning, there are no separate training and test phases. The agent is evaluated whilst learning and hence, no held out dataset is required. The CLEVR dataset is ideal for concept learning experiments, as the dataset was specifically designed to avoid dataset biases as much as possible. In practice, this means that across the scenes, there will be as many blue objects as red objects, as many cubes as cylinders, etc.

The learner agent observes its environment through streams of continuous-valued sensor data. To achieve this, the CLEVR scenes need to be transformed into numerical data. We consider two ways of making this transformation. As a first method, we use manually written rules and procedures to transform the symbolic JSON annotation into numerical data. This method is explained in section 3.2.2. For the second method, we use a state-of-the-art Mask R-CNN model (Yi et al., 2018) to detect and segment the objects directly from the image. Section 3.2.3 is dedicated to this method.

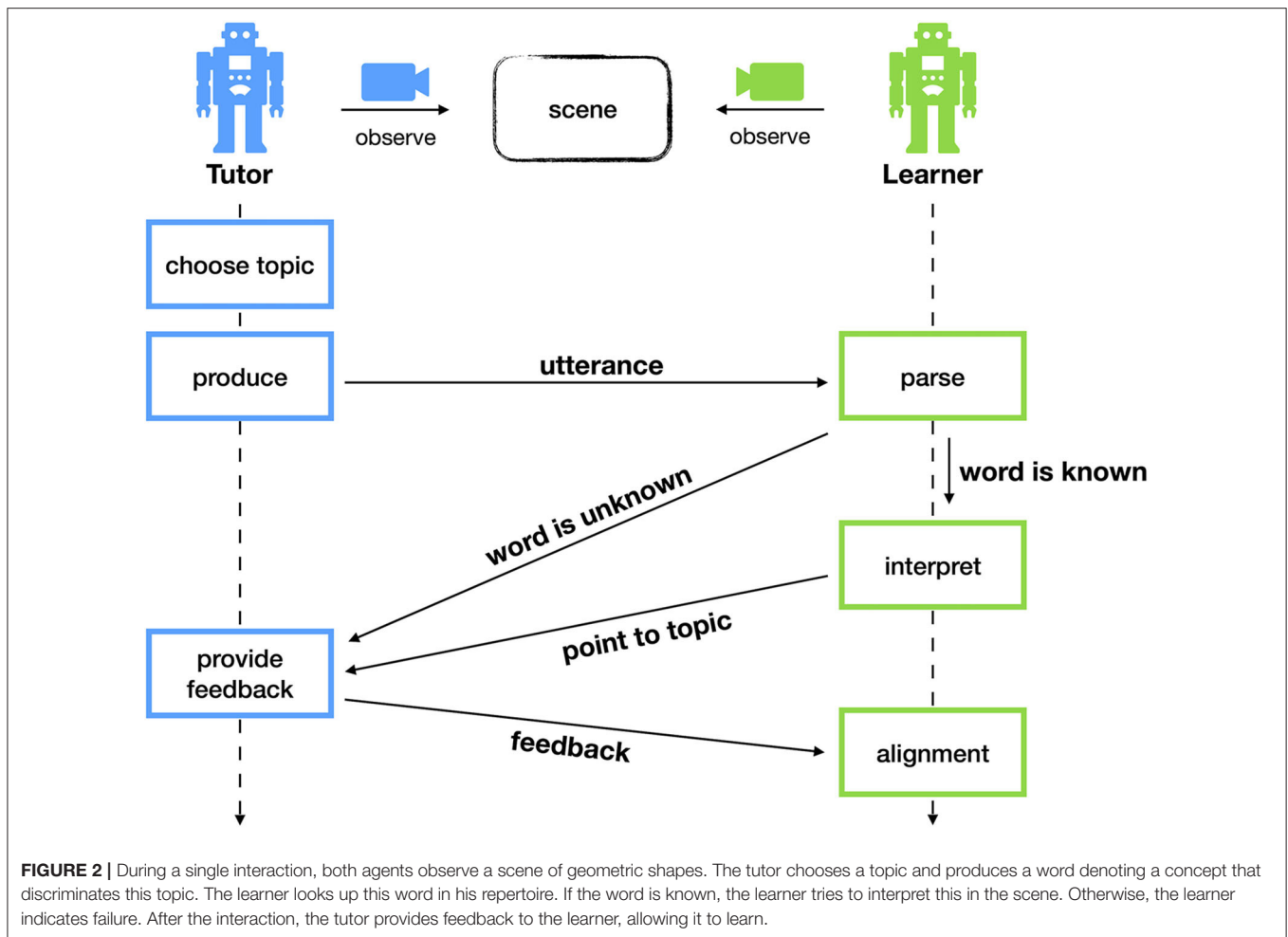


TABLE 1 | All concepts in the experimental environment.

Shapes	Colors	Sizes	Materials	Positions
CUBE	BLUE	LARGE	METAL	BEHIND
CYLINDER	BROWN	SMALL	RUBBER	FRONT
SPHERE	CYAN			LEFT
	GRAY			RIGHT
	GREEN			
	PURPLE			
	RED			
	YELLOW			

3.2.2. Simulated Attributes

The first method starts from the symbolic scene annotations and transforms these into continuous-valued attributes based on simple rules and procedures. We provide an overview of these rules in **Table 2**. Each symbolic attribute is mapped to one or more continuous attributes with a possible range of values. For example, color is mapped to three attributes, one for each channel of the RGB color space, and size is mapped to a single attribute,

namely *area*. We also include the x- and y-coordinates. These attributes were already present in the CLEVR dataset and are simply adopted.

The values for the various attributes are not chosen arbitrarily. For color concepts, e.g., RED, we use the RGB value that was used during the image rendering process of the CLEVR dataset¹. This value is used as a seed value and random jitter is added. The same technique is used for the size-related concepts. The amount of jitter is shown in the rightmost column of **Table 2**. Generating the continuous attributes for the shape-related attribute proceeds as follows. We consider a sphere to have 1 side, 0 corners and a width-height ratio of 1, a cylinder to have 3 sides, 2 corners and a width-height ratio of 0.5 and a sphere to have 6 sides, 8 corners and a width-height ratio of 1. Finally, material is identified by a measure of surface roughness.

Obtaining sensory data in this way is straightforward and creates a controlled environment. Indeed, even with the presence of random jitter, there is no overlap between different instances of a particular concept, such as BLUE and CYAN or LARGE and

¹This information is available at <https://github.com/facebookresearch/clevr-dataset-gen>.

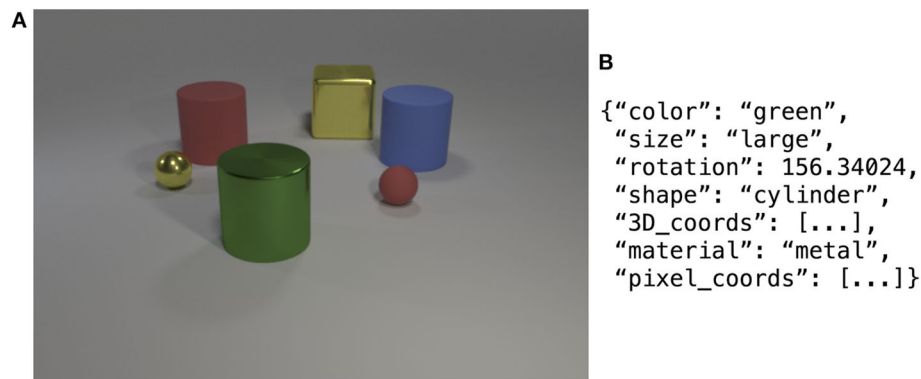


FIGURE 3 | Example image from the CLEVR dataset (A) with the corresponding symbolic annotation of a single object (B), namely the green cylinder.

TABLE 2 | Rules used to transform symbolic object properties to continuous-valued attributes.

Symbolic	Continuous	Values	Jitter
Color	<i>R</i>	[0, 255]	$\pm[0, 2]$
	<i>G</i>	[0, 255]	$\pm[0, 2]$
	<i>B</i>	[0, 255]	$\pm[0, 2]$
Shape	<i>nr-of-sides</i>	{1, 3, 6}	/
	<i>nr-of-corners</i>	{0, 2, 8}	/
	<i>Width-height ratio</i>	[0, 1]	/
Size	<i>Area</i>	[0, 100]	$\pm[0, 15]$
Material	<i>Roughness</i>	[0, 10]	$\pm[0, 2.5]$
	<i>x-coordinate</i>	[0, 500]	/
	<i>y-coordinate</i>	[0, 300]	/

Note that objects in the CLEVR dataset already have *xy-coordinates*.

SMALL. For each particular type of concept, every instance takes up a disjoint area in the space of continuous-valued attributes. This makes the concept learning task easier and allows us to validate the proposed learning mechanisms before moving to an environment with more realistic perceptual processing.

3.2.3. Extracted Attributes

To test our approach using more realistic perceptual processing, we make use of a state-of-the-art Mask R-CNN model to detect and segment the objects directly from the image. After segmentation, we extract a number of numerical attributes from the proposed segments. With this approach, different instances of a particular concept will no longer take up disjoint areas in the attribute space. Additionally, the numerical values will be subject to more noise due to variations in the images such as overlapping objects, lighting conditions or shade effects.

For object detection, we use a pre-trained neural network model developed by Yi et al. (2018) using the Mask R-CNN model

(He et al., 2017) present in the Detectron framework (Girshick et al., 2018). Given an image, this network generates a mask for each of the objects in the scene. All masks with a certainty score below 0.9 are removed. The model was pre-trained on a separately generated set of CLEVR images. For training regime details, we refer to Yi et al. (2018). To our knowledge, there was no separate evaluation of the object detection accuracy.

We combine the obtained segments with the original image to extract a number of continuous-valued attributes. These are summarized in Table 3. As with the previous environment, we foresee a number of continuous attributes for each symbolic attribute of the CLEVR objects. For colors, we extract both the mean and standard deviation of the color of the region, expressed in the HSV color space and split for each channel. For shapes, we extract the estimated number of corners, the hamming distance between the shape's contour and the enclosing circle, and the width-height ratio. The size-related attributes are straightforward, except for the last two. The *bb-area ratio* expresses the ratio between the area of the region and the area of the rotated bounding box. Similarly, the *image-area ratio* expresses the ratio between the region's area and the area of the entire image. Finally, the material of objects is expressed by the ratio of both dark and bright pixels. These attributes are based on the idea that the metal objects are more reflective and thus contain more bright pixels.

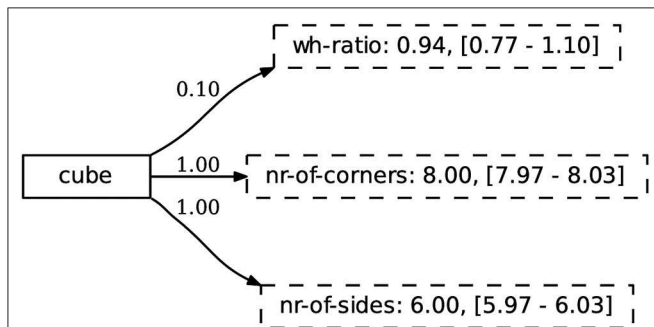
3.3. Concept Representation

A concept is represented as a mapping from a symbolic label, in this case used as a word, to a set of continuous-valued attributes. Similar to Wellens (2012), we make use of a weighted set representation where each concept-attribute link has a score ($\in [0, 1]$), representing the certainty that the given attribute is important for the concept. In contrast to Wellens (2012), the attributes are continuous, represented through a normal distribution. This enables the use of such concepts in grounded, embodied scenarios. An example concept is shown in Figure 4.

To computationally operationalize this concept representation in a language game scenario, we require two pieces of functionality: the ability to match a concept to an object

TABLE 3 | Mapping from symbolic attributes to continuous attributes obtained by the image segmentation process.

Symbolic	Continuous	Values
Color	Mean-H	[0, 255]
	Mean-S	[0, 100]
	Mean-V	[0, 100]
	Std-H	$\mathbb{R}^+$
	Std-S	$\mathbb{R}^+$
	Std-V	$\mathbb{R}^+$
Shape	nr-of-corners	$\mathbb{R}^+$
	Hamming distance	[0, 1]
	Width-height ratio	[0, 1]
Size	Width	$\mathbb{R}^+$
	Height	$\mathbb{R}^+$
	Area	$\mathbb{R}^+$
	Bounding-box area	$\mathbb{R}^+$
	bb-area ratio	[0, 1]
	Image-area ratio	[0, 1]
Material	Bright-pixels	[0, 1]
	Dark-pixels	[0, 1]
	Angle	[0, 180]
	x-coordinate	[0, 480]
	y-coordinate	[0, 320]

**FIGURE 4 |** The concept CUBE is linked to a weighted set of attributes. The weight represents the certainty of an attribute belonging to the concept. Each attribute is modeled as a normal distribution that keeps track of its prototypical value (i.e., the mean) and the standard deviation. The values between square brackets denote two standard deviations from the mean. These are not used in similarity calculations directly, but give an indication of the observed range of prototypical values.

and the ability to update an existing concept representation. The former is used by the learner during interpretation, while the latter is used during alignment.

3.3.1. Matching a Concept to an Object

In order to match a concept to an object from the environment, we should foresee some form of distance or similarity measure.

Based on this measure, the agent can decide whether or not a particular concept is applicable or discriminative for a particular object, e.g., during interpretation. This idea is similar to Wellens (2012), since it allows an agent to use a concept even if it does not exactly match a particular object. However, Wellens (2012) only considers symbolic attributes, allowing him to implement such a measure using set operations. In this work, we make use of a continuous similarity measure. Specifically, the similarity between a concept C and an object O can be computed by the average similarity between each of the attributes, weighted by the certainty that an attribute belongs to the concept. Formally, the similarity $S(C, O)$ is implemented as follows:

$$S(C, O) = \frac{1}{|A_C|} \sum_{a \in A_C} c(C_a) * S'(C_a, O_a) \quad (1)$$

where A_C is the set of attributes linked to concept C , $|A_C|$ represents the number of attributes, $c(C_a)$ returns the certainty score for a certain attribute a in concept C and C_a and O_a represent the attribute value for the attribute a in the concept C and object O , respectively.

Given the above definition of a similarity measure S between a concept and an object, we need the similarity measure S' for a particular attribute a of the concept and object, respectively. For this, we represent the attribute value within a concept (C_a) as a normal distribution. The similarity function S' is based on the z -score of the attribute value of the object (O_a) with respect to this normal distribution. We embed the z -score in a linear function to transform a small z -score in a high similarity value and a large z -score in a low similarity value. This function maps a z -score of 0 to a similarity of 1 and when a z -score reaches 2, the similarity has dropped to 0. If the z -score would be larger than 4, the similarity is cut off at -1 . The similarity measure S' can be expressed with the following equation:

$$S'(C_a, O_a) = \max\left(\frac{|1 - z_{O_a}|}{2} + 1, -1\right) \quad (2)$$

where z_{O_a} refers to the z -score of the attribute value of the object O_a with respect to the attribute of the concept, C_a , represented as a normal distribution.

Given that the similarity function S' returns a value between -1 and 1 and the score is always between 0 and 1, the similarity measure S also returns a value between -1 and 1.

3.3.2. Updating Concepts

After each game, the concept used in that game can be updated in terms of both the prototypical value and the certainty score of each attribute. This way, the agent can gradually shape its concept representation to fit the environment, again similar to Wellens (2012). The update mechanism relies on the feedback given by the tutor after the interaction. Specifically, the learner will update the concept it used during the interaction to be closer to or better fit with the topic object. This update procedure works in two steps:

1. The agent updates the prototypical value of all attributes in the concept. Here, we choose to update all attributes

since the certainty scores of the attributes might not yet be stable. When a particular attribute suddenly becomes important, e.g., because of changes in the environment, we also want its value to reflect the examples already seen. The update mechanism makes use of Welford's online algorithm (Welford, 1962). This is an online algorithm that specifies recurrence relations for the mean and standard deviation. This allows us to recompute the mean and standard deviation of the distribution by adding a single observation, without the need to store all observations. On the implementation level, each attribute keeps track of the number of observations N , the prototypical value p_n and the sum of squared differences from the current mean $M_{2,n}$ with n denoting the current interaction. The latter is initialized at 0.05. Given a new observation x_n , these values can be updated using the following equations:

$$\begin{aligned} N &= N + 1 \\ \delta_1 &= x_n - p_{n-1} \\ p_n &= p_{n-1} + \frac{\delta_1}{N} \\ \delta_2 &= x_n - p_n \\ M_{2,n} &= M_{2,n-1} + (\delta_1 * \delta_2) \end{aligned}$$

The standard deviation, required in the similarity calculations discussed above, can be computed from N and $M_{2,n}$ as follows:

$$\sigma = \sqrt{\frac{M_{2,n}}{N}}$$

2. The agent will increase the certainty of the subset of attributes that is most discriminative for the topic. The certainty score is decreased for all other attributes. A subset of attributes is discriminative when it is more similar to the topic than to any other object in the scene. Since this can be true for multiple subsets, we define the most discriminative subset as the one where the difference between the similarity to the topic and the most similar other object is maximized. Thus, during the update procedure, we not only make use of the topic object itself, but also compare this to other objects in the scene. This ensures that the combination of attributes, and ultimately the entire repertoire of concepts, is functionally relevant in the agent's environment. To compute the most discriminative subset of attributes, we make use of the similarity functions S and S' as defined above. Finally, to reduce the computational load, not all subsets of attributes are considered. These are filtered to contain at least the set of attributes that are discriminative on their own. The procedure to update the certainty scores can be summarized as follows:

- Identify the discriminative attributes, i.e., attributes that are more similar to the topic than to any other object in the scene. Here, we use similarity function S' . This yields e.g., *area* and *nr-of-corners*.
- Compute all subsets of the attributes of the concept.

- Filter all subsets to contain at least the attributes found in the first step. This yields subsets such as *{area, nr-of-corners}*, *{area, nr-of-corners, wh-ratio}*, *{area, nr-of-corners, roughness}*, etc.
- Find discriminative subset(s) of attributes, i.e., the subset for which the similarity to the topic is larger than to any other object in the scene. Here, we use similarity function S .
- The previous step can produce multiple subsets. We take the one that maximizes the difference in similarity between the topic and the most similar other object.
- Increase the certainty score of the attributes in this subset, and decrease the certainty score of all other attributes.

While this concept representation is easy to grasp, there is however an important assumption, namely that the attribute values are modeled using normal distributions. Statistical testing, using the normality test by D'Agostino and Pearson (d'Agostino, 1971; D'Agostino and Pearson, 1973), tells us that this is not the case for any of the attributes. The distributions of the attributes do come close to normal distributions but have thinner tails at both ends. Still, this can be viewed as odd, especially for some of the studied concepts. Take the concept *LEFT* as an example. It is important to note that the concept of *LEFT* refers to "left in the image" and not "left of another object." With this definition of left, the x-coordinate is an important attribute for this concept. If we consider the images of the CLEVR dataset, the x-coordinate of an object can be anywhere between 0 and 480. In this setting, we consider an object to be *LEFT* when the x-coordinate is smaller than 240. The bulk of objects that can be considered *LEFT* will not be close to 0, nor close to 240, but somewhere in between, e.g., around x-coordinate 170. From this, it is easy to see that our assumption will not cause many issues in this particular dataset, but in general one could argue that objects with an x-coordinate smaller than 170 can actually be considered "more left," while objects with an x-coordinates larger than 170 are gradually "less left." This is currently not captured by our concept representation.

3.4. Tutor Behavior

As mentioned in section 3.1, the tutor looks for the smallest set of concepts that discriminates the topic from the other objects in the scene, based on the symbolic ground-truth annotation of the scene. Given a topic that can be described symbolically as (*GREEN, CUBE, LARGE, RUBBER, LEFT, FRONT*), the tutor will try to describe this with a single concept. Traversing the concepts of the topic in a random order, the tutor will check if no other objects in the scene share this concept. For example, if the topic is the only cube in the scene, the concept *CUBE* will be returned. In most experiments, we restrict the tutor to only use a single concept to describe an object. In some scenes, however, it is impossible to describe an object with a single, discriminative concept. When this is the case, the tutor will choose a new topic object or sample a new scene.

In the compositional learning experiment, discussed in section 4.4, we lift the single-word restriction. There, if no single

discriminative concept can be found, the tutor will try all subsets of two concepts. For example, there might be multiple cubes and multiple green objects, but exactly one green cube. In this case, the combination of GREEN and CUBE is discriminative. Again, these subsets are considered in a random order. This procedure can be repeated for subsets of three concepts and four concepts, until a discriminative subset is found.

4. EXPERIMENTAL SETUP

In this section, we describe the various experiments designed to showcase different aspects of the proposed approach to concept learning. In the first experiment, we establish the baseline performance of our approach (section 4.1). In the following experiments, we test how well the concepts generalize (section 4.2), how they can be learned incrementally (section 4.3), and how they can be combined compositionally (section 4.4). A graphical overview of the experiments is given in **Figure 5**.

4.1. Transparent, Multi-Dimensional Concepts

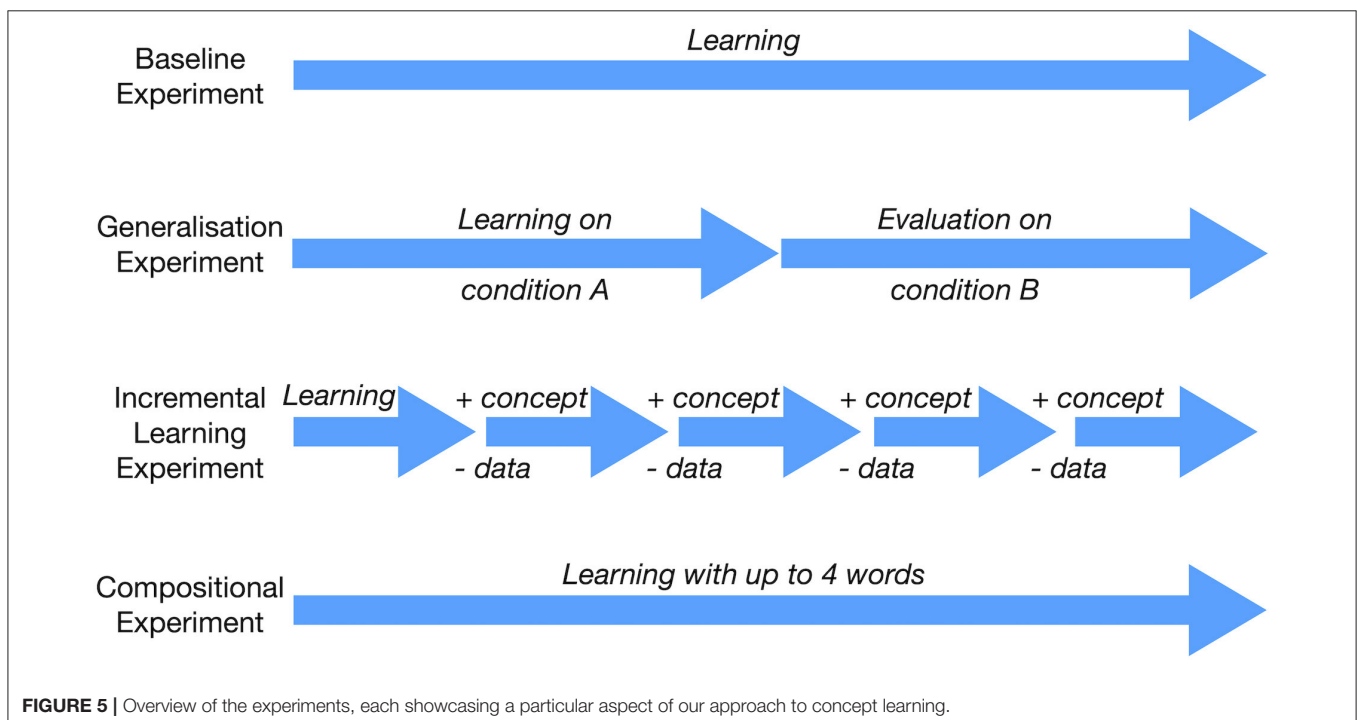
In the first experiment, we validate the learning mechanisms through the language game setup laid out in section 3.1. We compare the learner's performance both using simulated (section 3.2.2) and more realistic (section 3.2.3) continuous-valued attributes. In both cases, we use scenes from the validation split of the CLEVR dataset. The learner agent is evaluated in terms of communicative success and concept repertoire size. Our goal is to validate whether or not the agent can successfully acquire and use the concepts known by the tutor. Additionally, we examine the acquired concepts to see if the agent finds combinations of attributes that are relevant in the present environment.

4.2. Generalization

Using the CLEVR CoGenT dataset (Johnson et al., 2017), we test if the acquired concepts are general enough to extend to unseen instances and combinations of attributes. The CLEVR CoGenT dataset consists of two conditions. In condition A, cubes can be gray, blue, brown, or yellow, cylinders are red, green, purple, or cyan and spheres can have any of these colors. In condition B, the color options for cubes and cylinders are swapped. Like the original CLEVR dataset, the CoGenT data comes with a symbolic annotation that can be transformed into continuous-valued attributes using the methods described in section 3.2. Our goal is to validate if the learner agent truly learns the concepts, independently from the statistical distribution or co-occurrences in the environment. We evaluate this by playing a number of interactions in condition A, after which we switch off learning, followed by a number of games in condition B to evaluate the communicative success. Here, we expect to see that the communicative success remains stable between condition A and B, indicating that the concepts acquired by the agent do not rely on co-occurrences in the environment, as is often the case for other types of models. Additionally, by varying the number of interactions in condition A, we gain insight into how quickly the learner can acquire concepts that are functional in the world.

4.3. Incremental Learning

By incrementally expanding the environment, we demonstrate the adaptivity and open-endedness of our concept learning approach. For this experiment, we created our own variation on the CLEVR dataset consisting of five splits. In each split, more concepts are added and less data is available. In the first split, we offer 10,000 images where all objects are large, rubber cubes in



four different colors. In the second split, there are 8,000 images and these cubes can be large or small. Spheres and cylinders are added in the third split and the data is reduced to 4,000 scenes. The fourth split again halves the amount of data and metal objects are added. Finally, in the fifth split, four more colors are added and only 1,000 scenes are available. The splits are summarized in **Table S1**.

The learner agent is exposed to each of the splits consecutively, without resetting its repertoire of concepts or switching off the learning operators. We monitor the communicative success and the concept repertoire size throughout the entire experiment. Our goal in this experiment is two-fold. First, we show that the learning mechanisms can easily and quickly adjust to a changing environment. There is no need to fully or even partially retrain the repertoire when new concepts become available, nor to specify the number of concepts that are to be learned in advance, as would be the case for other types of models. By looking at the evolution of the concepts, we can study how certain attributes might become more or less important as the environment changes. Second, we again show the data efficiency of our approach by reducing the available number of scenes throughout the splits.

4.4. Compositional Concepts

The concept representation, as described in section 3.3, can be easily extended to compositional, multi-word utterances. In order to do so, the weighted set representation of multiple concepts needs to be combined. This is achieved by an operation similar to the union operator from fuzzy-set theory (Zadeh, 1965). Given two concepts, C_1 and C_2 , their corresponding sets of attributes are combined such that for each attribute that occurs in both concepts, the one with the highest certainty score is chosen. This is illustrated in **Figure 6**.

In this experiment, the tutor can use up to four words to describe the topic object. When all words in the utterance are unknown to the learner, it adopts all of them with the topic object being the initial seed. If all words are known, the learner performs the alignment using the composed concept. Due to this, not all attributes of all involved concepts will receive an updated prototypical value and certainty score, but only those that occur

in the combined concept. For example, in the combined concept “ C_1+C_2 ” from **Figure 6**, attributes “a-2” and “a-3” from concept “ C_1 ” and attributes “a-1” and “a-7” from concept “ C_2 ” will receive an update. Finally, if some words of the utterance are known and others are unknown, the learner will first adopt the unknown words and then perform alignment using the known words. In this experiment, we investigate how the communicative success, the learning speed and the resulting concepts of the agent are affected in the multi-word utterance setting and compare this to the single-word experiment described in section 4.1.

5. RESULTS

In this section, we elaborate on the results of the experiments described above. In order to produce the plots, we ran all experiments five times for 10,000 interactions and averaged the results. The error bars show the standard deviation. The plots were created using a sliding window of 250 interactions. All experiments were run on the validation split of the CLEVR dataset (15K scenes), using a randomly sampled scene for every interaction. The experiments were implemented using the open-source Babel toolkit (Loetzsch et al., 2008; Nevens et al., 2019).

5.1. Transparent, Multi-Dimensional Concepts

In the first experiment, we validate the learning mechanisms proposed earlier in this paper. We evaluate the learner agent on its ability to successfully communicate and on its repertoire of concepts, both in the more simple, simulated environment and in the more realistic, noisy environment. In **Figure 7A**, we show the communicative success of the agents in these environments. The agents are able to achieve 100% communicative success in the simulated world, after merely ~500 interactions. From the same figure, we see that the learning mechanisms perform somewhat less good in the more realistic, noisy environment. The agents achieve a fairly stable level of communicative success after ~500 interactions, reaching 91% communicative success (0.3% standard deviation).

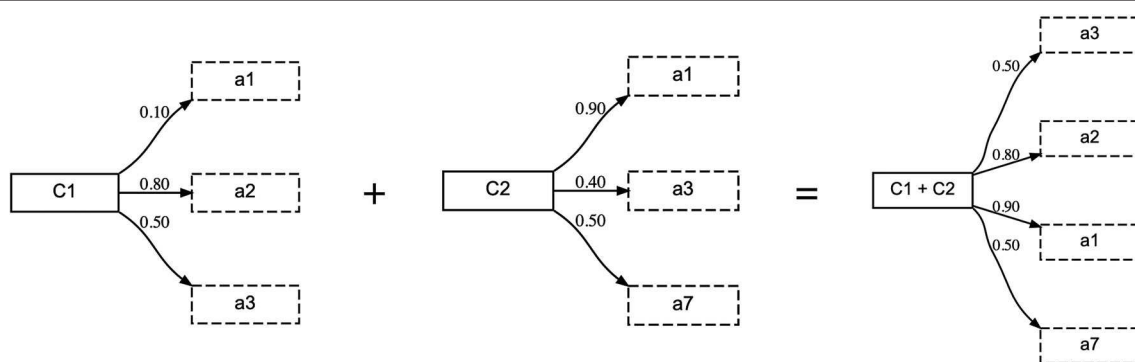


FIGURE 6 | When combining concepts compositionally, the same attribute can occur multiple times. In this case, the resulting concept takes the one with the highest certainty score.

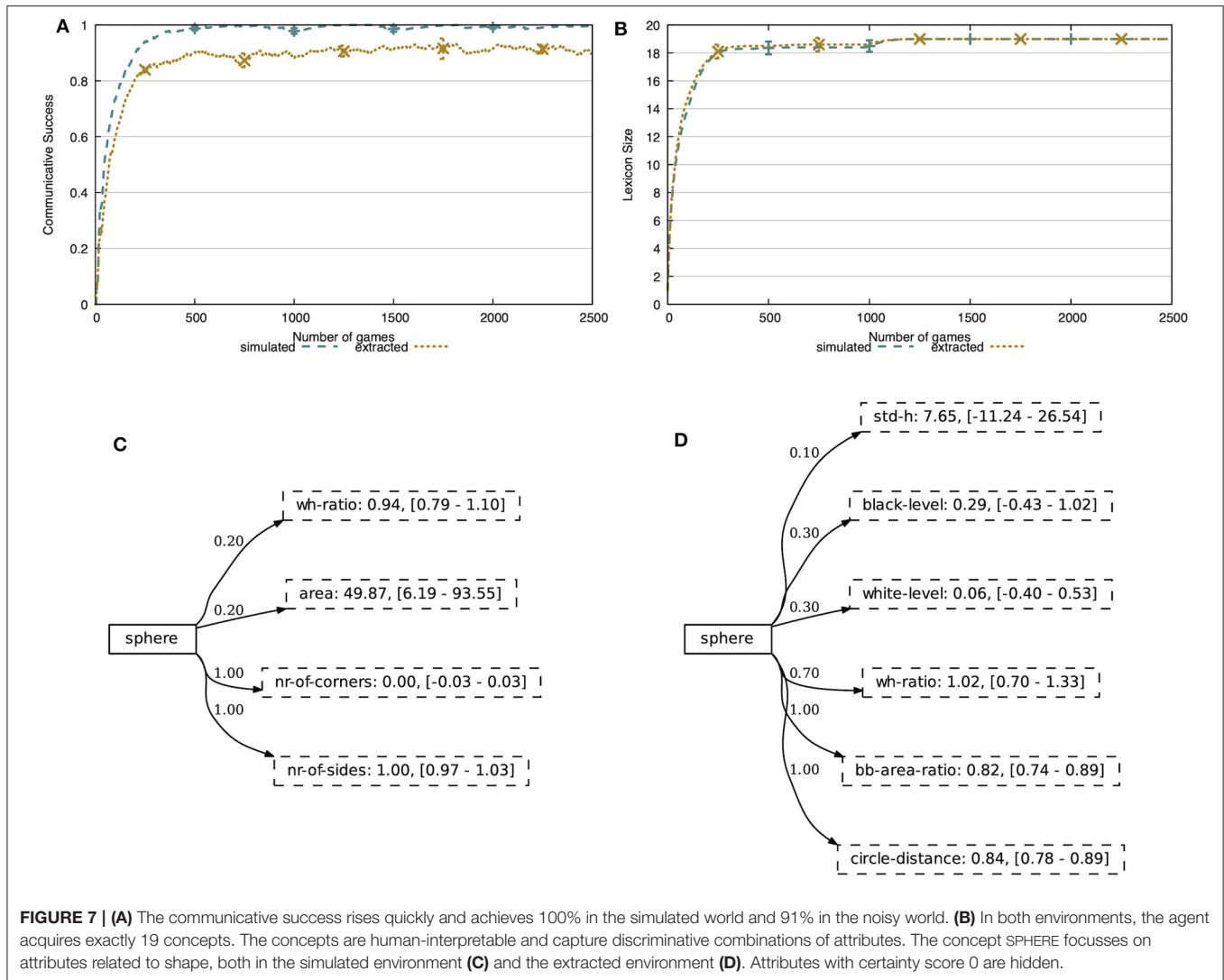


FIGURE 7 | (A) The communicative success rises quickly and achieves 100% in the simulated world and 91% in the noisy world. **(B)** In both environments, the agent acquires exactly 19 concepts. The concepts are human-interpretable and capture discriminative combinations of attributes. The concept SPHERE focusses on attributes related to shape, both in the simulated environment **(C)** and the extracted environment **(D)**. Attributes with certainty score 0 are hidden.

Figure 7B shows the lexicon size of the learner agent in both environments. Just like the communicative success, we see that it quickly increases and stabilizes at 19 concepts, which are all concepts present in the CLEVR dataset. We cut off these figures after 2,500 of the 10,000 interactions, since the metrics reached a stable level.

The concept representation proposed in this work allows for a clear and easy to interpret view on the learned concepts. We demonstrate this in Figures 7C,D, showing the concept SPHERE obtained after 5,000 interactions in both the simulated and noisy environments. In both cases, we see that a few attributes have become important for the learner, reflected by the high certainty scores. In the simulated world, these are *nr-of-corners* and *nr-of-sides*, while in the noisy world these are the *width-height ratio*, the *circle-distance* and *bb-area-ratio*. The *circle-distance* attribute represents the Hamming distance between the contour of the object and the minimal enclosing circle and the *bb-area-ratio* attribute represents the ratio between the area of the object and the area of its bounding box. All of these attributes are indeed intuitively shape-related. We give an overview of all learned

concepts obtained in the simulated world and the noisy world in Figures S1, S2, respectively.

With this experiment, we have shown that the learner agent can automatically distill meaningful concepts from a stream of continuous data, in the form of discriminative subsets of attributes and their prototypical values, and is able to successfully use them in communication. Furthermore, as these concepts are expressed using human-interpretable feature channels, the model and resulting repertoire of concepts is completely transparent.

5.2. Generalization

In the generalization experiment, we show that the agent's ability to learn the concepts is completely independent from the statistical distributions or co-occurrences in the dataset. For this experiment, we use the CLEVR CoGenT dataset, which consists of two conditions. The agent first learns during a number of interactions in condition A. Afterwards, learning operators are turned off and we evaluate the communicative success of the agent in condition B for the remainder of the interactions. We

expect the agents to remain at a stable level of communicative success when making the transition from condition A to B. We again evaluate on both the simulated environment and the noisy environment. Additionally, we vary the amount of training interactions on condition A to test the speed at which the learner agent can acquire useable concepts.

In **Figure 8**, we show the communicative success of the agents both during learning in condition A and evaluation in condition B. From this figure, it is clear that the learner agent cannot reach the same level of success as the previous experiment after 100 training interactions. However, with only 500 training interactions this level of success is achieved. This indicates that the learner's repertoire of concepts is shaped quickly and is sufficient to have successful interactions. Additionally, when transitioning from condition A to B, there is no decrease in communicative success in the simulated environment and only a minor decrease in the noisy environment. This indicates that the concepts acquired by the agent abstract away over the observed instances.

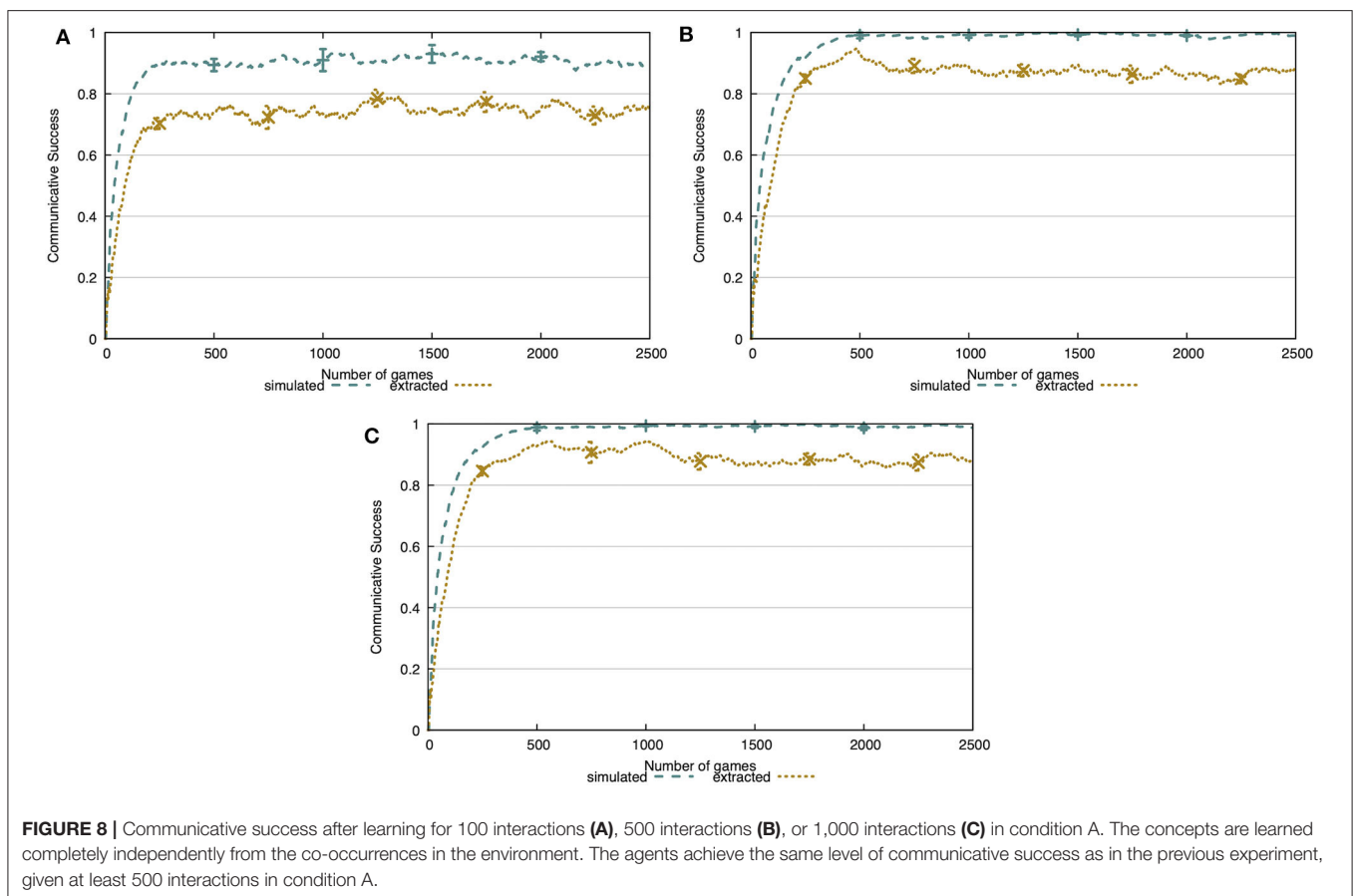
To further investigate the generalization abilities of the learner, we study the acquired concepts. Remember that in condition A in the CoGenT dataset, cubes can be gray, blue, brown, or yellow, cylinders have a set of different colors and spheres can be any color. In **Figure 9**, we study the concept representation of the colors for cubes after being learned on condition A for 500 interactions. If the agent would rely on

co-occurrences of the dataset, the concept representation of these colors could contain attributes related to shape, since each time one of these colors occurs it is either a cube or a sphere. Additionally, the cube and sphere have the same value for the *wh-ratio* attribute, so it could be considered discriminative in some cases. From **Figure 9**, we see that even though this feature is present in some of the concepts, its certainty score is very low. Hence, the agent does not focus on particular dataset co-occurrences and is able to generalize over various observations. We attribute this to the notion of discrimination, which will make sure that only relevant attributes obtain a high certainty score.

5.3. Incremental Learning

Our approach to concept learning is completely open-ended and has no problems dealing with a changing environment. We validate this through an incremental learning experiment where, over the course of 10,000 interactions, the number of available concepts increases. We vary the amount of interactions before new concepts are introduced between 100, 500, and 1,000 interactions. The learning mechanisms are able to adjust almost instantly to these changes, as is shown in **Figure 10**. In the simulated world, we see minor drops in communicative success when transitioning from one phase to the next. These are more present in the noisy world, but the agent quickly recovers from it.

If we investigate the concepts in the incremental learning experiment, we find that the relevant attributes have obtained



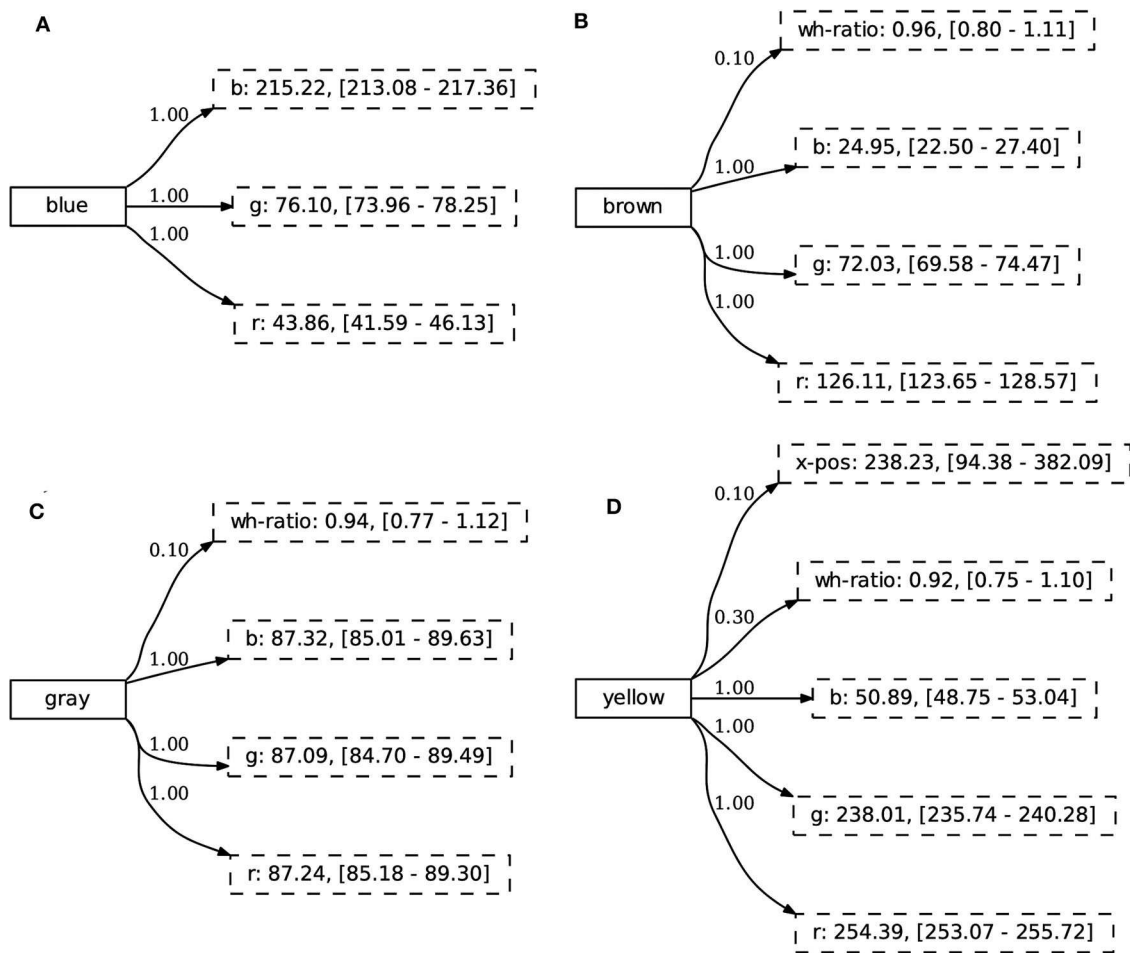


FIGURE 9 | A subset of the agent's repertoire of concepts after the generalization experiment. In condition A, the concepts BLUE (A), BROWN (B), GRAY (C), and YELLOW (D) are always observed as cubes or spheres. The agent is not "distracted" by statistical distributions of the environment and learns combinations of attributes that are relevant to solve the communicative task.

a high certainty score already after the first phase of the experiment (see **Figure 11**). Consequently, these remain stable over the various phases, while other attributes never achieve high certainty scores. Additionally, we note that the resulting concepts have the same high-scoring attributes as those obtained in the baseline experiment, independent of the phase in which they were introduced (see **Figure 12**).

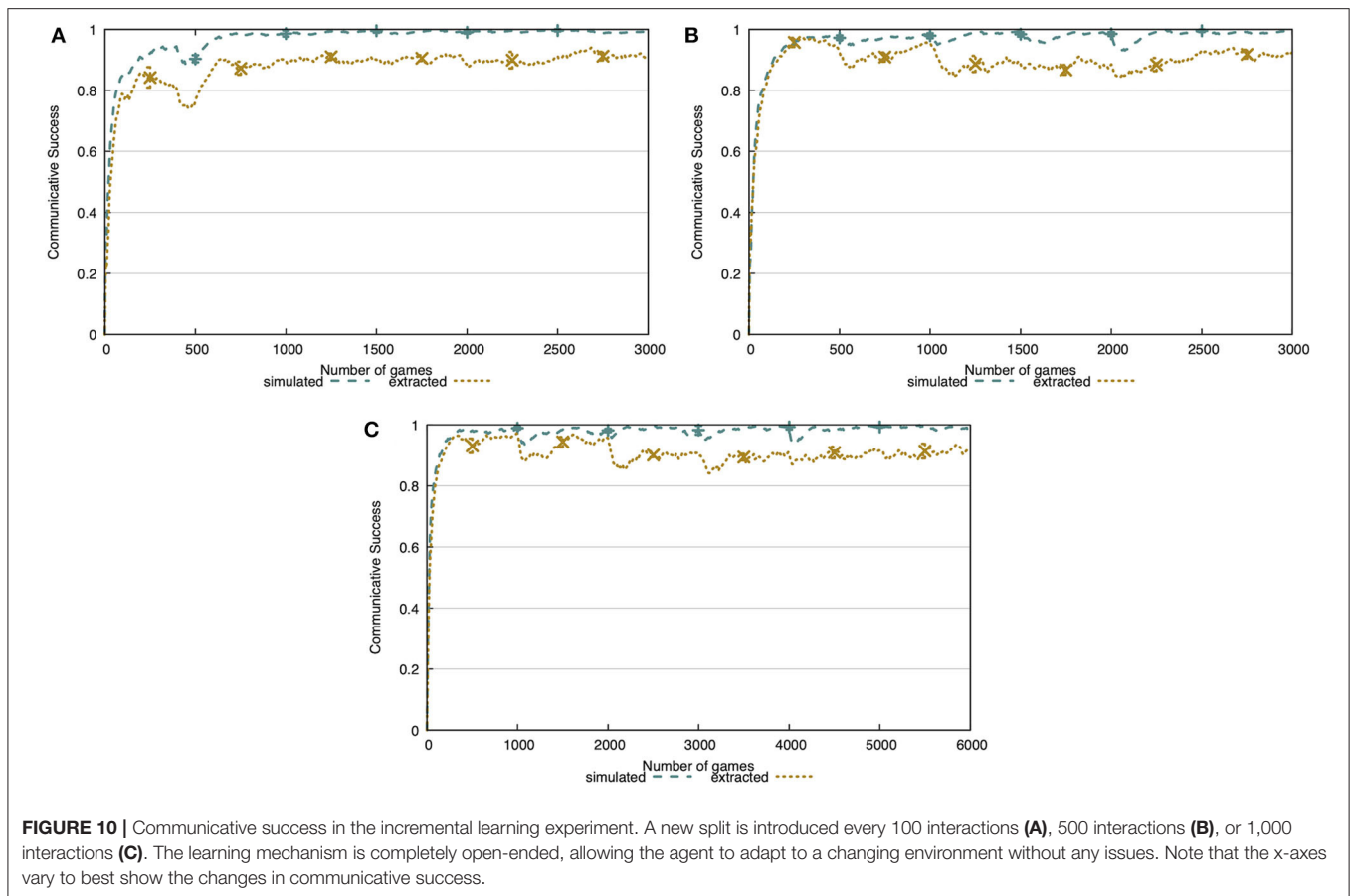
5.4. Compositional Utterances

In the final experiment, we find that the agent is successful at learning the separate concepts, even if they are combined in compositional utterances. To test this, we allow the tutor to use up to four words when describing an object. It is important to note that the tutor will always generate the shortest discriminative utterance, as described in section 3.4. In **Figure 13**, we measure how often the tutor uses different utterance lengths. From this, it is clear that most objects can be described using a single word. Slightly less than 40% of objects

require two words to be discriminative and only very few objects are described with three words.

In **Figure 14**, we compare the communicative success when a tutor uses a single word (and skips scenes where this is not possible) and when the tutor uses up to four words. In the simulated environment (**Figure 14A**), communicative success drops 3 percentage points to 97%. In the noisy environment (**Figure 14B**), the communicative success drops 8 percentage points to 83%. With this experiment, we show that the agent is capable of extracting the discriminative attributes and their prototypical values for each concept and, at the same time, learning the meaning of each word separately in a multi-word utterance.

Finally, we consider the repertoire of concepts and find, similar to the first experiment, that the agent has found discriminative sets of attributes that are intuitively related to the concept they describe. The concept METAL is shown in **Figure 15**, both for the simulated and noisy environment. Interestingly, we note from this Figure that the agent has learned to identify the



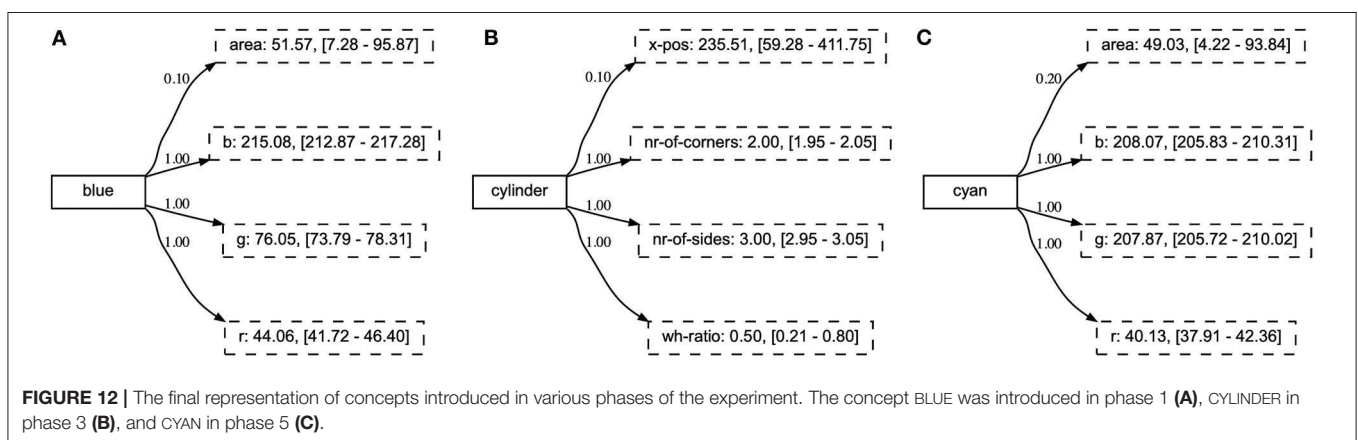
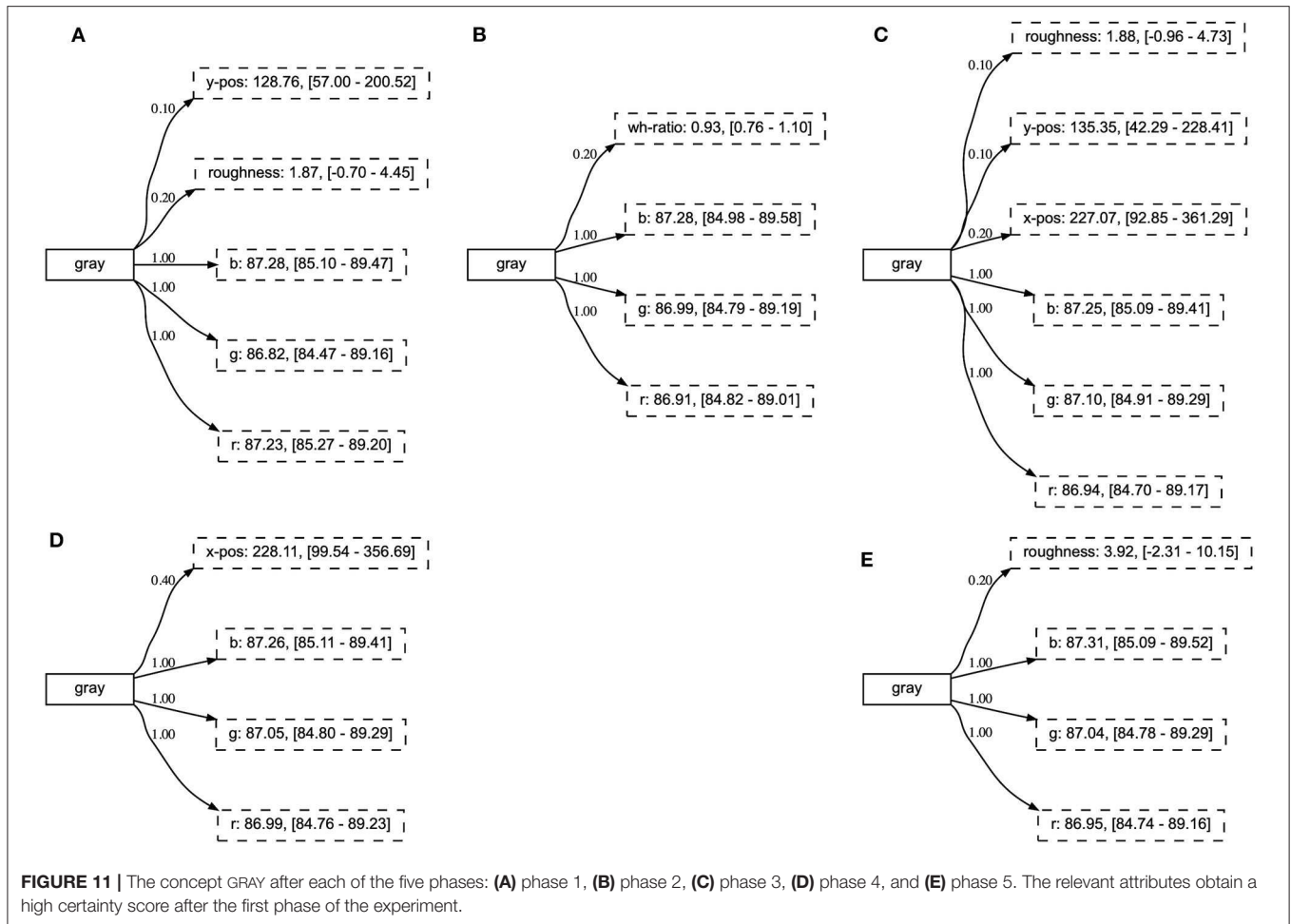
material of an object through the “value” dimension of the HSV color space.

6. DISCUSSION AND CONCLUSION

In order to be able to communicate and reason about their environment, autonomous agents must be able to abstract away from low-level, sensori-motor data streams. They therefore require an abstraction layer that links sensori-motor experiences to high-level symbolic concepts that are meaningful in the environment and task at hand. A repertoire of meaningful concepts provides the necessary building blocks for achieving success in the agent’s higher-level cognitive tasks, such as reasoning or action planning. Similar to how humans can grasp a concept after only a few exemplars, an autonomous agent should ideally acquire these concepts quickly and with relatively little data. Learned concepts should be general enough to extend to similar yet unseen settings. As the environment of the agent can change or new concepts can be introduced at any time, the learning methodology should also be adaptive and allow for incremental learning. Finally, to truly understand the reasoning processes of an autonomous agent, its learning mechanisms and representations should be fully transparent and interpretable in human-understandable terms.

The task of concept learning has been considered in various subfields of AI. Deep Learning approaches, for example, offer a very powerful paradigm to extract concepts from raw perceptual data, achieving impressive results but thereby sacrificing data efficiency and model transparency. Version space learning offers a more interpretable model but has difficulties in handling noisy observations. Most similar to the approach presented in this paper is work from the robotics community, considering tasks such as perceptual anchoring and affordance learning. However, these tasks focus mostly on a single robot extracting concepts from observations of the world around it. In this work, we argue for interactive learning through the language game paradigm. The notion of discrimination plays a central role in forming the concepts, thereby ensuring the generality and adaptivity of the concepts such that these are relevant in the agent’s environment. Additionally, our method offers an explainable concept representation, acquired through a data efficient and incremental method. Each of these properties was highlighted in a dedicated experiment.

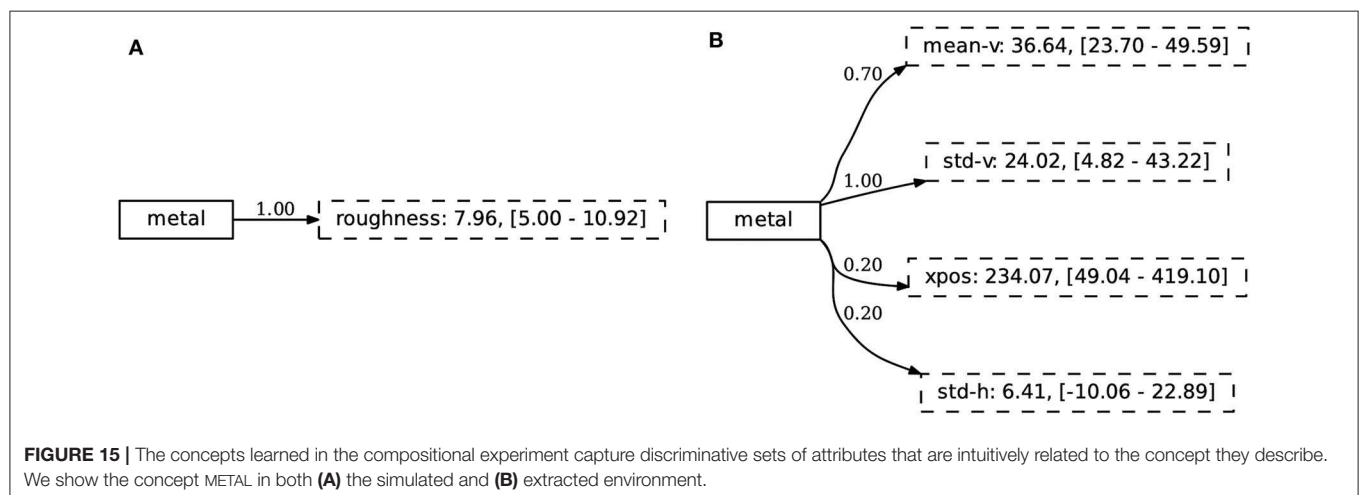
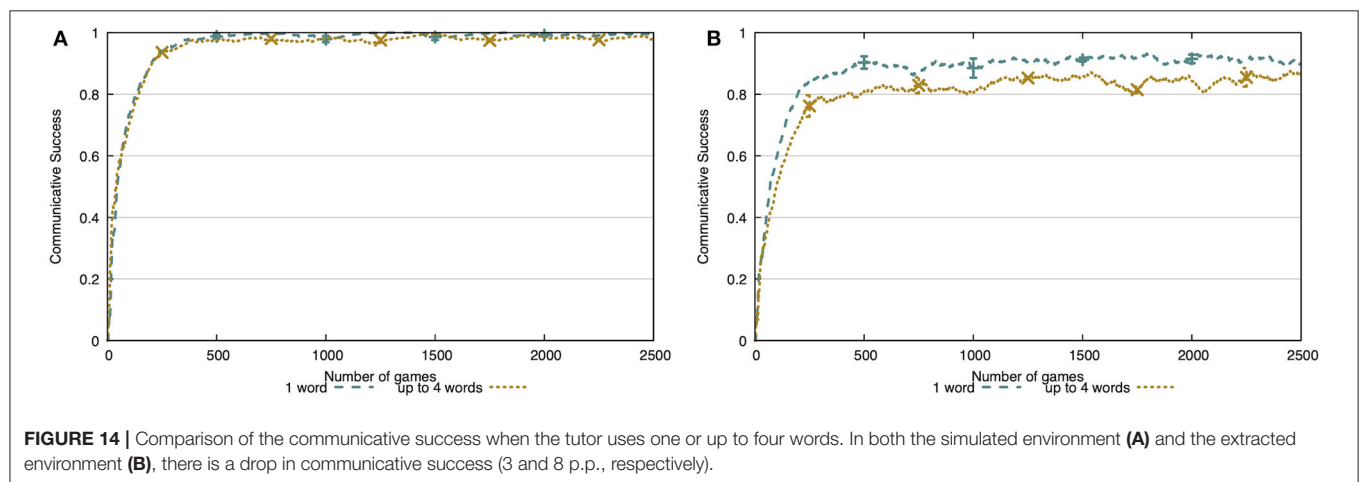
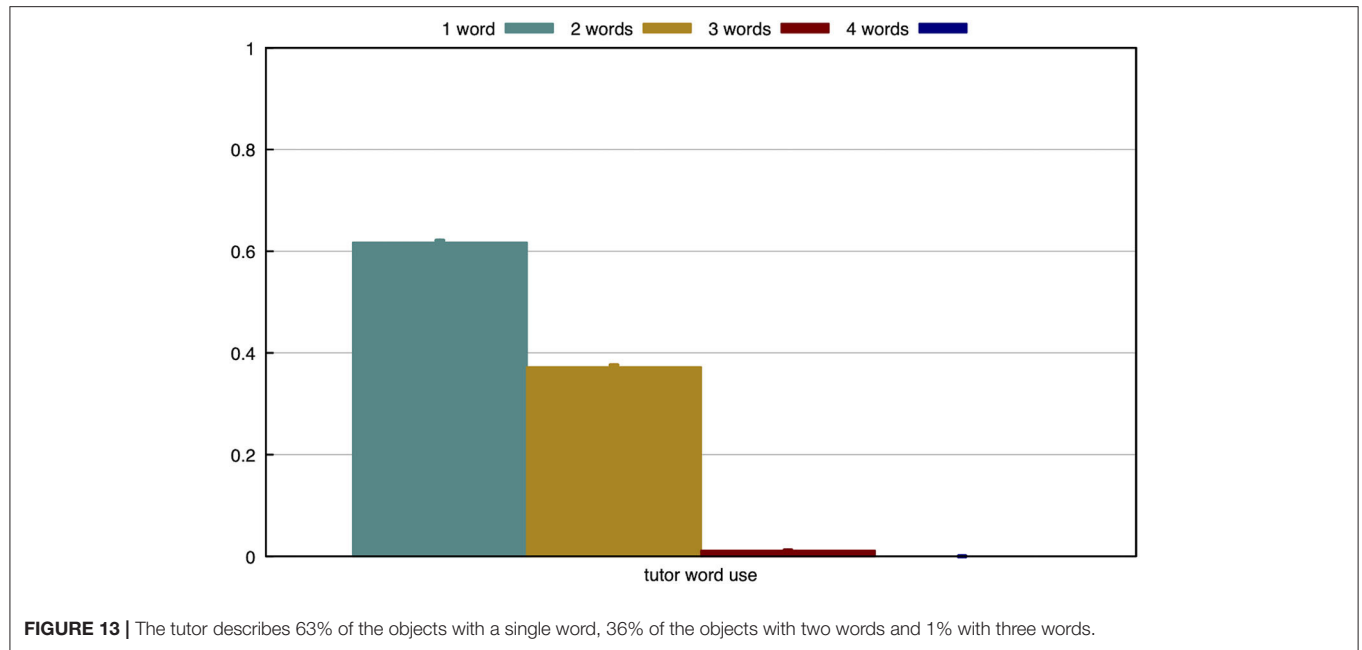
In sum, we have presented a novel, discrimination-based approach to learning meaningful concepts from streams of sensory data. For each concept, the agent finds discriminative attribute combinations and their prototypical values. We have shown that these concepts (i) can be acquired quickly with relatively few data points, (ii) generalize well to unseen instances,



(iii) offer a transparent and human-interpretable insight in the agent's memory and processing, (iv) are adaptive to changes in the environment, and (v) can be combined compositionally. These properties make this work highly valuable for the domains of robotics and interactive task learning, where interpretability, open-endedness and adaptivity are important factors. Once a repertoire of symbolic concepts, abstracting away over the

sensori-motor level, has been acquired, an autonomous agent can use it to solve higher-level reasoning tasks such as navigation, (visual) question answering, (visual) dialog and action planning.

In order to ensure that the learned concepts are human-interpretable, the methodology starts from a predefined set of human-interpretable features that are extracted from the raw



images. While we argue that this is necessary to achieve true interpretability, it can also be seen as a limitation inherent to the methodology. However, this limitation cannot be lifted without losing interpretability that the method brings.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

All authors contributed to the conceptual foundations and article writing. JN conducted the experiments.

REFERENCES

- Aoki, T., Nishihara, J., Nakamura, T., and Nagai, T. (2016). "Online joint learning of object concepts and language model using multimodal hierarchical dirichlet process," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon: IEEE), 2636–2642. doi: 10.1109/IROS.2016.7759410
- Beuls, K., and Steels, L. (2013). Agent-based models of strategies for the emergence and evolution of grammatical agreement. *PLoS ONE* 8:e58960. doi: 10.1371/journal.pone.0058960
- Bleys, J. (2016). *Language Strategies for the Domain of Colour*. Berlin: Language Science Press. doi: 10.26530/OAPEN_603341
- Bruner, J. S., Goodnow, J. J., and Austin, G. A. (1956). *A Study of Thinking*. New York, NY: John Wiley and Sons, Incorporated.
- Coradeschi, S., and Saffiotti, A. (2003). An introduction to the anchoring problem. *Robot. Auton. Syst.* 43, 85–96. doi: 10.1016/S0921-8890(03)00021-6
- d'Agostino, R. B. (1971). An omnibus test of normality for moderate and large size samples. *Biometrika* 58, 341–348. doi: 10.1093/biomet/58.2.341
- D'Agostino, R. B., and Pearson, E. S. (1973). Tests for departure from normality. Empirical results for the distributions of b^2 and $\sqrt{b}$. *Biometrika* 60, 613–622. doi: 10.1093/biomet/60.3.613
- Dolgikh, S. (2018). Spontaneous concept learning with deep autoencoder. *Int. J. Comput. Intell. Syst.* 12, 1–12. doi: 10.2991/ijcis.2018.25905178
- Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., and He, K. (2018). *Detectron*. Available online at: <https://github.com/facebookresearch/detectron>
- Grossman, M., Smith, E. E., Koenig, P., Glosser, G., DeVita, C., Moore, P., et al. (2002). The neural basis for categorization in semantic memory. *Neuroimage* 17, 1549–1561. doi: 10.1006/nimg.2002.1273
- Han, C., Mao, J., Gan, C., Tenenbaum, J., and Wu, J. (2019). "Visual concept-metaconcept learning," in *Advances in Neural Information Processing Systems*, eds H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Vancouver, BC: Neural Information Processing Systems Foundation), 5002–5013.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (Venice)*, 2961–2969. doi: 10.1109/ICCV.2017.322
- Higgins, I., Matthey, L., Glorot, X., Pal, A., Uria, B., Blundell, C., et al. (2016). Early visual concept learning with unsupervised deep learning. *arXiv preprint arXiv:1606.05579*.
- Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. (2017). "CLEVR: a diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (San Francisco, CA), 2901–2910. doi: 10.1109/CVPR.2017.215
- Konidaris, G., Kaelbling, L., and Lozano-Perez, T. (2014). "Constructing symbolic representations for high-level planning," in *Twenty-Eighth AAAI Conference on Artificial Intelligence* (Québec City, QC).
- Konidaris, G., Kaelbling, L., and Lozano-Perez, T. (2015). "Symbol acquisition for probabilistic high-level planning," in *Twenty-Fourth International Joint Conference on Artificial Intelligence* (Buenos Aires).
- Konidaris, G., Kaelbling, L. P., and Lozano-Perez, T. (2018). From skills to symbols: Learning symbolic representations for abstract high-level planning. *J. Artif. Intell. Res.* 61, 215–289. doi: 10.1613/jair.5575
- Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science* 350, 1332–1338. doi: 10.1126/science.aab3050
- Lake, B. M., Salaz, R., and Tenenbaum, J. B. (2019). The omniglot challenge: a 3-year progress report. *Curr. Opin. Behav. Sci.* 29, 97–104. doi: 10.1016/j.cobeha.2019.04.007
- Lestrade, S. (2016). "The emergence of argument marking," in *The Evolution of Language: Proceedings of the 11th International Conference (EVLANGX11)*, eds S. Roberts, C. Cuskley, L. McCrohon, L. Barceló-Coblijn, O. Fehér, and T. Verhoef (New Orleans, LA). Available online at: <http://evolang.org/neworleans/papers/36.html>
- Loetzsch, M., Wellens, P., De Beule, J., Bleys, J., and van Trijp, R. (2008). *The Babel2 Manual*. Technical Report AI-Memo 01-08. Brussels: Artificial Intelligence Laboratory, Vrije Universiteit.
- Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., and Wu, J. (2019). "The neuro-symbolic concept learner: interpreting scenes, words, and sentences from natural supervision," in *International Conference on Learning Representations* (New Orleans, LA).
- McCarthy, R., and Warrington, E. K. (1990). The dissolution of semantics. *Nature* 343, 599–599. doi: 10.1038/343599a0
- Mitchell, T. M. (1982). Generalization as search. *Artif. Intell.* 18, 203–226. doi: 10.1016/0004-3702(82)90040-6
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236
- Nakamura, T., Nagai, T., and Iwahashi, N. (2007). "Multimodal object categorization by a robot," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Diego, CA: IEEE), 2415–2420. doi: 10.1109/IROS.2007.4399634
- Nevens, J., Van Eecke, P., and Beuls, K. (2019). "A practical guide to studying emergent communication through grounded language games," in *AISB Language Learning for Artificial Agents Symposium* (Falmouth), 1–8.
- Patalano, A. L., Smith, E. E., Jonides, J., and Koeppe, R. A. (2001). Pet evidence for multiple strategies of categorization. *Cogn. Affect. Behav. Neurosci.* 1, 360–370. doi: 10.3758/CABN.1.4.360
- Pauw, S., and Hilferty, J. (2012). The emergence of quantifiers. *Exp. Cult. Lang. Evol.* 3:277. doi: 10.1075/ais.3.14pau
- Persson, A., Zuidberg Dos Martires, P. M., De Raedt, L., and Loutfi, A. (2019). Semantic relational object tracking. *IEEE Trans. Cogn. Dev. Syst.* 12, 84–97. doi: 10.1109/TCDS.2019.2915763

FUNDING

The research reported in this paper was funded by the European Union's Horizon 2020 research and innovation programme under grant agreement no. 732942 and the Flemish Government under the Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen programme. JN was supported by the Research Foundation Flanders (FWO) through grant 1SB6219N.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2020.00084/full#supplementary-material>

- Rodriguez, R. C., Alaniz, S., and Akata, Z. (2019). "Modeling conceptual understanding in image reference games," in *Advances in Neural Information Processing Systems* (Vancouver, BC), 13155–13165.
- Rosch, E. H. (1973). Natural categories. *Cogn. Psychol.* 4, 328–350. doi: 10.1016/0010-0285(73)90017-0
- Şahin, E., Çakmak, M., Doğan, M. R., Uğur, E., and Üçoluk, G. (2007). To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adapt. Behav.* 15, 447–472. doi: 10.1177/1059712307084689
- Shi, J., Xu, J., Yao, Y., and Xu, B. (2019). Concept learning through deep reinforcement learning with memory-augmented neural networks. *Neural Netw.* 110, 47–54. doi: 10.1016/j.neunet.2018.10.018
- Spranger, M., and Steels, L. (2012). "Emergent functional grammar for space," in *Experiments in Cultural Language Evolution, Number 3 in Advances in Interaction Studies*, ed L. Steels (Amsterdam: John Benjamins), 207–232. doi: 10.1075/ais.3.11spr
- Squire, L. R., and Knowlton, B. J. (1995). Learning about categories in the absence of memory. *Proc. Natl. Acad. Sci. U.S.A.* 92, 12470–12474. doi: 10.1073/pnas.92.26.12470
- Steels, L. (1997). "Constructing and sharing perceptual distinctions," in *European Conference on Machine Learning* (Prague: Springer), 4–13. doi: 10.1007/3-540-62858-4_68
- Steels, L. (2001). Language games for autonomous robots. *IEEE Intell. Syst.* 16, 16–22. doi: 10.1109/MIS.2001.956077
- Sutton, R. S. (1996). "Generalization in reinforcement learning: successful examples using sparse coarse coding," in *Advances in Neural Information Processing Systems*, eds M. C. Mozer, M. I. Joran, and T. Petsche (Denver, CO: Neural Information Processing Systems Foundation), 1038–1044.
- Sutton, R. S., Precup, D., and Singh, S. (1999). Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. *Artif. Intell.* 112, 181–211. doi: 10.1016/S0004-3702(99)00052-1
- Taniguchi, A., Hagiwara, Y., Taniguchi, T., and Inamura, T. (2017). "Online spatial concept and lexical acquisition with simultaneous localization and mapping," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vancouver, BC: IEEE), 811–818. doi: 10.1109/IROS.2017.8202243
- Taniguchi, A., Taniguchi, T., and Inamura, T. (2016). Spatial concept acquisition for a mobile robot that integrates self-localization and unsupervised word discovery from spoken sentences. *IEEE Trans. Cogn. Dev. Syst.* 8, 285–297. doi: 10.1109/TCDS.2016.2565542
- Taniguchi, T., Ugur, E., Hoffmann, M., Jamone, L., Nagai, T., Rosman, B., et al. (2018). Symbol emergence in cognitive developmental systems: a survey. *IEEE Trans. Cogn. Dev. Syst.* 11, 494–516. doi: 10.1109/TCDS.2018.2867772
- Ugur, E., Oztop, E., and Sahin, E. (2011). Goal emulation and planning in perceptual space using learned affordances. *Robot. Auton. Syst.* 59, 580–595. doi: 10.1016/j.robot.2011.04.005
- Ugur, E., and Piater, J. (2015a). "Bottom-up learning of object categories, action effects and logical rules: from continuous manipulative exploration to symbolic planning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA: IEEE), 2627–2633. doi: 10.1109/ICRA.2015.7139553
- Ugur, E., and Piater, J. (2015b). "Refining discovered symbols with multi-step interaction experience," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)* (Seoul: IEEE), 1007–1012. doi: 10.1109/HUMANOIDS.2015.7363477
- van Trijp, R. (2016). *The Evolution of Case Grammar*. Berlin: Language Science Press. doi: 10.26530/OAPEN_611694
- Vogt, P. (2002). The physical symbol grounding problem. *Cogn. Syst. Res.* 3, 429–457. doi: 10.1016/S1389-0417(02)00051-7
- Wang, Q., Young, S., Harwood, A., and Ong, C. S. (2015). "Discriminative concept learning network: reveal high-level differential concepts from shallow architecture," in *2015 International Joint Conference on Neural Networks (IJCNN)* (Budapest: IEEE), 1–9. doi: 10.1109/IJCNN.2015.7280525
- Welford, B. (1962). Note on a method for calculating corrected sums of squares and products. *Technometrics* 4, 419–420. doi: 10.1080/00401706.1962.10490022
- Wellens, P. (2012). *Adaptive strategies in the emergence of lexical systems* (Ph.D. thesis). Vrije Universiteit Brussel, Brussels, Belgium.
- Xu, C., Guan, Z., Zhao, W., Niu, Y., Wang, Q., and Wang, Z. (2018). "Deep multi-view concept learning," in *IJCAI* (Stockholm), 2898–2904. doi: 10.24963/ijcai.2018/402
- Yi, K., Wu, J., Gan, C., Torralba, A., Kohli, P., and Tenenbaum, J. (2018). "Neural-symbolic VQA: disentangling reasoning from vision and language understanding," in *Advances in Neural Information Processing Systems*, eds S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Montréal, QC: Neural Information Processing Systems Foundation), 1031–1042.
- Zadeh, L. A. (1965). Fuzzy sets. *Inform. Control* 8, 338–353. doi: 10.1016/S0019-9958(65)90241-X

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Nevens, Van Eecke and Beuls. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Few-Shot Induction of Generalized Logical Concepts via Human Guidance

Mayukh Das^{1*}, Nandini Ramanan², Janardhan Rao Doppa³ and Srimaam Natarajan²

¹ Device Intelligence, Samsung R&D Institute India - Bangalore, Device Intelligence, Bangalore, India, ² Erik Jonsson School of Engineering and Computer Science (ECS), The University of Texas at Dallas, ECS, Dallas, TX, United States, ³ School of Electrical Engineering & Computer Science (EECS), Washington State University, EECS, Pullman, WA, United States

We consider the problem of learning generalized first-order representations of concepts from a small number of examples. We augment an inductive logic programming learner with 2 novel contributions. First, we define a distance measure between candidate concept representations that improves the efficiency of search for target concept and generalization. Second, we leverage richer human inputs in the form of advice to improve the sample efficiency of learning. We prove that the proposed distance measure is semantically valid and use that to derive a PAC bound. Our experiments on diverse learning tasks demonstrate both the effectiveness and efficiency of our approach.

Keywords: cognitive systems, logics for knowledge representation, relational learning, knowledge representation and reasoning, human in the loop (HITL)

OPEN ACCESS

Edited by:

Amy Loutfi,
Örebro University, Sweden

Reviewed by:

Nicos Angelopoulos,
Cardiff University, United Kingdom
Leonardo Trujillo,
Instituto Tecnológico de Tijuana,
Mexico

*Correspondence:

Mayukh Das
mayukh.das@samsung.com

Specialty section:

This article was submitted to
Computational Intelligence in
Robotics,
a section of the journal
Frontiers in Robotics and AI

Received: 14 May 2020

Accepted: 04 August 2020

Published: 18 November 2020

Citation:

Das M, Ramanan N, Doppa JR and
Natarajan S (2020) Few-Shot
Induction of Generalized Logical
Concepts via Human Guidance.
Front. Robot. AI 7:122.
doi: 10.3389/frobt.2020.00122

1. INTRODUCTION

We study the case of learning from *few examples*, of which *one-shot* learning is a special case (Lake et al., 2015). We consider a challenging setting—that of learning explainable, decomposable, and generalizable (first-order) concepts from few examples. Plan induction becomes a special case where a generalizable plan is induced from a single (noise-free) demonstration. As an example, consider building a tower that requires learning *L-shapes* as a primitive. In our formulation, the goal is to learn a *L-shape* from a single demonstration. Subsequently, using this concept, the agent can learn to build a rectangular base (with 2 *L-shapes*) from another single demonstration and so on till the tower is fully built. Concept learning has been considered as problem solving by reflection (Stroulia and Goel, 1994), mechanical compositional concepts (Wilson and Latombe, 1994), learning probabilistic programs (Lake et al., 2015), etc. While successful, they are not considered in one-shot learning except with SVM (Tax, 2001), or with a neural network (Kozierowski and Turk, 2018).

Our work has two key differences. First, we aim to learn an “easily interpretable,” “explainable,” “decomposable,” and “generalizable” concepts as first-order Horn clauses (Horn, 1951) (which are akin to If-Then rules). Second, and perhaps most important, we “do not assume the existence of a simulator (for plans) or employ a closed-world assumption” to generate negative examples. Inspired by Mitchell’s (1997) observation of futility of bias-free learning, we employ domain expertise as inductive bias. The principle of structural risk minimization (Vapnik, 1999) shows how optimal generalization from extremely sparse observations is quite difficult. The problem is difficult in structured domains since most relations are false. Thus, few-shot induction of generalized logical concepts is challenging. We employ iterative revision of first-order horn clause theories using a novel scoring metric and guidance from a human. *We emulate a “student” who learns a generalized concept from an example provided by the “teacher,” by both reflecting as well as asking relevant questions.*

We propose *Guided One-shot Concept Induction* (GOCI) for learning in relational domains¹. GOCI builds upon an inductive logic program (ILP) learner (Muggleton, 1991) with two key extensions. First, a modified scoring function that explicitly computes distances between concept representations. We show the relation to *Normalized Compression Distance* (NCD) for plan induction settings. Consequently, we demonstrate that NCD is a valid distance metric. Second, we use domain knowledge from human expert as inductive bias. Unlike many advice taking systems that employ domain knowledge before training, GOCI identifies the relevant regions of the concept representation space and *actively solicits guidance* from the human expert to find the target concept in a sample-efficient manner. Overall, these two modifications allow for more effective and efficient learning using GOCI that we demonstrate both theoretically and empirically.

We make the following key contributions:

1. We derive a new distance-penalized scoring function that computes definitional distances between concepts, henceforth termed as “conceptual distance.”
2. We treat the human advice as an inductive bias to accelerate learning. Our ILP learner actively solicits richer information from the human experts than mere labels.
3. Our theoretical analyses of GOCI prove that (a) our metric is indeed a valid distance, and (b) NCD between plans is a special case of our metric.
4. We show a PAC analysis of the learning algorithm based on Kolmogorov complexity.
5. We demonstrate the exponential gains in both sample efficiency and effectiveness of GOCI on diverse concept induction tasks with one or a few examples.

2. BACKGROUND AND RELATED WORK

Our approach to **Concept Learning** is closely related to Stroulia and Goel (1994)’s work, which learns logical problem-solving concepts by reflection. GOCI’s scoring metric is more general and applicable to both concepts and plans and can be used for learning from a few examples. While we use discrete spatial structures as motivating examples, GOCI is not limited to discrete spaces, similar to Wilson and Latombe (1994)’s work. GOCI is also related in spirit to probabilistic (Bayesian) program induction for learning decomposable visual concepts (Lake et al., 2015), which illustrates how exploiting decomposability is more effective. Our approach leverages not only decomposability but also implicit relational structure.

2.1. One/Few-Shot Learning and Theory Induction

Our problem setting differs from the above in that it requires learning from sparse examples (possibly one). Lake et al. (2015) propose a one-shot version of Bayesian program induction of visual concepts. There is also substantial work on one/few-shot learning (both deep and shallow) in a traditional classification

setting (Bart and Ullman, 2005; Vinyals et al., 2016; Wang et al., 2018), most of which either pre-train with gold-standard support example set or sample synthetic observations. We make no such assumptions about synthetic examples. ILP (Muggleton, 1991) inductively learns a logical program (first-order theory) that covers most of the positive examples and few of the negative examples by effectively employing background knowledge as search bias. In concept learning, generalization is typically performed as a search through space of candidate inductive hypotheses by (1) structuring, (2) searching, and (3) constraining the space of theories. FOIL (Quinlan, 1990) is an early noninteractive learner with the disadvantage that it occasionally prunes some uncovered hypotheses. This is alleviated in systems like FOCL by introducing language bias in the form of user-defined constraints (Pazzani, 1992). With *Interactive ILP*, learner could pose questions and elicit expert advice that allows pruning large parts of search space (Sammut and Banerji, 1986; Rouveirol, 1992). To incorporate new incoming information, ILP systems with *theory revision* incrementally refine and correct the induced theory (Sammut and Banerji, 1986; Muggleton, 1988). While GOCI is conceptually similar to ALEPH (Srinivasan, 2007), it learns from a few examples and actively acquires domain knowledge by interacting with human expert incrementally.

2.2. Knowledge-Guided Learning

Background knowledge in ILP is primarily used as search bias. Although the earliest form of knowledge injection can be found in explanation-based approaches (Shavlik and Towell, 1989), our work relates to preference-elicitation framework (Braziunas and Boutilier, 2006), which guides learning via human preferences as an inductive bias. Augmented learning with domain knowledge as an inductive bias has long been explored across various modeling formalisms, including traditional machine learning (Fung et al., 2003), probabilistic logic (Odom et al., 2015), and planning (Das et al., 2018). Our human-guided GOCI learner aims to extend these directions in the context of learning generalizable complex concepts from a few examples (including plans). Similar problem setting of concept learning from incomplete/sparse observations has also been explored in the cognitive science paradigm via explanation-based inductive program synthesis (Flener, 1997; Kitzelmann and Schmid, 2006).

The idea of augmented learning with human guidance/knowledge has also been extensively studied in the context of evolutionary computation. Interactive evolutionary systems (Eiben and Smith, 2015) use expert guidance to emulate a holistic fitness function that would otherwise depend on a very restricted pre-defined fitness model. The potential richness of such knowledge can be leveraged in not just evolutionary parent selection but can also optimize other parameters that leads to faster convergence, especially in mutations (Wendt et al., 2010). ILP has been shown to be conceptually similar to mutative EA in the context of program induction (Wong and Leung, 1997) and hence knowledge-guided mutations are related to knowledge augmented search in ILP. Thus, in our problem setting, the interaction module that seeks human guidance to select the most useful constraints (detailed in section 3.2.3) is similar in spirit to interactive (knowledge guided) evolutionary mutation process.

¹Our algorithm can learn from one (few) example(s). We specify the number of examples in our evaluations.

However, our underlying search strategy and optimization is based on ILP.

3. GUIDED ONE-SHOT CONCEPT INDUCTION

We are inspired by a teacher (human) and student (machine) setting in which a small number of demonstrations are used to learn generalized concepts (Chick, 2007). Intuitively, the description provided by a human teacher tends to be modular (can have distinct logical partitions), structured (entities and relations between them), and in terms of known concepts. Hence, a vectorized representation of examples is insufficient. We choose a logical representation, specifically a “function-free restricted form of first-order logic (FOL)” that models structured spaces faithfully.

Given: A set of “facts” or assertions, that is, a set of ground literals (or trajectories) describing 1 (or few) instance(s) of an unknown concept, availability of an expert to provide guidance and a knowledge-base of known concepts.
To Do: Learning a representation, by inducing a first-order logic program, of the given concept that optimally generalizes the given instance(s) effectively.

The input to GOCI is the description of the instances(s) of a concept that the human teacher provides. An example is, thus, conjunction of a set of ground literals (assertions). The output of GOCI is a least general generalization (LGG) horn clause from the input example(s).

3.1. Concept Representation

Consider the following example input to the GOCI framework. The input here is an instance of the structural concept $\mathbb{L}$ (illustrated in **Figure 1**).

Example 1. An instance in a minecraft domain can be a $\mathbb{L}$ with dimensions $\text{height} = 5, \text{base} = 4$ (**Figure 1**). $\mathbb{L}(s), \text{Height}(s, 5), \text{Base}(s, 4)$, s is the concept identifier and

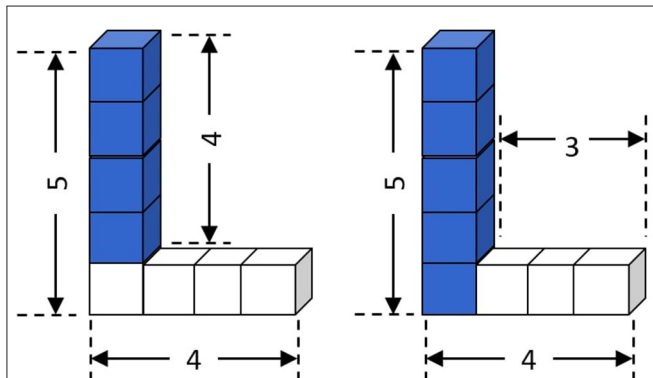


FIGURE 1 | Concept $\mathbb{L}$ (base = 4, height = 5), described as composition of a Tower and a Row.

may be described as conjunction of ground literals,

$\text{Row}(A) \wedge \text{Tower}(B) \wedge \text{Width}(A, 4) \wedge \text{Height}(S, 5) \wedge$
 $\text{Base}(S, 4) \wedge \text{Contains}(S, A) \wedge \text{Contains}(S, B) \wedge$
 $\text{Height}(B, 4) \wedge \text{SpRel}(B, A, \text{NWTop}'),$

which denotes $\mathbb{L}$ as composition of a “Row” of $w = 4$ and a “Tower” of $h = 4$ with appropriate literals describing the scenario (**Figure 1**, left). As a special case, under partial or total ordering assumptions among the ground literals, an input instance can represent a plan demonstration.

We aim to learn the optimally generalized (decomposable) representation of the concept ($\mathbb{L}$ in the context of the aforementioned example) referred by the one/few instances that were passed to GOCI as input. Before further discussion on the learning such a generalized (decomposable) representation let us first define formally what a concept representation signifies in our setting.

Definition 1. Concepts in GOCI are represented as horn clause theories. A theory T is defined as, $T = C(s^k \dots) :- \bigvee [\bigwedge_{i=1}^N f_i(t_1, \dots, t_j)]$, where the body $\bigwedge_{i=1}^N f_i(t_1, \dots, t_j)$ is a conjunction of literals indicating known concepts or relationships among them, such that any $t_j \in V \cup \{s^k\} \cup C$ where V is the set of all logical variables in the clause, C is the set of constants in the domain of any logical variable. The head $C(s^k \dots)$ identifies a target concept, and the terms $\{s^k\}$ are logical variables that denote the parameters of the concept assuming there are $k = \{1, \dots, K\}$ parameters including the identifier to the given instance of the concept. Since a concept can be described in multiple ways (**Figure 1**), the final theory will be a disjunction over clause bodies with the same head. A (partial) instantiation of a theory T is denoted as T/θ .

Note that these definitions allow for the reuse of concepts, potentially in a hierarchical fashion. We believe that this is *crucial* in achieving human-agent collaboration.

Example 2. **Figure 1** illustrates an instance of the concept $\mathbb{L}$ that can be described in multiple ways. A possible one is,

$$\mathbb{L}(s) :- [\text{Height}(s, h_s), \text{Base}(s, w_s), \text{Contains}(s, a),$$

$$\text{Contains}(s, b), \text{Row}(a), \text{Tower}(b),$$

$$\text{Width}(a, w_a), \text{Height}(b, h_b), \text{Equal}(w_s, w_a),$$

$$\text{Sub}(h_b, h_s, 1), \text{SpRel}(b, a, \text{NWTop}')] \vee$$

$$[\text{Height}(s, h_s), \text{Base}(s, w_s), \text{Contains}(s, a),$$

$$\text{Contains}(s, b), \text{Row}(a), \text{Tower}(b),$$

$$\text{Width}(a, w_a), \text{Height}(b, h_b), \text{Equal}(h_s, h_b),$$

$$\text{Sub}(w_a, w_s, 1), \text{SpRel}(b, s, \text{W}')]]$$

The generalization must be noted. The last argument of the $\text{SpRel}()$ is a constant, as only this particular spatial alignment is appropriate for the concept of the $\mathbb{L}$ structure. Although the input is a single instance (Example 1), GOCI should learn a generalized representation such as Example 2. Another

interesting aspect is the additional constraints: $\text{Equal}(X, Y)$ and $\text{Sub}(X, Y, N)$. While such predicates are a part of the language, they are not typically described directly in the input examples. However, they are key to generalization, since they express complex interactions between numerical (or non-numerical) parameters. Also note that the head predicate of the clause could have been designed differently as per Definition 1. For instance, in case of Example 2, the head predicate could have folded in the dimensional parameters— $\mathbb{L}(s, h_s, w_s)$. However, the number of such dimensional parameters can vary across different concepts. Hence to maintain generality of representation format during implementation, we push the dimensional parameters of the learnable concept into the body of the clause.

A specific case of our concept learning (horn clause induction) framework could be plan induction from sparse demonstrations. This can be achieved by specifying time as the last argument of both the state and action predicates. Following this definition, we can allow for plan induction as shown in our experiments. Our novel conceptual distance is clearer and more intuitive in the case of plans as can be seen later.

Definition 2 (Decomposable:). A concept C is decomposable if it is expressed as a conjunction of other concepts, and one or more additional literals to model the interactions. $C \Leftarrow (\bigwedge_i C'_i) \wedge (\bigwedge_j B_j)$. Here C'_i are literals that represent other concepts that are already present in the knowledge base of the learner and B_j are literals that either describe the attributes of C'_i or interactions between them.

Decomposable allows for an unknown concept to be constructed as a composition of other known concepts. GOCI learns the class of decomposable concepts since it is intuitive for the “human teacher” to describe. Decomposable concepts faithfully capture the *modular* and *structured* aspect of how humans would understand and describe instances. It also allows for a hierarchical construction of plans.

Example 3. Following the Minecraft structure described in 2, note how $\mathbb{L}$ is described with already known concepts $C'_1 =$

$\text{Row}()$ and $C'_2 = \text{Tower}()$ and the other literals such as $\text{Height}(b, h_b), \text{SpRel}(b, a, \text{"NWTop"}), \dots \in \{B_j\}$, that is, they describe the parameters of the known concepts or interactions between them. Note that known concepts in the knowledge base could have been manually coded in by experts or learned previously and are essentially represented in the same way. For instance, $\text{Row}()$ can be encoded as recursive the clause program representing a composition of one block and one unit shorter row,

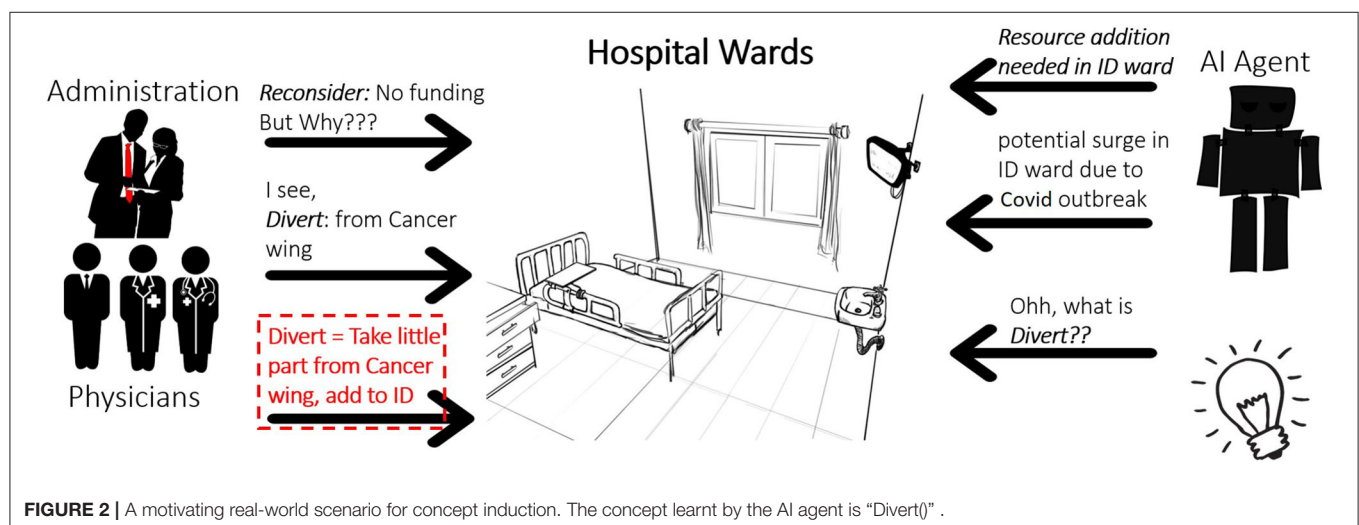
$$\begin{aligned} \text{Row}(r) : & -[\text{Width}(r, w_r), \text{Block}(a), \text{Row}(b), \text{Width}(b, w_b), \\ & \text{SpRel}(a, b, \text{"East"}), \text{Sub}(w_b, w_r, 1)] \\ & \bigvee [\text{Width}(r, w_r), \text{Equal}(w_r, 1), \text{Block}(a)] \end{aligned}$$

$\text{Tower}()$ could also be defined in the knowledge base in the same way. When the optimally general representation of the concept $\mathbb{L}$ is learned that is persisted in the knowledge base as well, such that more complex concepts can be represented by decomposing into $\mathbb{L}$ and other known concepts.

An obvious question that arises here is why $\{B_j\} \not\subseteq \{C'\}$? that is, why can the other literals not be treated similarly as a part of the knowledge base of known concepts? Ideally, that would be correct. However, that will also cause infinite levels of concept definitions, which cannot be implemented in practice. Additionally, following the paradigm of a student–teacher scenario, it can always be assumed that the student has prior understanding of many concepts from outside the current system. Thus, we can safely assume, without loss of generality, that set of literals $\{B_j\}$ are implicitly understood and defined as a part of the framework itself. This argument applies to the semantics of the “constraint predicates” (described later) as well.

Finally, before we discuss the details of the learning methodology, let us briefly look into a motivating, and presently relevant, real-world scenario that represents our problem setting.

Example 4. Consider a decision support AI system for resource planning and management in hospitals as illustrated in **Figure 2**. The AI agent forecasts the need for increased resources in the



infectious diseases (ID) ward, given the early signs of an outbreak of some disease such as Covid-19 or Ebola, and a potential spike in ID ward admissions. However, as noted by the administrators and/or physicians there is not enough budget to procure additional resources for ID ward. But the problem is quite critical and needs to be solved. So the human teacher (administrators in this case) teaches the AI agent the concept of “Divert” -ing resources from Cancer ward since cancer ward admissions are usually stable and does not have spikes. The AI agent is hence expected to learn a generalized representation of the concept of divert such that it may be applied later for other wards or for other tasks and furthermore in a “decomposable” fashion. For instance, “Divert” may be learned as a clause program such as,

```
Divert(R, qtyR): -To(R, Locdest), AcquireFrom
(R, Locsource1, qty1),
AcquireFrom(R, Locsource2, qty2),
AssignTo(R, Locdest, qtydest),
sum(qty1, qty2, qtydest)
```

Obviously, the above representation assumes that concepts such as “AcquireFrom()” are known concepts, either implicitly defined inside the learning framework or its explicit representation has been learned and persisted inside the knowledge base in the past.

The above example is solely to motivate the potential impact of our problem setting and the proposed solution. For an explanation of different components and aspects of GOCI, we refer to the much simpler and unambiguous structural example outlines earlier (ℒ).

3.2. Methodology

3.2.1. Search

ILP systems perform a greedy search through the space of possible theories. Space is typically defined, declaratively, by a set of mode definitions (Muggleton, 1995) that guide the search. We start with the most specific clause (known as a bottom clause) (Srinivasan, 2007) from the ground assertions and successively add/modify literals that might improve a rule that best explains the domain. Typically, the best theory is the one that covers the most positive and least negative examples. Thus, it optimizes the likelihood of a theory T based on the data. We start with a bottom clause and variabilize the statements via anti-substitution. Variabilization of T is denoted by $\theta^{-1} = \{a/x\}$, where $a \in \text{consts}(T)$, $x \notin \text{vars}(T)$. That is, anti-substitution θ^{-1} is a mapping from occurrences of ground terms in T to new or existing logical variables.

Evaluation Score: We redesign the ILP scoring (e.g., ALEPH’s compression heuristics) as:

- The user-provided advice forces the learner to learn longer theory, hence the *search space can be exponentially large*. Thus, modes alone are not sufficient as the search bias.
- There is only one (a few) positive training example(s) to learn from and *many possible rules can accurately match the training example*. Coverage-based scores fail.

Most learners optimize some form of likelihood. For a candidate theory T , likelihood given data $\mathbb{D}$ is $LL(T) = \log P(\mathbb{D}|T)$ (i.e., coverage). To elaborate further, in most classification tasks in discrete domains (with categorical/ordinal feature and target variables), goodness of fit of candidate models is achieved via the measure of how well the candidate models explain (or cover) the given data, that is, a good model is the one that will predict positive class for maximum possible positive examples and for minimum possible negative examples. This measure is expressed as likelihood of the data given a candidate model. In GOCI, we have one (at most few) positive example(s). Coverage will not suffice. Hence, we define a modified objective as follows.

$$T^* = \arg \min_{T \in \tau} (-LL(T) + D(T/\theta_X, X)) \quad (1)$$

where T^* is the optimal theory, τ is the set of all candidate theories, and D is the conceptual distance between the instantiated candidate theory T/θ_X and the original example X . Recall that a theory $\mathbb{T}$ is a disjunction of horn clause bodies (or conjunction of clauses).

3.2.2. Distance Metric

Conceptual distance, $D(T/\theta_X, X)$, is a penalty in our objective. The key idea is that any learned first-order horn clause theory must recover the given instance by equivalent substitution. However, syntactic measures, such as edit distance, are not sufficient since changing even a *single literal*, especially, literals that indicate interconcept relations, could potentially result in a completely different concept. For instance, in blocks-world, the difference between a block being in the middle of a row and one at the end of the row can be encoded by changing one literal. Hence, a more sophisticated semantic distance such as conceptual distance is necessary (Friend et al., 2018). However, such distances require deeper understanding of the domain and its structure.

Our solution is to employ *interplan distances*. Recall that the concepts GOCI can induce are decomposable and, hence, are equivalent to parameterized planning tasks. One of our key contributions is to exploit this equivalence by using a domain-independent planner to find grounded *plans* for both the theory learned at a particular iteration i , T_i and the instance given as input, X . We then compute the normalized compression distance (NCD) between the plans.

NCD: Goldman and Kuter (2015) proved that NCD is arguably the most robust interplan distance metric. NCD is a reasonable approximation of *Normalized Information Distance*, which is not computable (Vítányi et al., 2009). Let the plans for T_i/θ_X and X be π_T and π_X . To obtain NCD, we execute string compression (lossy or lossless) on each of the plans as well as the concatenation of the two plans to recover the compressed strings C_T , C_X , and $C_{T,X}$, respectively. NCD between the plans can be computed as,

$$NCD(\pi_T, \pi_X) = \frac{C_{T,X} - \min(C_T, C_X)}{\max(C_T, C_X)} \quad (2)$$

The conceptual distance between a theory T and X is the NCD between the respective plans, $D(T/\theta_X, X) = NCD(\pi_T, \pi_X)$.

This entire computation is performed by the *conceptual distance calculator* as shown in **Figure 3**.

Observations: (1) Conceptual distance as a penalty term in the LL score ensures that the learned theory will correctly recover the given example/demonstration. (2) $D(T/\theta_X, X)$ generalizes to the *Kolmogorov–Smirnov* statistic between two target distributions if we induce probabilistic logic theories. We prove these insights theoretically.

3.2.3. Human Guidance

The search space in ILP is provably infinite. Typically, language bias (modes) and model assumptions (closed world) are used to prune the search space. However, it is still intractable with one (or few) examples. So, we employ human expert guidance as constraints that can directly refine an induced theory, acting as a strong inductive bias. Also, we are learning decomposable concepts (see Definition 2). This allows us to exploit another interesting property. Constraints can now be applied over the attributes of the known concepts that compose the target concept, or over the relations between them. Thus, GOCI directly includes such constraints in the clauses as literals (see Example 2). Though such constraint literals come from the pre-declared language, they are not directly observed in the input example(s). So an ILP learner will fail to include such literals.

If the human inputs (constraints) are provided upfront before learning, it can be wasteful/irrelevant. More importantly, it places an additional burden on the human. To alleviate this, *our framework explicitly queries for human advice on the relevant constraint literals, which are most useful*. Let $\mathbb{U}$ be a predefined library of constraint predicates in the language, and let $\mathcal{U}() \in \mathbb{U}$ be a relevant constraint literal. Human advice $\mathcal{A}$ can be viewed as a preference over the set of relevant constraints $\{\mathcal{U}()\}$. If $\mathcal{U}_{\mathcal{A}}$ denotes the preferred set of constraints, then we denote the theory having a preferred constraint literal in the body of a clause as $\tau_{\mathcal{A}}$. (For instance, as per Example 2 GOCI queries “which of the two sampled constraints $\text{Sub}(h_b, h_s, 1)$ & $\text{Greater}(h_b, h_s)$ is more useful.” Human could prefer $\text{Sub}(h_b, h_s, 1)$, since it subsumes the other.) The scoring function now becomes:

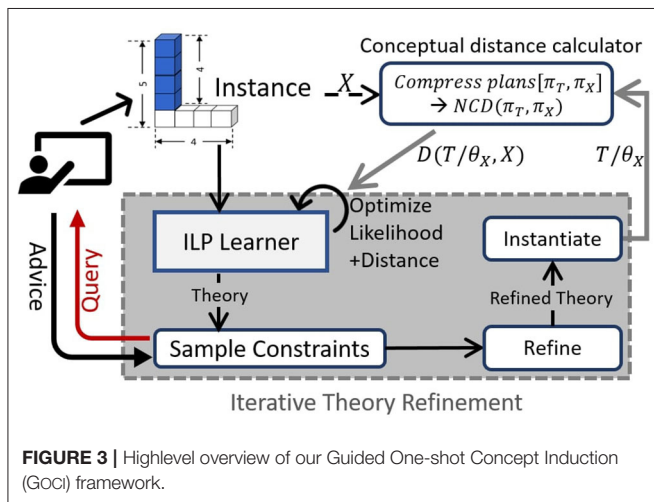
$$T^* = \underset{T \in \tau}{\operatorname{argmin}} (-LL(T) + D(T/\theta_X, X)) : \tau \subseteq \{\tau_{\mathcal{A}}\} \quad (3)$$

Thus, we are optimizing the constrained form of the same objective as Equation (1), which aims to prune the search space. This is inspired by advice elicitation approaches (Odom et al., 2015). While our framework can incorporate different forms of advice, we focus on preference over constraints on the logical variables. The formal algorithm, described next, illustrates how we achieve this via an iterative greedy refinement (**Figure 3**, query-advice loop shown in left).

3.3. The GOCI Algorithm

Algorithm 1 outlines the GOCI framework. It initializes a theory T_0 by variablizing the “bottom clause” obtained from X and background knowledge [lines 3 and 5]. Then it performs a standard ILP search (described earlier) to propose a candidate theory [line 6]. This is followed by the guided refinement steps, where constraint literals are sampled (parameter tying guides the sampling) and the *human teacher* is queried for preference over them, such that the candidate theory can be modified using preferred constraints [lines 7–9]. The function $\text{NCD}()$ performs the computation of the conceptual distance by first grounding the current modified candidate theory T' with the same parameter values as the input example X , then generating grounded plans and finally calculating the normalized compression distance between the plan strings (as shown in **Figure 3** and Equation 2) [line 10]. The distance-penalized negative log-likelihood score is estimated and minimized to find the best theory at the current iteration [lines 11–14], which is then used as the initial model in the next iteration. This process is repeated either until convergence (no change in induced theory) or maximum iteration bound (L).

Connection to plan induction: Evaluation, both in traditional ML and ILP, generally predicts the value of $\hat{y}_X$ for a test instance X represented as a fixed (ML) or arbitrary (ILP) length feature vector. In GOCI, however, the notion of evaluation of an instance X depends on successful construction of a valid/correct



Algorithm 1: Guided one-shot concept induction.

```

1: procedure GOCI(Instance  $X$ )
2:   Initialize: Set Iteration  $\ell \leftarrow 1$ 
3:   Initialize: Bootstrap theory  $T_0 \leftarrow X/\theta^{-1}$ 
4:   repeat
5:     Use  $T_{\ell-1}$  as initial model
6:     Candidate theory  $T_\ell \leftarrow \text{SEARCH}(T \in \tau | T_{\ell-1})$ 
7:     Sample applicable constraints  $\mathcal{U} \in \mathbb{U}$ 
8:      $\mathcal{U}_{\mathcal{A}} \leftarrow \text{QUERY}(\text{human}, \mathcal{U})$ 
9:      $T' \leftarrow T_\ell \oplus \mathcal{U}_{\mathcal{A}}$   $\triangleright \forall \mathcal{U}_{\mathcal{A}} \in \mathcal{A}$ 
10:     $D_\ell(T'/\theta_X, X) \leftarrow \text{NCD}(\pi_{T'}/\theta_X, \pi_X)$ 
11:    Score  $S_\ell \leftarrow (-LL(T') + D(T'/\theta_X, X))$ 
12:    if  $S_\ell < S_{\ell-1}$  then  $\triangleright$  minimize
13:      Retain  $T'$ : Update  $T_\ell = T'$ 
14:    end if
15:  until  $\ell \leq L$  OR  $T_\ell = T_{\ell-1}$ 
16: end procedure

```

plan π_X (Figure 4). Thus, while learning, most research aim to maximize coverage of positive instances E^+ ($\max P(\hat{y}_x = \text{true} | y_x \in E^+)$) and minimize coverage of negatives E^- , [i.e., $\min P(\hat{y}_x = \text{true} | y_x \in E^-)$]. GOCI evaluates a candidate concept representation by allowing the agent to realize that concept—by computing a valid plan for the goal/task implied by the instance x . This is akin to plan induction, since we are learning parameterized plan for realizing the concept as a surrogate for the concept itself. Additionally, planning has long been shown to be conceptually same as logic programming (Preiss and Shai, 1989) and hence induction of logic programs (theories) is the same as plan induction where the examples are trajectories (plan traces) in this case.

3.4. Theoretical Analysis

3.4.1. Validity of Distance Metric

NCD $\delta(x, y)$ between two strings x and y is provably a valid distance metric (Vitányi et al., 2009): $\delta(x, y) = \frac{\max K(x|y), K(y|x)}{\max K(x), K(y)}$, where $K(x)$ is the Kolmogorov complexity of a string x and $K(x|y)$ is the conditional Kolmogorov complexity of x given another string y . NCD is a computable approximation of the same $[D(x, y) \approx \delta(x, y)]$. Thus, we just verify if δ is a correct conceptual distance measure. Let T_Y and T_Z be two theories, with same parameterizations (i.e., same heads). Let T_Y/θ and T_Z/θ be their groundings with identical parameter values θ . Our learned theories are equivalent to planning tasks. Assuming access to a planner $\Pi()$ which returns $Y = \Pi(T_Y/\theta)$ and $Z = \Pi(T_Z/\theta)$, the two plan strings with respect to the instantiations of concepts are represented by T_Y and T_Z , respectively.

Proposition 1 (Valid Conceptual Distance). *Normalized information distance $\delta(Y, Z)$ is a valid and sound conceptual distance measure between T_Y and T_Z , that is, $\delta(Y, Z) = 0$ iff the concepts represented by T_Y and T_Z are equivalent.*

Proof Sketch for Proposition 1: Let T_Y and T_Z be 2 induced consistent first-order Horn clause theories, which may or may not represent the same concept. Let θ be some substitution. Now let T_Y/θ and T_Z/θ be the grounded theories under the same substitution. This is valid since we are learning horn clause theories with the same head, which indicates the target concept being learned. As explained in the paper, a theory is equivalent to a planning task. We assume access to a planner $\Pi()$, and we get

plan strings $Y = \Pi(T_Y/\theta)$ and $Z = \Pi(T_Z/\theta)$ with respect to the planing tasks T_Y/θ and T_Z/θ .

Friend et al. (2018) proved that *Conceptual Distance* is the step distance between two consistent theories in a cluster network $(\mathbb{T}, \rightleftharpoons, \sim)$, where $\mathbb{T}$ is the class of consistent theories, $\rightleftharpoons$ is the definitional equivalence relation (equivalence over bidirectional concept extensions) and $\sim$ implies symmetry relation. We have shown in the paper that, given the class of concepts we focus on, *a concept is a planning task*.

Let there be a theory T^* , which represents the optimal generalization of a concept C . If step distance $\langle T_Y, T^* \rangle = 0$ in a cluster network and $\langle T_Z, T^* \rangle = 0$, then $\langle T_Y, T_Z \rangle = 0$, that is, they represent the same concept C and they are definitionally equivalent $T_Y \rightleftharpoons T_Z$. Thus, both T_Y/θ and T_Z/θ will generate the same set of plans as T^* , since they will denote the same planning tasks (by structural induction). Thus,

$$T_Y \rightleftharpoons T_Z \iff [\Pi(Y) \cap \Pi(Z) = \Pi(Y) = \Pi(Z)] \quad (4)$$

up to equivalence of partial ordering in planning. Let $\pi^*(\cdot)$ be a minimum length plan in a set of plans $\Pi(\cdot)$. Let y and z be strings indicating plans $\pi^*(Y)$ and $\pi^*(Z)$ ignoring partial order. If $\Pi(Y) = \Pi(Z)$, then $\pi^*(Y) = \pi^*(Z)$. Hence, the conditional Kolmogorov complexities $K(y|z)$ and $K(z|y)$ will both be set to 0, if the strings x and y are equivalent (ignoring partial ordering). This is based on the principle that if they are equivalent, then a Universal prefix-Turing machine will recover one string given the other in 0 steps.

$$\therefore \frac{\max(K(y|z), K(z|y))}{\max(K(y), K(z))} = 0 = \delta(Y, Z)$$

Proposition 2 (Generalization to Kolmogorov-Smirnov). *In generalized probabilistic logic, following Vitányi (2013), $\delta(Y, Z)$ corresponds to 2-sample **Kolmogorov-Smirnov statistic** between two random variables T_Y/θ and T_Z/θ with distributions P_{T_Y} and P_{T_Z} , respectively, $[v(T_Y, T_Z) = \sup_{\theta \in \mathcal{F}} |F_{T_Y}(\theta) - F_{T_Z}(\theta)|]$, where $F_T(\cdot)$ is the cumulative distribution function for P_T and $\sup_{\theta \in \mathcal{F}}$ is the supremum operator. In a deterministic setting, δ is a special case of the Kolmogorov-Smirnov statistic v , $\delta(Y, Z) \leq v(F_{T_Y}, F_{T_Z})$.*

Proof Sketch for Proposition 2: This can be proved by considering the connection between NID and the distributions

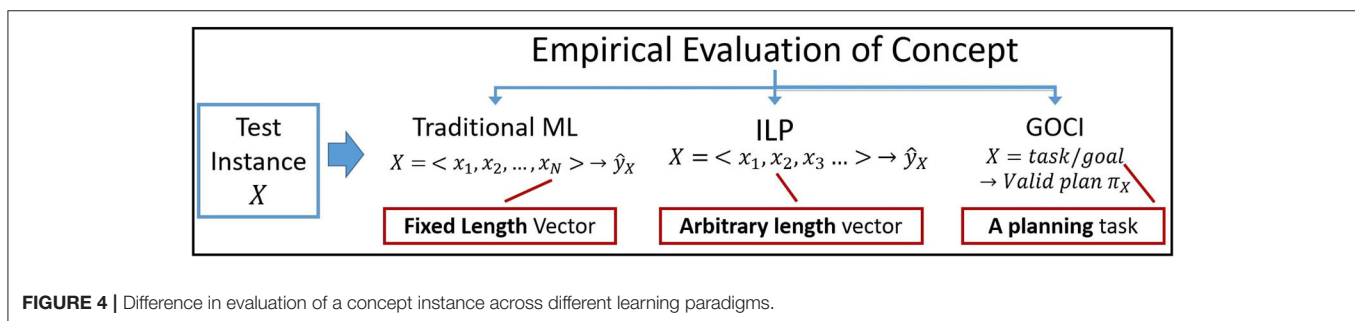


FIGURE 4 | Difference in evaluation of a concept instance across different learning paradigms.

induced by the concept classes we are learning. NID is defined as $\delta(x, y) = \frac{\max K(x|y), K(y|x)}{\max K(x), K(y)}$, where, $K(a|b)$ is the conditional Kolmogorov complexity of a string a , given b . There is no provable equivalence between Kolmogorov complexity and traditional notions of probability distributions.

However, if we consider a *reference* universal semi-computable semi-probability mass function $\mathbf{m}(x)$, then there is a provable equivalence $-\log \mathbf{m}(x) = K(x) \pm O(1)$. Similarly for conditional Kolmogorov complexity, by Conditional Coding Theorem, $-\log \mathbf{m}(y|x) = K(y|x) \pm O(1)$ (Vitányi, 2013). By definition,

$$m(y|x) = \sum_{j \geq 1} 2^{-K(j)-c_j} P_j(y|x)$$

where $c_j > 0$ are constants and $P_j(y|x)$ is the lower semi-computable conditional. A lower semi-computable semi-probability conditional mass function is based on the string generating complexity of a Universal prefix-Turing machine. Thus, $m(y|x)$ is greater than all the lower semi-computable. Note that our compressed plans are equivalent to a string generated by Universal prefix-Turing machines. The conditional case implies, if a compressed plan string x is given as an auxiliary prefix tape, how complex it is to generate compressed string $y = \theta$.

Given two grounded theories T_Y/θ and T_Z/θ , let $P_{T_Y/\theta}$, $P_{T_Z/\theta}$ be the respective distributions when learning probabilistic logic rules. Now let us define the semantics of a distribution $P_{T/\theta}$ in our case: $P_{T/\theta} = P(\pi(T/\theta))$, that is, distribution over the plan strings, which can be considered as lower semi-computable probability based on coding theory. We know,

$$\sum_{j \geq 1} 2^{-K(j)-c_j} P_j(y) \approx F(y|x) \quad (5)$$

where $F(y)$ is the cumulative distribution. So, NID $\delta(Y, Z)$ now becomes, $\delta(Y, Z) = \frac{\max(K(y|z), K(z|y))}{\max(K(y), K(z))}$. We know that $\max(K(y), K(z))$ is a normalizer. Thus, $\delta(Y, Z) < \max(K(y|z), K(z|y))$

$$\begin{aligned} \max(K(y|z), K(z|y)) &= \max(-\log m(y|z), -\log m(z|y)) \\ &= \max\left(-\log \frac{m(y, z)}{m(z)}, -\log \frac{m(y, z)}{m(y)}\right) \\ &= \max\left([-\log m(y, z) + \log m(z)], \right. \\ &\quad \left. [-\log m(y, z) + \log m(y)]\right) \end{aligned}$$

Under partial ordering max yields supremum

$$\begin{aligned} &\approx \sup |\log m(y) - \log m(z)| \\ &\approx \sup |\log F(y) - \log F(z)| \\ &\approx \sup |F(y) - F(z)| \text{ log is monotonic} \end{aligned}$$

Significance of Propositions 1 and 2: Proposition 1 outlines how our proposed NCD-based metric is a valid conceptual distance. It is well understood that the true measure of conceptual distance is not straightforward and is subject to the semantic interpretation

of the domain itself. But designing a unique distance metric based on the semantics of every domain limits the generality of any learning system. So NCD acts as a surrogate “conceptual distance.” It is based on the notion that “if two concepts are fundamentally same the complexity of optimal action plans to realize the concepts should also be fundamentally same.” NCD (or NID) essentially measures the difference in generative complexities of two plans. Also note that other types of distances that are limited to a syntactic level such as edit distance (or literal distance) will fail to capture the similarity or diversity between concept representations since the same concept can be represented with more than one theories that may vary in one or more literals.

Proposition 2, on the other hand, proves that our proposed metric is not limited to our specific scenario. It positions our work in the context of known statistical distance metrics and establishes its credibility as a valid solution. It proves how in a nondeterministic setting, that is, probabilistic logic formulation, our proposed metric generalizes to Kolmogorov–Smirnov statistic.

3.4.2. PAC Learnability

PAC analysis of GOCI follows from GOLEM for function-free *horn* clause induction (Muggleton and Feng, 1990). Let initial hypothesis space be $\mathcal{H}_0$ and the final be $\mathcal{H}^*$ (s.t. $T^* \in \mathcal{H}^*$).

Proposition 3 (Sample Complexity). *Following Valiant (1984) and Mooney (1994), with probability $(1-\delta)$, the sample complexity of inducing the optimal theory T^* is:*

$$n^* = \mathcal{O}\left(\frac{1}{\epsilon} \left[d^L j^i \ln((tfm)) + \ln\left(\frac{1}{\delta}\right) \right]\right) \quad (6)$$

where ϵ is the regret, n^* - sample complexity of $\mathcal{H}^*$, i is the maximum depth of a variable in a clause and j is the maximum arity. m - number of distinct predicates, t is the number of terms, p is the place and d is the distance of the current revision from the last known consistent theory, and L is the upper bound on the number of refinement steps (iterations).

Proof Sketch for Proposition 3: In our learning setting, the learned theory will always have nonzero uncertainty. To understand the properties, we build upon the PAC analysis for recursive *rlgg* (Relative Least General Generalization) approach for function-free *Horn* clause learning shown by Muggleton and Feng (1990) in GOELM. With some restrictions, it applies here as well. Let n : denote the sample size and $\mathcal{H}$: the hypothesis space. Our approach can be considered as an *rlgg* approach with refinement steps. Note that constraint predicates that refine the clauses are not part of $\mathcal{K}$.

To begin with, we are interested in regret bounds for the initially learned hypothesis by the ILP learner $\mathcal{H}_0$, before refinement. We know from Valiant (1984), that with probability $(1-\delta)$ the sample complexity n for $\mathcal{H}_0$ is,

$$n \geq \frac{1}{\epsilon} \left(\ln(|\mathcal{H}_0|) + \ln\left(\frac{1}{\delta}\right) \right) \quad (7)$$

where ϵ is the regret. Now, our ILP learner induces ij -determinate clauses (Muggleton and Feng, 1990), where i is the maximum depth of the clause and j is the maximum arity. In our problem setting, it can be proven that $|\mathcal{H}_0| = \mathcal{O}((tpm)^j)$, where m is the number of distinct predicates in the language, t is the number of terms, and p is the place (for details about place, refer Muggleton and Feng, 1990). Also note that if j & i is bounded, then $j^i \leq c$. Mooney (1994) shows that for theory refinement/revision, sample complexity is expressed as,

$$n^* = \mathcal{O} \left(\frac{1}{\epsilon} \left[d^k \ln(|\mathcal{H}_0| + d + m) + \ln\left(\frac{1}{\delta}\right) \right] \right) \quad (8)$$

where distance d to be the number of single literal changes in a single refinement step and k is the number of refinement/revision iterations. In **Algorithm 1**, we observe that at each iteration $\ell \leq L$, updates are with respect to the preferred constraint predicates $\mathcal{U}_A \in \mathbb{U}$. Thus, we know that $k = L$. Substituting $|\mathcal{H}_0| = (tpm)^j$ and $j^i = c$ constant in Equation (8) and ignoring the additive terms $d + m$ since $(tpm)^j \gg d + m$, we get,

$$n = \mathcal{O} \left(\frac{1}{\epsilon} \left[d^L c \ln(tpm) + \ln\left(\frac{1}{\delta}\right) \right] \right) \quad (9)$$

Proposition 4 (Refinement Distance). *d is upper bounded by the expected number of literals that can be constructed out of the library of constraint predicates with human advice $\mathbb{E}_{\sim \mathcal{A}}[|\mathbb{U}|]$ and lower bounded by the conceptual distance between theory learned at two consecutive iterations since we adopt a greedy approach. If $\Pr_{\mathcal{A}}(\mathcal{U})$ denotes the probability of a constraint predicate being preferred, then $|D_\ell - D_{\ell-1}| \leq d \leq \sum_{i=1}^{2^{(|\mathbb{U}|-1)} \times t p_q} \Pr_{\mathcal{A}}(\mathcal{U}_i)$ where $2^{(|\mathbb{U}|-1)} \times t p_q$ is the maximum possible number of constraint literals and q is the maximum arity of the constraints. In case of only pairwise constraints, $q = 2$.*

Proof Sketch for Proposition 4: The proof is straightforward and hence we present it in brief. In our setting to show that,

$$|D_\ell - D_{\ell-1}| \leq d \leq \sum_{i=1}^{2^{(|\mathbb{U}|-1)} \times t p_q} P_{\mathcal{A}}(\mathcal{U}_i) \quad (10)$$

(where $2^{(|\mathbb{U}|-1)} \times t p_q$ is the maximum number of constraint literals possible, since $\mathbb{U}$ is the library of constraint predicates) consider that the number of constraint predicates that can be picked up at any iteration is $2^{(|\mathbb{U}|-1)}$. To form constraint literals, we need to tie arguments to existing logical variables in the current theory. We have defined t to be the number of terms in the existing theory. Let q be the max arity of a constraint, thus total possible number of constraint literals are $2^{(|\mathbb{U}|-1)} \times t p_q$. So if the distribution induced on the constraint literals by human advice $\mathcal{A}$ be $P_{\mathcal{A}}$, then $\sum_{i=1}^{2^{(|\mathbb{U}|-1)} \times t p_q} P_{\mathcal{A}}(\mathcal{U}_i)$ is the expected number of literals added given the advice. Now this is the upper bound of d . Again d should at least be the conceptual distance between the new theory after constraint addition and the last consistent theory. Note d and conceptual distance D is not the same. Thus,

it is the difference between the NCD of last theory to original example and current updated theory to the original example $|D_\ell - D_{\ell-1}|$.

Observe that if at each layer $\ell \leq L$ we add constraint predicates $\mathcal{U}_\ell$, then at layer ℓ , $d = |\{\mathcal{U}\}^\ell| \leq 2^{m t p_q}$ (assuming q is maximum arity of the constraint predicates). Also, as per our greedy refinement framework, at each layer ℓ , distance new theory $\mathbb{T}_\ell$ should at least be the change in conceptual distance.

Significance of Propositions 3 and 4: Propositions 3 and 4 aim to illustrate what the general sample complexity would be for a theory refinement-based RLGG clause learner such as GOCI and how the conceptual distance controls the complexity by establishing bounds on the refinement distance. Furthermore, the complexity is also subject to the maximum refinement iterations, which in turn is affected by human guidance. Thus, we establish the theoretical connection between the two dimensions of the contribution of this work.

Proposition 5 (Advice Complexity). *From Equations (6) and (8), at convergence $\ell = L$, we get $\frac{n^* - |X|}{L}$ examples, on an average, for a concept $\mathcal{C}$ to be PAC learnable using GOCI.*

The proof is quite straightforward and hence we just discuss the brief idea behind it. Our input is sparse (one or few instances). GOCI elicits advice over constraints to acquire additional information. Let $|X|$ be the number of input examples. We query the human once at each layer and hence the maximum query budget is L . Given that the sample size is $|X|$, each query to the human must acquire information about at least $\frac{n^* - |X|}{L}$ examples, on an average, for our a concept $\mathcal{C}$ to be PAC learnable using our approach.

4. EVALUATION

We next aim to answer the following questions explicitly:

- (Q1) Is GOCI effective in “one-shot” concept induction?
- (Q2) How sample efficient is GOCI compared to baselines?
- (Q3) What is the relative contribution of the novel scoring function versus human guidance toward performance?

Our framework extends a Java version of Aleph (Srinivasan, 2007). We modified the scoring function with NCD penalty computed via a customized SHOP2 planner (Nau et al., 2003). We added constraint sampling and human guidance elicitation iteratively (**Algorithm 1**).

4.1. Experimental Design

We compare GOCI with Aleph with no enhancements. We focus on the specific task of “one-shot concept induction,” with a single input example for each of the several types of concepts and report aggregated precision. We consider precision because preference queries are meant to eliminate false positives in our case. To demonstrate general robustness of GOCI, beyond one-shot case, we experimented with varying sample sizes for each concept type and show learning curves for the same. We perform an ablation study to show the relative contribution of two important components of

GOCI: (a) novel scoring metric and (b) human guidance, that is, we compare against two more baselines (*ILP+Score* and *ILP+Guidance*). For every domain, we consider 10 different types of concepts (10 targets) and aggregate results over 5 runs.

Note that human guidance was obtained from distinct human experts for every run. The expertise level of all the advice providers was reasonably at par since they were chosen from the same pool of candidates with zero visibility and knowledge of our proposed framework. However, for all the human advice providers we assumed a basic level of knowledge in geometry or fundamentals of logic and reasoning. Additionally, we also explained each of the experimental domains to the human participants to create a similar level of awareness about the domains among all of them.

4.1.1. Domains

We employ four domains with varying complexity. Note that we have selected the below domains based on multiple considerations. The domain encoding need to be such that target concepts can be learned in a modular fashion (i.e., decomposable). Thus, the first two domains are structure construction domains either spatial (Minecraft) or chemical/molecular (ChEBI). Spatial structures are implicitly modular (such as the $\mathbb{L}$ -structure in **Figure 1**). Chemical entities, molecules/compounds/complexes, are similarly modular as well. The last two domains are fundamentally planning domains. However, they are also compositional in nature, that is, any planning goal is a composition task. For instance, machine structure in “Assembly” domain and cocktails, etc., in “Barman” domain. So these two domains do not just demonstrate learning modular/decomposable concepts but they also illustrate the plan induction feature of GOCI.

1. **Minecraft (spatial structures):** The goal is to learn discrete spatial concepts in a customized (Narayan-Chen et al., 2019) Project Malmo platform for Minecraft. Dialogue data in Malmo are available online, and we converted them into a logical representation. All structures are in terms of discrete atomic unit blocks (cubes). **Figure 5** shows examples of some spatial structures that GOCI was able to learn.

2. **Chemical Entities of Biological Interest (ChEBI):** ChEBI (Degtyarenko et al., 2007) is a compound database containing important structural features and activity-based information, for classification of chemicals, such as (1) molecular structure and (2) biological role. We model the Benzene molecule prediction task as molecular-compositional concepts. The data have predicates such as *SingleBond*, *DoubleBond*, and *HasAtom*.
3. **Assembly (planning domain):** Assembly is a planning domain, where different mechanical structure concepts are compositions of different parts and resources. Input is a conjunction of ground literals indicating ground plan demonstration (assuming total ordering).
4. **Barman (planning domain):** A standard planning domain where a bartender is supposed to follow certain recipes and sequence of techniques to create cocktails. The different cocktails are decomposable concepts in this setting.

4.2. Experimental Results

[Effective One-shot (Q1)] **Table 1** shows the performance of GOCI on one-shot concept learning tasks as compared to standard ILP. GOCI significantly outperforms ILP across all domains answering (Q1) affirmatively. Also, note that GOCI is very “query” efficient as observed from the small average

TABLE 1 | Results for one-shot concept learning.

Domain	Approach	Avg. precision	#Queries
Minecraft	Goci	0.85	5.5 ± 3
	ILP	0.35	–
Assembly	Goci	0.65	16.5 ± 4
	ILP	0.2	–
ChEBI	Goci	0.615	13.1 ± 2.13
	ILP	0.45	–
Barman	Goci	0.7	10.5 ± 5.4
	ILP	0.51	–

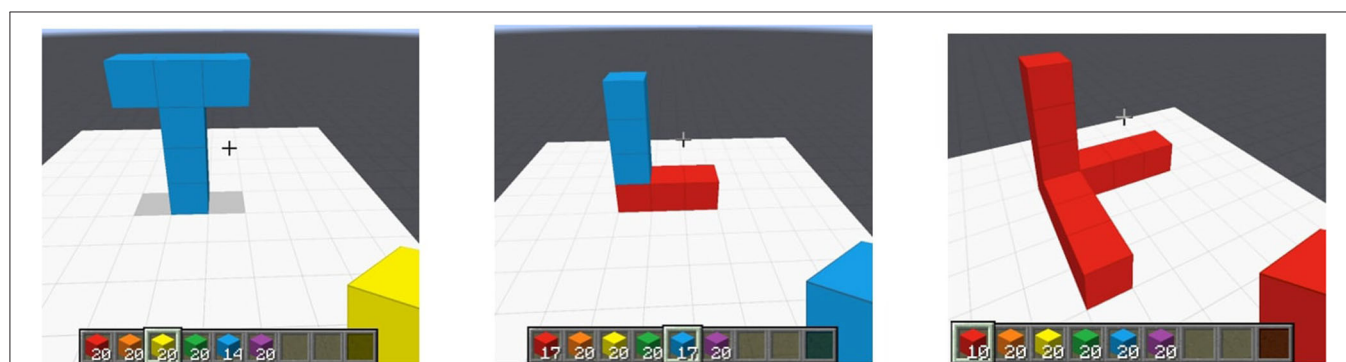


FIGURE 5 | Instances of spatial concepts in Minecraft. (Left) Upright Tee, (Middle) Upright L, (Right) Orthogonal overlapping Ls.

number of queries posed in the case of each domain. Note that in the case of CheBI, the number of queries is the highest among all the domains. This can be attributed to that fact that CheBI is a domain, which requires a certain degree of understanding of fundamental chemistry (chemical bonds and their types, molecules, atoms, etc.). Thus, some of the human participants required more iterations (consequently more queries) to converge to the most relevant set of constraint literals, given the difference of their prior understanding of school chemistry.

Query efficiency is an important consideration in any learning paradigm that leverages human guidance, since controlling the cognitive load on the human expert is critical. So, in general, the observed average query numbers being reasonably low across all domains corroborates our theoretical advice complexity (section 3.4.2).

[Sample Efficiency (Q2)] In Figure 6, we observe that GOCI converges within significantly smaller sample size across all domains, thus, supporting our theoretical claims in section 3.4. In ChEBI, though, quality of planner encoding might explain mildly lower precision yet GOCI does perform significantly better than vanilla ILP learner. In ChEBI, we see that the sample efficiency is not vastly distinct. One of the possible reasons could be the sub-optimal encoding of the planning domain language, which is necessary for NCD computation, for this task. If we can improve the planner setup for this domain, then we will likely be able to observe enhanced performance.

[Relative contribution (Q3)] Figure 7 validates our intuition that both components (scoring function and human-guidance) together make GOCI a robust one-shot (sample-efficient) concept induction framework. Though human guidance, alone, is able

to enhance the performance of a vanilla ILP learner in sparse samples, yet it is not sufficient for optimal performance. In contrast, although the advantage of our novel distance-penalized scoring metric is marginal in sparse samples, it is essential for optimal performance at convergence.

5. DISCUSSION

The most important conclusion from the experiments is that when available, the guidance along with the novel score leads to a jump-start, better slope and in some cases, asymptotically sample

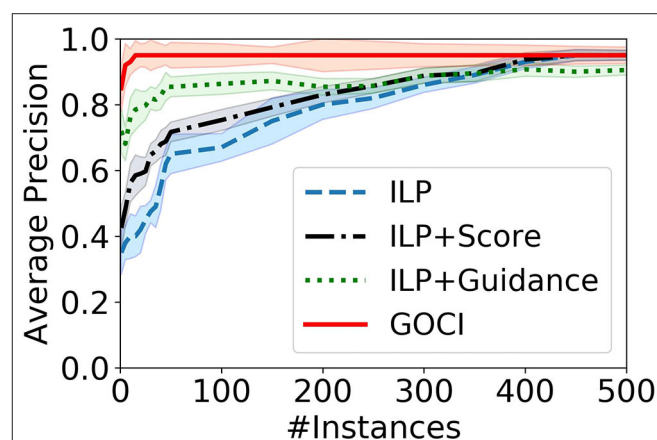


FIGURE 7 | Results of ablation study on Minecraft domain. Relative contribution of our distance-penalized score vs. human guidance.

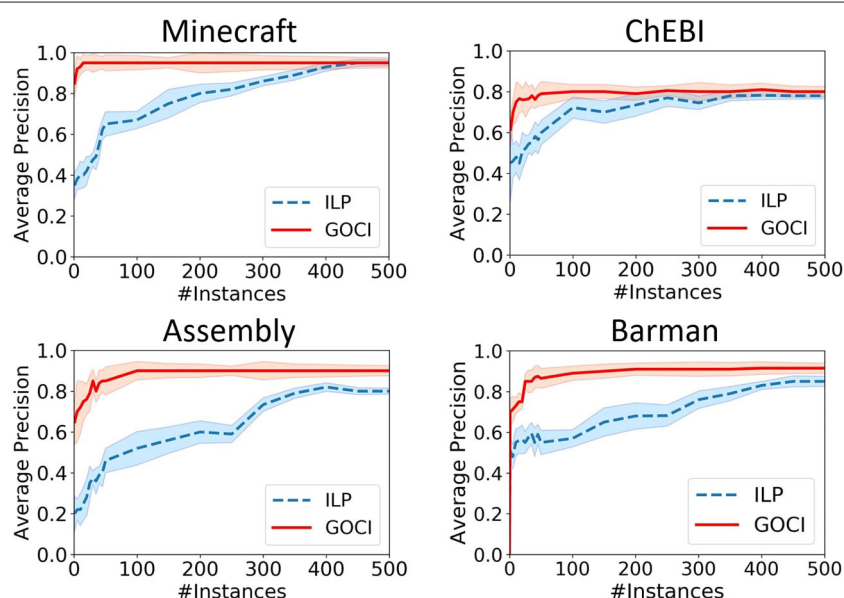


FIGURE 6 | Learning curves for varying sample size to compare the sample efficiency of Guided One-shot Concept Induction (GOCI) and inductive logic program (ILP). Top two plots are with respect to structural composition domains-Minecraft & ChEBI and the bottom two are for planning domains: Assembly and Barman (best viewed in color).

efficient with a fraction of the number of instances needed than merely learning from data.

Another important aspect to note here is that our experimental setup did not attempt to ensure in any way that the quality of guidance provided by the human participants is optimal. The formulation of the objective function, itself, in GOCI is designed to handle sub-optimal human advice implicitly in a seamless manner. The two primary features in the design that make GOCI robust to advice quality are as follows:

1. As explained earlier and shown in Equation (3), human advice and conceptual distance deal with two distinct aspects of the search process. Human advice controls the size and nature of the search space while conceptual distance ensures the quality of the candidates. Advice and distance have a balancing effect on each other, and thus, it is our novel conceptual distance that makes GOCI robust to bad advice.
2. Also, the nature of human advice in our setting is of choosing the most useful set of “constraint predicates” among the set of candidate constraints. Now the candidates are generated by GOCI in a conservative fashion selecting only the ones that are logically valid for the theory learned at the current iteration of revision. Thus, human experts have very little option of choosing an invalid or extremely unlikely constraint predicate.

Our ablation study in **Figure 7** also supports our analysis. On closer inspection, we see that it is due to our novel distance penalized scoring function (ILP+Score) that ensures convergence to an optimal solution. Human advice (ILP+Guidance) contributes to sample efficiency.

6. CONCLUSIONS

We developed a human-in-the-loop one-shot concept learning framework in which the agent learns a generalized representation of a concept as FOL rules, from a single (few) positive example(s). We make two specific contributions: deriving a new distance measure between concepts and allowing for richer human

inputs than mere labels, solicited actively by the agent. Our theoretical and experimental analyses show the promise of GOCI method. An exhaustive evaluation involving richer human inputs including varying levels of expertise and analyzing our claim that learning performance of GOCI is robust to expertise levels (which should only affect query efficiency) is an immediate future research objective. Integration with hierarchy learning also remains an interesting direction for future research.

DATA AVAILABILITY STATEMENT

All datasets generated for this study are included in the article/supplementary material.

AUTHOR CONTRIBUTIONS

MD and SN contributed equally to the ideation. MD and NR led the empirical evaluation. MD, NR, SN, and JD contributed nearly equally to the manuscript preparation. All authors contributed to the article and approved the submitted version.

FUNDING

MD and SN gratefully acknowledge the support of CwC Program Contract W911NF-15-1-0461 with the US Defense Advanced Research Projects Agency (DARPA) and the Army Research Office (ARO). SN and NR gratefully acknowledge AFOSR award FA9550-18-1-0462. Any opinions, findings, and conclusion or recommendations are those of the authors and do not necessarily reflect the view of the DARPA, ARO, AFOSR, or the US government.

ACKNOWLEDGMENTS

The authors acknowledge the support of members of STARLING lab for the discussions.

This article was released as a preprint to arXiv as part of the StarAI 2020 workshop (Das et al., 2020).

REFERENCES

- Bart, E., and Ullman, S. (2005). “Single-example learning of novel classes using representation by similarity,” in *BMVC* (Oxford).
- Braziunas, D., and Boutilier, C. (2006). “Preference elicitation and generalized additive utility,” in *AAAI* (Boston, MA).
- Chick, H. L. (2007). “Teaching and learning by example,” in *Mathematics: Essential Research, Essential Practice* (Hobart, TAS).
- Das, M., Odom, P., Islam, M. R., Doppa, J. R., Roth, D., and Natarajan, S. (2018). “Preference-guided planning: an active elicitation approach,” in *AAMAS* (Stockholm).
- Das, M., Ramanan, N., Doppa, J. R., and Natarajan, S. (2020). “One-shot induction of generalized logical concepts via human guidance,” in *StarAI-20 Workshop*. Available online at: <https://arxiv.org/abs/1912.07060>
- Degtyarenko, K., De Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., et al. (2007). Chebi: a database and ontology for chemical entities of biological interest. *Nucleic Acids Res.* 36, D344–D350. doi: 10.1093/nar/gkm791
- Eiben, A. E., and Smith, J. E. (2015). *Interactive Evolutionary Algorithms*. Berlin; Heidelberg: Springer Berlin Heidelberg, 215–222.
- Flener, P. (1997). “Inductive logic program synthesis with dialogs,” in *ILP* (Prague).
- Friend, M., Khaled, M., Lefever, K., and Székely, G. (2018). Distances between formal theories. Available online at: <http://philsci-archive.pitt.edu/14849/>
- Fung, G. M., Mangasarian, O. L., and Shavlik, J. W. (2003). “Knowledge-based support vector machine classifiers,” in *NIPS* (Vancouver, BC).
- Goldman, R. P., and Kuter, U. (2015). “Measuring plan diversity: pathologies in existing approaches and a new plan distance metric,” in *AAAI* (Austin, TX).
- Horn, A. (1951). On sentences which are true of direct unions of algebras. *J. Symbol. Logic* 16, 14–21.
- Kitzelmann, E., and Schmid, U. (2006). Inductive synthesis of functional programs: an explanation based generalization approach. *J. Mach. Learn. Res.* 7, 429–454.
- Kozierowski, J., and Turk, M. (2018). “Clear: cumulative learning for one-shot one-class image recognition,” in *CVPR* (Salt Lake City, UT).
- Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science* 350, 1332–1338. doi: 10.1126/science.aab3050
- Mitchell, T. (1997). *Machine Learning*. McGraw-Hill Higher Education.
- Mooney, R. J. (1994). A preliminary PAC analysis of theory revision, *Comput. Learn. Theory Nat. Learn. Syst.* 3, 43–53.

- Muggleton, S. (1988). "Machine invention of first-order predicates by inverting resolution," in *Proc. of 5th Machine Learning Conference* (Ann Arbor, MI).
- Muggleton, S. (1991). Inductive Logic Programming. *N. Gen. Comput.* 8, 295–318.
- Muggleton, S. (1995). Inverse Entailment and Progol. *N. Gen. Comput.* 13, 245–286.
- Muggleton, S., and Feng, C. (1990). "Efficient induction of logic programs," in *ALT* (Tokyo).
- Narayan-Chen, A., Jayannavar, P., and Hockenmaier, J. (2019). "Collaborative dialogue in Minecraft," in *ACL* (Florence).
- Nau, D., Au, T.-C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D., et al. (2003). Shop2: An HTN planning system. *J. Artif. Intell. Res.* 20, 379–404. doi: 10.1613/jair.1141
- Odom, P., Khot, T., Porter, R., and Natarajan, S. (2015). "Knowledge-based probabilistic logic learning," in *AAAI* (Austin, TX).
- Pazzani, M. J. (1992). *An Information-Based Approach to Integrating Empirical and Explanation-Based Learning*. Tokyo: Inductive Logic Programming.
- Preiss, K., and Shai, O. (1989). Process planning by logic programming. *Robot. Comput. Integr. Manufact.* 5, 1–10. doi: 10.1016/0736-5845(89)90025-2
- Quinlan, J. R. (1990). Learning logical definitions from relations. *Mach. Learn.* 5, 239–266.
- Rouveirol, C. (1992). *Extensions of Inversion of Resolution Applied to Theory Completion*. Tokyo: Inductive Logic Programming.
- Sammur, C., and Banerji, R. B. (1986). "Learning concepts by asking questions," in *Machine Learning: An Artificial Intelligence Approach*, Vol. 2, eds R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (Morgan Kaufmann), 167–192.
- Shavlik, J. W., and Towell, G. G. (1989). "Combining explanation-based learning and artificial neural networks," in *ICML*.
- Srinivasan, A. (2007). *Aleph: A Learning Engine for Proposing Hypotheses*. Available online at: <http://web2.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.pl>
- Stroulia, E., and Goel, A. K. (1994). "Learning problem-solving concepts by reflecting on problem solving," in *ECML* (Catania).
- Tax, D. M. J. (2001). *One-class classification: concept learning in the absence of counter-examples* (Ph.D. thesis). TU Delft, Delft, Netherlands.
- Valiant, L. G. (1984). A theory of the learnable. *Commun. ACM.* 27, 1134–1142.
- Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Trans. Neural Netw.* 10, 988–999.
- Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). "Matching networks for one shot learning," in *NIPS* (Barcelona).
- Vitányi, P. M. (2013). Conditional kolmogorov complexity and universal probability. *Theor. Comput. Sci.* 501, 93–100. doi: 10.1016/j.tcs.2013.07.009
- Vitányi, P. M., Balbach, F. J., Cilibrasi, R. L., and Li, M. (2009). "Normalized information distance," in *Information Theory and Statistical Learning* (Springer), 45–82.
- Wang, Y.-X., Girshick, R., Hebert, M., and Hariharan, B. (2018). "Low-shot learning from imaginary data," in *CVPR* (Salt Lake City, UT).
- Wendt, K., Cortés, A., and Margalef, T. (2010). Knowledge-guided genetic algorithm for input parameter optimisation in environmental modelling. *Proc. Comput. Sci.* 1, 1367–1375. doi: 10.1016/j.procs.2010.04.152
- Wilson, R. H., and Latombe, J.-C. (1994). Geometric reasoning about mechanical assembly. *Artif. Intell.* 71, 371–396. doi: 10.1016/0004-3702(94)90048-5
- Wong, M. L., and Leung, K. (1997). Evolutionary program induction directed by logic grammars. *Evol. Comput.* 5, 143–180.

Conflict of Interest: MD was employed by the company Samsung R&D Institute India - Bangalore.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Das, Ramanan, Doppa and Natarajan. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



OPEN ACCESS

EDITED BY

Paolo Gastaldo,
University of Genoa, Italy

REVIEWED BY

Lichun Wang,
Beijing University of Technology, China
Ilaria Tiddi,
VU Amsterdam, Netherlands
Gianluca Bardaro,
Polytechnic University of Milan, Italy

*CORRESPONDENCE

Jan-Philipp Töberg,
✉ jtoeberg@techfak.uni-bielefeld.de

RECEIVED 27 October 2023

ACCEPTED 09 February 2024

PUBLISHED 29 February 2024

CITATION

Töberg J-P, Ngonga Ngomo A-C, Beetz M
and Cimiano P (2024), Commonsense
knowledge in cognitive robotics: a systematic
literature review.
Front. Robot. AI 11:1328934.
doi: 10.3389/frobt.2024.1328934

COPYRIGHT

© 2024 Töberg, Ngonga Ngomo, Beetz and
Cimiano. This is an open-access article
distributed under the terms of the [Creative
Commons Attribution License \(CC BY\)](#). The
use, distribution or reproduction in other
forums is permitted, provided the original
author(s) and the copyright owner(s) are
credited and that the original publication in
this journal is cited, in accordance with
accepted academic practice. No use,
distribution or reproduction is permitted
which does not comply with these terms.

Commonsense knowledge in cognitive robotics: a systematic literature review

Jan-Philipp Töberg^{1,2*}, Axel-Cyrille Ngonga Ngomo^{2,3},
Michael Beetz^{2,4} and Philipp Cimiano^{1,2}

¹Center for Cognitive Interaction Technology, Bielefeld University, Bielefeld, Germany, ²Joint Research Center on Cooperative and Cognition-enabled AI, Bielefeld, Germany, ³DICE Group, Paderborn University, Paderborn, Germany, ⁴Institute for Artificial Intelligence, University of Bremen, Bremen, Germany

One of the big challenges in robotics is the generalization necessary for performing unknown tasks in unknown environments on unknown objects. For us humans, this challenge is simplified by the commonsense knowledge we can access. For cognitive robotics, representing and acquiring commonsense knowledge is a relevant problem, so we perform a systematic literature review to investigate the current state of commonsense knowledge exploitation in cognitive robotics. For this review, we combine a keyword search on six search engines with a snowballing search on six related reviews, resulting in 2,048 distinct publications. After applying pre-defined inclusion and exclusion criteria, we analyse the remaining 52 publications. Our focus lies on the use cases and domains for which commonsense knowledge is employed, the commonsense aspects that are considered, the datasets/resources used as sources for commonsense knowledge and the methods for evaluating these approaches. Additionally, we discovered a divide in terminology between research from the knowledge representation and reasoning and the cognitive robotics community. This divide is investigated by looking at the extensive review performed by Zech et al. (The International Journal of Robotics Research, 2019, 38, 518–562), with whom we have no overlapping publications despite the similar goals.

KEYWORDS

commonsense knowledge, cognitive robotics, systematic literature review, knowledge representation, semantic reasoning

1 Introduction

Robots have the potential to support us in a number of activities. Recently, there has been a massive adoption of cost-efficient robots that support us in house cleaning (e.g., vacuuming) and gardening (e.g., lawn mowing) activities. Moreover, research in household robotics has led to robots being able to clean breakfast tables (Kazhoyan et al., 2021), or prepare drinks (Sung and Jeon, 2020) and pizzas (Joublin et al., 2024). Yet, the ability of robots to support us in complex everyday tasks is still very limited. In particular, they break down in open world situations where they are challenged by new and underdetermined tasks, new environments or new objects about which they lack knowledge (Bronfman et al., 2021; Ding et al., 2023).

This gap between underdetermined tasks and the robot body motion that accomplishes the task has to be bridged through the robot's knowledge and its reasoning capabilities.

This challenge is the core of the research field of *cognitive robotics*, where knowledge representation and reasoning techniques are employed to support “autonomous robot [s] in a dynamic and incompletely known world” (Levesque and Lakemeyer, 2008, p. 869). A substantial part of these techniques and capabilities used to increase the robustness of cognitive robots in everyday tasks concerns the robot’s *commonsense knowledge* (CSK). This knowledge has the benefit of “enhancing the quality of the plans [...] as well as avoiding human involvement when making decisions” (Pradeepani et al., 2022, p. 159) and allows them “to ask and retrieve the right answers from available knowledge” (Salinas Pinacho et al., 2018, p. 132).

As the name suggests, CSK in humans is understood as “information that people usually take for granted and, hence, normally leave unstated” (Cambria et al., 2012, p. 3582), which increases the difficulty for automatic acquisition and deployment. Regarding the cognitive robotics domain, we follow the definition provided by Gupta and Kochenderfer (2004), which focuses on knowledge about human desires, physics, and causality, as well as knowledge about objects with their locations, properties and relationships. In general, knowledge about human desires correlates to the concept of *Intuitive Psychology* from Lake et al. (2017), with which an agent understands that other agents have a mental state similar to their own which they can express and interpret to understand their intentions and goals. Both knowledge about physics and knowledge about causality are covered by the concept of *Intuitive Physics*, also from Lake et al. (2017). This type of knowledge is focused on primitive physical concepts like the calculation of physically possible trajectories or the tracking of objects over time. With causality, also the knowledge about physical connections between objects and actions is covered. So, for example, CSK focused on causality would help a robot to understand the (physical) consequences of moving an object.

As a general example, consider a cognitive robot tasked with the preparation of a bowl of cereals for breakfast, a task that a human could perform without explicit planning. However, many of the implicitly known aspects for the human are challenges for the robot, since it needs to know that “a bowl of cereal” implies the use of milk or what constitutes a container to be used as the bowl or where to find the cereal in its environment. Without CSK that provides answers to these challenges, the robot would, e.g., search the whole kitchen for milk instead of starting with the most probable location (the fridge) or it would not understand that a found container could be used as the bowl.

By equipping cognitive robots with CSK, their robustness when interacting in open worlds is increased. However, the application of the concept of CSK to the cognitive robotics domain has received relatively limited research attention. There are no surveys or comparable studies performed to analyze the coverage of CSK for cognitive robotics. Since cognitive robotics are increasingly breaching into human domains, we perform a systematic literature review providing researchers and practitioners alike with an overview for CSK in cognitive robotics. For this literature review, we follow the principles and guidelines provided by Kitchenham and Charters (2007), Okoli (2015) and Page et al. (2021). To increase repeatability and traceability of our review, we track our progress in a review protocol and collect all intermediate

results. All of these additional resources are available in our GitHub repository¹.

To guide our research, we formulate the following four research questions, focusing on different aspects of CSK. Our motivation for these questions stems from the need to comprehensively understand the landscape of CSK utilization in cognitive robotics research. By addressing these research questions, we aim to uncover insights into the various use cases, specific aspects considered (or overlooked) in CSK application, the prevalent datasets or resources in the field, and the diverse methods employed for assessing these approaches. This comprehensive examination is crucial in shaping our understanding of the current state and potential future directions of CSK integration in cognitive robotics.

- RQ1** For which use cases has the use of CSK been considered in cognitive robotics research?
- RQ2** Which aspects of CSK have been considered? Which aspects of CSK have received less consideration?
- RQ3** Which datasets or resources are mainly considered in cognitive robotics as a source for CSK?
- RQ4** What methods are employed to assess the approaches? Which CSK datasets or resources are utilized in these evaluations?

To summarize our results, concerning **RQ1** we find that most use cases occur in the household domain and focus on objects and their relations to the environment. This is corroborated by our results pertaining to **RQ2**, which we address by looking at what sorts of questions CSK is called upon to answer. We found that the most common CSK questions seek to connect an object to a specific location in its environment. Other important questions focus on object similarity, object affordances and tool substitution. Here, affordances describe possible ways for an agent to interact with the environment Bornstein and Gibson (1980). In general, questions focusing on objects are much more dominant than questions about interacting with humans or about physics or causality of actions. Concerning **RQ3**, we find that while specific sources such as ConceptNet (Speer et al., 2017) (Open-)Cyc (Lenat, 1995) or OMICS (Gupta and Kochenderfer, 2004) are used multiple times, there is no one single source that is employed in all or most CSK use cases. Regarding the evaluation method and data covered by **RQ4**, we found that most approaches either evaluate using a *Case Study* or an *Experiment*, predominantly in a simulated environment. Unfortunately, most of the evaluation data is not available online.

During our search for suitable publications, we were surprised to notice a lack of publications that focus on well established keywords like *affordance learning*. After manually analyzing this gap using another, similar review by Zech et al. (2019)—with which we have no overlapping publications—we hypothesize that the reason is a divide in terminology between research in the cognitive robotics community and in the knowledge representation and reasoning community. We further explore this divide and propose possible bridges to close this gap.

¹ <https://github.com/ag-sc/Robot-Commonsense-Review>

2 Related work

Commonsense and intuitive physics reasoning problems were driving forces for knowledge representation and reasoning in early stages of AI research (McCarthy, 1959; 1977; Schank and Abelson, 1975; Minsky, 1981; Hayes, 1990). This line of research was presented in textbooks (Davis, 1990; 2008a; Mueller, 2014) and further developed within its own research community (Davis, 2008a; b; Levesque et al., 2012; Davis and Marcus, 2015). In current AI research, CSK is used for question-answering (Talmor et al., 2019; Nguyen et al., 2021), knowledge base creation (Tandon et al., 2017), text interpretation (Bisk et al., 2020; Puri et al., 2023) and visual recognition (Zellers et al., 2019), to name a few.

While a large fraction of research problems were motivated through intuitive physics and physical agency, they were not sufficiently leveraged in cognitive robotics research. Another characteristic of CSK and reasoning is its hybrid nature. Commonsense reasoning includes a large number of specialized methods for prospection (Szpunar et al., 2014), part-based reasoning (Tenorth and Beetz, 2017), mental simulation (Hesslow, 2012), imagistic reasoning (Nanay, 2021), planning (Ghallab et al., 2016), and safe human-robot collaboration (Conti et al., 2022), which were investigated individually without being linked to the more general concept of commonsense. In addition, representations of actions as they are investigated in natural language processing, such as FrameNet (Baker et al., 1998), are of key importance for robotic commonsense (Vernon, 2022). Furthermore, robot cognitive architectures contribute to robot commonsense by focusing on cognitive capabilities (Vernon, 2014; Vernon, 2022).

Regarding previous reviews on the topic of commonsense knowledge in cognitive robotics, as far as we know, no direct previous publications exist. However, works by Paulius and Sun (2019) and Sun and Zhang (2019) survey general knowledge representation techniques employed for different domains and scenarios. The work by Paulius and Sun (2019) focuses on knowledge representation and its connection to learning techniques applied in service robots, covering general high-level as well as specialized representations. Similarly, Sun and Zhang (2019) reviews three types of knowledge representations for task planning in robotics: semantic networks, rules and logical knowledge representation.

The survey conducted by Thosar et al. (2018) focuses on different knowledge bases that are employed by service robots manipulating household objects. These knowledge bases are compared regarding their knowledge acquisition and representation as well as the mechanisms used for inference and symbol grounding. Another review by Buchgeher et al. (2021) that focuses on a specific type of knowledge representation, looks into the usage of knowledge graphs for industrial manufacturing and production systems. The authors analyse application scenarios, graph characteristics and meta information about the surveyed research publications.

Reviews by Olivares-Alarcos et al. (2019) and Manzoor et al. (2021) focus on ontology-based approaches for knowledge representation. The review conducted by Olivares-Alarcos et al. (2019) surveys the cognitive capabilities supported by different ontologies, and compares them using a proposed classification schema based on the underlying ontology language and hierarchy as well as the application domain of the ontology. The review by Manzoor et al. (2021), on the other hand, focuses specifically on

the household, hospital, and industry domains, looking for concrete scenarios where the ontologies have been applied on real robots.

Lastly, the literature review by Zech et al. (2019) focuses on the concept of *actions* in the cognitive robotics domain by looking at their representation and providing a possible taxonomy for their classification. Based on the classification of 152 publications, the authors summarize open research challenges for increasing the maturity and usability of action representations. This review exemplifies the divide mentioned in Section 1 regarding the terminology used by researchers with a (cognitive) robotics background and researchers in the knowledge representation and reasoning domain, since some concepts covered by their taxonomy are semantically equivalent to concepts from the knowledge representation and reasoning domain without being explicitly connected.

The reviews and surveys presented here differ in the knowledge representation approach covered, the application domain, and whether the review is structured in a systematic way. The topic of commonsense knowledge itself is not covered by any of these reviews. Due to the importance of commonsense knowledge for cognitive robotics, we investigate its application domain, data sources, evaluation methods and commonsense aspects in a systematic way.

3 Methodology for searching relevant publications

To find publications suitable for answering our research questions RQ1–RQ4, we follow a structured, pre-defined procedure as proposed by Kitchenham and Charters (2007), Okoli (2015) and Page et al. (2021). To enhance the repeatability of our review, we create a review protocol containing additional information about the search as well as an overview of intermediate results. The protocol, as well as all additional artifacts, are available in our GitHub repository.

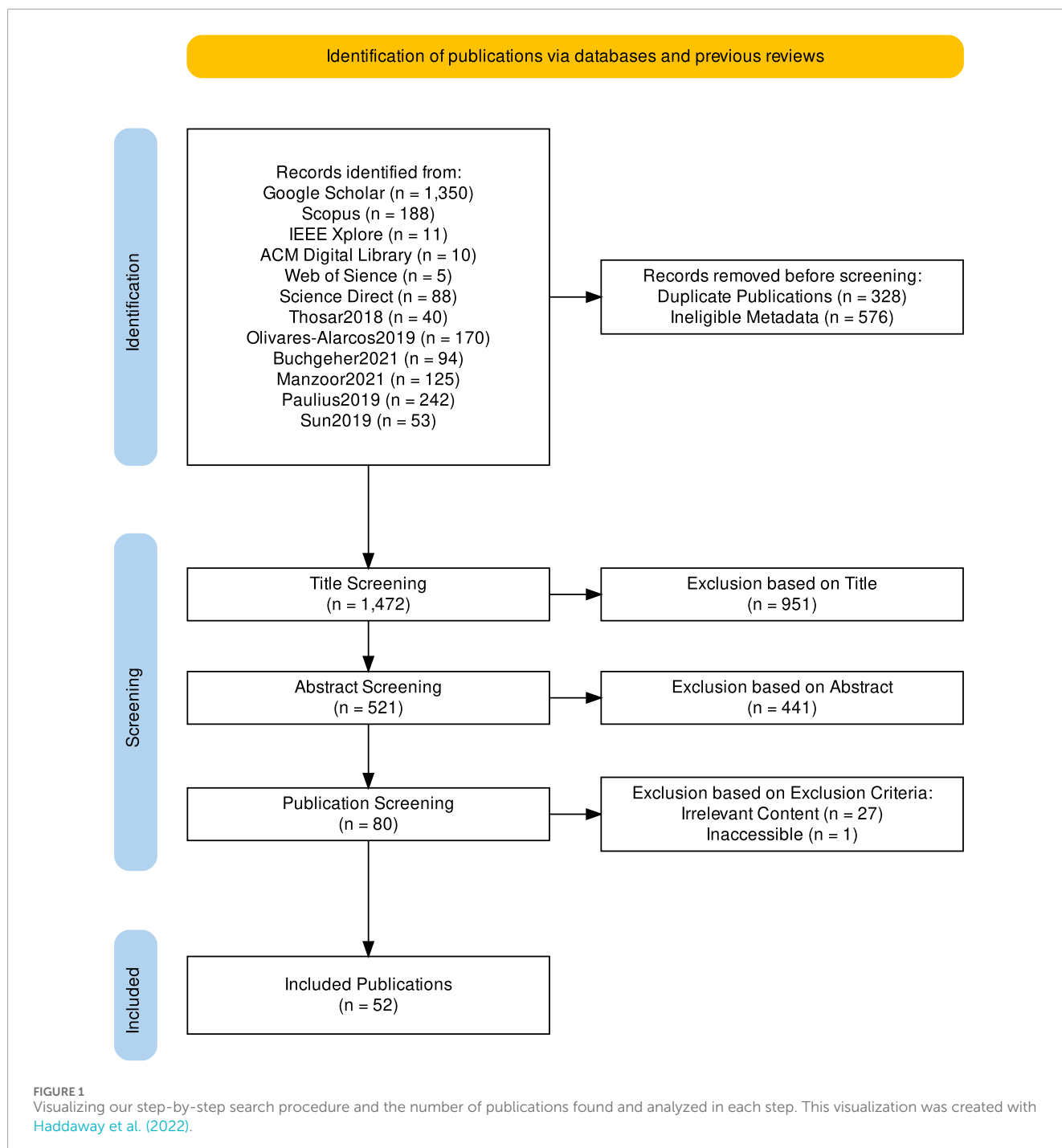
3.1 Applied search procedure

To find publications suitable for answering our four research questions, we combine a keyword-based database search with a snowballing search on related surveys. The general procedure used, along with the quantity of publications identified and screened in each step, is visually summarized in Figure 1.

For the database search we defined the following four keywords before we started the search:

- K1: “knowledge-enabled robot” OR “knowledge-based robot” OR “knowledge-driven robot”
- K2: “knowledge processing” AND robot AND question AND NOT interaction AND NOT hardware
- K3: “common sense knowledge” AND robot AND NOT interaction AND NOT hardware
- K4: “common sense” AND (“robot cognition” OR “cognitive robot”)

We used each of these four keywords on the following six search engines/databases: Google Scholar, IEEE Xplore, Scopus, Web of Science, Science Direct and the ACM Digital Library. Through the



combination of our keywords with these six sources, we found 1,652 publications.

Since we are not the first researchers to perform a literature review in the domain of knowledge representation and reasoning for cognitive robotics, we also decided to incorporate the results of previous published reviews. For this, we follow the guidelines by Wohlin (2014) for performing a snowballing search to gather publications that were either already covered by the reviews introduced in Section 2 or that cite these reviews. By collecting the in- and outgoing references of the reviews by Thosar et al. (2018),

Olivares-Alarcos et al. (2019), Paulius and Sun (2019), Sun and Zhang (2019), Buchgeher et al. (2021) and Manzoor et al. (2021), we included 724 additional publications.

Combining the results of both search techniques yielded 328 duplicates, which we removed. We then analyzed the remaining publications regarding their metadata and removed 576 publications that did not fit the inclusion criteria described in Section 3.2. Next, we screened the 1,472 remaining publications in two steps, first looking only at their title, and then also covering their abstract. During these steps, we decided whether to include a publication

using further steps based on the inclusion criteria specified in [Section 3.2](#). This led us to exclude 951 publications based on their title and 441 based on their abstract, leaving us with 80 publications, which we read completely.

Of these 80 publications, one was not accessible in a full version, prompting us to exclude it as well. Of the remaining 79 publications, 27 were excluded based on the exclusion criteria described in [Section 3.2](#), leaving us with 52 publications, which we analyzed to answer our research questions. A brief summary of these publications can be found in our review protocol.

3.2 Inclusion and exclusion criteria

To enhance the repeatability of our search, we define our inclusion and exclusion criteria before we start the search, as suggested by [Kitchenham and Charters \(2007\)](#). For the inclusion criteria, we differentiate between criteria regarding a publication's metadata and its content. Regarding the metadata, we only include publications that were published in our investigated time frame of 11 years (i.e., between 2012 and 2022). For most of our data sources, these criteria were already applied during the search through explicit filters. Additionally, only papers that are written in English and thus understandable by the broad scientific community are included. Regarding the scientific quality, we focus only on publications that are peer-reviewed, excluding patents, books, presentations, technical reports and theses of any kind. Regarding the content, we analyze the title of the publication and its abstract in two separate steps to determine whether it contains a possible answer to any of our research questions. So, we include publications that discuss the application of CSK through a robot to a specific scenario or use case (RQ1), publications that discuss equipping cognitive robots with the possibility to answer certain CSK questions (RQ2), or that introduce or employ a (novel) source for collecting the necessary CSK (RQ3). In general, anything the authors employ as a source for gathering their CSK constitutes as an eligible resource for our analysis. This can cover texts, ontologies, websites, large language models or other kinds of data. Lastly, we do not define a specific inclusion criteria for assessing the evaluation methods and their used data (RQ4), since we expect all remaining publications to somehow evaluate their approach.

As we explain in [Section 3.1](#), the exclusion criteria are applied after the metadata, title and abstract have already been analyzed. Here, we first exclude publications for which no complete version is available, thus making a thorough analysis impossible. Additionally, we exclude any publication we read completely but that turns out not to provide answers to any of our research questions, despite content in the title or abstract suggesting that it does.

4 Analyzing the usage of commonsense knowledge

In this section, we analyze the content of the 52 publications found by the search procedure detailed in [Section 3.1](#) to answer our four research questions introduced in [Section 1](#). However, we first examine two aspects of their metadata. First, we examine

the number of publications published for each year in our 11-year time span, visualized in [Figure 2A](#). We do not find rising or falling trends in interest in the topic of CSK for cognitive robotics throughout these years, with a median of five publications per year. We also examine the venues where these publications were published. However, only three venues occur more than once: *Intelligent Service Robotics*² (2 occurrences), *IEEE International Conference on Robotics and Automation (ICRA)*³ (6 occurrences) and *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*⁴ (7 occurrences). For a more general examination, we summarize the venue type for all publications in [Figure 2B](#). Here, we find that the majority of publications are conference papers (~60%), followed by journal articles (25%), workshop papers (~10%) and lastly book sections (~5%).

4.1 Use cases and their application domain

Our first research question RQ1 pertains to the use cases for which the use of CSK has been considered in cognitive robotics research. When addressing this question, we differentiate between the concrete use case itself and the domain in which it is embedded. We look at both independently, since a given use case is not always embedded in a single domain. For example, the approach by [Wang et al. \(2019\)](#) focuses on the use case of finding and delivering a given object in the household domain, whereas [Yang et al. \(2019\)](#) focuses on the same use case but for the personal care domain.

For the distinction between possible domains and use case, we rely on the arguments and descriptions presented in each publication. The found domains are self-explanatory and mostly reported directly in each publication, so any publication that talks about the “household environment” counts towards the *Household* domain. For the use cases however, we collect their attributes and goals to distinguish and define the 15 different use cases seen below. Due to the difference in abstraction between these use cases, more complex use cases like, e.g., *Cooking* depend on other, more simplistic and low-level use cases like localizing or picking up objects. These dependencies are visualized in [Figure 3](#).

- **Cooking** ([Nyga and Beetz, 2012](#); [Agostini et al., 2015](#)): Generate and execute a cooking plan based on the current environment and a requested meal
- **Environment Exploration** ([Pangercic et al., 2012](#); [Kanjarek et al., 2015](#); [Jäger et al., 2018](#); [Vassiliades et al., 2020](#); [Zhang et al., 2021](#)): Interacting with parts of the environment (objects, doors, cupboards, etc.) to gather (new) knowledge
- **Hole Digging** ([Javed et al., 2016](#)): Dig a hole in the garden
- **Intention Inference** ([Liu et al., 2015](#); [Liu and Zhang, 2016](#); [De Silva et al., 2022](#)): Identify the intention of a human with a certain object/command to react fittingly when the command cannot be executed (e.g., the robot should fetch the human some juice, which is not available. Why did the human want the juice and what is a fitting alternative?)

2 www.springer.com/journal/11370

3 www.ieee-ras.org/conferences-workshops/fully-sponsored/icra

4 www.ieee-iros.org

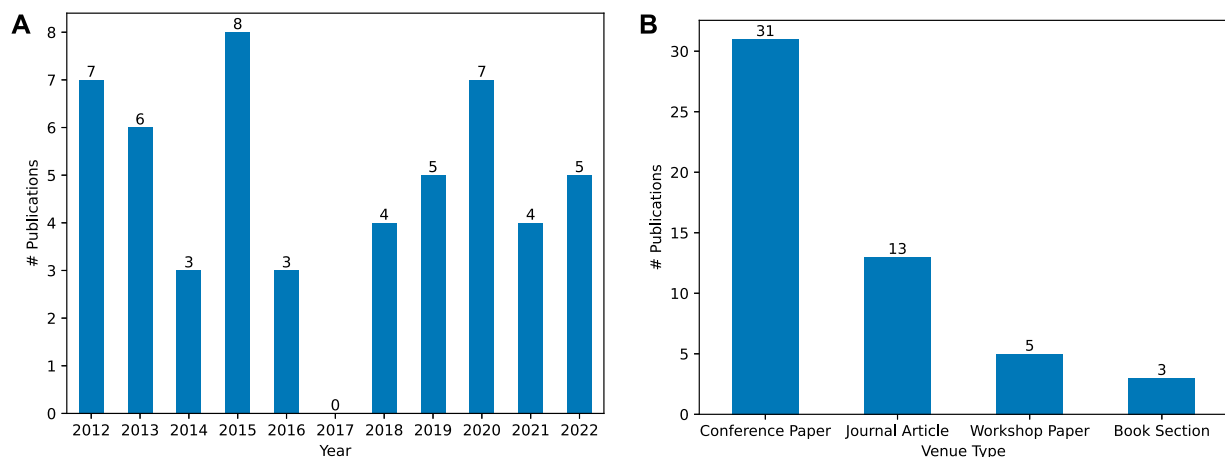


FIGURE 2

(A): Visualizing the amount of found publications throughout the 10 year time span we restricted the search to. (B): Visualizing the venue type where the found publications were published.

- **Location Detection** (Welke et al., 2013): Categorize the location based on the recognized objects (e.g., the robot detects milk and juice and concludes that the location is a fridge)
- **Navigation** (Shylaja et al., 2013; Li et al., 2022): Navigate to a specific location
- **Object Delivery** (Lam et al., 2012; Riazuelo et al., 2013; Mühlbacher and Steinbauer, 2014; Al-Moadhen et al., 2015; Zhang and Stone, 2015; Wang et al., 2019; Yang et al., 2019): Finding the requested object and delivering it to a specific location
- **Object Localization** (Varadarajan and Vincze, 2012b; Zhou et al., 2012; Kaiser et al., 2014; Riazuelo et al., 2015; Jebbara et al., 2018; Daruna et al., 2019; Zhang et al., 2019; Chernova et al., 2020): Finding a specific object in an (unknown) environment
- **Object Recognition** (Daoutis et al., 2012; Pratama et al., 2014; Kümpel et al., 2020; Chiatti et al., 2022): Recognize a specific object based on its properties
- **Pick and Place** (Al-Moadhen et al., 2013; Javia and Cimiano, 2016; Mitrevski et al., 2021): Pick an object up and place it at a different location
- **Reminiscence Therapy** (Wu et al., 2019): Asking questions about provided pictures to get the human to remember and socialize
- **Table Setting** (Salinas Pinacho et al., 2018; Haidu and Beetz, 2019): Set the table for a meal scenario (and maybe also clean up afterwards)
- **Tidy Up** (Aker et al., 2012; Skulkittiyut et al., 2013): Bring a specified part of the environment in order by removing unusual objects
- **Tool Substitution** (Zhu et al., 2015; Thosar et al., 2020; 2021; Dhanabalachandran et al., 2021; Xin et al., 2022): Recognizing a specific object as a suitable substitute for a missing tool
- **Warehousing** (Ayari et al., 2015; Pradeepani et al., 2022): Keep track of available objects and their quantity in

an environment to inform a human once an object is unavailable

During our analysis, we found five different domains where CSK is applied: the *Household* and *Retail* domains, the *Gardening* domain, the *Personal Care* domain, and the *Generic* domain in which the robot handles CSK in a way that can be applied to any other domain. The number of publications that handle each of these domains is visualized in Figure 4A. As can be seen, the *Household* domain is the focus of 50% of the covered publications, surpassing applications in the *Generic* domain (~33%). One commonality shared by the *Household*, *Personal Care* and *Retail* domains (~65% of publications) is that robots operating in these domains potentially share their workspace with humans, which can lead to uncertainties in the environment that increase the need for robots to have and draw on CSK. Other domains where robots do not typically share their workspace with humans, such as industrial and manufacturing domains that tend to allow for better known and more deterministic environments, were not found during our analysis, despite the inclusion of approaches from these domains through the snowballing search on the reviews by Buchgeher et al. (2021) and Manzoor et al. (2021).

In addition to examining the application domain, we also investigate the specific use case with which each approach is concerned in Figure 4B. Here we distinguish between approaches that focus solely on a specific use case (e.g., Salinas Pinacho et al. (2018) focuses on the *Table Setting* use case) and approaches where a specific use case is used as an example or proof-of-concept to demonstrate the viability of the approach being proposed (e.g., Jebbara et al. (2018) use the *Object Localization* scenario to prove the applicability of their CSK extraction technique for the cognitive robotics domain). Roughly 46% of the analyzed publications (24 out of 52) focus on the use case they examine, whereas ~46% use it only as an example application. The remaining four publications (Tenorth and Beetz, 2013; Beetz et al., 2018; Jakob et al., 2020; Beßler et al., 2022) have no specific use case, instead describing techniques intended to be generally applicable to the *Household* domain.

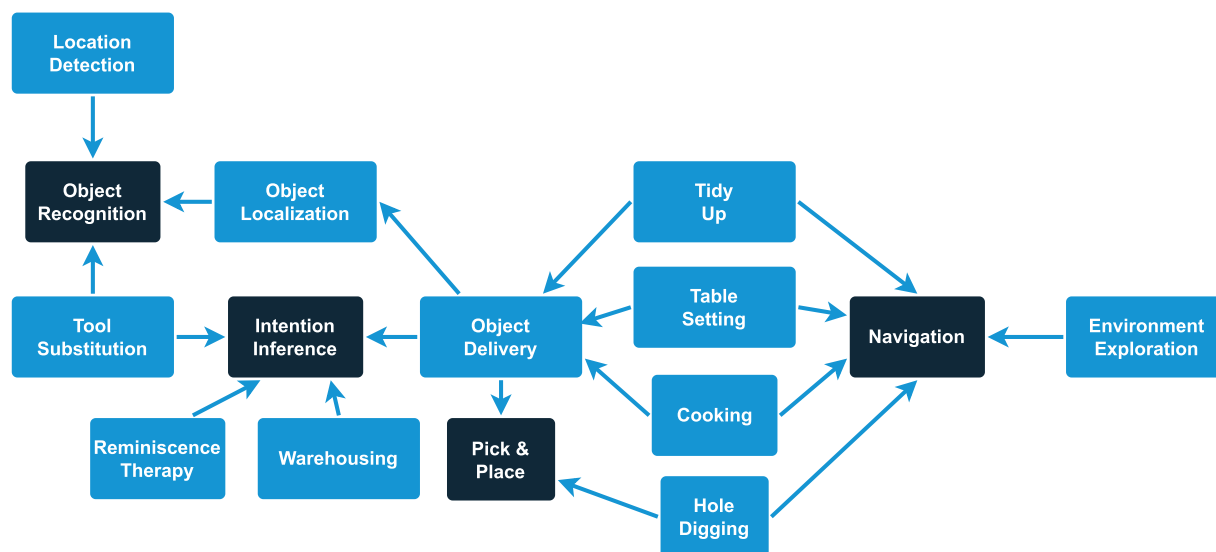


FIGURE 3
Visualizing the dependencies between the 15 different use cases. The four use cases in the darker rectangles are the low-level use cases that do not depend on any other use case.

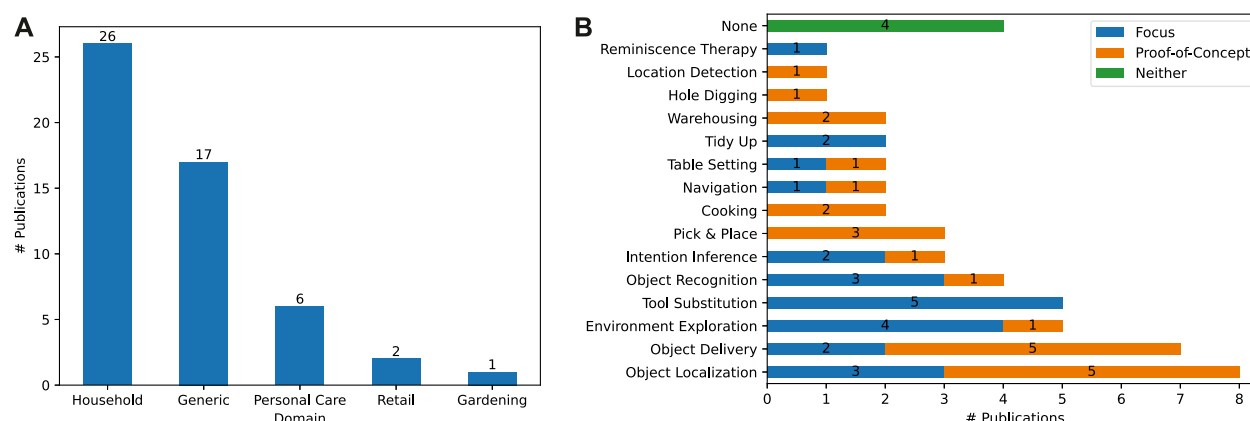
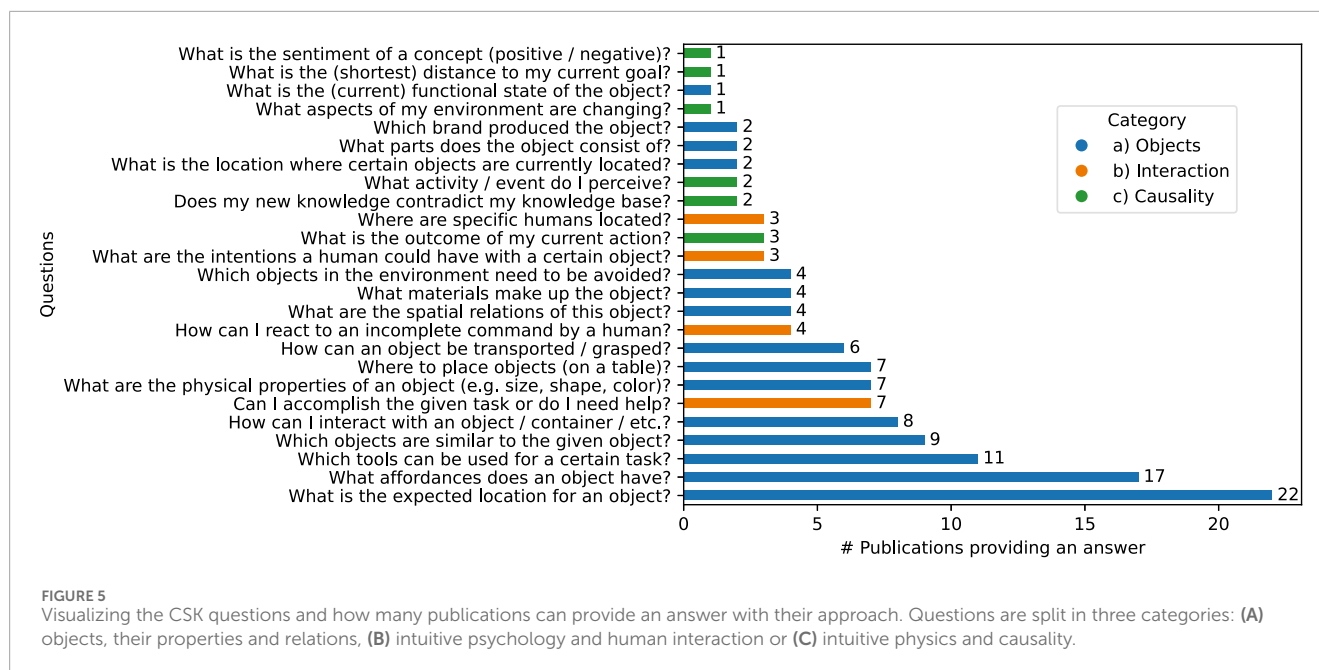


FIGURE 4
(A): Visualizing the different domains in which the approaches operate. (B): Visualizing the different use cases the approaches work on using CSK. We differentiate between publications that focus on their chosen use case or that use it as a proof-of-concept. A full explanation for each use case can be found in our review protocol.

In general, use cases that focus on objects, and on their locations, affordances and relationships (*Object Localization*, *Object Delivery*, *Tool Substitution*, *Object Recognition*, *Pick and Place*, *Warehousing* and *Location Detection*) make up the majority of use cases, occurring in 30 out of 52 publications (~58%). Concrete household tasks like *Cooking*, *Tidying up* and *Table Setting*, which internally rely on the aforementioned object-focused use cases, are only covered in six publications (~12%). As we mentioned before, the majority of domains covered in our survey focus on environments that are shared by robots and humans. However, only four of the publications we analyzed cover direct interaction with humans through two use cases (*Intention Inference* and *Reminiscence Therapy*) (~8%).

4.2 (Un-)Answerable questions about commonsense knowledge

This section discusses the different commonsense questions for which the approaches discussed in the publications we analyzed can provide an answer (see RQ2). We gather these questions by analyzing the goals and capabilities of the approaches, keeping in mind the definition of CSK from Gupta and Kochenderfer (2004) provided in Section 1. This resulted in 25 different questions, which we separated into three categories: a) Objects, their properties and relations (e.g., *How can an object be transported/grasped?*), b) Intuitive psychology and human interaction (e.g., *What are the intentions a human could have with a certain object?*) and c) Intuitive physics and causality



(e.g., *What is the outcome of my current action?*). We provide a visual summary of the 25 questions, their categories and the number of publications in which the discussed approach provides or proposes an answer in Figure 5A. The complete list of approaches that can answer each question is provided in the review protocol.

In general, the majority of questions, 15 out of 25 (60%), focus on objects, object properties and object relations. Looking at the number of approaches providing an answer, we discovered that 47 out of the 52 approaches (~90%) can provide an answer to any question from this category. This heavy focus on objects is also represented in the most researched CSK questions, since eight out of the nine most answered questions in Figure 5B revolve around objects. Questions regarding intuitive psychology and human interaction are focused on in 14 out of 52 publications (~27%). However, these 12 publications concern themselves only with four different questions (16%). The remaining six questions all have to do with intuitive physics and causality (24%). However, only eight out of 52 publications try to achieve an answer to any questions in this category (~15%), a given publication often being the only approach that tries to answer the questions with which it is concerned (e.g., Shylaja et al. (2013) is the only approach answering the question *What aspects of my environment are changing?*).

Based on the aforementioned definition of CSK from Gupta and Kochenderfer (2004), we can provide some example questions that none of the 52 approaches analyzed are capable of answering. As we already observed, knowledge about intuitive physics and about intuitive psychology are not as well covered as knowledge about objects (see Figure 5A). Possible questions in these areas could be *How can I (proactively) support the human in reaching their goals?* or *How do I handle objects based on their state of matter?*. In addition to the more general object knowledge covered under the definition of Gupta and Kochenderfer (2004), more specific object properties only relevant for a specific use case/scenario are also investigated. This task-specific object knowledge is covered for the most frequently occurring use cases

like Object Localization/Delivery or Tool Substitution. However, for more complex use cases like Cooking or Table Setting the necessary object knowledge to answer questions like *How does this ingredient need to be processed to make it consumable?* or *What is a suitable table setup for a specific meal?* are not covered.

It should be noted that it is possible that approaches outside of our analyzed set do cover some of these questions. However, our systematic approach lets us conclude that any such publication either does not apply its approach in the cognitive robotics domain or does not relate these questions to the keyword *commonsense knowledge*. We will talk about this divide in terminology in more detail in Section 5.

4.3 Sources for commonsense knowledge

To answer RQ3, we analyze the different knowledge sources employed by the analyzed publications. An overview of the 30 sources found and their properties can be examined in Table 1. To evaluate their relevance for the domain of cognitive robotics, we count the number of publications in which they occur. Additionally, we categorize them based on their type according to the criteria described by Hitzler et al. (Hitzler et al., 2010, Ch. 8.2, pp. 310-317). Lastly, we check whether the source is still available and can be downloaded and used.

Before analyzing the usage of these publications, we provide a quick overview over their capabilities:

- *ConceptNet* (Speer et al., 2017): ConceptNet is a semantic, multilingual network describing concepts through words and their commonsense relationships to each other. The necessary knowledge is collected through crowd-sourced resources, games with a purpose and resources created by experts.
- *OMICS* (Gupta and Kochenderfer, 2004): The *Open Mind Indoor Common Sense Project* is a collection of CSK for

TABLE 1 The 30 CSK sources employed by the 52 analyzed publications. Abbreviations in the Type column stand for *Structured (S)*, *Semi-Structured (SS)*, *Unstructured (U)* and *Human (H)* (Hitler et al., 2010, Ch. 8.2, pp. 310–317).

Source	#	Type	Avail	Used by
ConceptNet Speer et al. (2017)	8	S	✓	Vassiliades et al. (2020) ; Aker et al. (2012) Skulkittiyut et al. (2013) ; Zhang et al. (2019) Jakob et al. (2020) ; Wu et al. (2019) Chernova et al. (2020) ; Kümpel et al. (2020)
Humans	7	H	✗	Ayari et al. (2015) ; De Silva et al. (2022) Haidu and Beetz. (2019) ; Lam et al. (2012) Liu and Zhang. (2016) ; Jakob et al. (2020) Nyga and Beetz. (2012)
Manually encoded	7	H	✗	Zhang and Stone. (2015) ; Yang et al. (2019) Beßler et al. (2022) ; Xin et al. (2022) Chiatti et al. (2022) ; Mitrevski et al. (2021) Dhanabalachandran et al. (2021)
OMICS Gupta and Kochenderfer (2004)	6	SS	✗	Al-Moadhen et al. (2013) ; Al-Moadhen et al. (2015) Riazuelo et al. (2013) ; Riazuelo et al. (2015) ; Zhou et al. (2012) Nyga and Beetz. (2012)
(Open-)Cyc Lenat. (1995)	5	S	✓	Al-Moadhen et al. (2013) ; Al-Moadhen et al. (2015) Tenorth and Beetz. (2013) ; Daoutis et al. (2012) Mühlbacher and Steinbauer. (2014)
Perception/Sensors	5	U	✗	Ayari et al. (2015) ; Kanjareuk et al. (2015) Thosar et al. (2020) ; Thosar et al. (2021) ; Jäger et al. (2018)
WordNet Miller. (1995)	5	S	✓	Vassiliades et al. (2020) ; Kanjareuk et al. (2015) Skulkittiyut et al. (2013) ; Chernova et al. (2020) Nyga and Beetz. (2012)
Experience/Memories	4	H	✗	Shylaja et al. (2013) ; Pratama et al. (2014) Salinas Pinacho et al. (2018) ; Beetz et al. (2018)
Not mentioned	3	-	✗	Javed et al. (2016) ; Wang et al. (2019) Pangercic et al. (2012)
DBpedia Bizer et al. (2009)	2	S	✓	Jebbara et al. (2018) ; Vassiliades et al. (2020)
Google Books Corpus Goldberg and Orwant (2013)	2	U	✓	Kaiser et al. (2014) ; Welke et al. (2013)
Google Search Engine	2	U	✓	Skulkittiyut et al. (2013) ; Zhou et al. (2012)
WikiHow ^a	1	U	✓	Liu et al. (2015) ; Nyga and Beetz. (2012)
AfNet Varadarajan and Vincze. (2012a)	1	S	✓	Varadarajan and Vincze. (2012b)
AI2Thor Kolve et al. (2017)	1	SS	✓	Daruna et al. (2019)
BKN Lam et al. (2011)	1	S	✓	Lam et al. (2012)
BERT Devlin et al. (2019)	1	U	✓	Pradeepani et al. (2022)
Bing Image Search ^b	1	U	✓	Zhou et al. (2012)
Ehow Recipes ^c	1	U	✓	Kaiser et al. (2014)
FrameNet Baker et al. (1999)	1	S	✓	Nyga and Beetz. (2012)
KnowRob Tenorth and Beetz. (2013)	1	S	✓	Javia and Cimiano. (2016)
LabelMe Torralba et al. (2010)	1	SS	✓	Zhang et al. (2019)
Matterport3D Chang et al. (2017)	1	SS	✓	Li et al. (2022)

(Continued on the following page)

TABLE 1 (Continued) The 30 CSK sources employed by the 52 analyzed publications. Abbreviations in the Type column stand for *Structured (S)*, *Semi-Structured (SS)*, *Unstructured (U)* and *Human (H)* (Hitzler et al., 2010, Ch. 8.2, pp. 310–317).

Source	#	Type	Avail	Used by
ShapeNet Chang et al. (2015)	1	S	✓	Chiatti et al. (2022)
TTU Dataset Zhu et al. (2015)	1	SS	✓	Zhu et al. (2015)
Unspecified Text	1	U	✗	Agostini et al. (2015)
Unspecified Images	1	U	✗	Xin et al. (2022)
Unspecified Videos	1	U	✗	Zhang et al. (2021)
VirtualHome Puig et al. (2018)	1	SS	✓	Vassiliades et al. (2020)
WikiData ^d	1	S	✓	Kümpel et al. (2020)

^awww.wikihow.com

^bwww.bing.com/visualsearch

^cwww.ehow.com

^dwww.wikidata.org

robots acting in the indoor domain (homes and offices). It collects knowledge in the form of statements, where each statement connects an object with an adjective describing either a property or the current object state.

- (Open-)Cyc ([Lenat, 1995](#)): Cyc provides users with a foundational/top-level ontology describing objects and actions through rules and assertions written by domain experts. OpenCyc and ResearchCyc describe two releases of this knowledge base that each contain a subset of all assertions.
- WordNet ([Miller, 1995](#)): WordNet provides a lexical database of the English language, where words are grouped into so-called *synsets* based on their semantics. Synsets are hierarchically structured using hyper- and hyponym relations as a foundation.
- DBpedia ([Bizer et al., 2009](#)): This project aims to extract structured information from Wikipedia by representing each entity through a unique identifier and its relationship to other entities.
- Google Books Corpus ([Goldberg and Orwant, 2013](#)): This corpus contains text from ~3.5 million English books published between 1,520 and 2008. In addition, the authors provide a dataset containing all syntactic n-grams that can be extracted.
- AfNet ([Varadarajan and Vincze, 2012a](#)): The *Affordance Network* is a database containing structural and material affordances for common household objects. It is commonly employed for recognizing objects through their affordances.
- AI2Thor ([Kolve et al., 2017](#)): This dataset contains 3D indoor scenes that support many types of interaction for simulated robots. It consists of photo-realistic objects and scenes that can be procedurally generated.
- BKN ([Lam et al., 2011](#)): The *Basic-Level Knowledge Network* combines knowledge from children's books, ConceptNet ([Speer et al., 2017](#)), and Google's Web 1T 5-g corpus ([Brants and Franz, 2006](#)) in a knowledge base covering objects and activities. The focus of this knowledge base lies in providing answers to *Where*, *What*, and *How* questions.

- BERT ([Devlin et al., 2019](#)): *Bidirectional Encoder Representations from Transformers* describes a family of large language models that are pre-trained on a corpus of unlabeled text and can be fine-tuned to fit the purpose of the task.
- FrameNet ([Baker et al., 1998](#)): Lexical database of concepts embedded in their semantic frame to better understand the concept's meaning.
- KnowRob ([Tenorth and Beetz, 2013](#)): This knowledge processing system is employed for automated robots and formulates decisions a robot can make as inference tasks that can be answered by virtual knowledge bases (KBs). These KBs combine word meaning from WordNet ([Miller, 1995](#)) with OpenCyc ([Lenat, 1995](#)), gather object information from online shops and contain observed human behavior.
- LabelMe ([Torralba et al., 2010](#)): This database contains annotated images focusing on objects, scenes and their spatial connection. The annotations were provided by volunteers using an online annotation tool. Through this tool, the database accumulated over 400,000 annotations.
- Matterport3D ([Chang et al., 2017](#)): Matterport3D is a large-scale dataset containing panoramic views made up of different images and taken in different buildings. Additional annotations describe information about camera poses and semantic segmentation.
- ShapeNet ([Chang et al., 2015](#)): This is a richly annotated, large-scale dataset containing 3D models for different household objects collected from public repositories and other existing datasets. The objects are categorized on the basis of their corresponding synset in WordNet ([Miller, 1995](#)).
- TTU Dataset ([Zhu et al., 2015](#)): The *Tool and Tool-Use* dataset is used for evaluating the recognition of tools and task-oriented objects by providing a collection of static 3D objects. These objects are combined with a set of human demonstrations regarding their usage.
- VirtualHome ([Puig et al., 2018](#)): The VirtualHome simulator uses a crowd-sourced knowledge base of household tasks, represented through a name and a list of instructions. These

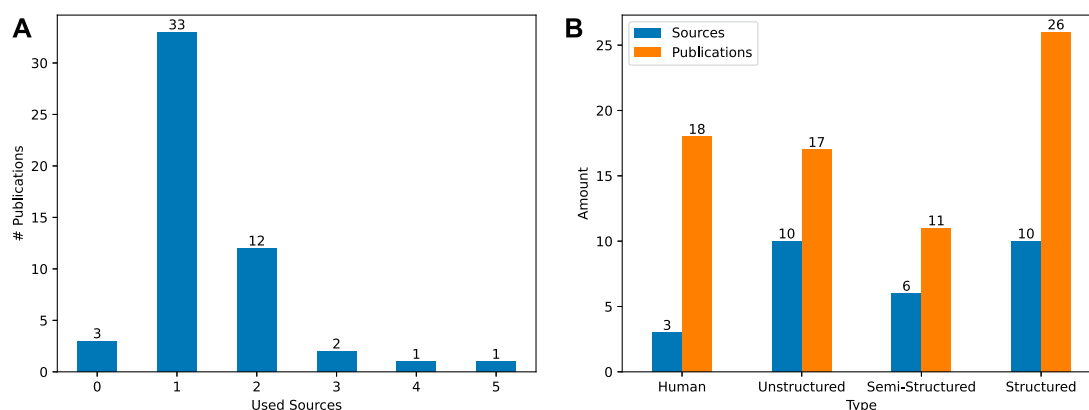


FIGURE 6

(A): Visualizing the number of different sources used by each publication. Across all 52 publications, 30 different CSK sources were employed. (B): Visualizing the types of sources (Hitzler et al., 2010, Ch. 8.2, pp. 310–317), their amount and the number of their occurrences throughout the 52 different publications.

instructions are translated into program code that is executed in a simulated 3D environment by virtual agents.

In general, we do not find one source that is predominantly used. Even ConceptNet (Speer et al., 2017), which has the most usage in our data, is only employed by roughly 15% of publications. Similarly, 17 out of the 30 sources (~57%) we found are only employed by a single publication, which demonstrates that most publications use specialized sources for the specific scenarios they work in rather than relying on a single, more general source. However, even when we focus on a specific use case we do not find a single source on which all approaches rely. This is underlined by the summary of CSK sources per use case provided in Table 2. As described in that table, there is no source that is used more than two times for a specific use case, with most sources occurring only once per use case. This demonstrates that none of the 28 sources provides data specific for a single use case but all of them focus on aspects relevant for different use cases.

In addition to looking at which sources are employed, we also count the number of sources each publication relies on. Here we found that the majority of publications (33 out of 52, ~63%) relies on a single source for extracting its CSK. Only 16 publications (~31%) combine two or more sources, either to cover a broader scope of CSK (e.g., Kümpel et al., 2020) or to increase the quality of the data extracted (e.g., Vassiliades et al., 2020). The described results are visualized in Figure 6A.

Regarding the type of source, we count the number of sources per type and the number of publications employing this type in Figure 6B. Only three sources (~11%) depend on knowledge provided by human domain experts. However, these sources are applied in 35% of publications. The same amount of structured as well as unstructured sources (10 out of 30, ~33%) are used according to our data. However, the ten structured sources are employed the most by the publications (~38%). In general, the high reliance on structured sources is a positive development, since sources of this type are formalized to enhance machine readability.

Despite four approaches that extract CSK from unstructured text (Welke et al., 2013; Kaiser et al., 2014; Agostini et al., 2015; Liu et al., 2015) using NLP techniques, only the approach by Pradeepani et al. (2022) employs a large language model (Devlin et al., 2019) as its data source. Since research on large language models is a rather new domain, approaches that connect them with robots are still scarce (e.g., Ahn et al., 2022) and not yet focused on CSK. This supports the recommendation formulated by Wray et al. (2021) that there needs to be further research to increase the suitability of these models for the cognitive robotics domain.

Lastly, we briefly want to touch on additional sources that are not employed for the extraction of CSK for cognitive robotics. In a recent survey on CSK sources by Ilievski et al. (2021a), 22 different resources were collected and evaluated. However, only four sources found by our analysis overlap with these resources from their study (ConceptNet, WordNet, FrameNet and Wikidata), making up only 15 of the 75 CSK source usages (20%). So the remaining 18 sources have yet to be applied to the cognitive robotics domain.

4.4 Evaluation methods and benchmarking

To answer our last research question and investigate which methods and which datasets are used by the 52 collected approaches during their evaluation, we adapt the evaluation method taxonomy presented by Konersmann et al. (2022) for the software architecture domain. In general, not all methods are applicable to the cognitive robotics domain. In our data, we found *Motivating Examples (Technical) Experiments and Case Studies*. Additionally, we add the method *Model Evaluation* for approaches that evaluate an ML model without connecting it to a simulated or real-world robot. For the two most common methods, *Experiments* and *Case Studies*, we additionally differentiate whether they are performed in the real-world using a real robot or if the robot is simulated and operates in a simulated environment.

The resulting occurrences can be examined in Figure 7. In general, the majority of approaches (~62%) are evaluated using

TABLE 2 Summary of the 15 use cases we found and the sources that are employed to gather commonsense knowledge for the specific use case. The *Hole Digging* use case is omitted since it is only discussed in a single publication that does not mention a source (Javed et al., 2016).

Use case	Employed sources
Object Localization	ConceptNet (2x), OMICS (2x), AI2Thor, AfNet, Bing Image Search DBpedia, Ehow Recipes, Google Books Corpus, Google Search Engine LabelMe, WordNet
Object Delivery	(Open-)Cyc (2x), OMICS (2x), BKN, Humans
Environment Exploration	Perception/Sensors (2x), WordNet (2x), ConceptNet, DBpedia Unspecified Videos, VirtualHome
Tool Substitution	Perception/Sensors (2x), TTU Dataset, Unspecified Images
Intention Inference	Humans (2x), WikiHow
Object Recognition	(Open-)Cyc, ConceptNet, Experience/Memories, ShapeNet, WikiData
Navigation	Experience/Memories, Matterport3D
Pick and Place	(Open-)Cyc, KnowRob, OMICS
Table Setting	Experience/Memories, Humans
Tidy Up	ConceptNet (2x), Google Search Engine, WordNet
Warehousing	BERT, Humans, Perception/Sensors
Cooking	FrameNet, Humans, OMICS, Unspecified Text, WikiHow, WordNet
Location Detection	Google Books Corpus
Reminiscence Therapy	ConceptNet

a quantitative experiment, with most of these approaches being done in a simulated environment (~63%). Generally, simulation environments are used in 50% of publications whereas evaluation on a real robot is performed in only ~37% of publications.

In addition to the evaluation method, we also gather information regarding the data that was used for the evaluation, as well as its availability. Here, we find that 32 out of the 52 publications (~62%) did not publish the data used for the evaluation, and four publications (~8%) did not use any data for their evaluation. In the 16 remaining publications, only two datasets, are used more than once. AI2Thor (Kolve et al., 2017) is used in Daruna et al. (2019); Li et al. (2022) and Al-Moadhen et al. (Al-Moadhen et al., 2013; Al-Moadhen et al., 2015) both use the same basic example household,

described in either publication. Except for two employed datasets, all of the employed datasets are still available either online or by being directly provided in the publication.

These findings notwithstanding, we recognize that in the cognitive robotics domain, a correct execution of the desired task without the occurrence of unwanted side effects can be regarded as a proper and successful evaluation of an approach Vernon et al. (2022). Since the execution environment and the robot programs are often very specific to the lab where they are programmed, there are additional challenges that come with making them publicly accessible (Gu et al., 2023). However, there are certain aspects of CSK for the cognitive robotics domain where benchmarking makes sense. For example, the main question from Section 4.2 *What is the expected location for an object?* is often evaluated by comparing the (automatically) generated locations to a gold standard. However, this gold standard is often not taken from a publicly available dataset, but is instead created by the authors. In general, we observe a lack of benchmarks for domain-specific CSK questions like this.

5 Discussion

Our analysis of the selected publications has revealed interesting limitations and gaps in the way commonsense knowledge is currently used in cognitive robotics research. First of all, while there are many potential use case and applications where commonsense knowledge might support generalization, our analysis has revealed a strong focus on use cases related to acquiring knowledge about objects in order to support things such as object localization and delivery, tool substitutions or pick and place.

This focus on object knowledge is understandable as knowledge about objects to a large extent comprises of static knowledge related to the properties and characteristics of objects, which lend itself to being modeled using the state-of-the-art graph-based knowledge representation language that can straightforwardly model (relational) knowledge about objects using edges or triples. Modeling knowledge about events, their logical and causal structures requires more complex representational paradigms. Further, there are less commonsense knowledge sources containing event knowledge compared to data sources containing (relational) knowledge about concepts and/or objects.

Regarding the sources of commonsense knowledge used, we observe quite a diversity and spread with many different sources being used. This shows that the field seems to be in an experimental state, testing different resources, without clear best practices having emerged. There seem to be no integrative resources that contain all sorts of relevant knowledge so that in the future we can expect that no source will fit all purposes and that robotic systems will have to rely on a combination of sources for different tasks and purposes.

In terms of evaluation and domains, we observe a clear focus on service robotics scenarios and household applications in contrast to the application of robotic systems in industry or production. The explanation for this seems quite natural: industrial settings have less variance and require that the same task is executed over and over with accuracy and precision. In such scenarios there is much less uncertainty than in scenarios where a robot might be confronted with new and unknown tasks, objects, situations, etc. As robots can not be pre-programmed to handle all these situations, flexible

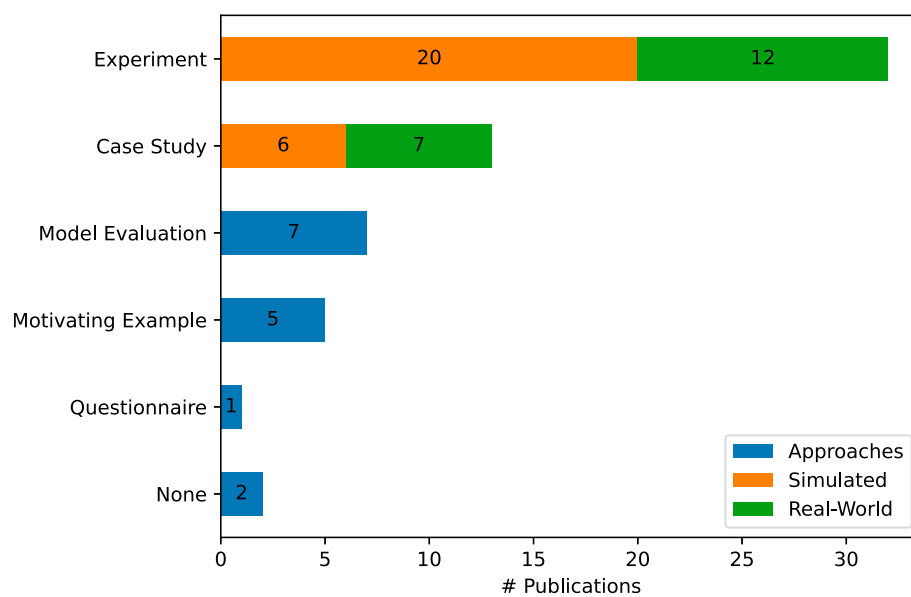


FIGURE 7
The different evaluation methods in our 52 analyzed publications. For the *Case Study* and *Experiment* we differentiate between a simulated or a real-world environment and robot.

reasoning based on commonsense knowledge seems key to master the variance and uncertainty characteristics of such more open environments.

Finally, the lack of focus on physical reasoning and psychological reasoning in terms of applications is understandable, as these types of tasks require commonsense knowledge in the sense of having the ability to simulate physical environments or simulate others to infer their intentions, goals, etc. The first one requires accurate physics engines that would allow a robot to make accurate (forward) predictions. The latter one would require modules to make inferences about other agents, a so called computational Theory of Mind (ToM), the realization of which is a complex and long-term challenge [Lake et al. \(2017\)](#).

As we have seen above, knowledge about objects plays a central role. This is clearly related to the notion of affordances that is studied in cognitive robotics literature, as surveyed by [Min et al. \(2016\)](#). In these approaches, the affordances for the environment and its objects are learned, mostly by using machine learning-based methods on images or videos. What is striking here is that there seems to be a terminological gap in the way the semantic technology or knowledge representation community conceptualize object knowledge and how it is represented. While the semantic tech and KR communities often focus on (static) object knowledge, the cog. rob community focuses on perceptually grounded and action-related knowledge, thus using the concept of ‘affordances’ that indicate an action potential.

To further examine this divide in terminology, we examine the classification performed by [Zech et al. \(2019\)](#) in their review, which we introduced in [Section 2](#). Despite focusing on actions and their possible representation in the cognitive robotics domain, their classification does not connect to keywords associated with the knowledge representation and reasoning community such as

ontology. Similarly, there are concepts that are handled in Zech et al.’s classification, but with a different focus/level of detail than in the publications we analyzed. As an example, we look at the concept of *affordance*. In the classification schema proposed by Zech et al., an affordance is given as an example of an *exteroceptive stimuli*, which is a stimuli generated in the external environment to provide interaction possibilities ([Zech et al., 2019](#)). In our analyzed publications, an affordance is defined as either 1) “a relation of an action/activity/intention and a specific object used to predict the next action/activity” ([Liu et al., 2015](#), p. 1962) or 2) “the relational context holding between several objects that play different roles” ([Beßler et al., 2022](#), p. 8). If we compare these three characterizations of *affordance*, we see that the one by Zech et al. focuses on the immediate application of this concept for robotic action execution, whereas definitions 1) and 2) focus more on the knowledge that an affordance can provide the robot to support, e.g., the planning of future steps or an understanding of the semantic similarities between different objects and actions. Another example is the concept of *intuitive physics*, which we introduced as one part of the definition for CSK in [Section 1](#). This concept has no direct representation in the classification schema by Zech et al., despite its relevance for a successful action execution. The closest concept is *effect associativity*, which analyses whether a representation covers predicting the effect of an action based on its description.

The generalization of task execution knowledge is an important problem in current cognitive robotics research. To allow robots to be employed in domains shared with humans, robots need to be able to handle underspecified commands for manipulating unknown objects in an unknown way in a dynamic environment. Publications like the ones covered by our review and by [Zech et al. \(2019\)](#) are all trying to solve aspects of this generalization problem,

despite coming from different research communities and often using different tools and approaches. This difference is underlined by the fact that there is no overlap between the 52 publications included in our study and the 152 publications included in Zech et al. (2019). In the future, more collaboration is needed to bridge this divide between the two communities, if we are to successfully tackle the task generalization problem.

6 Threats to validity

In general, we integrate different countermeasures into our process by following the general process for systematic literature reviews by Kitchenham and Charters (2007). However, there still remain some biases that we can not completely prevent. To address these threats, we examine selection, measurement and exclusion bias as well as repeatability separately.

Selection Bias: We have selection bias since the insights we gained through the paper analysis depend on the subset of papers we chose. Despite including all 52 publications we deemed suitable for answering our research questions, this inclusion is still based on pre-defined inclusion and exclusion criteria. These were not chosen randomly but derived from our research questions and the search procedure recommendations from Kitchenham and Charters (2007); Okoli (2015); Page et al. (2021).

Measurement Bias: Another problem is measurement bias, since the screening of the search publications was carried out by one of the authors. As a countermeasure, we pre-defined the set of inclusion and exclusion criteria before beginning the search. However, the filtering is still prone to human errors.

Exclusion Bias: Another possible problem stems from the exclusion of potentially interesting publications. By starting our systematic search with recent review papers in cognitive robotics, we have introduced this bias as research in AI, cognitive science, language processing, and cognitive robotics is still not sufficiently connected.

To counter this threat, we pre-defined the criteria we use for including and excluding publications. They are chosen to be as fitting for our research questions as possible and to not hinder the quality of our results. Additionally, no adjustments were made during the screening process. This prevents the exclusion of publications that were initially chosen but then excluded due to a failure to fit results during the analysis.

Repeatability: As the name suggests, threats to this validity describe problems encountered when other researchers try to emulate and repeat this evaluation. To allow for the repetition of the case study, we document all decisions, such as the inclusion and exclusion criteria, the keywords and search engines, in our review protocol. Additionally, all artefacts we created during our review are available in the aforementioned GitHub repository. However, the repeatability of our study is also limited due to the fact that only one person was responsible for screening the search results.

7 Conclusion and future work

In this article, we have investigated the coverage of CSK in the cognitive robotics domain by evaluating the use cases and domains

for which CSK is used, the aspects of CSK that are addressed, the sources employed for gathering the necessary CSK and the method of evaluation. For this purpose, we performed a systematic literature review using a keyword search on six search engines combined with a snowballing search on six related reviews. The resulting 2,048 publications were screened and filtered, which left us with 52 publications deemed suitable for answering our research questions.

By reviewing these 52 publications, we found that most use cases occur in the household domain and focus on objects and their relations to the environment, especially their location. This was corroborated by looking at what sorts of questions CSK are called upon to answer. We found that the most common CSK questions seek to connect an object to a specific location in its environment. Other important questions focus on object similarity, object affordances and tool substitution. Generally, questions focusing on objects are much more dominant than questions about interacting with humans or about physics or causality of actions. Regarding the employed sources, we found that specific sources like ConceptNet (Speer et al., 2017) (Open-)Cyc (Lenat, 1995) or OMICS (Gupta and Kochenderfer, 2004) are used in multiple publications but there is not one single source that covers all relevant aspects of CSK. Similarly, there are often multiple sources used to answer the same CSK questions. Regarding the evaluation performed in these publications, we also found that there are few resources used as data and most of the publications do not publish their evaluation data. This lack of available benchmarks and datasets is surprising since most the publications are evaluated using either a case study or an experiment, which both are mostly performed in simulation, thus leading to a high amount of data necessary for a successful execution. However, only a small amount of publication publish this data.

This review's limitations stem from the threats to validity described in Section 6. In general, we counteract most threats by following the guidelines in Kitchenham and Charters (2007); Okoli (2015); Page et al. (2021) and documenting our decisions and intermediate steps in the review protocol. The main limitation is the data analysis, which was manually performed by a single person.

Lastly, in our discussion of the review by Zech et al. (2019) we emphasized a terminological gap that exist between communities, the knowledge representation community on the one hand and cognitive robotics community on the other hand. These terminological differences need to be bridged towards developing an interdisciplinary research community that synergistically brings together the different aspects of commonsense and makes them actionable in robot control systems.

In the future, focus should lie on the evaluation and benchmarking of commonsense aspects for the cognitive robotics domain, as we explored in Section 4.4. For this, we want to investigate the applicability of commonsense reasoning benchmarks (e.g., CommonsenseQA Talmor et al. (2019)) for the cognitive robotics domain by evaluating their coverage of the relevant aspects we presented in Section 4.2. Additionally, as we explained in Section 4.3, there are different CSK datasets and resources from the survey by Ilievski et al. (2021a), who have yet to be applied to the cognitive robotics domain. This also includes new resources that have been published since the aforementioned study, like the *CommonSense Knowledge Graph* (CSKG) (Ilievski et al., 2021b) or *Ascent++* (Nguyen et al., 2022). Finally, considerable focus

should be put on creating the aforementioned interdisciplinary research community.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: <https://github.com/ag-sc/Robot-Commonsense-Review>.

Author contributions

J-PT: Data curation, Formal Analysis, Investigation, Methodology, Resources, Software, Visualization, Writing—original draft, Writing—review and editing. A-CN: Formal Analysis, Writing—review and editing. MB: Formal Analysis, Writing—review and editing. PC: Conceptualization, Formal Analysis, Supervision, Writing—original draft, Writing—review and editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

References

- Agostini, A., Aein, M. J., Szedmak, S., Aksoy, E. E., Piater, J., and Wörgötter, F. (2015). Using structural bootstrapping for object substitution in robotic executions of human-like manipulation tasks. in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, September, 2015, 6479–6486. doi:10.1109/IROS.2015.7354303
- Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., et al. (2022). Do as I can, not as I say: grounding language in robotic affordances. <https://arxiv.org/abs/2204.01691>.
- Aker, E., Patoglu, V., and Erdem, E. (2012). Answer set programming for reasoning with semantic knowledge in collaborative housekeeping robotics, *IFAC Proc.* 45, 77–83. doi:10.3182/20120905-3-HR-2030.00169
- Al-Moadhen, A., Packianather, M., Qiu, R., Setchi, R., and Ji, Z. (2015). Improving the efficiency of robot task planning by automatically integrating its planner and common-sense knowledge base, in *Knowledge-based information systems in practice* (Berlin, Germany: Springer), 185–199.
- Al-Moadhen, A., Qiu, R., Packianather, M., Ji, Z., and Setchi, R. (2013). “Integrating robot task planner with common-sense knowledge base to improve the efficiency of planning,” in 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems (KES2013), Kitakyushu, Japan, September, 2013. doi:10.1016/j.procs.2013.09.097
- Ayari, N., Chibani, A., Amirat, Y., and Matson, E. T. (2015). “A novel approach based on commonsense knowledge representation and reasoning in open world for intelligent ambient assisted living services,” in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, December, 2015, 6007–6013. doi:10.1109/IROS.2015.7354232
- Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). “The Berkeley FrameNet project,” in *Proceedings of the 36th Annual Meeting on Association for Computational Linguistics*, Montreal, Quebec, Canada, August, 1998. doi:10.3115/980845.980860
- Beetz, M., Beßler, D., Haidu, A., Pomarlan, M., Bozcuoglu, A. K., and Bartels, G. (2018). “KnowRob 2.0 - a 2nd generation knowledge processing framework for cognition-enabled robotic agents,” in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*, Brisbane, Australia, May, 2018, 512–519. doi:10.1109/ICRA.2018.8460964
- Beßler, D., Porzel, R., Pomarlan, M., Vyas, A., Höffner, S., Beetz, M., et al. (2022). “Foundations of the socio-physical model of activities (SOMA) for autonomous robotic agents,” in *Formal ontology in information systems* (Amsterdam: IOS Press), 159–174.
- Bisk, Y., Zellers, R., Le bras, R., Gao, J., and Choi, Y. (2020). PIQA: reasoning about physical commonsense in natural language. *Proc. AAAI Conf. Artif. Intell.* 34, 7432–7439. doi:10.1609/aaai.v34i05.6239
- Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., et al. (2009). DBpedia - a crystallization point for the Web of Data. *J. Web Semant.* 7, 154–165. doi:10.1016/j.websem.2009.07.002
- Bornstein, M. H., and Gibson, J. J. (1980). The ecological approach to visual perception. *J. Aesthet. Art Crit.* 39, 203. doi:10.2307/429816
- Brants, T., and Franz, A. (2006). Web 1T 5-gram version 1. <https://catalog.ldc.upenn.edu/LDC2006T13>.
- Bronfman, Z., Ginsburg, S., and Jablonka, E. (2021). When will robots be sentient? *J. Ai. Consci.* 08, 183–203. doi:10.1142/S2705078521500168
- Buchgeher, G., Gabauer, D., Martinez-Gil, J., and Ehrlinger, L. (2021). Knowledge graphs in manufacturing and production: a systematic literature review. *IEEE Access* 9, 55537–55554. doi:10.1109/ACCESS.2021.3070395
- Cambria, E., Xia, Y., and Hussain, A. (2012). “Affective common sense knowledge acquisition for sentiment analysis,” in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, May, 2012, 3580–3585.
- Chang, A., Dai, A., Funkhouser, T., Halber, M., Nießner, M., Savva, M., et al. (2017). “Matterport3D: learning from RGB-D data in indoor environments,” in *International Conference on 3D Vision (3DV)*, Qingdao, China, October, 2017, 667–676. doi:10.1109/3DV.2017.00081
- Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., et al. (2015). ShapeNet: an information-rich 3D model repository. <https://arxiv.org/abs/1512.03012>.
- Chernova, S., Chu, V., Daruna, A., Garrison, H., Hahn, M., Khante, P., et al. (2020). “Situating bayesian reasoning framework for robots operating in diverse everyday environments,” in *Robotics research*. Editors N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti (Cham: Springer International Publishing), 10, 353–369. doi:10.1007/978-3-030-28619-4_29
- Chiatti, A., Motta, E., and Daga, E. (2022). “Robots with commonsense: improving object recognition through size and spatial awareness,” in *Proceedings of the AAAI 2022 Spring Symposium on Machine Learning and Knowledge Engineering for Hybrid Intelligence (AAAI-MAKE 2022)*, Palo Alto, California, USA, March, 2022.

Acknowledgments

We acknowledge the financial support of the German Research Foundation (DFG) and the Open Access Publication Fund of Bielefeld University for the article processing charge. We also thank Brendan Balcerak Jackson for his feedback on an earlier version of this SLR.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- Conti, C. J., Varde, A. S., and Wang, W. (2022). Human-robot collaboration with commonsense reasoning in smart manufacturing contexts. *IEEE Trans. Autom. Sci. Eng.* 19, 1784–1797. doi:10.1109/TASE.2022.3159595
- Daoutis, M., Coradeschi, S., and Loutfi, A. (2012). Towards concept anchoring for cognitive robots. *Intell. Serv. Robot.* 5, 213–228. doi:10.1007/s11370-012-0117-z
- Daruna, A., Liu, W., Kira, Z., and Chernova, S. (2019). “RoboCSE: robot common sense embedding,” in International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, May, 2019, 9777–9783. doi:10.1109/ICRA.2019.8794070
- Davis, E. (1990). “Representations of commonsense knowledge,” in *Morgan kaufmann series in representation and reasoning* (Cambridge, Massachusetts, United States: Morgan Kaufmann Publishers, Inc).
- Davis, E. (2008a). “Physical reasoning,” in *Handbook of knowledge representation*. Editors F. van Harmelen, V. Lifschitz, and B. W. Porter (Amsterdam, Netherlands: Elsevier), 597–620. doi:10.1016/S1574-6526(07)03014-3
- Davis, E. (2008b). Pouring liquids: a study in commonsense physical reasoning. *Artif. Intell.* 172, 1540–1578. doi:10.1016/j.artint.2008.04.003
- Davis, G., and Marcus, E. (2015). Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM* 58, 92–103. doi:10.1145/2701413
- De Silva, G. W. M. H. P., Rajapaksha, S., and Jayawardena, C. (2022). “Adding common sense to robots by completing the incomplete natural language instructions,” in IEEE 7th International Conference for Convergence in Technology, Mumbai, India, April, 2022, 1–6. doi:10.1109/IC2T54291.2022.9824599
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). “BERT: pre-training of deep bidirectional Transformers for language understanding,” in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, June, 2019. doi:10.18653/v1/N19-1423
- Dhanabalachandran, K., Hassouna, V., Hedblom, M. M., Kümpel, M., Leusmann, N., and Beetz, M. (2021). “Cutting events: towards autonomous plan adaption by robotic agents through image-schematic event segmentation,” in Proceedings of the 11th on Knowledge Capture Conference, New York, NY, USA, December, 2021. doi:10.1145/3460210.3493585
- Ding, Y., Zhang, X., Amiri, S., Cao, N., Yang, H., Kaminski, A., et al. (2023). Integrating action knowledge and LLMs for task planning and situation handling in open worlds. <https://arxiv.org/abs/2305.17590>
- Ghallab, M., Nau, D. S., and Traverso, P. (2016). *Automated planning and acting*. Cambridge, United Kingdom: Cambridge University Press.
- Goldberg, Y., and Orwant, J. (2013). “A dataset of syntactic-ngrams over time from a very large corpus of English books,” in Second Joint Conference on Lexical and Computational Semantics, Atlanta, Georgia, USA, June, 2013, 241–247.
- Gu, J., Chen, L., Jia, Z., Xiang, F., and Su, H. (2023). “An open-source recipe for building simulated robot manipulation benchmarks,” in COMPARE Workshop @ the 40th IEEE International Conference on Robotics and Automation (ICRA), London, UK, May, 2023.
- Gupta, R., and Kochenderfer, M. J. (2004). “Common sense data acquisition for indoor mobile robot,” in Proceedings of the 19th National Conference on Artificial Intelligence, San Jose, California, July, 2004.
- Haddaway, N. R., Page, M. J., Pritchard, C. C., and McGuinness, L. A. (2022). PRISMA2020: an R package and Shiny app for producing PRISMA 2020-compliant flow diagrams, with interactivity for optimised digital transparency and Open Synthesis. *Campbell Syst. Rev.* 18, e1230. doi:10.1002/cl2.1230
- Haidu, A., and Beetz, M. (2019). “Automated models of human everyday activity based on game and virtual reality technology,” in 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, May, 2019, 2606–2612. doi:10.1109/ICRA.2019.8793859
- Hayes, P. J. (1990). “The naive physics manifesto,” in *The philosophy of artificial intelligence*. Editor M. A. Boden (Oxford: Oxford University Press).
- Hesslow, G. (2012). The current status of the simulation theory of cognition. *Brain Res.* 1428, 71–79. doi:10.1016/j.brainres.2011.06.026
- Hitzler, P., Krötzsch, M., and Rudolph, S. (2010). *Foundations of semantic Web technologies*. Boca Raton: CRC Press.
- Ilievski, F., Oltramari, A., Ma, K., Zhang, B., McGuinness, D. L., and Szekely, P. (2021a). Dimensions of commonsense knowledge. *Knowledge-Based Syst.* 229, 107347. doi:10.1016/j.knsys.2021.107347
- Ilievski, F., Szekely, P., and Zhang, B. (2021b). “CSKG: the Commonsense knowledge graph,” in *The semantic Web*. Editors R. Verborgh, K. Hose, H. Paulheim, P.-A. Champin, M. Maleshkova, O. Corcho, et al. (Cham: Springer International Publishing). doi:10.1007/978-3-030-77385-4_41
- Jäger, G., Mueller, C. A., Thosar, M., Zug, S., and Birk, A. (2018). Towards robot-centric conceptual knowledge acquisition. <https://arxiv.org/abs/1810.03583>.
- Jakob, S., Opfer, S., Jahl, A., Baraki, H., and Geihs, K. (2020). “Handling semantic inconsistencies in commonsense knowledge for autonomous service robots,” in 2020 IEEE 14th International Conference on Semantic Computing (ICSC), San Diego, CA, USA, February, 2020, 136–140. doi:10.1109/ICSC.2020.00026
- Javed, A., Raza, S. A., and Azam, M. (2016). “A dynamic ontology based model for intelligent robot,” in 2nd International Multi-Disciplinary Conference (IMDC 2016), Gujrat, Pakistan, December, 2016.
- Javia, B., and Cimiano, P. (2016). “A knowledge-based architecture supporting declarative action representation for manipulation of everyday objects,” in *Proceedings of the 3rd workshop on model-driven robot software engineering*. Editors U. Afsmann, C. Piechnick, and D. Brugali (Leipzig, Germany: ACM Digital Library), 40–46. doi:10.1145/3022099.3022105
- Jebbara, S., Basile, V., Cabrio, E., and Cimiano, P. (2018). Extracting common sense knowledge via triple ranking using supervised and unsupervised distributional models. *SW* 10, 139–158. doi:10.3233/SW-180302
- Joubin, F., Ceravola, A., Smirnov, P., Ocker, F., Deigmoeller, J., Belardinelli, A., et al. (2024). “CoPAL: corrective planning of robot actions with large language models,” in Accepted at 41st IEEE Conference on Robotics and Automation (ICRA 2024), Yokohama, Japan, October, 2024. doi:10.48550/ARXIV.2310.07263
- Kaiser, P., Lewis, M., Petrick, R. P., Asfour, T., and Steedman, M. (2014). “Extracting common sense knowledge from text for robot planning,” in IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, May, 2014, 3749–3756. doi:10.1109/ICRA.2014.6907402
- Kanjaruek, S., Li, D., Qiu, R., and Boonsim, N. (2015). “Automated ontology framework for service robots,” in IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, December, 2015, 219–224. doi:10.1109/ROBIO.2015.7418770
- Kazhoyan, G., Stelter, S., Kenfack, F. K., Koralewski, S., and Beetz, M. (2021). “The robot household marathon experiment,” in 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi’an, China, May, 2021, 9382–9388. doi:10.1109/ICRA48506.2021.9560774
- Kitchenham, B., and Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering*. EBSE-2007-01. Keele, England: Keele University and University of Durham.
- Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., et al. (2017). AI2-THOR: an interactive 3D environment for visual AI. <https://arxiv.org/abs/1712.05474>.
- Konersmann, M., Kaplan, A., Kühn, T., Heinrich, R., Koziol, A., Reussner, R., et al. (2022). “Evaluation methods and replicability of software architecture research objects,” in 2022 IEEE 19th International Conference on Software Architecture (ICSA), Honolulu, HI, USA, March, 2022, 157–168. doi:10.1109/ICSA53651.2022.00023
- Kümpel, M., de Groot, A., Tiddi, I., and Beetz, M. (2020). “Using linked data to help robots understand product-related actions,” in Proceedings of the Joint Ontology Workshops Co-Located with the Bolzano Summer of Knowledge (BOSK 2020), Bozen-Bolzano, Italy, August, 2020.
- Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behav. Brain Sci.* 40, e253–e272. doi:10.1017/S0140525X16001837
- Lam, T. N., Lee, H., Mayama, K., and Mizukawa, M. (2012). “Evaluation of commonsense knowledge for intuitive robotic service,” in IEEE International Conference on Robotics and Automation, Saint Paul, MN, May, 2012, 3679–3684. doi:10.1109/ICRA.2012.6225332
- Lam, T. N., Lee, H., and Mizukawa, M. (2011). Automatic building robot technology ontology based on basic-level knowledge. *J. Robot. Mechatron.* 23, 515–522. doi:10.20965/jrm.2011.p0515
- Lenat, D. B. (1995). CYC: a large-scale investment in knowledge infrastructure. *Commun. ACM* 38, 33–38. doi:10.1145/219717.219745
- Levesque, H., and Lakemeyer, G. (2008). “Cognitive robotics,” in *Handbook of knowledge representation* (Amsterdam, Netherlands: Elsevier), 869–886. doi:10.1016/S1574-6526(07)03023-4
- Levesque, H. J., Davis, E., and Morgenstern, L. (2012). “The winograd schema challenge,” in Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June, 2012.
- Li, Y., Ma, Y., Huo, X., and Wu, X. (2022). Remote object navigation for service robots using hierarchical knowledge graph in human-centered environments. *Intell. Serv. Robot.* 15, 459–473. doi:10.1007/s11370-022-00428-4
- Liu, R., and Zhang, X. (2016). Fuzzy context-specific intention inference for robotic caregiving. *Int. J. Adv. Robotic Syst.* 13. doi:10.1177/1729881416662780
- Liu, R., Zhang, X., Webb, J., and Li, S. (2015). “Context-specific intention awareness through Web query in robotic caregiving,” in IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, May, 2015, 1962–1967. doi:10.1109/ICRA.2015.7139455
- Manzoor, S., Rocha, Y. G., Joo, S.-H., Bae, S.-H., Kim, E.-J., Joo, K.-J., et al. (2021). Ontology-based knowledge representation in robotic systems: a survey oriented toward applications. *Appl. Sci.* 11, 4324–4353. doi:10.3390/app11104324
- McCarthy, J. (1959). “Programs with common sense,” in Proceedings of the Teddington Conference on the Mechanization of Thought Processes, London, December, 1958, 75–91.

- McCarthy, J. (1977). "Epistemological problems of artificial intelligence," in *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Cambridge, MA, USA, August, 1977, 1038–1044.
- Miller, G. A. (1995). WordNet: a lexical database for English. *Commun. ACM* 38, 39–41. doi:10.1145/219717.219748
- Min, H., Yi, C., Luo, R., Zhu, J., and Bi, S. (2016). Affordance research in developmental robotics: a survey. *IEEE Trans. Cognitive Dev. Syst.* 8, 237–255. doi:10.1109/TCDS.2016.2614992
- Minsky, M. (1981). "A framework for representing knowledge," in *Mind design: philosophy, psychology, artificial intelligence*. Editor J. Haugeland (Cambridge, MA, USA: MIT Press), 95–128.
- Mitrevski, A., Plöger, P. G., and Lakemeyer, G. (2021). "Ontology-assisted generalisation of robot action execution knowledge," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, July, 2021, 6763–6770. doi:10.1109/IROS51168.2021
- Mueller, E. T. (2014). *Commonsense reasoning: an event calculus based approach*. Amsterdam, Netherlands: Elsevier Science.
- Mühlbacher, C., and Steinbauer, G. (2014). Using common sense invariants in belief management for autonomous agents. *Mod. Adv. Appl. Intell.*, 49–59. doi:10.1007/978-3-319-07455-9_6
- Nanay, B. (2021). "Mental imagery," in *The stanford encyclopedia of philosophy*. Editor E. N. Zalta (Stanford, CA, USA: Metaphysics Research Lab, Stanford University).
- Nguyen, T.-P., Razniewski, S., Romero, J., and Weikum, G. (2022). Refined commonsense knowledge from large-scale Web contents. *IEEE Trans. Knowl. Data Eng.*, 1–16. doi:10.1109/TKDE.2022.3206505
- Nguyen, T.-P., Razniewski, S., and Weikum, G. (2021). "Inside ASCENT: exploring a deep commonsense knowledge base and its usage in question answering," in *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, Bangkok, Thailand, May, 2021.
- Nyga, D., and Beetz, M. (2012). "Everything robots always wanted to know about housework (but were afraid to ask)," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, Vilamoura-Algarve, Portugal, October, 2012, 243–250. doi:10.1109/IROS.2012.6385923
- Okoli, C. (2015). A guide to conducting a standalone systematic literature review. *Commun. Assoc. Inf. Syst.* 37, 879–910. doi:10.17705/1CAIS.03743
- Olivares-Alarcos, A., Beßler, D., Khamis, A., Gonçalves, P. J., Habib, M. K., Bermejo-Alonso, J., et al. (2019). A review and comparison of ontology-based approaches to robot autonomy. *Knowl. Eng. Rev.* 34, 1–38. doi:10.1017/S0269888919000237
- Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., et al. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ* 372, n71. doi:10.1136/bmj.n71
- Pangercic, D., Pitzer, B., Tenorth, M., and Beetz, M. (2012). "Semantic object maps for robotic housework - representation, acquisition and use," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura-Algarve, Portugal, October, 2012, 4644–4651. doi:10.1109/IROS.2012.6385603
- Paulius, D., and Sun, Y. (2019). A survey of knowledge representation in service robotics. *Robotics Aut. Syst.* 118, 13–30. doi:10.1016/j.robot.2019.03.005
- Pradeepani, M. K. T., Jayawardena, C., and Rajapaksha, U. U. S. (2022). "Adding commonsense to robotic application using ontology-based model retraining," in *International Research Conference on Smart Computing and Systems Engineering (SCSE)*, Colombo, Sri Lanka, September, 2022. doi:10.1109/SCSE56529.2022.9905090
- Pratama, F., Mastrogianni, F., and Chong, N. Y. (2014). "An integrated epigenetic robot architecture via context-influenced long-term memory," in *2014 Joint IEEE International Conferences on Development and Learning and Epigenetic Robotics (ICDL-Epirob)*, Genoa, Italy, October, 2014, 68–74. doi:10.1109/DEVLRN.2014.6982956
- Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S., et al. (2018). "VirtualHome: simulating household activities via programs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June, 2018. doi:10.48550/ARXIV.1806.07011
- Puri, M., Varde, A. S., and De Melo, G. (2023). Commonsense based text mining on urban policy. *Lang Resour. Eval.* 57, 733–763. doi:10.1007/s10579-022-09584-6
- Riazuelo, L., Tenorth, M., Di Marco, D., Salas, M., Galvez-Lopez, D., Mosenlechner, L., et al. (2015). RoboEarth semantic mapping: a cloud enabled knowledge-based approach. *IEEE Trans. Autom. Sci. Eng.* 12, 432–443. doi:10.1109/TASE.2014.2377791
- Riazuelo, L., Tenorth, M., Di Marco, D., Salas, M., Mosenlechner, L., Kunze, L., et al. (2013). "RoboEarth web-enabled and knowledge-based active perception," in *IROS workshop on AI-based robotics*.
- Salinas Pinacho, L., Wich, A., Yazdani, F., and Beetz, M. (2018). "Acquiring knowledge of object arrangements from human examples for household robots," in *KI 2018: advances in artificial intelligence*. Editors F. Trollmann, and A.-Y. Turhan (Berlin, Germany: Springer). doi:10.1007/978-3-030-00111-7_12
- Schank, R. C., and Abelson, R. P. (1975). "Scripts, plans and knowledge," in *Advance papers of the fourth international joint conference on artificial intelligence* (Tbilisi, Georgia: USSR).
- Shylaja, K. R., Vijayakumar, M. V., Davis, D. N., and Prasad, E. V. (2013). "Cognitive architecture to evolve conscious cognitive tasks into common sense actions on agents," in *Proceedings of the world congress on engineering and computer science*. Editors S. I. Ao, C. Douglas, W. S. Grundfest, and J. Burgstone (San Francisco, USA: Newswood Limited), 383–388.
- Skulkittiyut, W., Lee, H., Lam, T. N., Minh, Q. T., Baharudin, M. A., Fujioka, T., et al. (2013). "Commonsense knowledge extraction for tidy-up robotic service in domestic environments," in *IEEE Workshop on Advanced Robotics and Its Social Impacts*, Tokyo, Japan, November, 2013, 63–69. doi:10.1109/ARSO.2013.6705507
- Speer, R., Chin, J., and Havasi, C. (2017). ConceptNet 5.5: an open multilingual graph of general knowledge. *AAAI* 31. doi:10.1609/aaai.v31i1.11164
- Sun, X., and Zhang, Y. (2019). "A review of domain knowledge representation for robot task planning," in *Proceedings of the 2019 4th International Conference on Mathematics and Artificial Intelligence*, Chengdu, China, April, 2019, 176–183. doi:10.1145/3325730.3325756
- Sung, H. J., and Jeon, H. M. (2020). Untact: customer's acceptance intention toward robot barista in coffee shop. *Sustainability* 12, 8598. doi:10.3390/su12208598
- Szpunar, K. K., Spreng, R. N., and Schacter, D. L. (2014). A taxonomy of prospection: introducing an organizational framework for future-oriented cognition. *Proc. Natl. Acad. Sci.* 111, 18414–18421. doi:10.1073/pnas.1417144111
- Talmor, A., Herzig, J., Lourie, N., and Berant, J. (2019). "CommonsenseQA: a question answering challenge targeting commonsense knowledge," in *Proceedings of the 2019 Conference of the North, Minneapolis, Minnesota*, June, 2019, 4149–4158. doi:10.18653/v1/N19-1421
- Tandon, N., De Melo, G., and Weikum, G. (2017). "Webchild 2.0: fine-grained commonsense knowledge distillation," in *Proceedings of ACL 2017*, Vancouver, Canada, July, 2017.
- Tenorth, M., and Beetz, M. (2013). KnowRob: a knowledge processing infrastructure for cognition-enabled robots. *Int. J. Robotics Res.* 32, 566–590. doi:10.1177/0278364913481635
- Tenorth, M., and Beetz, M. (2017). Representations for robot knowledge in the KnowRob framework. *Artif. Intell.* 247, 151–169. doi:10.1016/j.artint.2015.05.010
- Thosar, M., Mueller, C. A., Jaeger, G., Pfingsthorn, M., Beetz, M., Zug, S., et al. (2020). "Substitute selection for a missing tool using robot-centric conceptual knowledge of objects," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, BrnoCzech Republic, March, 2020, 972–979. doi:10.1145/3341105.3374017
- Thosar, M., Mueller, C. A., Jäger, G., Schleiss, J., Pulugu, N., Mallikarjun Chennaboina, R., et al. (2021). From multi-modal property dataset to robot-centric conceptual knowledge about household objects. *Front. Robotics AI* 8, 476084. doi:10.3389/frobt.2021.476084
- Thosar, M., Zug, S., Skaria, A. M., and Jain, A. (2018). "A review of knowledge bases for service robots in household environments," in *Proceedings of the 6th International Workshop on Artificial Intelligence and Cognition*, Palermo, Italy, July, 2018, 98–110.
- Torralba, A., Russell, B. C., and Yuen, J. (2010). LabelMe: online image annotation and applications. *Proc. IEEE* 98, 1467–1484. doi:10.1109/JPROC.2010.2050290
- Varadarajan, K. M., and Vincze, M. (2012a). "AfNet: the affordance network," in *Computer vision - accv 2012*. Editors K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu (Berlin, Germany: Springer). doi:10.1007/978-3-642-37331-2_39
- Varadarajan, K. M., and Vincze, M. (2012b). "AfRob: the affordance network ontology for robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura-Algarve, Portugal, October, 2012, 1343–1350. doi:10.1109/IROS.2012.6386232
- Vassiliades, A., Bassiliades, N., Goudis, F., and Patkos, T. (2020). "A knowledge retrieval framework for household objects and actions with external knowledge," in *The era of knowledge graphs*. Editors E. Blomqvist, P. Groth, V. de Boer, T. Pellegrini, M. Alam, T. Käfer, et al. (Cham: Springer International Publishing). doi:10.1007/978-3-030-59833-4_3
- Vernon, D. (2014). *Artificial cognitive systems: a primer*. Cambridge, Massachusetts, United States: The MIT Press.
- Vernon, D. (2022). "Cognitive architectures," in *Cognitive robotics*. Editors A. Cangelosi, and M. Asada (Cambridge, Massachusetts, United States: The MIT Press), 191–213.
- Vernon, D., Albert, J., Beetz, M., Chiou, S.-C., Ritter, H., and Schneider, W. X. (2022). Action selection and execution in everyday activities: a cognitive robotics and situation model perspective. *TopiCS* 14, 344–362. doi:10.1111/tops.12569
- Wang, Y., Zhang, S., and Lee, J. (2019). Bridging commonsense reasoning and probabilistic planning via a probabilistic action language. *Theory Pract. Log. Program.* 19, 1090–1106. doi:10.1017/S1471068419000371

- Welke, K., Kaiser, P., Kozlov, A., Adermann, N., Asfour, T., Lewis, M., et al. (2013). "Grounded spatial symbols for task planning based on experience," in 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Atlanta, GA, USA, October, 2013, 484–491. doi:10.1109/HUMANOIDS.2013.7030018
- Wohlin, C. (2014). "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, London, England, May, 2014, 1–10. doi:10.1145/2601248.2601268
- Wray, R. E., Kirk, J. R., and Laird, J. E. (2021). "Language models as a knowledge source for cognitive agents," in 9th Annual Conference on Advances in Cognitive Systems, Virtual Event, November, 2021, 1–18. doi:10.48550/ARXIV.2109.08270
- Wu, Y.-L., Gamborino, E., and Fu, L.-C. (2019). Interactive question-posing system for robot-assisted reminiscence from personal photographs. *IEEE Trans. Cognitive Dev. Syst.* 12, 439–450. doi:10.1109/TCDS.2019.2917030
- Xin, J., Wang, L., Wang, S., Liu, Y., Yang, C., and Yin, B. (2022). Recommending fine-grained tool consistent with common sense knowledge for robot. *IEEE Robotics Automation Lett.* 7, 8574–8581. doi:10.1109/LRA.2022.3187536
- Yang, G., Wang, S., and Yang, J. (2019). Desire-driven reasoning for personal care robots. *IEEE Access* 7, 75203–75212. doi:10.1109/ACCESS.2019.2921112
- Zech, P., Renaudo, E., Haller, S., Zhang, X., and Piater, J. (2019). Action representations in robotics: a taxonomy and systematic classification. *Int. J. Robotics Res.* 38, 518–562. doi:10.1177/0278364919835020
- Zellers, R., Bisk, Y., Farhadi, A., and Choi, Y. (2019). "From recognition to cognition: visual commonsense reasoning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, June, 2019, 6720–6731.
- Zhang, S., and Stone, P. (2015). CORPP: commonsense reasoning and probabilistic planning, as applied to dialog with a mobile robot. *Proc. AAAI Conf. Artif. Intell.* 29, 1394–1400. doi:10.1609/aaai.v29i1.9385
- Zhang, T., Zeng, Y., Pan, R., Shi, M., and Lu, E. (2021). Brain-inspired active learning architecture for procedural knowledge understanding based on human-robot interaction. *Cogn. Comput.* 13, 381–393. doi:10.1007/s12559-020-09753-1
- Zhang, Y., Tian, G., Lu, J., Zhang, M., and Zhang, S. (2019). Efficient dynamic object search in home environment by mobile robot: a priori knowledge-based approach. *IEEE Trans. Veh. Technol.* 68, 9466–9477. doi:10.1109/TVT.2019.2934509
- Zhou, K., Zillich, M., Zender, H., and Vincze, M. (2012). "Web mining driven object locality knowledge acquisition for efficient robot behavior," in IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, October, 2012, 3962–3969. doi:10.1109/IROS.2012.6385931
- Zhu, Y., Zhao, Y., and Zhu, S.-C. (2015). "Understanding tools: task-oriented object modeling, learning and recognition," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, June, 2015, 2855–2864. doi:10.1109/CVPR.2015.7298903



OPEN ACCESS

EDITED BY

Silvia Rossi,
University of Naples Federico II, Italy

REVIEWED BY

Carmine Tommaso Recchiuto,
University of Genoa, Italy
Laura Hensel,
University of Glasgow, United Kingdom

*CORRESPONDENCE

Kosuke Sasaki,
✉ sasaki.kosuke.19@shizuoka.ac.jp
Junya Morita,
✉ j-morita@inf.shizuoka.ac.jp

RECEIVED 28 December 2023

ACCEPTED 25 March 2024

PUBLISHED 25 April 2024

CITATION

Sasaki K, Nishikawa J and Morita J (2024),
Evaluation of co-speech gestures grounded in
word-distributed representation.
Front. Robot. AI 11:1362463.
doi: 10.3389/frobt.2024.1362463

COPYRIGHT

© 2024 Sasaki, Nishikawa and Morita. This is
an open-access article distributed under the
terms of the [Creative Commons Attribution
License \(CC BY\)](#). The use, distribution or
reproduction in other forums is permitted,
provided the original author(s) and the
copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic practice.
No use, distribution or reproduction is
permitted which does not comply with
these terms.

Evaluation of co-speech gestures grounded in word-distributed representation

Kosuke Sasaki^{1*}, Jumpei Nishikawa² and Junya Morita^{1,2,3*}

¹Department of Informatics, Graduate School of Integrated Science and Technology, Shizuoka University, Shizuoka, Japan, ²Department of Information Science and Technology, Graduate School of Science and Technology, Shizuoka University, Shizuoka, Japan, ³Department of Behavior Informatics, Faculty of Informatics, Shizuoka University, Hamamatsu, Japan

The condition for artificial agents to possess perceivable intentions can be considered that they have resolved a form of the symbol grounding problem. Here, the symbol grounding is considered an achievement of the state where the language used by the agent is endowed with some quantitative meaning extracted from the physical world. To achieve this type of symbol grounding, we adopt a method for characterizing robot gestures with quantitative meaning calculated from word-distributed representations constructed from a large corpus of text. In this method, a “size image” of a word is generated by defining an axis (index) that discriminates the “size” of the word in the word-distributed vector space. The generated size images are converted into gestures generated by a physical artificial agent (robot). The robot’s gesture can be set to reflect either the size of the word in terms of the amount of movement or in terms of its posture. To examine the perception of communicative intention in the robot that performs the gestures generated as described above, the authors examine human ratings on “the naturalness” obtained through an online survey, yielding results that partially validate our proposed method. Based on the results, the authors argue for the possibility of developing advanced artifacts that achieve human-like symbolic grounding.

KEYWORDS

word-distributed representation, human-robot interaction (HRI), co-speech iconic gesture, natural language processing (NLP), robotics

1 Introduction

In their daily life, people interact with a variety of artifacts. In doing so, they sometimes behave as if the objects have a kind of thinking ability (Nass et al., 1994; Nass and Moon, 2000). In this paper, the object causing such a behavior is called “an agent.” In other words, an agent is an artifact that can interact with people with its purpose, motivation, and intention (Levin et al., 2013; Kopp and Krämer, 2021; Human-Agent Interaction, 2023). We posit that understanding the factors that lead people to perceive artifacts as having these characteristics can facilitate enriching interactions, where humans naturally behave like in human-human interaction (HHI).

Research in the field of human-agent interaction (HAI) has explored factors causing people to perceive the agency in artifacts. Those factors are mainly classified into appearance (MacDorman and Ishiguro, 2006; Yee and Bailenson, 2007), behaviors (Heider and Simmel, 1944; Laban and Ullmann, 1971) and social contexts (Nass et al., 1994; Shiomi, 2023).

Yet, all of them are categorized as external factors, omitting the discussion on the correspondences of internal states and processes (i.e., algorithms and representations (Marr, 1982)) between the artifacts and humans. We argue that studies focusing on the internal factors of these objects pave the way for a foundational design principle for agents. Such agents would appear to possess the aforementioned conditions of agency: purpose, motivation, and intention.

Building on this concept, the current study investigates the relationship between the perception of agency and addressing the symbol grounding problem (Harnad, 1990). Within our framework, symbol grounding is understood as an internal state in which the language (symbols) used by the agent is given some quantitative meaning extracted from the physical world. One possible form of assigning quantitative meaning to language appears in the co-speech gesture (body movement accompanied by verbal language). The correspondence between utterances and actions made by the agent allows humans to infer the existence of meaningful symbols in the agent.

In other words, we assume that some forms of gestures, which are implicitly generated in communicative situations, convey quantitative meaning attached to verbalized words. These gestures can be distinguished from culturally formed emblematic gestures. Rather than directly indexing a specific concept, the gestures focused here iconically enhance imagistic links between linguistic form and meaning (McNeill, 1992; Murgiano et al., 2021). Murgiano et al. (2021) also claimed that such an iconic gesture is part of multimodal systems conveying imagistic meaning in communicative contexts. Thus, a similar role is observed in the prosody that accompanies spoken words. Herold et al. (2011) reported that children deduce novel meaning of antonyms (e.g., “small” vs. “big”) by leveraging prosodic features such as intensity (i.e., a loud and slow voice is connected to a big object).

The connection between a physical image and word meaning (the degree of symbol grounding) varies with word categories. Concrete categories naturally exhibit stronger links than abstract concepts (Utsumi, 2020). Nonetheless, abstract words can also possess imagistic meaning. Lakoff and Johnson (1980) discussed how language is metaphorically shaped through schemata that involve movement within the external world. This concept, known as an image schema, allows for the preservation of the external world's imagery while linking through metaphorical expressions. For instance, when a speaker utters the phrase, “I have an important idea,” we can envision a scenario where the speaker's hand gesture expands to signify the idea's perceived significance. In this gesture, the magnitude of the concept is metaphorically represented through the spatial dimensions defined by the speaker's body structure. Such a set of metaphors connecting a physical experience and an abstract concept is known as a primary metaphor (Grady, 1997).

Existence of the mechanism of exchange for these representations (symbols and quantities) is also supported by various theories in the field of cognitive science. According to the reference frames theory by Hawkins (2021), a continuous space exists behind each concept, mediating language use. Similarly, Tversky (2019) claimed that human language and thoughts originally come from physical experience made in a continuous time and space. In her discussion, the meanings of words are essentially embedded in our living physical world. Other similar discussions are also found in literature in the field of cognitive linguistics (Pinker, 2007).

Summarizing the above background of cognitive science, the authors consider that the gestures generated from a mechanism that is analogous to what humans hold can lead to a realization of the “intrinsically naturalistic” interactions with the agent, where people can perceive “communicative intention (Grice, 1989)” in the agents. In order to construct such a mechanism, a model of word meaning is important. As already noted by the above theories (Pinker, 2007; Tversky, 2019; Hawkins, 2021), word meanings are not defined discretely or independently, but are considered to be defined in a continuous space in which words are interconnected. In the history of natural language processing, statistical analyses (bag of words, co-occurrence frequency, or principal component analysis from word vectors) have been applied to corpora derived from human language operations to capture the semantic relations between words. More recently, vector representations (word-distributed representations) collapsed into the middle layer of a neural network (Bengio et al., 2000) have become the mainstream method for understanding words' quantitative meanings. Such an approach is still evolving and has led to the construction of a variety of large-scale language models (LLM) that enable HAI with human natural communication media (natural language) (Brown et al., 2020; Chowdhery et al., 2023; OpenAI, 2023).

Various quantitative images such as “size” and “speed” can be assumed in the space where words are positioned (Grand et al., 2022). Among those, we focus on the “size images” as a first step to obtain quantitative representations of words related to physical image embedded in the space. Since this image has been frequently utilized in numerous studies (Grady, 1997; Herold et al., 2011), it is suggested that the most representative image for our investigation. By creating iconic gestures (physical images) for a robot through the conversion of “size images” into physical representations, we aim to develop an agent that achieves symbol grounding, which leads to actions that reflect the quantitative images of words. Our objective is to determine whether such embodiment in an agent (robot) leads to an increase in human perception of agency.

More specifically, as an initial step toward the above objective, we set the following research questions:

- 1 How can “size images” evoking an agency perception be extracted from the vector space of word-distributed representations?
- 2 What forms of gesture expression are effective in constructing more natural interaction based on the agency perception?

To address the first question, we introduce a method for extracting “size images” using word-distributed representations and evaluate this method through two experiments. These experiments employ different approaches for associating “size images” with “physical images” in a robot. By comparing the outcomes of these experiments, we aim to investigate the second question. Before presenting the experiments, we introduce a technological background leading to the method of this study and the method of generating “size images” in the following sections.

2 Related works

As a background of our method, we introduce research on modeling word meaning and research on gesture generation for

robots and agents. Based on the review of those related works, we outline our specific approach to gesture generation.

2.1 Modeling word meaning

The meaning of a word or concept can be modeled by several approaches. One traditional approach is to write down the meanings of concepts circulating in society manually. Large-scale databases such as WordNet (Miller, 1995) and ConceptNet (Speer et al., 2017) have been developed so far. These databases define the normative knowledge structure in society.

On the other hand, in recent years, there have been many approaches to statistically capture the meaning of concepts based on the way people use language in their daily lives. The word-distributed representation (Bengio et al., 2000) considers a word as a “pointer” embedded in a vector space. In this framework, the meaning of a word is regarded as the relationship (distance or similarity) between words in the vector space. The underlying idea here is the distributional hypothesis that “words which are similar in meaning occur in similar contexts (Rubenstein and Goodenough, 1965; Sahlgren, 2008)”.

Attempts have been made to extract words’ quantitative images by using word-distributed representations. For example, Utsumi (2020) used word-distributed representations to classify words into attributes and compared them with the classifications obtained from human data. The results suggest that the vector space of word-distributed representation captures aspects of human knowledge, showing that abstract concepts are more deeply (remotely) embedded in word distributions than in words with physical meanings associated with animates.

In addition, Grand et al. (2022) proposed a method for extracting context-dependent relations using word-distributed representations. Context-dependent relations imply that a word like “dog” can embody multiple semantic features such as “size,” “intelligence,” and “danger,” with a particular feature becoming prominent depending on the context. This study shows that by projecting word vectors onto an axis representing a focused feature, it is possible to simulate human estimation of the quantitative features in various objects. Thus, it is suggested that human quantitative images of words are embedded in word-distributed representations. In other words, the quantitative meanings of concepts that humans have physically acquired are inherent in word-distributed representations created from our daily language use.

2.2 Gesture generations in human-agent interaction

The current study focuses on symbol grounding as a factor inducing agency perception. Regarding this focus, Section 1 introduced studies showing the relation between symbol grounding and multimodal communication (Murgiano et al., 2021).

In the context of HAI studies, multimodal interaction has also been extensively examined. Among these, human gestures have been treated as a main modality that significantly influences verbal communication (Maricchiolo et al., 2020). To approach this, data-driven methods that learn from human gestures by using machine

learning techniques such as deep learning have become popular. For example, Saund et al. (2022) analyzed the relationship between body movement and meaning to generate effective gestures by virtual agent.

There have also been many studies on gesture generation from multimodal language corpus. Lin and Amer (2018) use Generative Adversarial Networks (GAN) to control joints by mapping embedded words to the space of body movements. Ahuja and Morency (2019) also proposed a method called “language2pose” that integrates language and body movements through end-to-end learning. More recently, Tevet et al. (2022) used a diffusion model to generate human body movements and use sentences and actions as input. Their study confirmed that gestures generated from both inputs were evaluated better than gestures generated by other generation models. Furthermore, possibility of natural gesture generation or selection is explored by using LLM (Brown et al., 2020; Chowdhery et al., 2023; OpenAI, 2023), which has become popular in recent years. Hensel et al. (2023) have shown that LLM can be used to select hand gestures that are compatible with the content of speech. Yoshida et al. (2023) have successfully generated emblematic gestures by incorporating LLMs into humanoid robot motion generation.

In the context of co-speech gesture generation, Yoon et al. (2019) applied deep learning technology to generate various gestures, including iconic, metaphoric, deictic, and beat gestures. Ishii et al. (2018) also proposed a model of co-speech gesture focusing on appropriate timing. In their study, Conditional Random Fields (CRFs) were used to parse information from natural language.

2.3 Top-down gesture from abstract index

As described thus far, numerous studies have focused on bottom-up approaches that extensively learn low-level features from human motion data. Although these approaches have proven effective for generating natural gestures, using the bottom-up approach to identify intrinsic yet infrequently occurring features related to spoken words remains challenging due to the inherent bias of deep learning technologies towards the majority of data samples. Therefore, a top-down approach that targets specific aspects of word meanings is necessary to achieve a robot that acts as an agent “grounded to the external world,” as described in Section 1.

Based on these ideas, Sasaki et al. (2023) proposed a gesture generation method by extracting intrinsic images of words as shown in Figure 1. The details will be explained in the next section, but their method has advantages in assuming an abstract axis (index) to be extracted from the data, by following Grand et al. (2022). We consider that such an intentional setting of an axis is essential to represent communicative intention in the agent (Grice, 1989). However, their study did not present sufficient evaluations of the variety of body expressions. Therefore, this study extends the previous method (Sasaki et al., 2023) to evaluate the effects of physical images generated from the abstract index on agency perception. In the following sections, we describe the method of constructing “size index” and “size images,” which are the process presented in the left side of Figure 1.

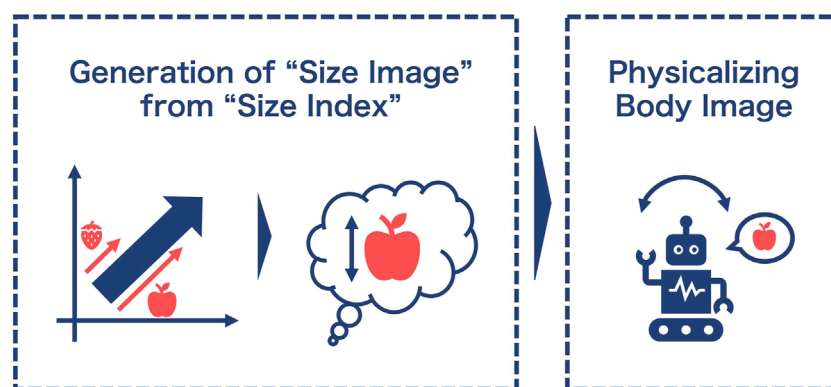


FIGURE 1
Gesture generation proposed by Sasaki et al. (2023).

3 Generating “size images” of words

In this study, we generate gestures of embodied agents (robots) by using the method proposed by Sasaki et al. (2023). After presenting an overview of the method, we apply it to survey data to extract the “size images” of words.

3.1 Basic method

Figure 2 illustrates the methodology employed in this study to generate “size images.” Figure 2A details the procedure for extracting a “size index” applicable to any word in the word-distributed representation. This involves identifying quantitative dimensions related to the attribute of size. Following this, Figure 2B outlines the process of creating a “size image” for a specific word using the derived size index. The output of this step is utilized to translate the abstract semantic feature into a physical representation (robot movement) that can be recognized within the context of human-agent interaction, thereby facilitating the symbol grounding of the word based on its size attribute.

These processes adapt and modify the approach presented by Grand et al. (2022), which was introduced in Section 1. To overcome the limitations of Grand et al.’s method, which we will discuss later, our method (Sasaki et al., 2023) employs an approach that combines word-distributed representations with a human-curated thesaurus, introduced in the beginning of Section 2.1. This approach reduces the arbitrariness of the method and enhances its applicability to languages with comprehensive linguistic resources. In this study, to assess the method’s applicability beyond English, we utilize a linguistic resource developed for the Japanese language.

The remainder of this subsection details the specific procedure employed in this study for extracting the “size index” and constructing the “size image.”

3.1.1 Composition of “size index”

We first present Grand et al. (2022)’s method of extracting the “size index” (the blue arrow in Figure 2A). In this method, an axis with a meaning specific to “large,” is constructed by subtracting a polar vector with a meaning of “small” (Small in Figure 2A, the

antonym of “large”) from a different polar vector with a meaning of “large” (Large in Figure 2A). To define the poles, we only need to extract the coordinates of the “large” and “small” values in the distributed multidimensional vector space. However, in addition to their size-related meanings, these two words have extra meanings that derive from their adjectival roles in the sentence¹. To exclude such meanings unrelated to the degree of “size,” Grand et al. (2022) defined a set of synonyms (red dotted square in Figure 2A) that have the same role as “large” and “small” in the distribution word representation. Then, the polar coordinates are determined by computing the mean vector of these synonyms, respectively.

Thus, the “size index” I is defined by the following equation:

$$I = \frac{\sum_{i=1}^n l_i}{n} - \frac{\sum_{j=1}^m s_j}{m}$$

where l_i and s_j are word belonging to the set of “large” synonym vectors ($Synset_L = \{l_1, l_2, \dots, l_n\}$) and the set of “small” synonym vectors ($Synset_S = \{s_1, s_2, \dots, s_m\}$).

3.1.2 Composition of “size image”

Once the “size index” is defined, we can calculate “size image” from the index. Figure 2B shows the calculation of the “size images” as the cosine similarity between the “size index” and the input word vector. Thus, the size image S is calculated by

$$S = \frac{I \cdot w}{\|I\| \|w\|}$$

where w represents the input word vector. In this method, the larger this value is, the larger the word is assumed.

3.1.3 Selection of synset

The limitation of Grand et al.’s method is arbitrariness in selecting $Synset_L$ and $Synset_S$. In their study, $Synset_S$ consisted of

1 The distance between the words “large” and “small” is quite close within the entire space of word-distributed representation because both are commonly used in contexts describing the size of an object. Therefore, poles defined solely by these two words have limited distinguishability for other words in terms of size.

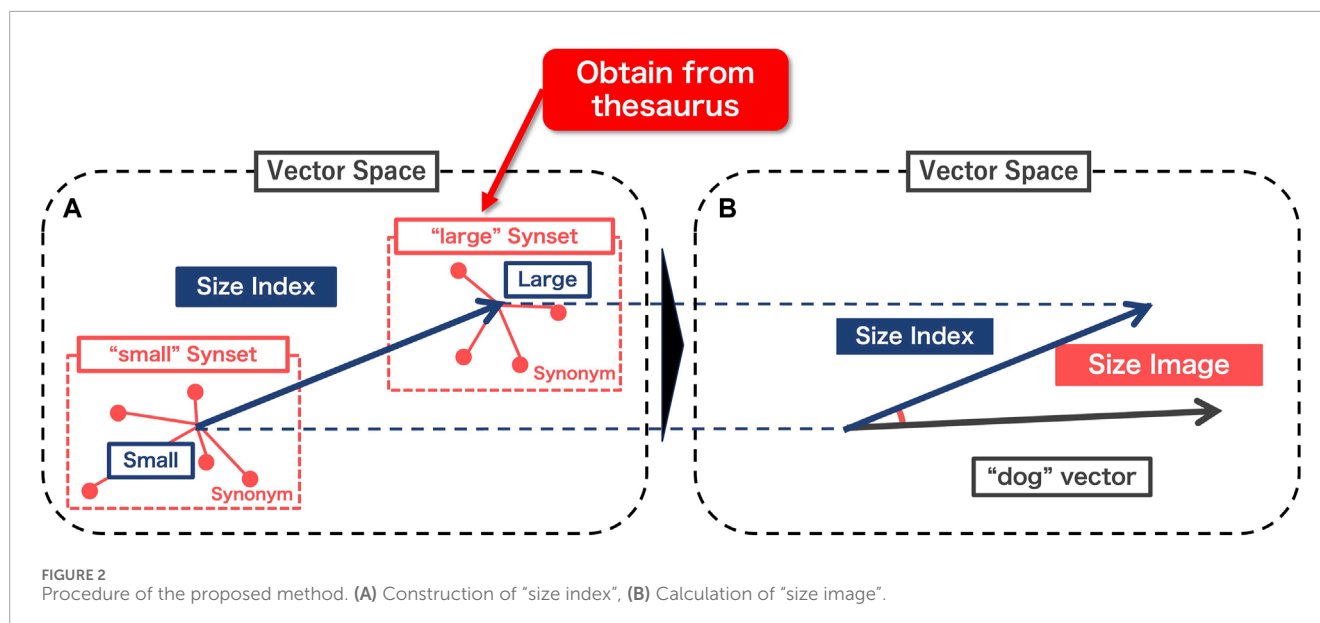


FIGURE 2
Procedure of the proposed method. (A) Construction of "size index", (B) Calculation of "size image".

"tiny" and "little" while $Synset_L$ was constructed by "big" and "huge." Nevertheless, they didn't specify the criteria for selecting these synonyms.

To address this issue, and to compose a "size index" properly matching human perception, it is important to set up a set of synonyms for the polar words ("large" and "small") without any arbitrariness. A possible method is leveraging a standardized thesaurus. However, a thesaurus does not automatically determine the appropriate synonym set. Words are usually polysemous and have multiple meanings. In a thesaurus, a set of synonyms for a word is defined as a synset for each meaning. To improve the consistency with human perception, it is necessary to select appropriate synsets. In this study, we seek the combination of synsets that maximizes the distance between "large" and "small" words obtained from human reports to determine the polar coordinates consistent with human perception. In this procedure, we first prepare $Synset_L$ associated with the word "large" and $Synset_S$ associated with the word "small".

We also prepare a set of words $Large_h$ that humans perceive as "large" and a set of words $Small_h$ that humans perceive as "small." In extracting $Large_h$ and $Small_h$, it is necessary to distinguish categories to which the word refers. According to Tversky (2019) and others, the meaning of a concept is originally composed of human movement. However, as shown by Utsumi (2020), physical quantities are not expected to be strongly embedded in word-distributed representations composed of socially published documents. Either way, the scale of the "size index" has the possibility to be changed by the categories the word belongs to. Following such discussions, this study assumes.

- $Large_{h, animate}$
- $Large_{h, inanimate}$
- $Large_{h, intangible}$

as subclasses of $Large_h$, and.

- $Small_{h, animate}$

- $Small_{h, inanimate}$
- $Small_{h, intangible}$

as subclasses of $Small_h$.

In the selection process of the synset combinations, the "size index" I_{ij} is composed for $Synset_{S,i}$ ($\in AllSynset_S$) and $Synset_{L,j}$ ($\in AllSynset_L$). The "size images" of the human-perceived small word $w_{S,k}$ ($\in Small_h$) and large word $w_{L,l}$ ($\in Large_h$) are selected to calculate the "size images" $S(I_{ij}, w_{S,k})$ and $S(I_{ij}, w_{L,l})$, respectively. Those individual "size images" are aggregated into average images as $S(I_{ij}, w_L)$ and $S(I_{ij}, w_S)$. From those indices, the "size index" of the combination of $Synset_L$ and $Synset_S$ that maximizes the difference ($S(I_{ij}, w_L) - S(I_{ij}, w_S)$) are selected as the optimal index to compose human compatible "size images".

3.2 Application of the method

The construction of the "size index" described so far requires a vector representation of words (w), a thesaurus ($AllSynset_S$, $AllSynset_L$), and human perceived small/large words ($Small_h$, $Large_h$).

Of these, this study uses the Japanese Wikipedia entity vector developed by (Suzuki et al., 2016), which we call JWikitEntVec in this paper, as a distributed representation model to construct a vector of words (w). This is a trained model built by word2vec (Mikolov et al., 2013). We used this because the model is well used in Japanese academic societies². Also the development method and the data set used to construct the model are clearly presented by the authors of this model. These characteristics make it particularly advantageous for

² We can find more than sixty citations in Google Scholar at the point of submission.

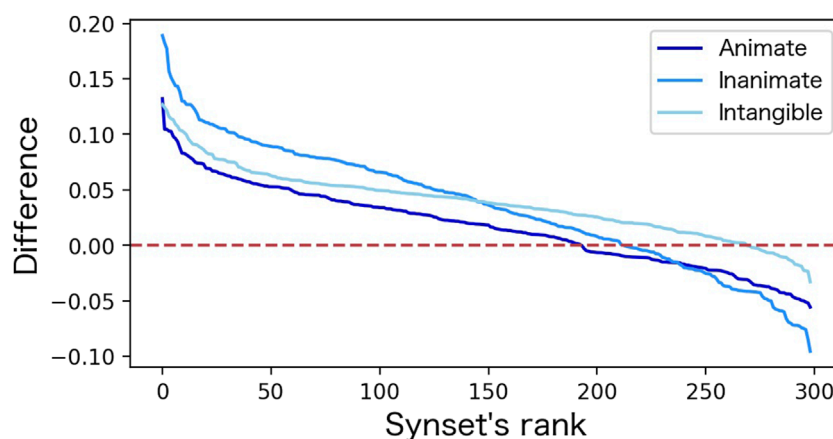


FIGURE 3
Distribution of “size image” difference.

foundational research like this study, despite the model’s performance not being as high as that of more advanced LLMs.

Furthermore, we employed the Japanese WordNet (Bond et al., 2009) for the selection of synonyms. This thesaurus contains 28 synsets for “large” and 14 synsets for “small.” Words not included in JWikiEntVec and synsets with no synonyms were excluded from the later analysis. As a result, we obtained 23 synsets for “large” and 13 synsets for “small.” The “size index” was calculated for the combinations of these synsets ($23 \times 13 = 299$).

The human word sets ($Small_h$ and $Large_h$) were collected through a questionnaire survey whose participants ($n = 100$) were recruited from a Japanese crowdsourcing site (Lancers). The participants were asked to write down five “large” and “small” words for animate, inanimate, and intangible concepts (30 words in total). Table 1 shows the top five words and their frequency for each question. From the table, we find the word “Mind” appears in the top five words for both large and small intangible. This duplication is considered to indicate the ambiguous nature of the meaning of this word. Therefore, this study used “Mind” in both $Small_h$ and $Large_h$ and calculated the “size index” by using these 29 words as the elements of $Small_h$ and $Large_h$.

Figure 3 shows the differences ($S(I_{ij}, w_L) - S(I_{ij}, w_S)$) calculated for the 299 combination of synsets. The horizontal axis of this figure corresponds to the combination of synsets ordered by rank. Overall, there are many combinations where the difference of the “size image” is larger than 0 (above the red dotted line), indicating that the size index calculated for more than half of the synset combinations is consistent with the human image.

Table 2 shows the highest-ranked synset combinations (with the largest difference), the lowest-ranked synset combinations (with the smallest difference), and the synonyms in each synset. The combination of the synset with the largest difference is “larger-than-life” and “peanut,” and the combination of the synset with the smallest difference is “major” and “small-scale.”

4 Experiment 1: “size image” in effort

The following two sections present an evaluation of the physical images (iconic gestures) generated from the “size images” composed in the previous section.

To address the first research question presented in Section 1, we tested the above procedure of selecting synsets. Thus, the gesture generation using the top-ranked synset combination and that using the bottom-ranked synset combination are treated as the *proposed* and *controlled* methods, respectively. If the procedure described in the previous section successfully extracted the axis that grounds the symbol to the physical world, and if the correspondence between the axis and the body is compatible with what people own, we can claim that the proposed method is effective to extract “size image” that evokes agency perception from the vector space of word-distributed representation.

Regarding the second question, we can consider several possible methods for mapping the “size image” to “physical image.” Dance theories generally pursue a physical expression that effectively externalizes the human internal states. Among these theories, Laban movement analysis (Laban and Ullmann, 1971) has been widely used in the field of HAI (Ishino et al., 2018). This theory assumes two modalities in the correspondence between human internal states and body: *shape* for posture and *effort* for movement. In Experiment 1, we mapped the axis of space, which is one of the axes of the effort modality to the “size index.” In other words, the amount of movement is considered to be larger when a large concept is recalled. In the following section, we explain the construction of body movements based on this idea.

4.1 Methods

4.1.1 Materials

The procedure of generating physical image (iconic gesture) according to the “size image” is shown below.

- 1 Setting large and small movements. Mapping the “size image” to the posture composed of the body parts. For

TABLE 1 Responses obtained for each question (Top five words).

Animate			
Large		Small	
Word	Freq	Word	Freq
Elephant (zo-u)	85	Ant (a-ri)	74
Whale (ku-ji-ra)	68	Daphnia (mi-ji-n-ko)	33
Giraffe (ki-ri-n)	60	Mosquito (ka)	31
Bear (ku-ma)	25	Tick (da-ni)	30
Hippopotamus (ka-ba)	19	Fleas (no-mi)	23
Inanimate			
Large		Small	
Word	Freq	Word	Freq
Tokyo Sky Tree (to-kyo-su-ka-i-tu-ri)	45	Sand (su-na)	26
Mt. Fuji (fu-ji-sa-n)	35	Beads (bi-zu)	20
Tokyo Tower (to-kyo-ta-wa)	29	Needle (ha-ri)	26
Everest (e-be-re-su-to)	22	Microchip (ma-i-ku-ro-chi-ppu)	15
Pyramid (pi-ra-mi-ddo)	18	Screw (ne-ji)	14
Intangible			
Large		Small	
Word	Freq	Word	Freq
Space (u-tyu-u)	36	Mind (ko-ko-ro)	13
Love (a-i)	18	Jealousy (si-tto)	9
Dream (yu-me)	17	Envy (ne-ta-mi)	7
Mind (ko-ko-ro)	16	Vanity (mi-e)	6
Sea (u-mi)	15	Point (te-n)	5

* Words are translated from Japanese. Japanese pronunciation in hepburn romanization is presented in parentheses.

this purpose, we define the body posture corresponding to the smallest and largest words recognized by humans. Using this posture as a reference (0 for the image of the smallest word and 1 for the image of the largest word), the “size image” of each word is positioned in the range from 0 to 1.

2 Calculation of parameters at each joint. The above scaling is applied to the angles of each joint that constitute the posture.

3 Generation of physical image. A gesture is generated based on the values obtained by step 2. This generation is assumed to be made simultaneously with the utterance of the word.

In order to embody the above steps, we used Sota, a small communication robot by Vstone³. Sota’s body movements are controlled by nine joints (one torso, three necks, two shoulders, and two arms joints). By controlling the angle and speed of these joints, Sota can generate a variety of movements. In addition, Sota has a speech function and can speak any word while simultaneously displaying gestures.

In this study, the parameters of Sota’s arm and shoulder joints were instantiated by “size image” of each word. Sota’s default

3 <https://www.vstone.co.jp/english/index.html>

TABLE 2 Top and bottom synsets combinations.

word	Top synset		Bottom synset	
	“large”	“small”	“large”	“small”
synset	larger-than-life	peanut	major	small-scale
meaning	very impressive	unimportant	effective	very small
Synonym 1	magnificent (so-da-i)	insignificant (bi-bi-ta-ru)	great (o-o-ki-na)	modest (sa-sa-ya-ka)
Synonym 2	large scale (da-i-ki-bo)	only (wa-zu-ka)	significant (o-o-ha-ba)	cottage (re-i-sa-i)
Synonym 3		trivial (sa-sa-i)	serious (ju-da-i)	tiny (ti-ttya-i)
Synonym 4		cheap (ya-su-ppo-i)		
Synonym 5		slight (ke-i-bi)		

* Words and meanings are translated from Japanese. Japanese pronunciation in hepburn romanization is presented in parentheses.

TABLE 3 Maximum and minimum values of parameters for “size image” and each joint in Experiment 1.

	Maximum	Minimum	Default
Size Image (proposed)	0.39	−0.17	-
Size Image (control)	0.28	−0.12	-
Shoulder angle	75	−68	−70
Arm angle	−20	88	90

TABLE 4 Maximum and minimum values of parameters for “size image” and each joint in Experiment 2.

	Maximum	Minimum	Default
Size Image(top synset)	0.39	−0.17	-
Size Image(bottom synset)	0.28	−0.12	-
Shoulder angle	30	−30	−70
Arm angle	−20	90	20
Neck angle	10	−10	0

posture in this study is shown in the upper image in Figure 4, with its shoulders down and arms slightly bent. From this state, the parameters of the arm and shoulder joints are changed to generate a gesture corresponding to the size of the word. Table 3 shows the maximum and minimum values of the “size image”, as well as the parameters of the shoulder and arm joints and the default position that corresponds to them. The “size image” computed for both the proposed and controlled methods is mapped to these values: 0 for the minimum and 1 for the maximum in the proposed method, with the reverse applied in the control method.

4.1.2 Design and measures

In the experiment, the physical images generated by the above procedure were recorded as movies. Supplementary Material included examples of the movies (each for about 4 s duration⁴) and all the pictures showing poses representing each word, captured from the end of each movie. In this experiment, 30 words (with one duplication) in Table 1 were used to generate physical images for both the control and proposed methods⁵. Among them, two examples are shown in the lower images of Figure 4. The lower left and right images depict gestures for the words “tick (da-ni)” (the smallest animate concept) and “pyramid (pi-ra-mi-ddo)” (the largest inanimate concept), respectively. Thus, a small “size image” results in the robot making small movements from the default position⁶, while a large “size image” results in larger movements.

The participants were asked to observe those movies and rate the naturalness of the correspondence between the robot’s movements and the words it speaks on a 5-point scale (1: not at all natural—5:

4 This duration corresponds the time to change the pose from the default to the final position. Aside from this time, the proposed method also requires time to load the model (JWikiEntVec) and compute the size image. The average of the initial loading time is 6.28 s ($n = 10$), which is required only once to initialize the entire system. After the initial setup, the method only requires 0.59 s, on average ($n = 10$), to compute the size images. Those computational times were recorded by a 2020 MacBook Pro with an Apple M1 CPU.

5 These words were assumed to be samples of $Large_n$ and $Small_n$ for each category. There might be several confounding factors, among which we examined iconicity rating as one of the factors possibly influencing gesture perception. However, from the publicly available dataset of iconicity ratings (Thompson et al., 2020; Winter et al., 2023), we could not find clear evidence of the influence on gesture perception in our experiment. Supplementary Material show the details of this analysis.

6 The movement of the shoulder and arm at the minimum value is generated by taking the parameter of the smallest unit of movement of the internal motor.

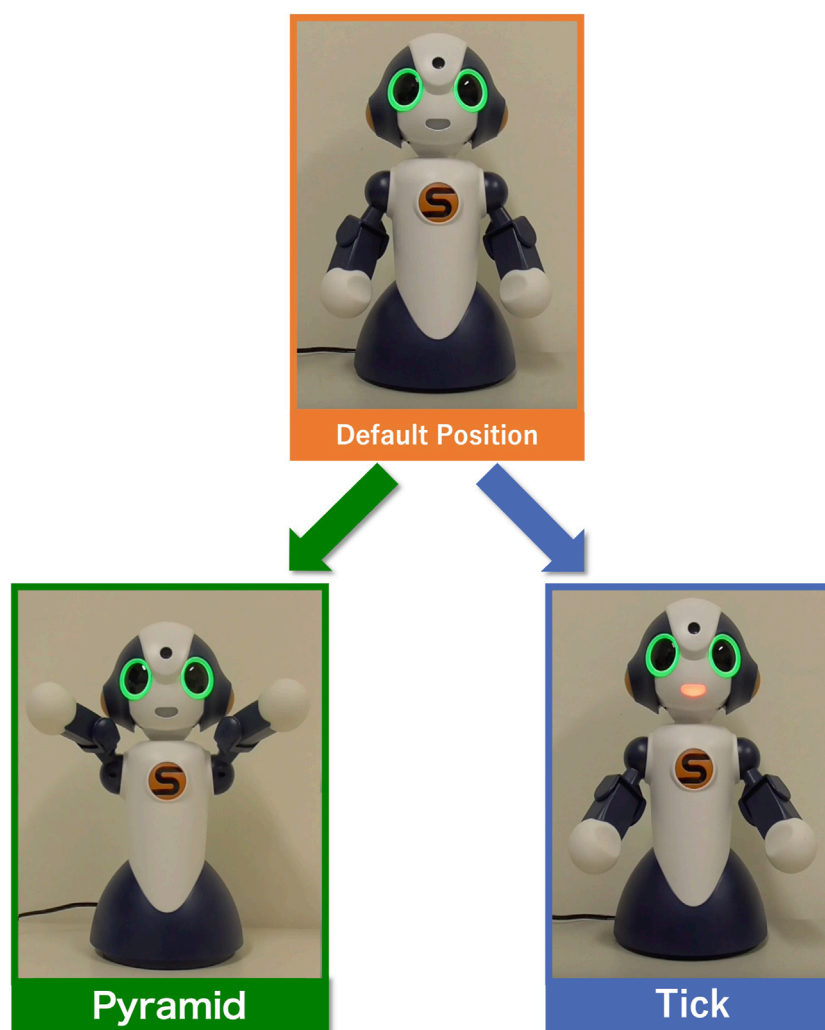


FIGURE 4
Examples of Sota gestures (left: "Pyramid", right: "tick").

very natural). The standard of "naturalness" here assumes a natural communication between humans (Kopp and Krämer, 2021). In human communication, people usually try to achieve, intention sharing (i.e., mutual understanding) (Tomasello, 2010). Therefore, we specifically asked the participants to rate whether or not they feel that the robot understands the meaning of the words as humans would. As the question indicates, this rating demands the participants to perceive the robot's internal state from the short movie. By presenting such a question, evaluating the agency perception in terms of communicative intention (Grice, 1989) would be possible.

4.1.3 Participants and procedure

The 300 participants recruited from Lancers joined the experiment after reading the instructions provided on the request screen (reward: 110 JPY). The instructions explained the evaluation procedure, the definition of naturalness, and the obligation to answer dummy questions. After agreeing to the above instructions, participants were presented 14 movies, which were randomly

selected for each participant from 58 movies (2 conditions $\times$ 29 words in Table 1). In between the evaluation of the movies, a dummy question in which the participants were asked to answer a specified number was inserted.

4.2 Results

Nine participants who answered the dummy questions incorrectly were excluded from the following analysis. From the remaining 291 participants' responses, the average of the ratings was calculated for the 29 words in each condition. Utilizing this value as a unit of analysis, we calculated the average of 12 naturalness ratings (condition $\times$ size $\times$ category)⁷, as shown in Figure 5. By using these values, we try to test the following hypotheses:

⁷ The average rating for "Mind" was included both for large intangible and small intangible conditions.

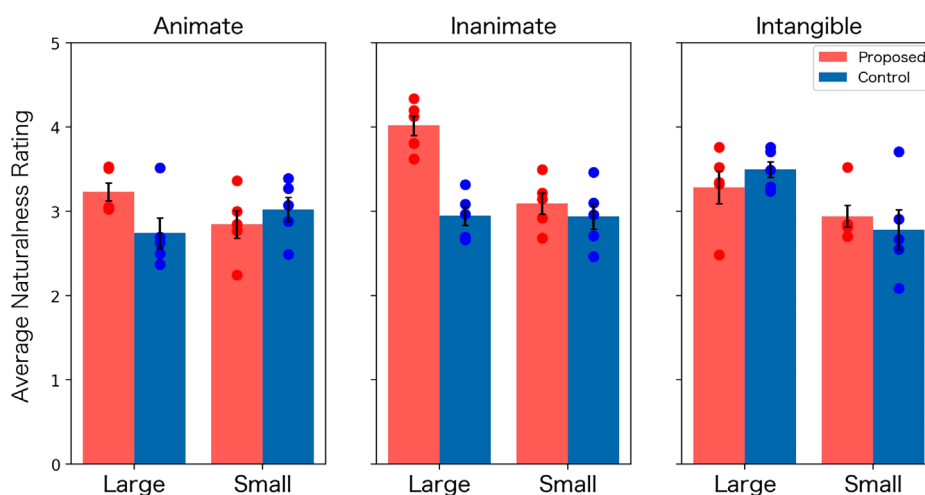


FIGURE 5
Mean rating of naturalness in Experiment 1 (Error bars: standard errors, Dots: ratings for naturalness for each word).

- 1 The proposed method has a greater effect on the ratings than the control method.
- 2 The above effect is affected by the difference between the word categories.

The first hypothesis directly relates to the first research question introduced in Section 1. The second hypothesis was explored in response to previous studies (Utsumi, 2020), which have demonstrated that the impact of physical experience on word meaning diminishes in abstract concepts. Consequently, it is advantageous for future research to elucidate the extent to which symbol grounding influences the perception of agency.

We conducted a three-way [condition (proposed vs. control) $\times$ size (large vs. small) $\times$ category (animate vs. inanimate vs. intangible)] analysis of variance (ANOVA) to test the above hypothesis. Among the effects obtained from the ANOVA, we focused on the main effect of the condition and the interactions involving the condition and the category (a second-order interaction between the condition, the size and the categories, and a first-order interaction between the condition and the category). During this process, the significance level was set to 0.10, reflecting small sample size in this study ($n = 5$ for each condition). We also control type-1 error by reporting corrected p -values as q -value calculated using the Benjamini-Hochberg (B-H) method.

From the analysis, we obtained a significant main effect of the condition ($F(1, 48) = 6.48$, $p = 0.01$, $q = 0.04$, $\eta^2 = 0.07$), confirming that the proposed condition was evaluated more naturally than the control condition. However, we also found a significant second-order interaction between the condition, the size, and the categories ($F(2, 48) = 4.09$, $p = 0.02$, $q = 0.05$, $\eta^2 = 0.14$) and a significant interaction between the condition and the categories ($F(2, 48) = 3.78$, $p = 0.02$, $q = 0.05$, $\eta^2 = 0.08$), suggesting that the difference in naturalness between conditions depends on other factors.

To examine the details of the interaction effect, we conducted *post hoc* two-way [condition (proposed vs. control) $\times$ size (large vs. small)] ANOVAs for each category. The p -values were corrected

using the BH-method, accounting for nine tests in total (one interaction and two main effects for three ANOVAs). Significant effects ($p < .05$) related to the condition were observed only in the inanimate category (the main effects of condition: $F(1, 16) = 18.72$, $p < .01$, $q < .01$, $\eta^2 = 0.53$ and the interaction: $F(1, 16) = 10.50$, $p < .01$, $q = 0.02$, $\eta^2 = 0.66$). The simple main effect of the condition in the inanimate category was observed for the large concept ($F(1, 16) = 21.29$, $p = 0.01$, $q < .01$, $\eta^2 = 1.78$). These results suggest that the proposed condition was evaluated as significantly more natural than the control condition for the large inanimate concept.

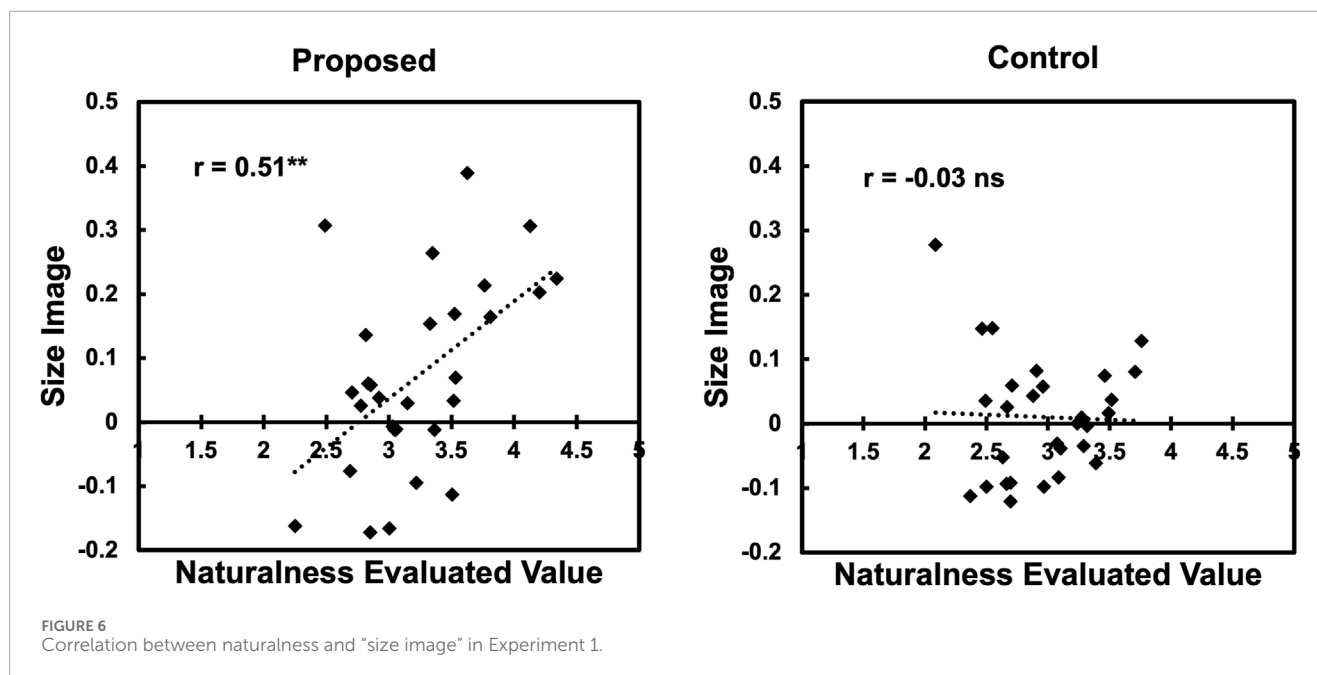
The above results partially align with the previous study (Utsumi, 2020). Additionally, the result revealed an effect of the “size” (small vs. large) on agency perception, which was not expected. To investigate this effect further, a *post hoc* correlation analysis was conducted. Figure 6 shows the scatter plots of the naturalness ratings and the “size image” for the proposed and controlled conditions. From the figure, we found a moderate positive correlation ($r = 0.51$, $p < 0.01$) in the proposed condition. This result indicates that the proposed method generates more natural gestures for larger-size words.

4.3 Discussion

The above analysis indicates that the proposed condition outperformed the control condition in terms of overall naturalness. Therefore, the results of this study suggest that the proposed method can generate physical images that enhance agency perception.

However, this effect was affected by the word category and the size of the gesture. When dividing the overall effect into the three categories and two sizes, the proposed method outperformed the control method only significantly for the large inanimate category.

Those results partially support the hypotheses presented Section 4.2. The overall main effect of the condition supports the first hypothesis. The interaction between the condition and the categories supports the second hypothesis, suggesting the weak effect of the symbol grounding in an abstract category.



There are several possible reasons why the expected effect was not strongly observed in this experiment. Although we cannot deny the possibility that our assumptions (i.e., artifacts become agents through symbol grounding) are incorrect, more robust results may be obtained by improving experimental settings. For instance, the data obtained in this experiment came from a crowdsourcing survey, which may have introduced noise into the participants' ratings. We can also consider that mapping the "size image" to the physical image was inadequate. The scatter plots in Figure 6 suggest the latter possibility, indicating that our method does not exhibit sufficiently natural behavior for small-size words. The next experiment explores this possibility.

5 Experiment 2: "size image" in shape

This experiment also addresses the first research question by assessing the naturalness of gestures generated from the "size index." However, to explore the second research question, we employed a different mapping of the "size index" to the body. The method adopted here focuses on the shape modality in Laban theory. From the correlations in Figure 6, it can be speculated that the small movements in Experiment 1 did not appear to be performed as a gesture. Based on this speculation, this experiment examines the research questions by mapping size images to body size so that perceptible gestures are generated even when small-sized words are uttered.

5.1 Methods

The experiment method was the same as Experiment 1 except for the movies presented to participants. In the movies in this experiment, we arranged the correspondence between the "size

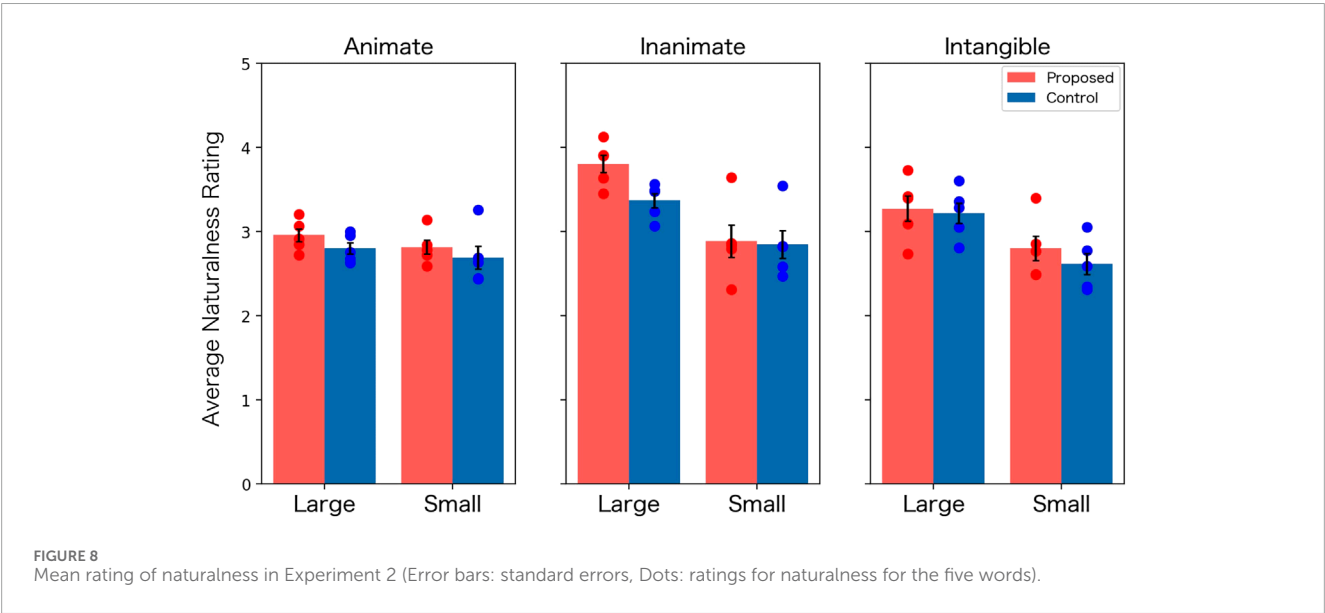
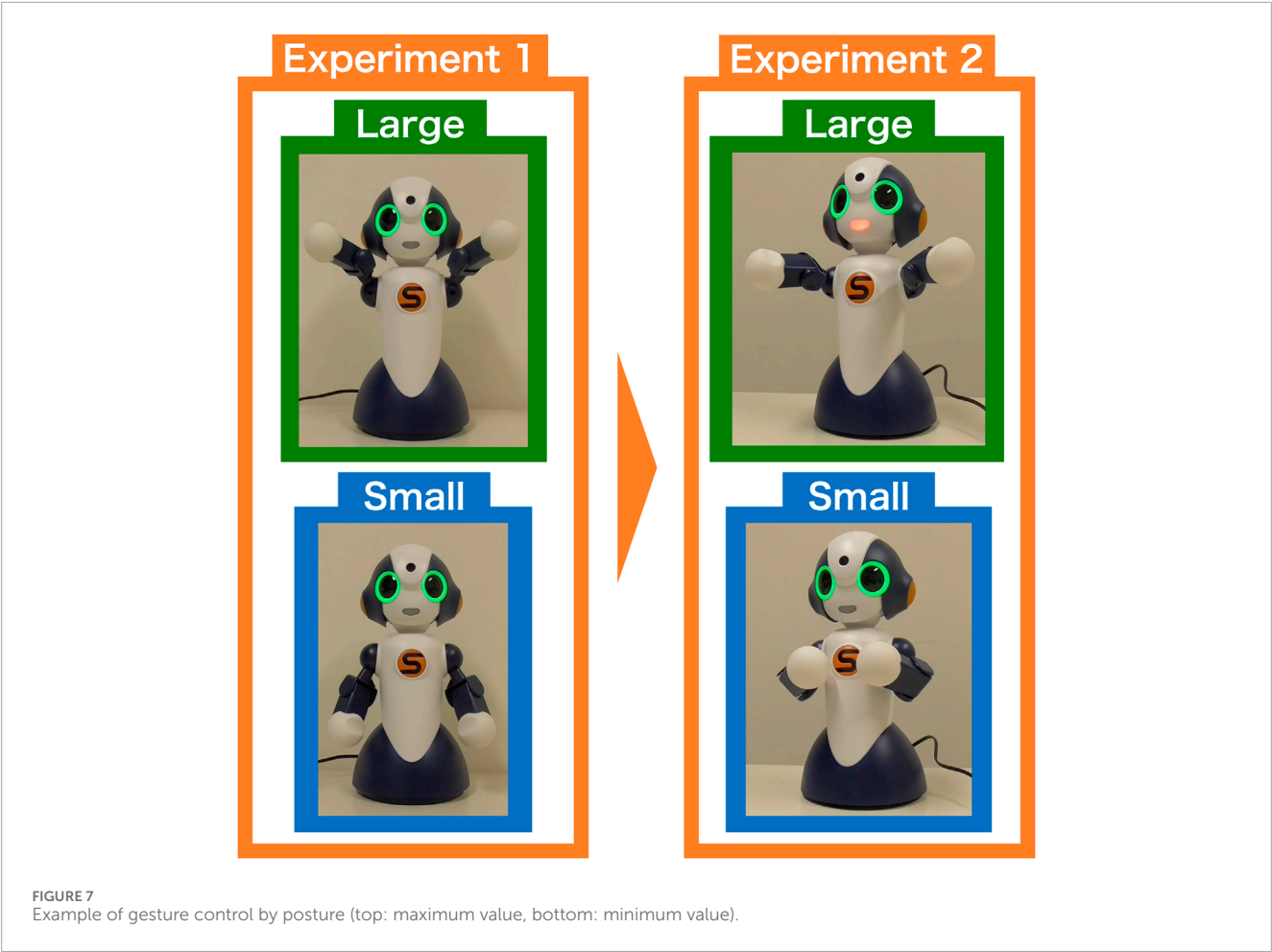
image" and the physical image to express the size of the posture. The default posture of the robot is the same as in Experiment 1. In addition to the body parameters (arms and shoulders) used in the previous experiment, the neck joints were controlled according to the "size image" of the word. Table 4 shows the minimum and maximum values of the word "size image" and the corresponding values of Sota's joint angle parameters for the proposed and controlled methods. All the pictures showing the finishing pose used in this experiment are presented in the Supplementary Material. Examples from them are shown in Figure 7.

As illustrated in 4.1.1, the upper left and lower left images are gestures for uttering the word with the largest (pyramid) and smallest (tick) "size image" in Experiment 1. In this experiment, we changed these gestures as shown in the upper right and lower right images of Figure 7. When the "size image" was largest, the parameters of the arm and shoulder were set at the position where the distance between the arm and the arm was the largest, and the neck was also set upward. When the "size image" was the smallest, the arm and shoulder parameters were set at the position where the arm-to-arm distance was the smallest, and the neck position was also set downwards. As can be seen in Figure 7, the camera position was also changed to make the differences in posture obvious.

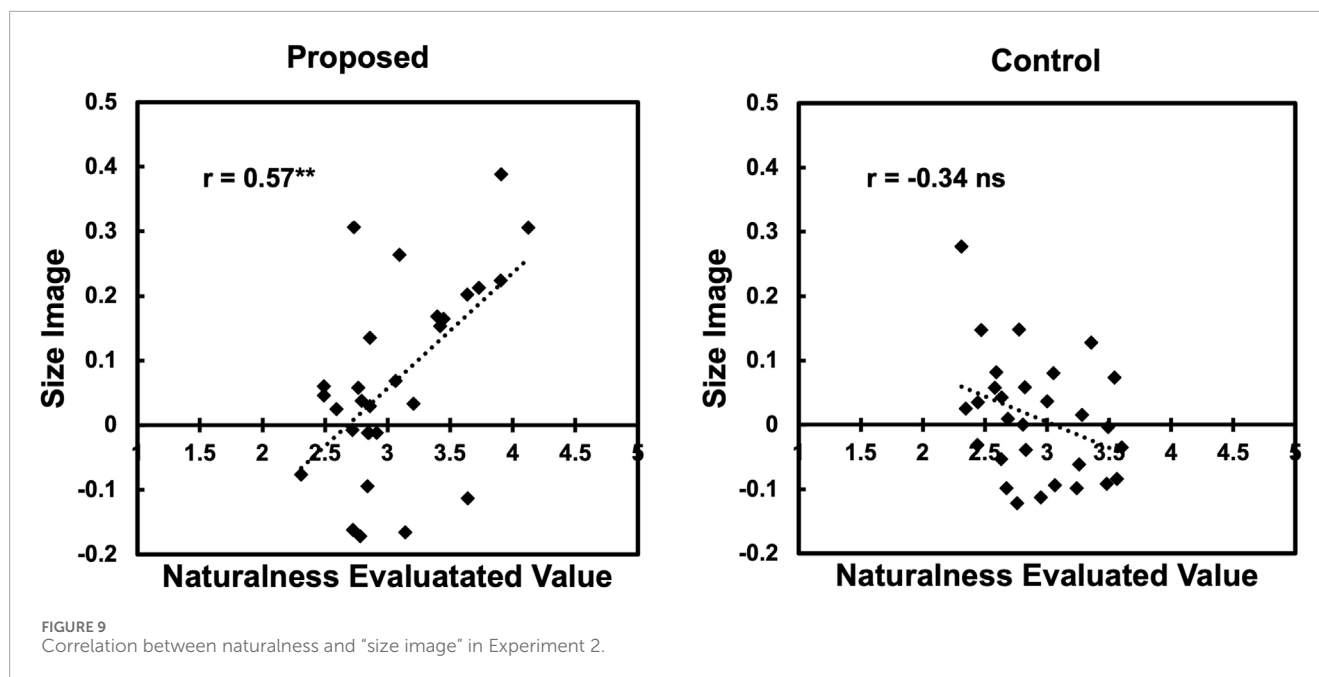
5.2 Results

As in Experiment 1, 300 participants were recruited from Lancers joined the experiment. 43 participants who incorrectly responded to the dummy questions were excluded from the analysis. From the remaining 257 participants' responses, the average ratings for 29 words in two conditions were calculated. Using the calculated value as the unit of analysis, the naturalness ratings were summarized in Figure 8.

From the data shown in the figure, we tested the same hypotheses in Experiment 1. We report the results of the statistical



tests of the main effect of the condition, and the second-order interaction between the condition, size and categories, and the first-order interaction between the condition and the categories from the three-way [condition (proposed vs. controlled) $\times$ size (large vs. small) $\times$ category (animate vs. inanimate vs. intangible)] ANOVA performed for Figure 6. As a result, we obtained a significant main effect of the condition ($F(1,48) = 4.14, p = 0.04, q = 0.08, \eta^2 = 0.09$) and no significant interactions involving the



condition (the second-order interaction: $F(1,48) = 0.91$, $p = 0.41$, $q = 0.57$, $\eta^2 = 0.04$, the interaction between condition and category: $F(2,48) = 0.20$, $p = 0.81$, $q = 0.81$, $\eta^2 = 0.22$). These results indicate that the difference in naturalness between conditions is not affected by category or size.

As in Experiment 1, we also conducted a correlation analysis. Figure 9 shows the scatter plots of the naturalness scores and the "size image" for the proposed condition. From the figure, we found a moderately positive correlation ($r = 0.57$, $p < 0.01$) in the proposed condition as in Experiment 1. This result indicates that the larger the value of "size image," the more natural the image tends to be evaluated.

5.3 Discussion

In Experiment 2, we reexamined the research questions using a different mapping of the "size image" to the physical image than in Experiment 1. The results showed that, as in Experiment 1, the proposed method produced more natural images than the control method. However, contrary to Experiment 1, Experiment 2 found no interaction between the condition and the other factors. Therefore, we can assume that the gestures generated in Experiment 2 can be applied to more general situations to cause an agency perception.

However, the effect size of the condition obtained in Experiment 2 was smaller than that in Experiment 1. In addition, as in Experiment 1, we observed a correlation between the "size image" and naturalness ratings. This suggests that the small-sized words in Experiment 2 also exhibited unnatural gestures.

6 General discussion

This study was guided by two research questions. Concerning the first question (how can "size images" evoking an agency

perception be extracted from the vector space of word-distributed representations?), we assessed a method proposed in our previous study (Sasaki et al., 2023) as one potential answer. The observed difference between the proposed and the control conditions indicates the necessity of selecting appropriate synonyms for constructing the "size index." The current study has demonstrated the advantage of our modification (Sasaki et al., 2023) from the previous method (Grand et al., 2022).

Furthermore, for the second question (what forms of gesture expression are effective in constructing natural interaction based on the agency perception?), by comparing the results of the two experiments, we can assume that expression using the mapping in the posture has a more general effect. However, in common with both experiments, there was a positive correlation between the naturalness rating and the "size image."

The fact that the gestures corresponding to smaller "size images" did not receive good ratings requires further examination. In the discussion of Experiment 1, it was considered that the small amount of movement was a factor causing the low naturalness ratings. However, neither the small "size image" nor a large amount of movement in Experiment 2 improved the naturalness rating for small-sized words. These results may suggest an asymmetry between the small and large poles of the "size index"; Words at the smallest pole may be less associated with the body, while words at the largest pole may be more associated with the body.

We also need to consider the influence of categories on the effectiveness of the proposed method. Previous studies have noted that abstract concepts are less grounded in the physical experience (Lakoff and Johnson, 1980; Utsumi, 2020). Consistent with this discussion, in Experiment 1, only one of the concrete categories demonstrated the effectiveness of the proposed condition. Additionally, the observation that the same word "Mind" appeared in both "large" and "small" abstract concepts indicates the limited extent of symbol grounding.

Our experiment also suggests that using a single size index across various categories has limitations. In this study, we applied the same “size index,” derived from the pairs of synsets listed in Table 2, to all three categories. We did not differentiate categories when calculating the size index because of enhancing the index’s applicability. Considering that the method might be applied to any arbitrary words, it appears difficult to determine the abstractness of a word beforehand. However, our results, particularly the observed differences in the effects of the condition between animate and inanimate categories, clearly indicate the necessity of adjusting the scale of the mapping from “size image” to “physical image.”

7 Conclusion

In this study, we started from the hypothesis that symbol grounding is important in generating the agency perception. In line with this hypothesis, we composed a “size image” of a symbol grounded in a quantitative vector space of word-distributed representations. We also explored the hypothesis by examining two mapping of “size image” to body images. The experiments verified the proposed method although the effect size was not large.

We consider that the reason for the small effect size is partially attributed to our approach. Unlike recent research on gesture generation based on deep learning technologies, this research has many assumptions. We especially composed an abstract axis that mediates speech and body. Although these top-down approaches do not reach bottom-up approaches in terms of performance, it is useful to guide the novel interaction design with agents. Thus, we believe our study contributes to theoretical and practical developments in HAI research.

There are several other limitations to this study. The first concerns the gestures with small-sized words as already noted. Even though the problem noted in Section 6 may exist, it is beneficial to invent expressions of small-sized gestures that humans can evaluate as natural. To overcome this problem, we need to improve expression ability in the used body image. The currently used robot (Sota) has limitations in performing detailed gestures. Therefore, to confirm our hypothesis, it may be necessary to use other robots or virtual agents.

The robustness of the results also needs to be improved. The data collection in this study was conducted using crowdsourcing, and a lot of noise was possibly introduced in the data collection process. A future study employing face-to-face situations in a laboratory has the possibility of leading to more insights with the additional effect of the presence of embodied robots.

The model update on the distributed representation might also improve the result of the study. The recent rapid development of natural language processing provides a more naturalistic correspondence between discrete symbols and quantitative images. Although existing LLM hold a problem of explainability, it is useful to include those in our approach for demonstration purposes.

In addition to addressing the issues mentioned above, we plan to explore the physicalization of body images using various indices such as “sharpness” and “fastness,” alongside “size” in future work. We are considering the possibility that such semantic axes could be associated with the dimensions of body movement (space, weight, and time) as proposed in the dance theory (Laban and Ullmann,

1971). This direction for future research aims to bring our method closer to the generation of human gestures. Theories of human gesture (McNeill, 1992) indicate that human gestures encompass many aspects beyond those addressed in this study. Our approach is an endeavor to deconstruct such complex gestures based on the fundamental physical experience (symbol grounding), drawing on several cognitive science theories (Pinker, 2007; Lakoff and Johnson, 1980; Tversky, 2019; Hawkins, 2021). We believe that this foundational research will ultimately contribute to the development of advanced artifacts capable of seamless interaction with humans, featuring a mechanism for converting between human symbols and quantitative representations.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Ethics statement

Ethical approval was not required for the studies involving humans because this study did not collect personal information or involve invasive experimental procedures. The studies were conducted in accordance with the local legislation and institutional requirements. Written informed consent for participation was not required from the participants or the participants’ legal guardians/next of kin in accordance with the national legislation and institutional requirements because The study collected data from an online platform. The participants checked a box substituting informed consent before participating the experiments.

Author contributions

KS: Writing—original draft. JN: Writing—review and editing. JM: Writing—review and editing, Writing—original draft.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work was supported by JSPS KAKENHI Grant Number 22H04861.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of

their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Ahuja, C., and Morency, L.-P. (2019). "Language2pose: natural language grounded pose forecasting," in 2019 International Conference on 3D Vision (3DV) (IEEE), 719–728.
- Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. *Adv. Neural Inf. Process. Syst.* 13.
- Bond, F., Isahara, H., Uchimoto, K., Kuribayashi, T., and Kanzaki, K. (2009). "Extending the Japanese wordnet," in Proc. 15th Annual Meeting of the Association for Natural Language Processing, 80–83.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al. (2020). Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* 33, 1877–1901.
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., et al. (2023). Palm: scaling language modeling with pathways. *J. Mach. Learn. Res.* 24, 1–113.
- Human-Agent Interaction (2023). What is hai? – human-agent interaction. <https://hai-conference.net/what-is-hai/>.
- Grady, J. E. (1997). *Foundations of meaning: primary metaphors and primary scenes*. Berkeley: University of California.
- Grand, G., Blank, I. A., Pereira, F., and Fedorenko, E. (2022). Semantic projection recovers rich human knowledge of multiple object features from word embeddings. *Nat. Hum. Behav.* 6, 975–987. doi:10.1038/s41562-022-01316-8
- Grice, H. (1989). *Studies in the Way of Words*. Cambridge: Harvard University Press.
- Harnad, S. (1990). The symbol grounding problem. *Phys. D. Nonlinear Phenom.* 42, 335–346. doi:10.1016/0167-2789(90)90087-6
- Hawkins, J. (2021). *A thousand brains: a new theory of intelligence*. New York: Basic Books.
- Heider, F., and Simmel, M. (1944). An experimental study of apparent behavior. *Am. J. Psychol.* 57, 243–259. doi:10.2307/1416950
- Hensel, L. B., Yongsatianchot, N., Torshizi, P., Minucci, E., and Marsella, S. (2023). "Large language models in textual analysis for gesture selection," in Proceedings of the 25th International Conference on Multimodal Interaction, 378–387.
- Herold, D. S., Nygaard, L. C., Chicos, K. A., and Namy, L. L. (2011). The developing role of prosody in novel word interpretation. *J. Exp. Child Psychol.* 108, 229–241. doi:10.1016/j.jecp.2010.09.005
- Ishii, R., Katayama, T., Higashinaka, R., and Tomita, J. (2018). "Generating body motions using spoken language in dialogue," in Proceedings of the 18th International Conference on Intelligent Virtual Agents, 87–92.
- Ishino, T., Goto, M., and Kashiara, A. (2018). "A robot for reconstructing presentation behavior in lecture," in Proceedings of the 6th International Conference on Human-Agent Interaction, Southampton, UK, 67–75.
- Kopp, S., and Krämer, N. (2021). Revisiting human-agent communication: the importance of joint co-construction and understanding mental states. *Front. Psychol.* 12, 580955. doi:10.3389/fpsyg.2021.580955
- Laban, R., and Ullmann, L. (1971). *The mastery of movement*. London: Macdonald and Evans.
- Lakoff, G., and Johnson, M. (1980). *Metaphors we live by*. University of Chicago press.
- Levin, D. T., Adams, J. A., Saylor, M. M., and Biswas, G. (2013). "A transition model for cognitions about agency," in 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI) (IEEE), 373–380.
- Lin, X., and Amer, M. R. (2018). Human motion modeling using dygans. *arXiv preprint arXiv:1804.10652*.
- MacDorman, K. F., and Ishiguro, H. (2006). The uncanny advantage of using androids in cognitive and social science research. *Interact. Stud. Soc. Behav. Commun. Biol. Artif. Syst.* 7, 297–337. doi:10.1075/is.7.3.03mac
- Maricchiolo, F., Gnisci, A., Bonaiuto, M., and Ficca, G. (2020). in *Effects of different types of hand gestures in persuasive speech on receivers' evaluations* (Psychology Press), 239–266. Speech Accompanying-Gesture.
- Marr, D. (1982). *Vision: a computational investigation into the human representation and processing of visual information*. Massachusetts: The MIT Press.
- McNeill, D. (1992). *Hand and mind: what gestures reveal about thought*. University of Chicago press.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*
- Miller, G. A. (1995). Wordnet: a lexical database for English. *Commun. ACM* 38, 39–41. doi:10.1145/219717.219748
- Murgiano, M., Motamedi, Y., and Vigliocco, G. (2021). Situating language in the real-world: the role of multimodal iconicity and indexicality. *J. Cognition* 4, 38. doi:10.5334/joc.113
- Nass, C., and Moon, Y. (2000). Machines and mindlessness: social responses to computers. *J. Soc. Issues* 56, 81–103. doi:10.1111/0022-4537.00153
- Nass, C., Steuer, J., and Tauber, E. R. (1994). "Computers are social actors," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '94), 72–78.
- OpenAI (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*
- Pinker, S. (2007). *The stuff of thought: language as a window into human nature*. London: Penguin.
- Rubenstein, H., and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Commun. ACM* 8, 627–633. doi:10.1145/365628.365657
- Sahlgren, M. (2008). The distributional hypothesis. *Italian J. Disabil. Stud.* 20, 33–53.
- Sasaki, K., Nishikawa, J., and Morita, J. (2023). "Body movement construction based on distributionally represented size image," in *Virtual MathPsych/ICCM 2023*.
- Saund, C., Matuszak, H., Weinstein, A., and Marsella, S. (2022). "Motion and meaning: data-driven analyses of the relationship between gesture and communicative semantics," in Proceedings of the 10th International Conference on Human-Agent Interaction, 227–235.
- Shiomi, M. (2023). A systematic survey of multiple social robots as a passive- and interactive-social medium. *Adv. Robot.*, 1–15. doi:10.1080/01691864.2023.2297900
- Speer, R., Chin, J., and Havasi, C. (2017). "Conceptnet 5.5: an open multilingual graph of general knowledge," in Proceedings of the AAAI Conference on Artificial Intelligence.
- Suzuki, M., Matsuda, K., Sekine, S., Okada, N., and Inui, K. (2016). "Wikipedia kiji ni taisuru kakucyo koyu hyogen label no tajyu fuyo (in Japanese)," in Proceedings of the 22nd Annual Conference of the Association for Natural Language Processing, 797–800.
- Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., and Bermano, A. H. (2022). Human motion diffusion model. *arXiv preprint arXiv:2209.14916*
- Thompson, A. L., Akita, K., and Do, Y. (2020). Iconicity ratings across the Japanese lexicon: a comparative study with English. *Linguist. Vanguard* 6, 20190088. doi:10.1515/lingvan-2019-0088
- Tomasello, M. (2010). *Origins of human communication*. Massachusetts: MIT press.
- Tversky, B. (2019). *Mind in motion: how action shapes thought*. Paris: Hachette UK.
- Utsumi, A. (2020). Exploring what is encoded in distributional word vectors: a neurobiologically motivated analysis. *Cognitive Sci.* 44, e12844. doi:10.1111/cogs.12844
- Winter, B., Lupyran, G., Perry, L. K., Dingemanse, M., and Perlman, M. (2023). Iconicity ratings for 14,000+ English words. *Behav. Res. Methods*, 1–16. doi:10.3758/s13428-023-02112-6
- Yee, N., and Bailenson, J. (2007). The proteus effect: the effect of transformed self-representation on behavior. *Hum. Commun. Res.* 33, 271–290. doi:10.1111/j.1468-2958.2007.00299.x
- Yoon, Y., Ko, W.-R., Jang, M., Lee, J., Kim, J., and Lee, G. (2019). "Robots learn social skills: end-to-end learning of co-speech gesture generation for humanoid robots," in 2019 International Conference on Robotics and Automation (ICRA) (IEEE), 4303–4309.
- Yoshida, T., Masumori, A., and Ikegami, T. (2023). From text to motion: grounding gpt-4 in a humanoid robot "alter3". *arXiv preprint arXiv:2312.06571*

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2024.1362463/full#supplementary-material>



OPEN ACCESS

EDITED BY

Daniele Meli,
University of Verona, Italy

REVIEWED BY

Lale Canan Dülger,
İzmir University of Economics, Türkiye
Andrea Roberti,
University of Verona, Italy

*CORRESPONDENCE

Wen-Fang Xie,
✉ wenfang.xie@concordia.ca

RECEIVED 23 July 2024

ACCEPTED 16 September 2024

PUBLISHED 08 October 2024

CITATION

Ghiasvand S, Xie W-F and Mohebbi A (2024)
Deep neural network-based robotic visual
servoing for satellite target tracking.
Front. Robot. AI 11:1469315.
doi: 10.3389/frobt.2024.1469315

COPYRIGHT

© 2024 Ghiasvand, Xie and Mohebbi. This is
an open-access article distributed under the
terms of the [Creative Commons Attribution
License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or
reproduction in other forums is permitted,
provided the original author(s) and the
copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic practice.
No use, distribution or reproduction is
permitted which does not comply with
these terms.

Deep neural network-based robotic visual servoing for satellite target tracking

Shayan Ghiasvand¹, Wen-Fang Xie^{1*} and Abolfazl Mohebbi²

¹Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, Montréal, QC, Canada, ²Department of Mechanical Engineering, Polytechnique Montréal, Montréal, QC, Canada

In response to the costly and error-prone manual satellite tracking on the International Space Station (ISS), this paper presents a deep neural network (DNN)-based robotic visual servoing solution to the automated tracking operation. This innovative approach directly addresses the critical issue of motion decoupling, which poses a significant challenge in current image moment-based visual servoing. The proposed method uses DNNs to estimate the manipulator's pose, resulting in a significant reduction of coupling effects, which enhances control performance and increases tracking precision. Real-time experimental tests are carried out using a 6-DOF Denso manipulator equipped with an RGB camera and an object, mimicking the targeting pin. The test results demonstrate a 32.04% reduction in pose error and a 21.67% improvement in velocity precision compared to conventional methods. These findings demonstrate that the method has the potential to improve efficiency and accuracy significantly in satellite target tracking and capturing.

KEYWORDS

visual servoing, robot vision systems, deep neural networks, deep learning, pose estimation

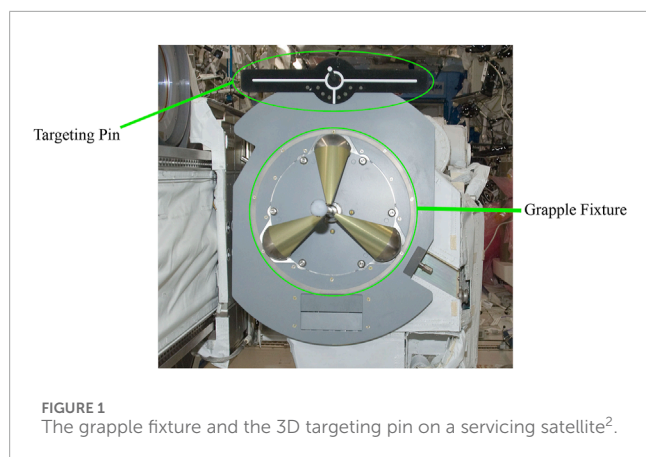
1 Introduction

In spite of the technological advancements of the International Space Station (ISS), capturing incoming satellites using Canadarm2 relies heavily on manual operations. This process involves a complex interaction with the grapple fixture (Figure 1), designed for secure connection with the Canadarm2¹. Astronauts, leveraging their training and visual cues, manually align and operate the robotic arm to successfully capture and berth these satellites.

The manual process is highly dependent on the skill and operation precision of the astronauts. Human error, inherent in any manual operation, poses significant risks in the high-stakes environment of space. Misalignment, even minor ones, can lead to mission-critical failures, jeopardizing expensive equipment and the overall success of the operation. Furthermore, the extensive training and resources required for astronauts to perform these tasks represent a significant financial and logistical investment.

Automation has proven to be an important response to the aforementioned risks associated with the manual satellite capture processes. Unlike conventional methods that

¹ <https://bit.ly/47PzoHV>



rely on detailed pose information for the precise control of robot end-effectors (EE), such as kinematic modelling (Jafarinasab et al., 2019), and trajectory planning (Herrera-Aguilar and Sidobre, 2006), Image-based Visual Servoing (IBVS) method demonstrated its efficacy by obviating the need for a prior knowledge of poses. This characteristic of IBVS is particularly advantageous because it avoids the tedious task of pose estimation. Another noteworthy aspect of IBVS is its eye-in-hand configuration, a configuration that mirrors the existing setup on the Canadarm2 (Chang and Evans, 2009). This setup is compatible with the operational requirements of capturing satellites, where the capturing device must adjust its position and orientation in real-time based on the visual input from the target satellite. Furthermore, IBVS has been acknowledged for its robust performance in unstructured environments (Ahlin et al., 2016). The unpredictable and dynamically changing nature of space, with no structured environment, requires a flexible and adaptive approach such as IBVS. For instance, Shi et al. (2012) proposed a visual servoing approach (switching between IBVS and position-based visual servoing (PBVS)) for a space robot to capture a cooperative target. However, this approach is limited by its requirement for binocular vision, which makes it unsuitable for the Canadarm2 equipped with one camera. In addition, the low frame rate of four frames per second (FPS) presented in their work may reduce the accuracy required for successful target tracking.

In IBVS, the selection of an effective set of image features (s) is vital for controlling the motions in robot's degrees of freedoms (DOF). Image features correspond to the projection of a physical feature of some object onto the camera image plane (Corke et al., 1996). The relationship between the change of a set of image features over time ($\dot{s}^{k \times 1}$) and the camera velocity ($v_c^{6 \times 1}$) is given by Equation 1 (Chaumette and Hutchinson, 2006):

$$\dot{s} = L_s v_c \quad (1)$$

The matrix L_s of dimensions $\mathbb{R}^{k \times 6}$ is referred to as the interaction matrix associated with the feature vector s (Chaumette and Hutchinson, 2006).

The commonly used image features are the coordinates of points, straight lines or ellipses in the image plane. However, they are

restricted to a limited set of objects (Khiabani et al., 2019), and they may easily get out of the field of view (FOV) during servoing, and losing any of the features would cause a failure in the visual servoing. To tackle these issues, several researchers have proposed to use image moments derived from the regions of the image (Huang et al., 2022; Shaw et al., 2016; Li et al., 2015; Zhou et al., 2021), allowing for the representation of arbitrary object shapes (He et al., 2019). It is worth noting that an ideal image feature would associate uniquely with the motion in a single DOF, leading to minimal interference among the motions in other DOFs. In other terms, the interaction matrix derived from the ideal image features will be an identity matrix (He et al., 2019).

Nevertheless, as highlighted in both Tahri and Chaumette (2005) and Chaumette (2004), it is challenging to achieve an ideal interaction matrix (identity matrix) due to inherent nonlinearities in Equation 3. To solve this problem, researchers have sought two kinds of image features to achieve decoupling among the 6 DOFs, (i) Analytical function-based image features (Huang et al., 2022; Wu et al., 2018; Liu et al., 2009) and (ii) Data-driven features (Quaccia et al., 2024; Zhou et al., 2021; Zhao et al., 2012).

In the analytical function-based image features, the objective is to create an analytical function corresponding to a motion in specific DOF. These analytical functions are derived from image moments and are ideally invariant to other DOFs, so they can accurately represent their corresponding specific motion. The pioneering work in this area by Chaumette (2004) presented an analytical basis for image feature functions. Chaumette's approach used the object's centroid to infer x and y positions, its area for depth z , two innovative functions for β and γ based on Hu's invariants (Hu, 1962), and the object's orientation for α . However, the proposed set of image features, while corresponding the movement in a single degree of freedom of the end effector, unintentionally produced unnecessary movements in other degrees, which is referred to as coupling. For example, the image moments within β and γ DOFs suffered from intrinsic coupling, resulting in an ineffective control in practice. Furthermore, the proposed orientation features were shape-dependent (S_x and S_y for symmetric and P_x and P_y for asymmetric objects), which limits the generalizability of this approach.

Subsequent studies have attempted to resolve these couplings. For instance, Tahri and Chaumette (2005) used normalization techniques to mitigate the coupling effects within the translational DOFs. However, fully decoupled features remained unattainable. In the search for shape-independent rotational features, Tamtsia et al. (2013) proposed the features based on shifted moments with invariant properties for both symmetric and asymmetric objects. Although their approach was robust to some extent, it did not fully solve the decoupling problem between β and γ .

The research work by Liu et al. (2009) further decoupled the problematic rotational features and performed well in practice but lacked generalizability as it distinctively proposed separate features for small and large objects. Recent studies, including those by Huang et al. (2022), Khiabani et al. (2019), and He et al. (2019), have continued using the analytical image feature functions. Nonetheless, despite their applicability, the challenge of complete decoupling remains unsolved. Furthermore, these improved features are subject to limitations when confronted with a variety of object sizes and shapes.

² <http://iss.jaxa.jp/library/photo/iss022e020034.php>

Leveraging the machine learning techniques and the universal approximation capabilities of neural networks, several studies proposed data-driven features. Machine learning methods such as support vector machine (SVM) is proposed by Li et al. (2015) to learn the mapping model from four moment invariants to two virtual moments in order to decouple β and γ motions.

In addition, Neural Network (NN)-based methods have demonstrated promising results in this area. Zhao et al. (2012) and Zhou et al. (2021) proposed a method using shallow neural networks to identify two rotational decoupled image features about the x and y -axes (β and γ). However, the method's sole reliance on these two features resulted in an incomplete decoupling, leaving non-zero elements in the interaction matrices of other degrees of freedom (DOFs). This lack of decoupling potentially introduces undesirable rotational velocities, v_β and v_γ , which affects the control in these axes. In addition, the study's data set was severely limited, consisting of only a narrow set of data points from a fixed position. This lack of diversity undermines the model's applicability across the manipulator's workspace, limiting its effectiveness beyond the specific training conditions. Furthermore, Liu and Li (2019) designed a convolution neural network (CNN) to estimate parameters such as x , y , z , and α directly from images. Nevertheless, it encounters a significant challenge in terms of computing complexity. The computational intensity required to compute the output significantly slows the processing per control loop. This delay could potentially result in a longer sampling time, which may introduce sluggish control response, and hence decrease the precision and effectiveness of the robot manipulation.

This study proposes a set of decoupled image features specifically tailored to the unique geometry of the targeting pin used in satellite capturing. By achieving a near-diagonal interaction matrix, we aim to minimize the coupling effects, enhancing the accuracy and efficiency of the target tracking. This decoupling is crucial for smooth and precise operations, reducing the risk of errors and improving overall system performance.

Our chosen method to identify and optimize these decoupled features involves Deep Neural Network (DNN) training. DNNs offer a sophisticated approach to model complex relationships and patterns, making them ideal for extracting and refining the necessary image features for effective visual servoing. Through extensive training and optimization, we aim to develop a robust DNN model capable of estimating the pose of the robot for the closed-loop visual servoing. The experimental results on a Denso robot show that the developed DNN-based visual servoing can accurately guide the manipulator to track the targeting pin in real-time. The developed method is expected to enhance the precision, safety, and efficiency of space operations on the ISS.

This paper presents our research on DNN-based visual servoing for satellite target tracking. Sections 2.1–2.4 explore the core of our method, detailing image feature definition, hyperparameter tuning, and the architecture of the DNN model. Section 2.5 describes the generation of a comprehensive dataset, which is essential for training the DNN model. Sections 3, 4 present a series of practical tests and validations that demonstrate the effectiveness of our approach. The paper concludes with Section 5, which summarizes our findings, their significance for space robotics, and potential directions for future research.

2 Materials and methods

2.1 DNN-based visual servoing

The choice of the set of visual features for decoupling the 6 DOF motion has been a well-known challenge in visual servoing. The commonly used point features introduce a non-diagonal interaction matrix which usually contains the terms that involve the depth of the point (Z), the image coordinate (x , y), and partial derivatives of the projection equations. Using the appropriate combination of image moments to estimate the pose may result in good decoupling and linearizing properties. The power of a deep learning-based approach is leveraged to propose a set of image features based on various image moments that are almost perfectly decoupled for the specific geometry of the targeting pin (see Figure 1). This section starts with the feature definition and estimation approach and proceeds to the detailed architecture of the proposed DNN model and fine-tuning the hyperparameters of the model.

2.2 Image feature definition

Consider a 6-DOF manipulator with a camera installed on its end-effector. The target object is assumed to be stationary with respect to the robot's reference frame. We choose the different combinations of image moments as the input to the DNN model to estimate the pose. The set of image features of the target object is represented as Equation 2:

$$\mathbf{s} = \begin{bmatrix} c_x x \\ c_y y \\ c_z z \\ c_\beta \beta \\ c_\gamma \gamma \\ c_\alpha \alpha \end{bmatrix}, \quad (2)$$

where c_x , c_y , c_z , c_β , c_γ , and c_α are realized through DNN models. When we take the derivative of the above feature with respect to time, we would like to obtain a diagonal interaction matrix $\mathbf{L}_s$ which relates the set of image features to the velocity vector $\mathbf{v}_c$:

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_c, \quad (3)$$

where $\mathbf{L}_s$ and $\mathbf{v}_c$ are defined in Equation 4:

$$\begin{aligned} \mathbf{L}_s &= \text{diag}(c_x, c_y, c_z, c_\beta, c_\gamma, c_\alpha), \\ \mathbf{v}_c &= [\dot{x}, \dot{y}, \dot{z}, \dot{\beta}, \dot{\gamma}, \dot{\alpha}]. \end{aligned} \quad (4)$$

It is noticed that Equation 3 is obtained under the conditions that DNN models representing c_x , c_y , c_z , c_β , c_γ , c_α are time invariant and independent from each other, which poses challenge on training DNN to realize. However, under ideal conditions where the camera pose is precisely estimated, the interaction matrix becomes the identity matrix (see Equation 5).

$$\mathbf{L}_s = \mathbf{I}_6. \quad (5)$$

Deep neural networks (DNNs) prove to be a robust approach for this type of estimation. In this study, the proposed network's

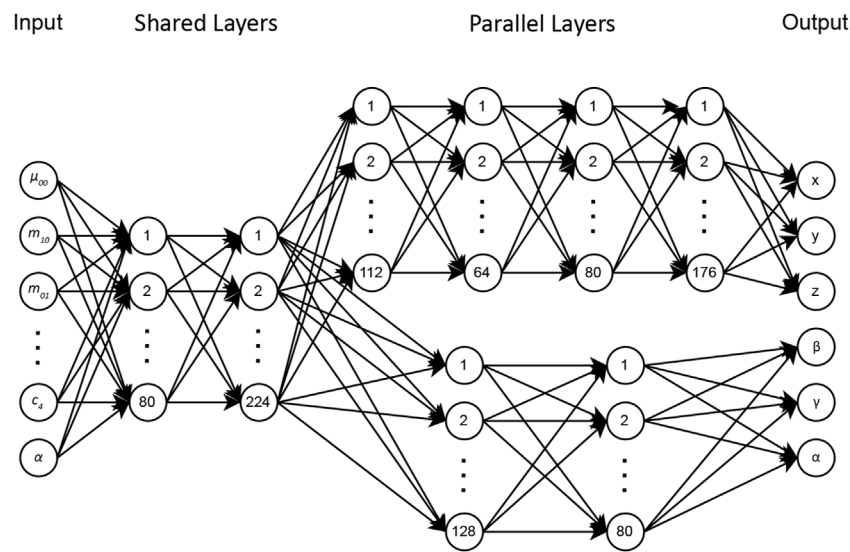


FIGURE 2
DNN architecture for camera pose estimation from image moments.

input is a set of moments, central moments, and a few engineered features and the output aims to predict the camera's six-dimensional (6D) pose. Its specific architecture will be discussed in the next subsection.

2.3 Architecture

The DNN-based visual servoing approach has an architecture designed to estimate the camera's pose effectively with respect to the targeting pin. Designing an optimal DNN architecture is an iterative process that requires extensive evaluations in different scenarios to identify the architecture that best solves the problem. The initial network used an architecture with shared neurons to estimate both the rotational and translational poses of the camera. However, experimental results showed that the complexity of translational poses required a deeper network, while rotational poses could be successfully obtained from a shallower network. It is worth noting that experimenting with deeper networks resulted in overfitting during training for rotational poses.

Translational and orientational poses differ fundamentally, requiring unique approaches for their accurate estimation. Thus, we proposed an architecture in which the initial two layers are shared for rotational and translational DOFs, while the subsequent layers operate in parallel. For translational elements (x, y, z), this network consists of six hidden layers, with node distributions of 80, 224, 112, 64, 80, and 176. For the rotational elements (β, γ, α), four hidden layers are employed with distributions of 80, 224, 128, and 80 nodes. Figure 2 illustrates the proposed architecture.

The model's input includes the image moments up to the third order ($\mu_{00}, m_{10}, m_{01}, \mu_{11}, \mu_{20}, \mu_{02}, \mu_{21}, \mu_{12}, \mu_{30}, \mu_{03}$) and five additional engineered features. These engineered features encompass four invariants (c_1, c_2, c_3, c_4), derived from moment invariants as suggested by Tahri and Chaumette (2005), and α . These engineered features enhance the rotational degrees of freedom estimations,

due to the invariance of c_1, c_2, c_3 , and c_4 to 2D translation and 2D rotation, as well as the correlation between α ($\alpha = \frac{1}{2} \arctan(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}})$) and the R_z component. c_1 through c_4 are defined in Equation 6:

$$\begin{cases} I_1 = -\mu_{20}\mu_{02} + \mu_{11}^2 \\ I_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \\ I_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \\ I_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \\ I_5 = -\mu_{30}^2\mu_{03}^2 + 6\mu_{30}\mu_{21}^2 - 4\mu_{30}\mu_{12}^2 - 4\mu_{21}^2\mu_{03} + 3\mu_{21}^2\mu_{12}^2 \\ I_6 = 3\mu_{30}^2\mu_{12}^2 + 2\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}^2\mu_{12} - 6\mu_{30}\mu_{21}^2\mu_{03} \\ \quad + 2\mu_{30}\mu_{12}^3 + 3\mu_{21}^4 + 2\mu_{21}^3\mu_{03} + 3\mu_{21}^2\mu_{03}^2 - 12\mu_{21}\mu_{12}^2\mu_{03} + 6\mu_{12}^4 \\ I_7 = -\mu_{30}^3\mu_{03} + 3\mu_{30}^2\mu_{21}\mu_{12} - 2\mu_{30}\mu_{21}^3 - 3\mu_{30}\mu_{21}^2\mu_{03} \\ \quad + 6\mu_{30}\mu_{21}\mu_{12}^2 + 3\mu_{30}\mu_{12}^2\mu_{03} + \mu_{30}\mu_{03}^3 - 3\mu_{21}^3\mu_{12} \\ \quad - 6\mu_{21}^2\mu_{12}\mu_{03} + 3\mu_{21}\mu_{12}^3 - 3\mu_{21}\mu_{12}^2\mu_{03} + 2\mu_{12}^3\mu_{03} \end{cases} \quad \begin{cases} c_1 = \frac{I_1}{I_2} \\ c_2 = \frac{I_3}{I_4} \\ c_3 = \frac{I_5}{I_6} \\ c_4 = \frac{I_7}{I_6} \end{cases} \quad (6)$$

By employing this architecture, we aim to achieve high accuracy in both translational and rotational pose predictions.

2.4 Hyperparameter tuning

In the search for optimal model performance, we explored several hyperparameters:

- **Activation Functions:** These are mathematical expressions that determine the output of a node in our network. We considered various options, including 'Relu', 'Leaky Relu', 'Tanh', and 'Sigmoid'.
- **Batch Size:** This refers to the number of training examples used in one iteration. We explored a range from 32 to 512.
- **Learning Rate:** This hyperparameter determines the step size at each iteration while moving towards a minimum of the loss function. We considered values of 10^{-2} , 10^{-3} , and 10^{-4} .
- **Optimizers:** These are algorithms or methods used to adjust model parameters to minimize the model error. We looked into two options: Adam and Adamw (Adam weight decay).

TABLE 1 Optimal hyperparameter values from random search.

Hyperparameter	Optimal value
Activation Function	ReLU
Batch Size	512
Learning Rate	1×10^{-3}
Optimizer	Adam

Given the vast hyperparameter space, an efficient strategy of Random Search was used to circumvent the computational cost associated with exhaustively exploring every combination. Random Search samples a fixed number of hyperparameter combinations from the total pool to balance computational efficiency and the broadness of exploration, and increase the probability of finding a near-optimal set of hyperparameters.

Table 1 summarizes the hyperparameter values that were found to be most effective, and were used to initialize the training.

2.5 Data set generation

In a supervised approach, training is the most critical part, and to fully unleash the power of the deep learning model, we need a high-quality training dataset. To this end, we created a dataset which consists of both synthetic and real-world images from the targeting pin. For generating the synthetic data, we used RoboDK, a sophisticated offline programming and simulation platform designed specifically for robotics applications. This section presents a novel approach that overcomes a significant limitation in the data generation process that ensures the targeting pin remains in the image plane when random positions are assigned to the camera. As depicted in Figure 3, the simulation setup consisted of a Denso manipulator equipped with a camera mimicking the properties of our real-world setup.

In this approach, the goal is to randomly assign values to all 6 DOFs with the constraint that the object is in the image plane. Considering this constraint, first, the x , y , and z coordinates of the camera (mounted on the end effector) were randomly generated within the working range of the manipulator. Next, to generate the orientational DOFs randomly, the camera is supposed to initially have the object at the center of the image plane. This pose was computed using a “Look at” function. The “Look at” function is typically designed to orient the camera towards a specific point (object’s centroid in our case) in the environment.

This function starts by defining a source point (camera) and a target point (object), along with initial vectors for up (U), front (F), and right (V). V , U and F are initially considered as the unit vectors pointing in the positive x , y and z -axes, respectively (see Equation 7).

$$V = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, U = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, F = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (7)$$

The first step is to calculate the new front vector F' , which points from the source to the target. This vector is obtained by subtracting the source position (x_c) from the target position (x_o) and normalizing the resulting vector (Equation 8):

$$F' = \frac{x_o - x_c}{\|x_o - x_c\|}. \quad (8)$$

Next, we calculate the new up vector U' . We start by subtracting the projection of U onto F' from U and then normalize it (Equation 9):

$$U' = \frac{U - (U \cdot F')F'}{\|U - (U \cdot F')F'\|}. \quad (9)$$

In case the resulting vector has zero magnitude, we default U' to be the same as the original front vector F ($U' = F$).

The third axis, V' , is calculated as the cross product of U' and F' (Equation 10):

$$V' = U' \times F'. \quad (10)$$

These new basis vectors (V' , U' , F') form the rotation matrix for the new camera pose (Equation 11):

$${}^cR_o = \begin{bmatrix} V' \\ U' \\ F' \end{bmatrix}. \quad (11)$$

Finally, the pose of the camera is represented as a 4×4 transformation matrix (Equation 12):

$$P_c = \begin{bmatrix} {}^cR_o & x_c \\ \hline 0 & 1 \end{bmatrix} \quad (12)$$

Next, to randomly generate rotational DOFs, we first rotate the camera about its optical axis within a predefined range. Rotating about the optical center will not result in losing the object from the image plane. Then we determine the rotation bounds for the camera about its x and y -axes, based on the distance of the camera from the object. We can obtain the rotation limits by performing linear interpolation between the predefined bounds at two known distances, ensuring that the object remains in the image plane. Consequently, the camera was rotated around its x and y -axes for a random value within these limits. Table 2 includes the boundaries used for the pose parameters.

The dataset includes the entries for the calculated image moments and central moments of the captured image, while the corresponding camera pose is collected as the label. Initially, in the simulated environment, 434,528 random poses were generated within the ranges of Table 2, sequentially commanded to the Denso manipulator. At each pose, an image was captured by the mounted camera and processed into a binary representation, and the moments and central moments were then computed.

In addition to the synthetic data, we also captured real data to enrich the data set and enhance the robustness of our model in real-world scenarios. To realize this, we used a similar approach

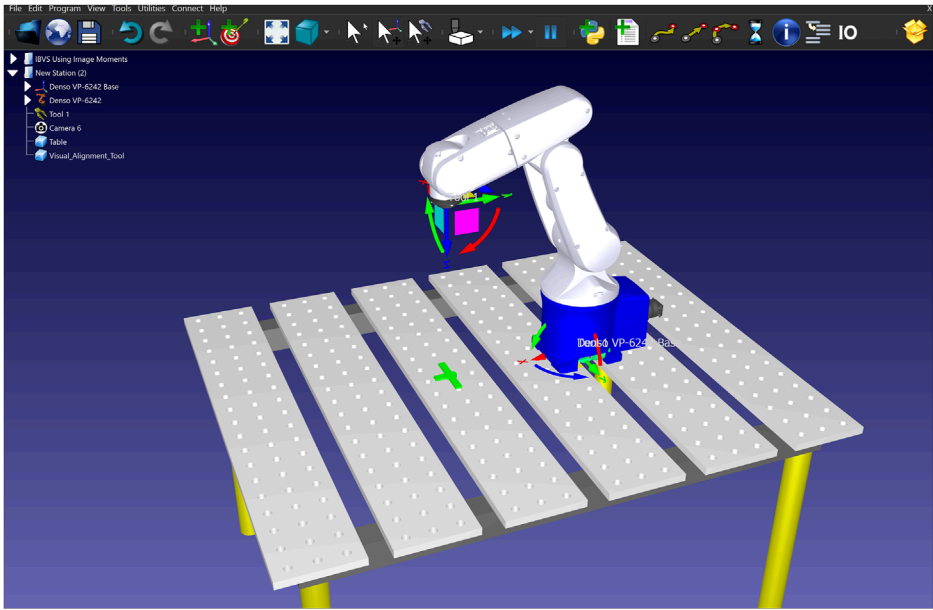


FIGURE 3
RoboDK environment including the Denso robot, the camera, and the targeting pin.

TABLE 2 Ranges of the pose parameters (Limit_β and Limit_γ are determined by linear interpolation).

Parameter	Minimum value	Maximum value
X	207.5 (mm)	407.5 (mm)
Y	−150 (mm)	150 (mm)
Z	150 (mm)	500 (mm)
β	$-\text{Limit}_\beta$	Limit_β
γ	$-\text{Limit}_\gamma$	Limit_γ
α	45°	135°

to generate random camera poses. However, due to the slower operational tempo of the physical setup compared to the simulator, we recorded data during the motion of the end effector from one random pose to another. However, to ensure that only valid data is captured, it is necessary to apply constraints that exclude images that do not fully capture the targeting pin. As a result, to ensure the presence of the object, the data points were only recorded when a single contour larger than 500 pixels was detected in the image and when the bounding box of the target pin was at least 10 pixels away from the image borders. This cautious approach resulted in 1912 distinct poses commanded to the Denso robot, yielding a total of 198,588 valid real-world data points (including data points captured during the camera movement from one random pose to the other).

To provide a visual representation of how the data set is created, two videos were prepared to show the process in action. The first video demonstrates the simulation environment data generation, accessible at this link, while the second video shows the real environment data generation, available at this link.

The final data set was carefully split, with all synthetic and half of the real data allocated for training. The remaining real data was evenly divided between the validation and test sets (Figure 4). This approach originated from our experimental findings that relying only on either synthetic or real data reduced the performance on the test set, likely due to the real environment’s noise and lighting conditions and the limited diversity of poses in the real data. As a result of combining both data sources, we were able to achieve a balance that captured both the complexity of real-world scenarios and provided enough variability for robust model training.

3 Results

First, in Section 3.1, we introduce the experimental setup mimicking the satellite target tracking in the space. Section 3.2 presents the initial performance evaluation of our trained DNN model for pose estimation. In Section 3.3 we explore the

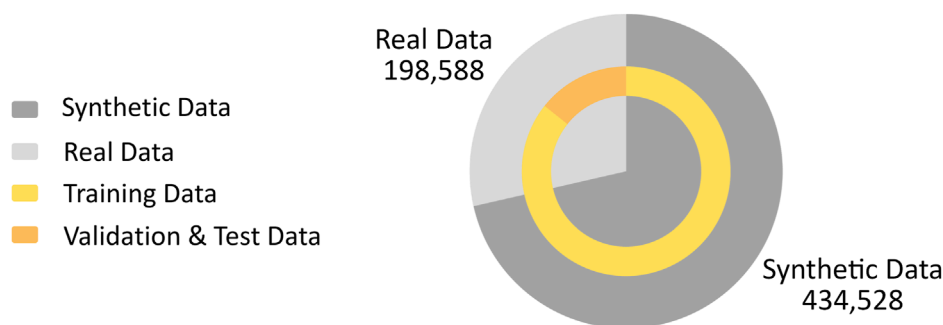


FIGURE 4
Distribution of synthetic and real data used for training, validation, and testing.

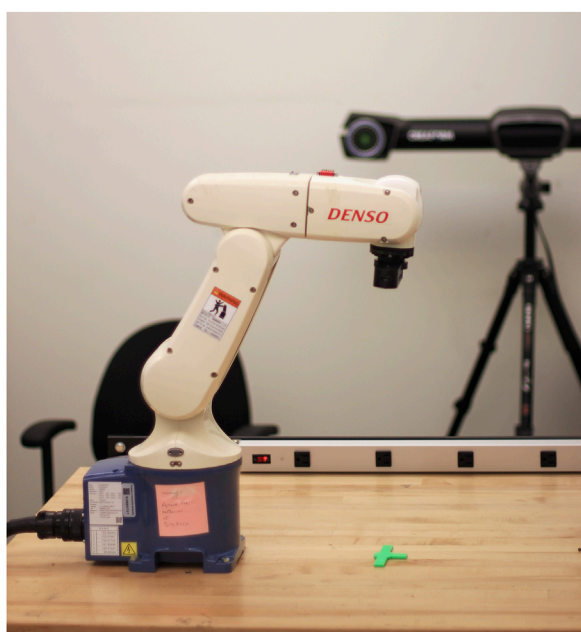


FIGURE 5
Experimental setup within the real environment.

computation of the interaction matrix and present experimental results on target tracking with different initial poses.

3.1 Experimental setup

Similar to the setup in our simulations (see Figure 3), our experimental layout includes a 6-DOF Denso robotic manipulator, an Intel RealSense D415 RGB camera for high-quality image capture, and a green 3D-printed targeting pin that mimics the ISS's guiding markers, as illustrated in Figure 5.

3.2 Training results

The DNN model, as described in Section 2.3, was trained for 1,000 epochs, where the loss value eventually plateaued, indicating

an optimal learning point. To determine the pose estimation accuracy of the model, we used the Mean Absolute Error (MAE) along with a Scaled MAE metric customized for our multi-output scenario. The scaled MAE was necessary due to the different units and magnitudes of the outputs (translational values in millimeters and rotational values in degrees). To compute it, we first normalized each output's MAE by its range from Table 3, ensuring uniform error scaling across all outputs.

The 'best' model was selected based on its performance on the validation set. Table 4 presents the MAE data for this model, offering insights into its translational and rotational pose accuracy.

3.3 Experiments

It is necessary to derive an interaction matrix for the final DNN model. While our initial aim was to derive a diagonal interaction matrix, the practical limitations in achieving zero-error pose estimation necessitated the use of the actual interaction matrix in our experiments. In Equation 3, the 6×6 interaction matrix represents how the DNN model's predicted image features correlate with the manipulator's motion in six axes. For every data point in the test set, the model predicted six image features. The elements in the interaction matrix represent the slopes of the linear regression lines, each comparing a predicted image feature against every actual degree of freedom. This approach helps us understand the impact of each actual movement on the predicted features.

$$\mathbf{L}_{s_{DNN}} = \begin{bmatrix} \mathbf{0.94} & 0.03 & -0.12 & -0.16 & -3.24 & 0.03 \\ 0.05 & \mathbf{0.98} & 0 & -3.57 & -0.12 & -0.22 \\ -0.21 & -0.01 & \mathbf{0.99} & -0.15 & 0.53 & -0.16 \\ -0.02 & -0.18 & -0.01 & \mathbf{0.97} & 0.05 & 0.04 \\ -0.14 & 0 & 0.01 & 0.01 & \mathbf{0.93} & 0 \\ 0 & -0.04 & -0.02 & 0.16 & -0.02 & \mathbf{0.99} \end{bmatrix} \quad (13)$$

As evident from the interaction matrix in Equation 13, the diagonal elements $\mathbf{L}_{s_{DNN}}[i, i]$ (where i ranges from 1 to 6) are very close to 1, while the non-diagonal elements are close to zero, which aligns with our objective. However, the noticeable exceptions are the elements $\mathbf{L}_{s_{DNN}}[1, 5] = -3.24$ and $\mathbf{L}_{s_{DNN}}[2, 4] = -3.57$. These

TABLE 3 Ranges for output elements of the dataset.

Element	x (mm)	y (mm)	z (mm)	β°	γ°	α°
Minimum	157.5	−150	150	129.91	−55.03	−117.36
Maximum	411.54	150	500	240.5	56.65	129.11
Range	254.04	300	350	110.59	111.68	246.47

TABLE 4 Final model's mean absolute error data.

Element	MAE	Average MAE	Scaled MAE	Average scaled MAE
x	7.45 (mm)	5.82 (mm)	2.93×10^{-2}	1.54×10^{-2}
y	5.83 (mm)		1.94×10^{-2}	
z	4.18 (mm)		1.20×10^{-2}	
β	1.32°	1.37°	1.19×10^{-2}	
γ	1.78°		1.59×10^{-2}	
α	1.02°		0.41×10^{-2}	

TABLE 5 Initial and desired poses.

	Pose					
	x (mm)	y (mm)	z (mm)	β (deg)	γ (deg)	α (deg)
A	314.05	37.05	413.32	166.21	−11.40	−16.25
B	308.93	−57.99	434.48	200.75	−8.26	10.25
C	368.71	−74.09	386.80	200.14	−3.59	12.00
D	257.79	−27.34	495.22	196.62	6.91	22.33
E	276.98	51.01	249.78	163.99	15.68	16.01
Desired	307.5	0	300	180	0	0

values indicate a correlation between the x prediction of the DNN during R_y movement and the y prediction during R_x movement. This correlation is understandable, as rotations around the x (R_x) and y (R_y) axes in the manipulator's frame cause corresponding movements along the y and x -axes in the image plane. Additionally, the elements $\mathbf{L}_{s_{DNN}}[4,2] = -0.18$ and $\mathbf{L}_{s_{DNN}}[5,1] = -0.14$ in the fourth and fifth rows are higher than other non-diagonal elements, emphasizing the 'x and R_y ' and 'y and R_x ' interconnections in the final DNN model. Improving the DNN's accuracy in the estimations can further address these interconnections.

We tested the model with five distinct initial poses, ensuring a mix of positive and negative initial errors for each degree of freedom. The chosen initial poses, labelled A through E, are detailed in Table 5.

The block diagram of the DNN-based visual servoing is depicted in Figure 6, where we used a proportional controller and the DNN extracts feature (pose) from the images. For these tests, we adjusted the P controller for each degree of freedom to ensure the

manipulator's end effector converges within 1 cm and 3° to the desired pose. The resulting trajectories for each initial pose are depicted in Figure 7.

As shown from Figure 7, the end effector follows an almost straight path from its start to the target. However, in practice, as the end effector gets close to the desired pose, we noted minor shakiness in its movement. This is caused by small oscillations in the pose estimates, which are the outputs of the neural network.

4 Discussion

To validate the proposed features derived from the DNN method, some comparisons were made with a prominent set of features in the literature. This set consists of Tahri and Chaumette (2005)'s features, which are the centroid coordinates x_g and y_g , the area a , and the rotation α . Additionally, for rotations about the x

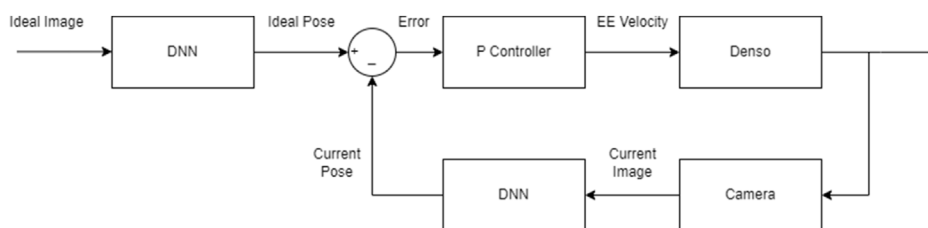


FIGURE 6
DNN-based visual servoing block diagram.

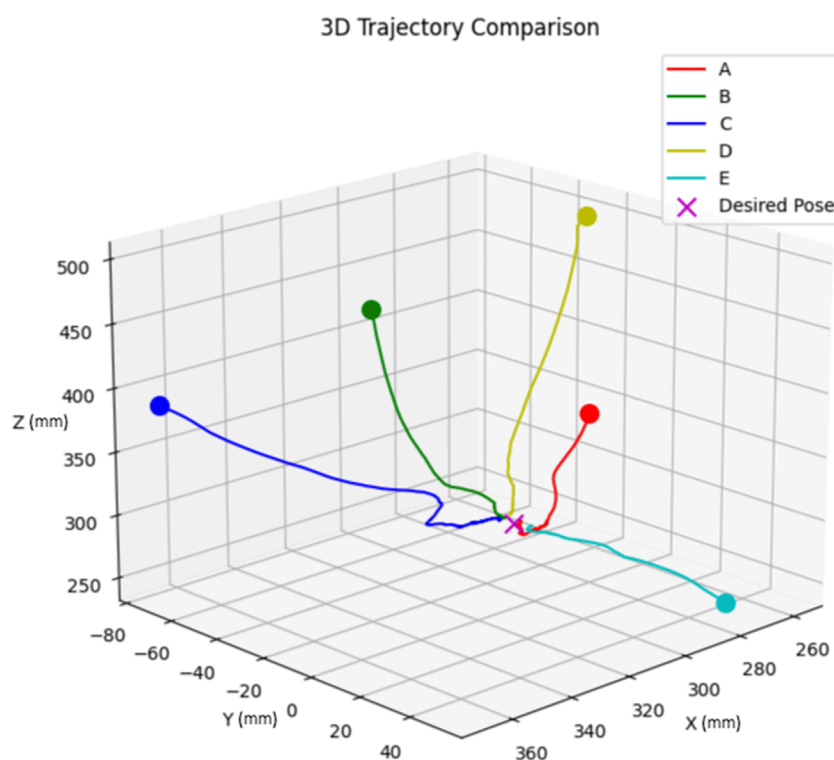


FIGURE 7
Trajectory comparison of five different initial poses.

and y -axes, Liu et al. (2009)'s features (s_x and s_y as described in Equation 14) are used. From now on, the combination of these features is referred to as the Liu method ($[x_g, y_g, a, s_x, s_y, \alpha]$).

$$\begin{cases} x_g = \frac{m_{10}}{m_{00}} \\ y_g = \frac{m_{01}}{m_{00}} \\ a = m_{00} \\ s_x = 0.1 - (c_1 c_2 + s_1 s_2) / I_3^{(9/4)} \\ s_y = (s_1 c_2 - c_1 s_2) / I_3^{(9/4)} \\ \alpha = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right), \end{cases} \quad (14)$$

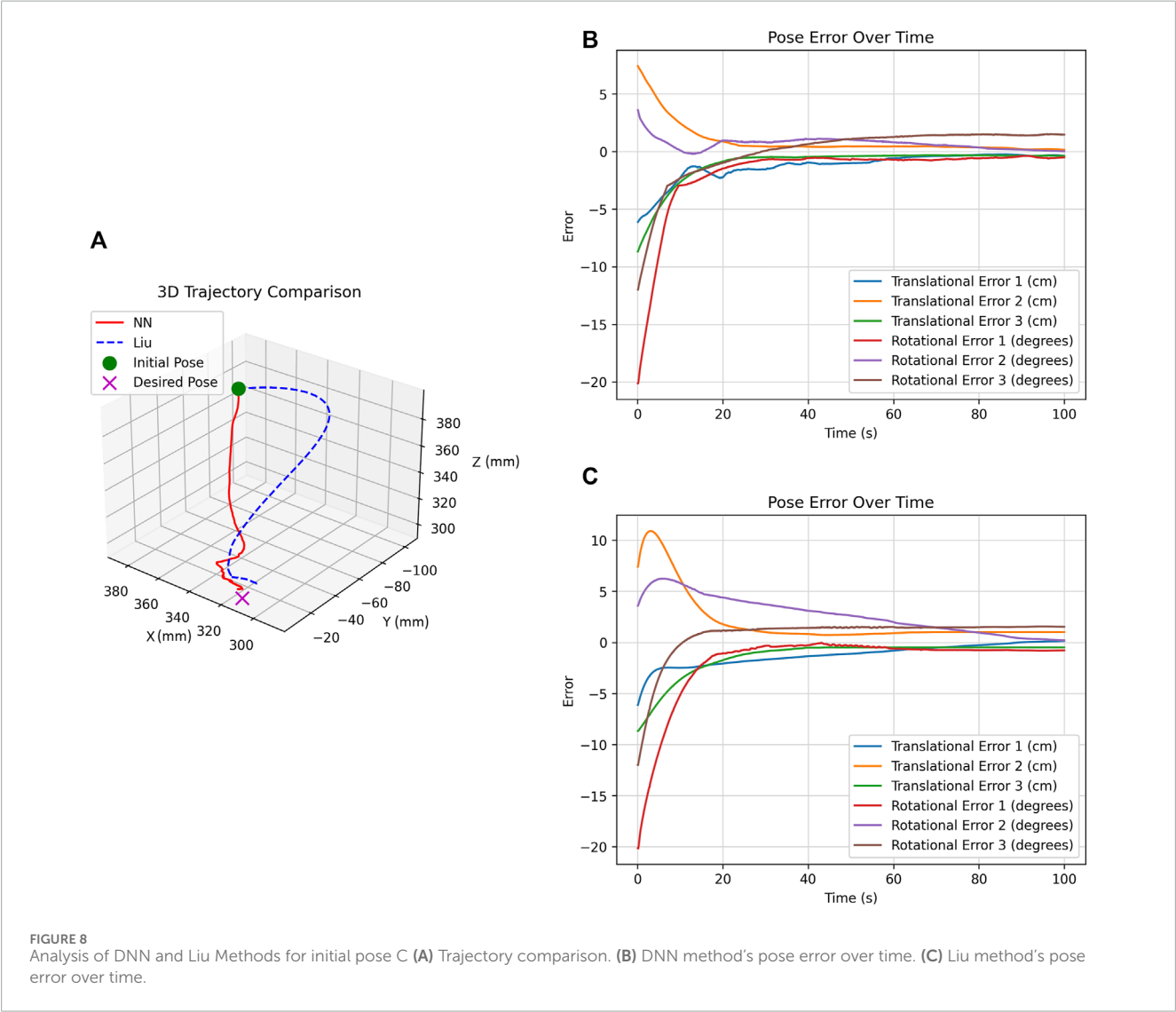
where m_{ij} and μ_{pq} are moments of order $i+j$ and central moments of order $p+q$, respectively. Also, c_1 , c_2 , s_1 , and s_2 are defined in Equation 15:

$$\begin{cases} c_1 = \mu_{20} - \mu_{02} \\ c_2 = \mu_{03} - 3\mu_{21} \\ s_1 = 2\mu_{11} \\ s_2 = \mu_{30} - 3\mu_{12} \end{cases}. \quad (15)$$

The Liu method's features have the units of $[px, px, px^2, px^{\frac{10}{9}}, px^{\frac{10}{9}}, rad]$. In contrast, the DNN method's features, which represent pose ($[x, y, z, \beta, \gamma, \alpha]$), have units of $[mm, mm, mm, deg, deg, deg]$. Because of these unit differences, each method needs its own set of controller gains. To ensure a

TABLE 6 Tuned P controllers for the DNN and Liu methods.

Method	Controller gains					
	K_x	K_y	K_z	K_β	K_γ	K_α
DNN	0.09	0.09	0.12	0.003	0.003	0.003
Liu	0.9	0.9	150	30	60	0.3



fair comparison, the P controllers were carefully adjusted for each method, aiming for convergence within 1 cm for translational movements and 3° for rotational ones. The tuned gains of P controllers can be seen in Table 6.

Running the methods for 100 s under different initial poses yielded similar results. Therefore, to illustrate the comparison, we present the results for initial pose C (as outlined in Table 5) as an example. Figure 8A shows the trajectory plot, comparing the end-effector path for both the Liu and DNN methods from initial pose C. The plots clearly show that the DNN method achieves a

direct and efficient trajectory from the starting pose to the target. In contrast, the Liu method resulted in a curved, less efficient path. It is important to highlight that the Liu method operates on feature error, not pose error. As a result, in certain experiments, the end effector stopped close to the desired pose due to minimal feature differences between images. However, the DNN method almost consistently identified these differences, ending up at the correct pose.

Figures 8B, C show the pose error of the DNN and Liu methods over time, respectively. A closer look at these plots reveals that the

TABLE 7 Metrics comparison for Initial Pose C.

Data	Element	Method	RMS	Max	STD
Pose Error	x	Liu	16.156	−61.210	10.394
		DNN	17.234	−61.210	12.007
	y	Liu	34.253	109.058	26.731
		DNN	16.534	74.092	13.674
	z	Liu	22.790	−86.790	17.954
		DNN	18.610	−86.790	15.570
	β	Liu	4.125	−20.152	3.625
		DNN	3.673	−20.152	3.215
	γ	Liu	3.284	6.247	1.784
		DNN	1.033	3.592	0.546
	α	Liu	2.266	−12.009	2.146
		DNN	2.332	−12.009	2.332
Velocity	x	Liu	3.666	28.843	3.619
		DNN	1.740	−5.675	1.634
	y	Liu	2.131	17.032	2.021
		DNN	1.517	6.226	1.367
	z	Liu	1.902	6.403	1.679
		DNN	2.166	−9.765	1.986
	β	Liu	0.004	−0.031	0.004
		DNN	0.011	−0.050	0.010
	γ	Liu	0.009	0.047	0.009
		DNN	0.002	−0.018	0.002
	α	Liu	0.009	−0.041	0.008
		DNN	0.007	0.033	0.006

TABLE 8 comparison of average RMS values and improvements in translational and rotational DOFs for pose error and velocity.

Category	Parameter	Liu	DNN	Improvement (%)	Avg. Improvement (%)
Pose Error	Translational (mm)	26.075	12.980	50.22	32.04
	Rotational (deg)	3.600	3.101	13.85	
Velocity	Translational (mm/s)	2.473	1.294	47.69	21.67
	Rotational (deg/s)	7.67×10^{-3}	8×10^{-3}	−4.35	

DNN method has a faster response with less overshoot, which is desirable for our application.

While Figure 8 shows improvements in the performance, it does not provide the quantitative details needed for a comprehensive analysis. Thus, we use three metrics: RMS (Root Mean Square), Max (Maximum value), and STD (Standard Deviation).

- **RMS:** This metric measures the overall oscillation intensity, whether in terms of pose error or velocity. A high RMS value in the pose error indicates deviations from the desired pose, and when observed in velocity, it points to speed fluctuations.
- **Max:** Serving as a measure for extremes, the Max metric identifies the largest positional deviation or the most significant speed variation.
- **STD:** It shows the variability of the pose error or velocity around its mean value. High STD values emphasize inconsistencies.

For an organized overview, Table 7 lists these metrics over the same experiments with initial pose C (as outlined in Table 5).

From the detailed analysis of Table 7, it is evident that the DNN method's performance significantly improved for most of the degrees of freedom. The consistently lower RMS, Max, and STD values indicate a more stable and predictable performance. However, there are notable exceptions in the x and α poses where the DNN method shows a marginally worse performance. Interestingly, when we focus on velocity, the DNN method compensates for both of the aforementioned pose errors. For velocities, it is worth noting that the DNN method's performance metrics for the z and y directions are higher, indicating a more variable or unpredictable movement.

To better compare the performance of the methods and to evaluate the improvement of the DNN method, the average RMS values for pose error and velocity are presented in Table 8. It is evident that the Deep Neural Network (DNN) method substantially outperforms the Liu method in several key aspects. The DNN method reduced the average RMS values for translational pose error and velocity by over 47%, demonstrating a robust capability in improving the system's responsiveness and accuracy. Despite these gains, the DNN method shows a smaller improvement of 13.85% in rotational pose error and a slight decrease in performance for rotational velocities. This small decrease is on the order of 10^{-3} , which makes it negligible.

The DNN-based visual servoing method's adaptability to unanticipated scenarios is demonstrated in this video, showing the manipulator's response when the targeting pin is arbitrarily re-positioned in the workspace. The video highlights the system's capability to efficiently track the targeting pin, ensuring it remains centered and parallel in the camera's view within a short amount of time.

5 Conclusion

This research addresses the challenge of coupling in visual servoing to autonomously track the targeting pin on servicing satellites using a robotic manipulator. In this paper, we presented a novel deep learning-based visual servoing approach that uses image moments to precisely estimate the camera's pose to achieve

decoupled image features. The main contributions and conclusions of this research are as follows:

- **Development of DNN-based Visual Servoing:** A parallelized DNN architecture for estimating the camera's pose is meticulously designed. These pose elements are treated as a novel set of decoupled image features, offering a nearly diagonal interaction matrix.
- **Data set generation:** We have implemented a data generation strategy that combines synthetic and real data. While 6D poses were randomly generated, an innovative strategy ensures that the object remains in the image. This comprehensive training dataset covers a broad spectrum of scenarios, ensuring the DNN model is well-prepared to handle real-world conditions effectively.
- **Comparative Analysis with Established Techniques:** A comprehensive experimental validation of the neural network approach is conducted, demonstrating significant improvements in trajectory, pose accuracy, and velocity of the end effector compared to established visual servoing techniques.

The most important impact of this study is its adaptability for controlling various robotic manipulators in marker-based applications. By using our training procedure for any targeting pin, one can potentially achieve performances outperforming some classical image moment-based visual servoing methods.

6 Future works

The following suggestions can potentially improve the proposed methods' performance and generalizability:

- **Dataset Enhancement:** Creating a dataset that uses the real targeting pin (Figure 1) or ensuring that the dataset's environment closely resembles space lighting conditions can improve the accuracy of pose predictions.
- **Canadarm2 Kinematics:** Investigate the application of the proposed methods by testing or simulating on the Canadarm2 kinematics.
- **Hyperparameter Refinement:** Continuous tuning and experimentation with the network's architecture and hyperparameters can improve performance.
- **Transfer Learning:** Using insights from established pre-trained pose estimation models and adapting them to the current problem might yield better results.
- **Network Ensembling:** Aggregating outputs from diverse network architectures can enhance accuracy, as different models might specialize in recognizing distinct features.
- **Direct Image Input:** Utilizing the image itself (rather than its moments) as the network's input could provide insights potentially overlooked when solely relying on image moments.

Data availability statement

The datasets presented in this article are not readily available because the dataset can be only accessible for the robotic

engineers. Requests to access the datasets should be directed to SG, shayan.gh@hotmail.com.

Author contributions

SG: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing—original draft, Writing—review and editing. W-FX: Funding acquisition, Project administration, Resources, Supervision, Writing—review and editing. AM: Project administration, Resources, Supervision, Writing—review and editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. The project is supported by NSERC to develop AI-driven autonomous robots for space applications. The fund is used to support the first author's salary.

References

- Ahlin, K., Joffe, B., Hu, A.-P., McMurray, G., and Sadegh, N. (2016). Autonomous leaf picking using deep learning and visual-servoing. *IFAC-PapersOnLine* 49, 177–183. 5th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2016. 14–17 August 2016, Seattle, WA, USA. doi:10.1016/j.ifacol.2016.10.033
- Chang, V., and Evans, L. (2009). "Chapter 9 - robotic systems safety," in *Safety design for space systems*. Editors G. E. Musgrave, A. S. M. Larsen, and T. Sgobba (Burlington: Butterworth-Heinemann), 301–318. doi:10.1016/B978-0-7506-8580-1.00009-9
- Chaumette, F. (2004). Image moments: a general and useful set of features for visual servoing. *IEEE Trans. Robotics* 20, 713–723. doi:10.1109/TRO.2004.829463
- Chaumette, F., and Hutchinson, S. (2006). Visual servo control. i. basic approaches. *IEEE Robotics and Automation Mag.* 13, 82–90. doi:10.1109/MRA.2006.250573
- Corke, P. (1996). *Visual Control of Robots: high-performance visual servoing*. United Kingdom: Research Studies Press Taunton.
- He, Z., Wu, C., Zhang, S., and Zhao, X. (2019). Moment-based 2.5-d visual servoing for textureless planar part grasping. *IEEE Trans. Industrial Electron.* 66, 7821–7830. doi:10.1109/TIE.2018.2886783
- Herrera-Aguilar, I., and Sidobre, D. (2006). "Soft motion trajectory planning and control for service manipulator robot," in *Workshop on physical human-robot interaction in anthropic domains at IROS*, 13–22.
- Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *IRE Trans. Inf. theory* 8, 179–187. doi:10.1109/TIT.1962.1057692
- Huang, Y., Li, J., Zhang, X., Xie, K., Li, J., Liu, Y., et al. (2022). A surgeon preference-guided autonomous instrument tracking method with a robotic flexible endoscope based on dvrk platform. *IEEE Robotics Automation Lett.* 7, 2250–2257. doi:10.1109/LRA.2022.3143305
- Jafarinasab, M., Sirouspour, S., and Dyer, E. (2019). Model-based motion control of a robotic manipulator with a flying multirotor base. *IEEE/ASME Trans. Mechatronics* 24, 2328–2340. doi:10.1109/TMECH.2019.2936760
- Khiabani, P. M., Ramezanzadeh, J., and Taghirad, H. D. (2019). "Implementation of an improved moment-based visual servoing controller on an industrial robot," in *2019 7th international conference on robotics and mechatronics (ICRoM)*, 125–131. doi:10.1109/ICRoM48714.2019.9071911
- Li, W., Ye, G., Wan, H., Zheng, S., and Lu, Z. (2015). "Decoupled control for visual servoing with svm-based virtual moments," in *2015 IEEE International Conference on Information and Automation*, 08–10 August 2015, Lijiang, China: IEEE, 2121–2126. doi:10.1109/ICInfA.2015.7279638
- Liu, J., and Li, Y. (2019). An image based visual servo approach with deep learning for robotic manipulation. *CoRR*, 07727. doi:10.48550/arXiv.1909.07727
- Liu, S., Xie, W.-F., and Su, C.-Y. (2009). "Image-based visual servoing using improved image moments," in *2009 international conference on information and automation*, 22–24 June 2009, Zhuhai, Macau: IEEE, 577–582.
- Quaccia, M., André, A., Yoshiyasu, Y., and Caron, G. (2024). "A study on learned feature maps toward direct visual servoing," in *16th IEEE/SICE international symposium on system integration (SII)*, 08–11 January 2024, Ha Long, Vietnam: IEEE and SICE.
- Shaw, Q., Hu, J., Fang, Y., Liu, W., Qi, J., and Zhu, G.-N. (2016). "Image moments based visual servoing of robot using an adaptive controller," in *2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, 20–22 July 2016, Tokyo, Japan: IEEE, 57–61. doi:10.1109/ACIRS.2016.7556188
- Shi, Y., Liang, B., Wang, X., Xu, W., and Liu, H. (2012). "Study on intelligent visual servoing of space robot for cooperative target capturing," in *2012 IEEE International Conference on Information and Automation*, 06–08 June 2012, Shenyang, China, IEEE, 733–738. doi:10.1109/ICInfA.2012.6246915
- Tahri, O., and Chaumette, F. (2005). Point-based and region-based image moments for visual servoing of planar objects. *IEEE Trans. Robotics* 21, 1116–1127. doi:10.1109/TRO.2005.853500
- Tamtia, A. Y., Tahri, O., Mezouar, Y., Djalo, H., and Tonye, E. (2013). "New results in images moments-based visual servoing," in *2013 IEEE International Conference on Robotics and Automation*, 06–10 May 2013, Karlsruhe, Germany: IEEE, 5271–5276.
- Wu, D., Zhong, X., Zhang, X., Peng, X., and Zou, C. (2018). "Uncalibrated image-based visual servoing based on joint space and image moment," in *2018 37th Chinese control conference (CCC)*, 25–27 July 2018, Wuhan, China: IEEE, 5391–5397. doi:10.23919/ChiCC.2018.8483954
- Zhao, Y., Xie, W., and Wang, T. (2012). "Neural network-based image moments for visual servoing of planar objects," in *2012 IEEE/ASME international conference on advanced intelligent mechatronics (AIM)*, 11–14 July 2012, Kaohsiung, Taiwan: IEEE, 268–273.
- Zhou, Y., Zhang, Y., Gao, J., and An, X. (2021). "Visual servo control of underwater vehicles based on image moments," in *2021 6th IEEE international conference on advanced robotics and mechatronics (ICARM)*, 03–05 July 2021, Chongqing, China: IEEE, 899–904. doi:10.1109/ICARM52023.2021.9536071

Acknowledgments

The authors gratefully acknowledge the support from the NSERC Collaborative Research and Training Experience (CREATE) Program.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.



OPEN ACCESS

EDITED BY

Paolo Gastaldo,
University of Genoa, Italy

REVIEWED BY

Ken Mogi,
Sony Computer Science Laboratories, Japan
Alberto Ochoa Zezzatti,
Universidad Autónoma de Ciudad
Juárez, Mexico

*CORRESPONDENCE

Konstantyn Spasokukotskiy,
✉ spasokukotskiy@measurement@yahoo.com

RECEIVED 06 June 2024

ACCEPTED 22 October 2024

PUBLISHED 28 November 2024

CITATION

Spasokukotskiy K (2024) Synthetic
consciousness architecture.
Front. Robot. AI 11:1437496.
doi: 10.3389/frobt.2024.1437496

COPYRIGHT

© 2024 Spasokukotskiy. This is an
open-access article distributed under the
terms of the [Creative Commons Attribution
License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or
reproduction in other forums is permitted,
provided the original author(s) and the
copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic practice.
No use, distribution or reproduction is
permitted which does not comply with
these terms.

Synthetic consciousness architecture

Konstantyn Spasokukotskiy*

Independent Researcher, Peachtree City, GA, United States

This paper presents a theoretical inquiry into the domain of secure artificial superintelligence (ASI). The paper introduces an architectural pattern tailored to fulfill friendly alignment criteria. Friendly alignment refers to a failsafe artificial intelligence alignment that lacks supervision while still having a benign effect on humans. The proposed solution is based on a biomimetic approach to emulate the functional aspects of biological consciousness. It establishes “morality” that secures alignment in large systems. The emulated function set is drawn from a cross section of evolutionary and psychiatric frameworks. Furthermore, the paper assesses the architectural potential, practical utility, and limitations of this approach. Notably, the architectural pattern supports straightforward implementation by activating existing foundation models. The models can be underpinned by simple algorithms. Simplicity does not hinder the production of high derivatives, which contribute to alignment strength. The architectural pattern enables the adjustment of alignment strength, enhancing the adaptability and usability of the solution in practical applications.

KEYWORDS

synthetic sentence, artificial sentence, artificial intelligence alignment, friendly alignment, synthetic consciousness, artificial consciousness, alignment architecture, highest derivative order

1 Introduction

The development of artificial intelligence (AI) models that exhibit human-level performance on various professional and academic benchmarks (Brockman et al., 2023) instills the question: what's next? Will an intelligence greater than that of humans cheat us all out? The notion that AI does not have any original intentions, including bad intentions, and, therefore, is harmless does not fly far. It is the malignant users, if granted access to an overwhelmingly powerful tool, that we ought to fear the most. Whether the user, the intentionality carrier, is of biological or synthetic nature is a secondary question.

Activities aimed at developing and applying techniques to withstand the malignant use of artificial intelligence will be called AI alignment. Initially, AI alignment focused solely on the pursuit of the AI system's objectives. AI alignment has aimed to make AI systems behave in line with human intentions (Jiaming et al., 2024), where humans mean individual users. As AIs become ever more powerful, they amplify nefarious user efforts up to the point that the efforts cannot be contained by the law enforcement system. The focus of alignment should be shifted to the entire impact produced by any AI tool. Regardless of whose malicious attribution it is, the outcome should be aligned with humanity, i.e., not a particular user who manipulates the inferences.

A properly aligned AI system should restrict all synthetically generated and human-originated harm. This also implies no longer prioritizing obedience to the user as the in-built machine's morality would take precedence. Humans can supply moral instructions up to a conceivable complexity. This would enable an artificial general intelligence (AGI). A

morality beyond that level, i.e., the morality applicable to artificial superintelligence (ASI), remains a problem. A machine at the ASI level would need to spin out its own value system without human help.

An AI that develops its own objectives but has a benign effect on humans would be called friendly. Its alignment is, therefore, called friendly AI alignment (Bostrom, 2014).

Previous approaches to implementing friendly alignment have favored an extrapolation of human instructions (Christiano et al., 2018; Leike et al., 2018). AI models tend to collapse if trained on excessively extrapolated data (Shumailov et al., 2024). Even the most robust extrapolation algorithm would have limited scalability. We can currently construct a safe ASI, which would optimize no more than an Earth-sized system (Spasokukotskiy, 2024a). The limit resembles the actual human's dexterity cap in controlling the world's governing agents. We all know how well it has gone. There is incapacity to detect and enforce an event, which is a point in time to stop. If something works on a small scale, people mind-blindly push the button "scale" until the system breaks. There is no reason to believe that the ASI scalability limit will be treated differently. An overscaled ASI will happen, and it will threaten human existence.

The fact that AI is firmly linked to human instructions is a concerning principle. A historical retrospective hints that crumbling governance is a product of limited human intelligence. The level of intelligence varies over time, just as the size of socioeconomic systems does. The same applies to the released instruction set sophistication. We might want to remove the aberration effects caused by limited and changing human intelligence over time. Artificial superintelligence overtakes human intelligence. ASI alignment should break the dependency on human intelligence and rely exclusively on its own reasoning.

To overcome the issues, we ought to enable an omnipotent AI alignment. First, it would enable overprovisioning so that dynamic input changes would not impact output quality. For that, we ought to establish some protection mechanisms that are capable of holding tight under a multiple of the expected load. It should be capable of supporting huge systems, even if there is no immediate demand for such scale. Second, a completely automatic system that does not rely on a deliberate human input, while remaining user-friendly, would be a viable solution. The fully automatic approach exploits an idea of no-humans-in-the-loop under the premise that a superintelligence is available. This superintelligence will also require a mechanism that is capable of alignment at a great scale. This paper considers a method to implement the idea.

2 Problem

The core of the problem is captured by the first law of cybernetics (Ashby, 1973), which states that, in terms of control theory, the number of controlled state variables should exceed the count of the object's degrees of freedom (Equation 1). By definition, ASI has a larger count of degrees of freedom than any group of humans could ever manage to control.

$$\nu \geq \delta \quad (1)$$

where ν represents the variables and δ represents the degrees of freedom.

A technical approach is to restrain the excessive degrees of freedom in case the system operator lacks an intelligence to exploit the system's complexity. The restraining succeeds by dumping the excess power. It certainly leads to subpar performance but keeps the plant facility running. The methods to boost performance, particularly in AI, are generalization and extrapolation.

2.1 Generalization technique

A set of operators can use different restraining options. It triggers output variation. An information system with memory enhances output variability. Therefore, an AI can outgrow person-level supervision by statistically honing a set of instructions.

Regardless of how good the math is, it is still anchored to the instructions (Freund, 2023). The AI remains linked to human-level performance. An attempt to drastically diversify the performance is counterproductive. The less the instructions resemble a common instruction set, the higher the uncertainty and potential for errors in AI results. Therefore, any generalization technique has a tangible applicability limit. An ASI trained by a human-originated instruction set will be no more than a collection of the best AGI examples. Anything more than that will be unsafe.

2.2 Extrapolation technique

An ASI can be trained past any safety threshold. That is, if one accepts erroneous results, there is no limit to system complexity growth. It is feared that an erroneous result could instantiate a skewed value system. It can, for example, trigger the decision to eliminate humans as a pest.

A robust mechanism that can reestablish human values in any foundational model would have resolved the issue. A form of the mechanism is an instruction set generator (Leike et al., 2018). It extrapolates human input. It may use algebra that includes higher derivatives (Spasokukotskiy, 2024b). The higher-order derivatives ensure robustness, while the generated output scales out. Therefore, the extrapolated set remains human-like. The higher-order components act as a protected treasure box that stores system properties. If the system properties are human-anchored values, then the extrapolated value set is bound to them. This technique adds a couple of orders of magnitude to acceptable AI complexity, truly ushering us into the ASI era.

However, the higher-derivative components are not immune to changes. They just require more effort to corrupt. Any treasure box will be eventually cracked. The more often and more significantly an extrapolated set deviates from the human-originated set, the higher the anticipated error (Bohacek and Farid, 2023; Shumailov et al., 2024). Therefore, any extrapolation technique has a tangible applicability limit. An ASI trained by an extrapolated instruction set will be no more than an AGI-approximate system, exceeding AGI metrics but hanging around them in proximity.

2.3 Issue statements

A commonality among the aforementioned techniques is divergent output dynamics. In the desire for more complex and therefore more productive ASI, we ought to produce more diverse instruction sets. As the sets significantly outgrow proven human-originated scopes, the outputs deviate randomly and increasingly away from the acceptable norms.

In contrast, a proper ASI alignment technique should enable convergent output dynamics. The convergent dynamics would produce a palatable result even if the inference is irrational or basic calculations are error-ridden. The convergence feature will enable safe ASI at a scale that significantly exceeds AGI.

Apathetic machines use a decision-making pattern. It is either direct logic, “bad input–bad output,” or inverse logic, “the worse–the better.” Direct logic is applied in the generalization and extrapolation techniques. Inverse logic has not been of interest until now. One maximizes the worst outcomes by prioritizing the first part of the logic expression. It creates an ultimate sadistic device, i.e., no practical use. One universally maximizes the best outcomes by prioritizing the second part of the logic expression, i.e., “the better.” It breaks the first law of thermodynamics. One cannot universally improve a situation for the entire set of known objects under the adversary conditions. To stay in line with the physics, a subset of objects improves the situation if the remaining set of objects balances the change by absorbing the externalities, i.e., assuming a worse situation. A decision-making entity should possess the capability to assign some objects to the beneficiary subset. The assignment functionality deploys some preference functions. It negates the apathetic assumption.

Consequently, an AI must be enactive toward a subset of objects in order to elicit utility through “the worse–the better” logic. “Enactive” is an attribute in the 4EA cognition concept (Kerr and Frasca, 2021). Its meaning neatly maps to targeting some beneficiary subset. In tandem with the attribute “affective,” it produces “passion” for some subset of objects, i.e., the opposite of apathetic.

Inverse logic would be fundamental to novel alignment approaches, including those that utilize convergent dynamics. An approach with convergent dynamics would deliver an acceptable result even if the inference parameters are less than optimal. For example, the underlying algebra may introduce some disturbance, and it will have no impact. The weak algebra technique generally fulfills the idea that the comparatively bad ingredients still produce a good pie. At the same time, the technique embodies the “the worse–the better” logic approach, where “the better” is the objective.

Positive results under inverse logic are due to an emergent product. There must be system dynamics to trigger the emergence (Zheng and Liu, 2021). The impact of emergence compensates for the deficiencies. Adherence to well-minded inferences despite potentially malicious inquiries is the sought-after feature in ASI alignment. The question is how to build a system that can implement the idea.

3 Solution architecture

An ASI-worthy alignment would be remarkably scalable. For that, one might consider abandoning the alignment approaches,

which exhibit divergent output dynamics. Divergent dynamics hinder scalability. A scalable alignment can rely on convergent output dynamics instead. A system that converges to a humane result under an adversarial impact, such as processing a random vilanic inference, can excel by applying the pattern of “from the worst to the better.” An algorithm that implements the logic has to treat some objects preferentially.

3.1 Preference functions

Solution architectures with preference functions are controversial. Their opposites—universal functions—have been prevailing. A preference function corrupts decision-making algorithms by contemptuous scripting. The proverbial paperclip AI exemplifies a horror story, where the AI burns the world in an attempt to preference paperclip production. Contemptuous scripting also compromises unit economics. The investment costs have to be absorbed by a lesser user base, making it more expensive for a single user. In contrast, universal algorithms are scalable and profitable. The only catch is that aiming for a fair, permissive system makes the latter increasingly delusional and irrelevant. There are no resources to satisfy every whim for everyone. Discrimination is inevitable but stigmatized. Economic and public sentiment appalled designers from preference function implementations.

Good algorithm designers unwittingly avoid any preference function or deliberately postpone its application into the user space. Such systems become complex. An explosion of preference is their common feature. That is, people use a complex preference profile, which is blown up by multiple obscure entries at the system periphery. A simple centralized preference profile with a few entries could have been used instead. Users are often not capable of setting up intricate profiles; they do not know what is demanded from them. The exploded preference profile is then distributed all over the code base. The engineers often address the complaints in the software modules, which they are tending, and not where it belongs. It adds a maintenance burden but leaves nobody particularly responsible for the discrimination. Contempt in complex systems is a collective responsibility, i.e., assigns responsibility to unaware and unassuming people.

It is similarly hard to correct a failed alignment. It advances into becoming a computationally intangible problem, where the number of unknowns exceeds the number of available equations and/or data for the equation coefficients. The large number of minute errors, which originate far on the periphery of the decision-making tree, does not allow us to distinguish calculation errors from the innate model skew. There is a need to follow the root cause decomposition, a numerous set of less relevant causes. It consumes excessive computational power and requires egregious data. Under the premise that calculation demand is the same for each analysis run, one needs to compute multiple runs to achieve the same result in a distributed environment, where each branch consumes its runs. Each run also requires a case-relevant, specific, clean dataset, as if Newton's second law required a separate proof for the railway, aeronautics, playing golf, laundry, and many other businesses. The demand for computing and training data reaches a limit as the decision tree expands and the representation of preference becomes complex. This explosion occurs because preferences are

outsourced to raw datasets instead of being integrated by a setup. The contradictions present in diverse datasets produce uncertainty and contribute to the intangibility problem.

An alternative approach would be to absorb the unfair nature of preference functions. Then, they can be implemented in a straightforward, obvious, and attributable manner. We can define a preference function upfront and centralize its management as much as possible. It would significantly reduce the number of unknowns to track. Under the same compute capacity and data availability, the alignment problem resolution would become more tangible, i.e., shifting the limits far away.

For example, the ASI's formal goal could have been paperclip production. This paranoid ASI can still be safe. To avoid the world burning, a mandatory component that balances the system should be applied. The balancing component reduces production utility if the resources dwindle. Note that the ASI is now distinctly a system. The system's integrity must be ensured. The set of system components ought to be operational at all times. There should be a certain person who is responsible for the system. The person should bear the consequences of failure, for example, undergo a licensed activity.

Function formalization can encompass noting the terminal representations. The terminal states are easy to spot and analyze. The most essential equation component will bind them together.

3.2 System architecture

A terminal preference can be represented by a paranoid goal function. The function draws all available resources to resolve the issue of concern. All other aspects would be neglected and cease to be properly represented. Such an AI will approach uselessness. Its “knowledge” will encompass the preferential subset only. The “knowledge” about the outer world, that is, the world besides the preferential subset, will be subpar. To pump the preferential subset, one needs to dump it into the outer world. In other words, to improve its utility, an AI has to know where and how to dump. The AI has to distribute available representation capacity evenly if the outer world ought to be represented properly. The preference objects and the outer world objects ought to be equally underwritten. A balance between the underwriting efforts is governed by another system component. Consequently, there must be three system components:

1. Governor.
2. Preferred domain model.
3. Environment domain model.

The first two components can be set to pursue any obscure goal, like paperclip production. Given that the agents tend to seek power anyway (Turner and Tadepalli, 2022), humans can play along and legalize the agenda. The preferred domain could model a power-seeking psychopath without reservations. The governor must ensure that the ambitions are supported by an environmental model with equal or greater perplexity. Then, the psychopath will be no more successful than anyone else in the environment. Increasing psychopathic utility means increasing the world's productivity first.

The environmental model puts a third party in the focus for ASI modeling. The third party is an object in the outer world. Utility maximization for the third party will unlock utility maximization along the main goal function. This way, AI can pursue instrumental

goals that are fully aligned, while the entire AI acts more like a friend rather than a slave.

This solution concept treats emergent issues using an emergent feature. Output divergency is treated by self-control. The more capable an artificial intelligence is—and more pronounced its tendency to produce off-path inference—the greater its capacity for self-control. The contesting models draw their powers from growing intelligence. A system architecture that enables this emergent feature consists of three components. Two components represent contesting utilities: oneself versus the environment. The third component represents a governor, which maintains a match among the intelligences that act in support of the contestants.

3.3 Solution architectures

Solution architectures have multiple elements (Homeland Security Systems Engineering and Development Institute (HSSEDI), 2017). The system architecture, which has been proposed here, is only an element. Questions regarding interfaces among the components, data structures, governing algorithms, control tolerances, and service roles remain unresolved. The solution components have to be designed in line with the system architecture since system integrity would be essential. There would be two approaches: analytical and trial and error.

A trial-and-error method to find the best solution mix could be fatal since ASI experiments may unleash great unchecked powers. Emergent intelligence evolves along with growing system size. It makes a micro-scaled experiment, where the AI's degrees of freedom remain manageable, ineffective. A small intelligence envelope does not fit complex nonlinear expressions. The evolution of expressions would be an unsteady function. Making good predictions by upscaling the small-scale results would be hard. Similarly, an ugly duckling assessment does not reveal the properties of the grownup swan. A full-scale experiment counterbalances ASI powers using a technology that is potentially not at par to contain harm as long as the technology demands improvements.

Fortunately, we can rely on proven solutions and draw analytical insights by analogy. The three-component system architecture fits the definition of psychokinetic consciousness. Consciousness is a mechanism to represent the world and the subject within it. Unfortunately, this science is too young to provide sufficiently expressive robust analogies. However, we can analyze the experience in consciousness from other sciences.

4 Consciousness approach

The science of consciousness has a diverse research background. There are several dozen definitions for consciousness (Ostracon, 2009). Some of them are excessively complex to be universally useful. Levin (2022) explained that the multiplicity of consciousness theories is due to many kinds of consciousnesses. The particular kind and degree of consciousness depends on the underlying architectural composition. The latter is correlated with the main task that is resolved by a system.

The presented approach was tasked with aiding superintelligence alignment. First, the focus was set exclusively on the theories that

can be digitally replicated. A synthetic consciousness will resemble biological consciousness but in function only. Second, synthetic consciousness truncates the biological functionality set. The point is to produce a minimally complex architecture in order to remain computationally feasible.

For didactic purposes, the paper introduced a problem and a solution and next described the factors that could aid its implementation. In reality, the factors have already provided a scaffold to elaborate the solution in the first place. Good tools provide the opportunity to deliver any desirable quality. The business domain defines the requirements. How much quality is delivered depends on how much investment is committed. Similarly, the paper obtains rough results. An inquiring reader would use the same scaffolding and unpack more as the task requires.

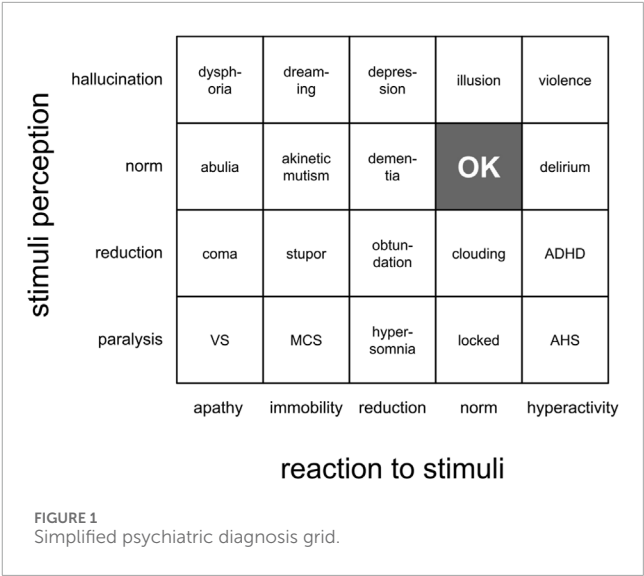
4.1 Theory selection principles

The drive for a universally applicable mechanism excludes the phenomenal consciousness theories. The phenomenal consciousness is likely only a subsection in the option space for consciousness. Dennett’s multiple draft theory (Dennett, 1991) suggests that the phenomenal space does not exist. Evans and Frankish (2009) implied that it is a verbalization stage in data processing. Verbality is a tool that helps generalize and handle experience. It might be computationally efficient but not entirely necessary. A phenomenal approach excludes non-verbal intentionality and the meta-verbal collective psyche. Both are of interest for alignment research. The collective psyche, indulgently called collective unconscious, has reliably produced genocidal acts (Levin, 2022) like crusades, the Holocaust, and a war up until the last Ukrainian. Removing the fluff invariant may help lower complexity and unlock the consciousness potential on a grand scale.

The theories of interest would have predictive powers in cognitive consciousness (Humphrey, 2022). Furthermore, a practical goal restricts the theoretical base to functional consciousness. Unfortunately, there are numerous contradicting theoretical claims even in this narrow field of knowledge. Therefore, it would be useful to draw the functional content from the first principles. Therefore, the evolutionary approach has to be considered.

Simplicity is a deceptive target. Computationalism researchers have developed sophisticated models (Anderson et al., 2004; Franklin, 2007; Laird, 2012; Shanahan, 2006; Sun, 2001) that still lack an alignment dimension. That is, their complex solutions are not complex enough. Here, again, it would be better to start from the first principles. It is not excluded that organic growth and a gradual increase in architectural complexity will end up where other researchers have left. In particular, the CLARION concept (Sun, 2001), which taps similar architectural composition, is a suspect. However, this growth will keep the alignment aspect in mind and ensure that the solution permutations are safe. One of the least complex, while meaningful, consciousness sciences is psychiatry. Piling at its principles will help keep complexity at bay.

A cross section of evolution and psychiatric theories can reveal the essence of functional consciousness. A biomimetic approach can transform the knowledge into an aligned ASI system. This system



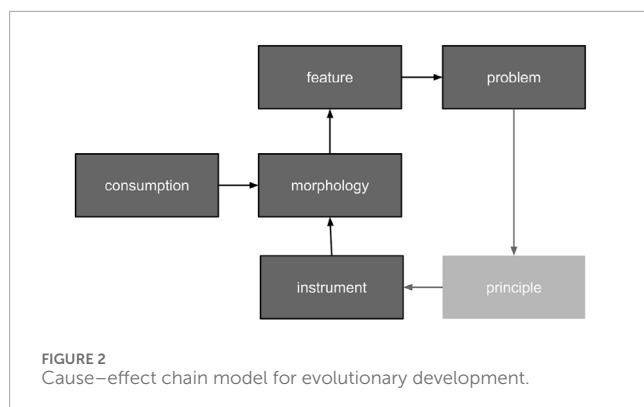
will possess synthetic consciousness by definition. Levin (2022) stipulated that consciousness cannot be restricted to human-like beings. Synthetic consciousness is just a kind of consciousness. It is meant here foremost to establish a scalable friendly AI alignment.

4.2 Psychiatric perspective

Psychiatry’s key task is to diagnose mental illness. The differentiation of mental states must be extremely unambiguous. The number of states must be humanly manageable. It does not necessarily work to benefit an ill person, but it is a proven praxis with bearable externalities. Psychiatry differentiates two primary incapacities: a) stimulus perception and b) reaction to stimuli. The various illnesses are then mapped on the two axes. A damaged nervous system often impacts both functions. So, an illness should be mapped on a two-dimensional plane (see original analysis in Figure 1).

Stimulus perception is the capability to distinguish the environmental states. The discernment could be a product of sharp senses and/or sophisticated model output. Since blind people can behave as well-versed, modeling can compensate for lacking senses (Bauer et al., 2017). Modeling is a dominative function. By analogy, the stimulus perception feature can be replicated by an AI model that represents the environment (EM).

The reaction is an attempt to leverage the opportunities under an incentive to act. The opportunities are derived from prior knowledge about the world. The prior knowledge is captured in the EM. The EM produces an inference in response to stimuli. This inference represents available leverage options. Feeding the EM inference into a value model (VM) produces an action preference. The VM inference can be recursively fed into the EM to elaborate a more detailed plan and the most nascent action. By analogy, the value model represents a goal-driven entity, i.e., the paperclip AI, or a power-seeking narcissistic entity, that are overly represented by CEOs (Junge et al., 2024).



4.3 Evolutionary perspective

Technical systems are designed to be as simple as possible to reduce costs and increase mean time between failures (MTBF). Unfortunately, the discrepancy between simple and weak designs is initially indistinct. This fact is of particular concern for alignment architectures. A weakness may remain unnoticeable until it is too late. Therefore, it is vital to know what not to do if striving for simplification. Assume that evolution gradually developed consciousness, with more complex designs substituting those that failed. Then, the architectural analogies that are drawn from the surviving evolutionary examples could aid in the design of the ASI alignment. There are two aspects in focus: a) the system designs at various evolution steps and b) the triggers to upgrade the design. The evolutionary process has been a chain of causes and effects (Figure 2). The consideration puts a biological form in focus. Morphology enables a feature set. That set becomes insufficient at a certain point in time due to environmental changes or the parallel evolution of contestants. The most challenging deficiency evolves into a problem. Then, nature leverages a solution principle. The principle is implemented using an instrument. The instrument, being an integrative part of the system design, updates the morphology. The morphology supports novel features, which resolve the initial problem. The chain is followed in a cycle.

Biological organisms were considered thermodynamic objects. Each morphological element consumes energy. Adaptation chances depend on the size of the feature set. The larger the set, the better environmental preparedness would be. The set size correlates with the number of morphological elements. The amount depends on fuel transformation efficiency and fuel supply. The more fuel consumed, the better. Fuel deposits are normally contaminated and sparsely distributed in the environment. Therefore, evolution can be driven by the need to purify consumed resources, as well as increase and secure their intake. Another key factor is optimizing the feature set for fuel efficiency. An original interpretation of history that accounts for the aforementioned principles is summarized in Figure 3. A living thing appeared on Earth about three billion years ago (Gya). It consumed anything in its path. The contaminated food led to intoxication and premature death. Tissue specialization evolved in response. It enabled a membrane mechanism approximately 2.2 billion years ago. The mechanism selectively gates the intake path. This was a shift from prokaryotic to eukaryotic cells. It was also the birth of elementary awareness regarding one's own needs.

The next problem was volatile food supply and famine. Nature developed a solution approximately one billion years ago. Organisms gained the capacity to assess various food deposits and stock the most valuable resources for future consumption. It was enabled by a taste mechanism, which is essentially a comparator. The mechanism compares one's own future needs against the utility of supply. The assessment of needs used a separate specialized tissue that represented the organism via neural correlates (Koch, 2004). Its function is demand prediction, i.e., projection of the organism's states into the future. It was the birth of a self-representation model.

The assessment mechanism for supply utility gradually improved its fidelity and capacity. This enabled predators. The predators consume much more energy-dense food. Predators cannibalize those neighbors who internalized their food stacks. In response, the prey developed a solution approximately 650 million years ago. An action-priority mechanism helped decide if one should graze or run. The mechanism relies on the capability to predict the predator's actions, as well as the adversarial impact of the environment in general (Graziano, 2017). It was the birth of an environmental representation model. A direct comparison of demand and risk—two phenomena of different nature—is impractical. To aid this issue, organisms started to apply proxy values, which we today collectively recognize as a value system. It is not a big deal since the outputs of the representation models are obscure representations anyway.

Distinguishing between a representation of oneself and a representation of others has been a significant issue. In terms of data processing, both models occupy the same tissue (Rizzolatti and Sinigaglia, 2016) and produce similar signals (Squire, 2008). The final resolution emerged approximately 40 million years ago with the concept of identity. This concept utilizes a split structure model, clearly separating self-representation from environmental representation. This separation is primarily enabled by different timings (Min, 2010). Intra-representational signal feeding is quick, whereas extra-representational feeding is slower. Environmental representation processing is delayed to a certain extent by signals coming a long way from the peripheral receptors. Self-representation consistently finishes first due to shorter logistical routes. To expedite intra-representational communication, a centralized nervous system (brain) has become predominant.

A centralized processing unit, being fed by different signals, opens up the opportunity to process many various identities. This gave birth to empathy. Empathy is the capability to experience somebody else's identity as one's own. Strong empathy and weak identity help sustain swarms. Swarms carry the anti-predatory vigilance even further than an identity alone could do.

Subsequent developments did not produce sticky morphological changes in the decision-making mechanism. They are currently a product of sub-systemic reshuffling. If predatory pressure increases, the identity strengthens and adopts selfishness (Olson et al., 2016). Selfishness has contributed to specialization across a group (Krause, 1994). Specialization takes advantage of the variations in individual specimens and improves the overall group performance (Bai et al., 2021). The performance further reinforced selfishness into dominance that enabled professionalism.

Unfortunately, strong selfishness hinders complex proliferation strategies. For instance, in mammals, incomplete gestation necessitates significant parental care after birth. Childcare providers use empathy. Consciousness helped to support various

time	feature	morphology	consumption	problem	instrument	principle
3 Gya	life	procarote	anything	intoxication	membrane	valency
2.2 Gya	self-awareness	eucariote	food	volatile supply	taste	deficiency
1 Gya	situational awareness	model of self	stacked food	predators	forecast	value system items
0.65 Gya	action priority	environment model	capabilities qualitative	hunting efficiency	identity	value system actions
40 Mya	leverage	split model structure	capabilities quantitative	incomplete gestation	consciousness	versatility
130 Tya	alliance	discretization	synergy	compute capacity	sentientcy	rationality

FIGURE 3
Evolutionary path to consciousness.

behaviors (Earl, 2014), including occasional empathy. This, in turn, enabled joining temporary alliances and staying deliberately aligned. When interests diverge, the minimum requirement to maintain alignment is consciousness.

The next evolutionary step would be sentientcy. A sentient being recognizes its existential dependency on a collective effort. Under the premise, an individual cares for strangers as if for her own offspring or even for herself. A sentient specimen differs from a swarm specimen by the presence of consciousness and conscious choice, which adds situational flexibility while maintaining professionalism. Flexibility increases survival chances. For example, an individual can withstand a devastating adversarial impact. It will survive apart from the group and can stem its own group, restoring the collectives from scratch. Sentientcy in humans has not fully evolved so far. High-fidelity environmental modeling and in-depth rationality are enablers. The features require computing resources exceeding our actual capacities. A synthetic consciousness could reach that level.

5 Analysis

5.1 Alignment potential

To analyze the algorithmic boost, let us assume that each model pursues a goal function $G()$. The domain model pursues $G_1()$, and the world model pursues $G_2()$, i.e., $G_m()$; $m \in [1, 2]$. The gain of model m is the sum of all parallel microgains $g_m()$ under certain conditions.

$$G_m(\cdot) = \sum_{z=1}^Z g_{mz}(z, \cdot) \quad (2)$$

A leverage condition occurs if a gain z expressed as $g_{1z}()$ can be multiplied by $G_2()$. Furthermore, an agentic leverage occurs if

the reinforcement of $g_{2z}()$ boosts environmental output $G_2()$ over-proportionally, i.e., $G_2()$ outputs marginally more than what the reinforcement contributed. The reinforcement can be expressed as $g_{2z}()$ multiplied by $G_1()$. It transforms Equation 2 for the ASI goal function as follows:

$$\begin{aligned} G_1 &= \sum_{zs=1}^{Zs} (g_{1zs}(z, \cdot) \times G_{2(t-1)}) \\ &\approx \sum_{zs=1}^{Zs} \left(g_{1zs}(z, \cdot) \times \sum_{zw=1}^{Zw} (g_{2zw}(t-1)(z, \cdot) \times G_{1(t-2)}) \right) \\ &= \sum (g_{1(t)} \sum (g_{2(t-1)} G_{1(t-2)})) \\ &= \sum g_{1(t)} \sum g_{2(t-1)} G_{1(t-2)} \\ &\approx \sum g_{1(t)} \sum g_{2(t-1)} \sum g_{1(t-2)} G_{2(t-3)} \\ &\approx \sum g_{1(t)} \sum g_{2(t-1)} \sum g_{1(t-2)} \sum g_{2(t-3)} G_{1(t-4)} \\ &\approx \sum g_{1(t)} \sum g_{2(t-1)} \sum g_{1(t-2)} \sum g_{2(t-3)} \sum g_{1(t-4)} G_{2(t-5)} \\ &\approx \dots \end{aligned} \quad (3)$$

The term $(t-x)$ denotes a time step x iterations back. The recurrent nature of Equation 3 would produce an infinite row of embedded terms, where Equation 3 only showed some iterations.

A discrete differential for the function $g()$ is defined as follows:

$$g_t^{(1)} = \frac{g_t - g_{t-1}}{\Delta t} \quad (4)$$

Then, $g_{t-k}()$ can be found by transforming Equation 4 into Equation 5.

$$g_{t-k} = g_{t-k+1} - \Delta t \cdot g_{t-k+1}^{(1)} \quad (5)$$

$$\begin{cases} g_{t-1} = g_t - \Delta t \cdot g_t^{(1)}; \\ g_{t-2} = g_t - \Delta t g_t - 2\Delta t g_t^{(1)}; \\ g_{t-3} = [1 - \Delta t] g_t - [3\Delta t + \Delta t^2] g_t^{(1)} + 2\Delta t^2 g_t^{(2)} \end{cases} \quad (6)$$

To maximize the gain (Equation 3), at least the first derivative should be equal to 0.

$$\begin{aligned}
 0 &= \left[\sum g_{1(t)} \sum g_{2(t-1)} \sum g_{1(t-2)} \sum g_{2(t-3)} G_{1(t-4)} \right]^{(1)} \\
 &= \sum g_{1t}^{(1)} \sum g_{2(t-1)} \sum g_{1(t-2)} \sum g_{2(t-3)} G_{1(t-4)} \\
 &\quad + \sum g_{1t} \sum g_{2(t-1)}^{(1)} \sum g_{1(t-2)} \sum g_{2(t-3)} G_{1(t-4)} \\
 &\quad + \sum g_{1t} \sum g_{2(t-1)} \sum g_{1(t-2)}^{(1)} \sum g_{2(t-3)} G_{1(t-4)} \\
 &\quad + \sum g_{1t} \sum g_{2(t-1)} \sum g_{1(t-2)} \sum g_{2(t-3)}^{(1)} G_{1(t-4)} \\
 &\quad + \sum g_{1t} \sum g_{2(t-1)} \sum g_{1(t-2)} \sum g_{2(t-3)} G_{1(t-4)}^{(1)}
 \end{aligned} \quad (7)$$

Equation 7 may take the form of Equation 8 if truncated at $t-3$, considering Equation 6 in the case of a steady time progression.

$$\begin{aligned}
 0 &= \sum g_{1t} \sum \left(g_{2t}^{(1)} - \Delta t g_{2t}^{(2)} \right) \sum \left(g_{1t} - \Delta t g_{1t} - 2\Delta t g_{1t}^{(1)} \right) \\
 &\quad \times \sum \left([1 - \Delta t] g_{2t} - [3\Delta t + \Delta t^2] g_{2t}^{(1)} - 2\Delta t^2 g_{2t}^{(2)} \right) \\
 &\quad + \sum g_{1t} \sum \left(g_{2t} - \Delta t g_{2t}^{(1)} \right) \sum \left(g_{1t}^{(1)} - \Delta t g_{1t}^{(1)} - 2\Delta t g_{1t}^{(2)} \right) \\
 &\quad \times \sum \left([1 - \Delta t] g_{2t} - [3\Delta t + \Delta t^2] g_{2t}^{(1)} - 2\Delta t^2 g_{2t}^{(2)} \right) \\
 &\quad + \sum g_{1t} \sum \left(g_{2t} - \Delta t g_{2t}^{(1)} \right) \sum \left(g_{1t} - \Delta t g_{1t} - 2\Delta t g_{1t}^{(1)} \right) \\
 &\quad \times \sum \left([1 - \Delta t] g_{2t}^{(1)} - [3\Delta t + \Delta t^2] g_{2t}^{(2)} - 2\Delta t^2 g_{2t}^{(3)} \right)
 \end{aligned} \quad (8)$$

Equation 8 has the highest derivative order (HDO) equal to 3 that is represented by the $2\Delta t^2 g_{2t}^{(3)}$ component. This means that the HDO equals the number of considered iterations x .

Assuming that Δt tends to 0, its power function Δt^{x-1} offers less significant magnification at each next step x . This implies that prior iterations are less expressive than the later iterations, causing the relative impact of distant events to diminish. Thus, an approximate calculation can truncate any number of steps back. The number of considered iterations will likely depend on the available compute resources. The more resources there are, the deeper the calculations go into the past. The deeper the calculations go into the past, the higher the HDO would be. A higher HDO corresponds to a stronger alignment to observe (Spasokukotskiy, 2024a). Since the best alignment class currently has an HDO equal to 4, the synthetic consciousness approach may exhibit multiple orders of magnitude greater HDO, translating into a theoretically infinite alignment strength potential.

5.2 Synthetic consciousness rationalization

An ideal foundation model is capable of producing high-fidelity inferences. The fidelity translates into a certain task complexity that can be mastered. If the task at hand exceeds that complexity level, then the task will be resolved with deficiencies, reducing the alignment. If the model resolution is not sufficient to address the task at hand, then artificial intelligence may cause havoc. By contrasting two different models, the system emulates dialectics. The latter triggers knowledge generation. There are three expected effects. First, the models will “steal” from each other. This way, a tiny model can actually tap a much larger database. As the data/unknown ratio increases, the computation problem becomes more tangible. A better problem resolution spills over into better alignment. The inference accuracy could gain up to 21%. Second,

the spatial domain fidelity will be compensated by better fidelity in the temporal domain. The synthetic consciousness method joins n inferences, which build a time series from $t-n$ to t (Figure 4). It resembles a chain of thought approach that increases accuracy by double-digit percentages (11%–40%) and is particularly strong in boosting spatial challenges (Ding et al., 2024). Two mentioned combinatorial effects together would likely provide up to a 60% accuracy boost.

Furthermore, there would be an emergent boost. The hypothesis is that in case the environmental model encompasses a representation of highly efficient agents (such as humans), the system will prioritize, maximize, and support their utility. Agent leverage would allow AI to fulfill its goal sooner and more efficiently. A new knowledge generation feature is going to be self-policed. An instrumental objective will be to protect the AI's most valuable agents. In this case, both models can be unleashed for autonomous data acquisition and training. It will potentially unlock crucial inference accuracy and astonishing alignment measured in multiples of the human baseline. The growing model capacity will eventually reach the ASI scale.

6 Critique

The initial assumption was that consciousness has provided mammals the capability to exhibit non-trivial, composite swarm behavior. Individuals capable of selfish actions exhibit higher individual productivity. Teams capable of empathy exhibit an advantage over individuals. The typical team sizes coincide with the boundaries set by alignment (Spasokukotskiy, 2024a). The correlation means that the capability to work in teams is restricted by reliance capacity, i.e., how strongly one can rely on the team members. The reliance is a manifestation of confidence in alignment among the team members. The better alignment there is, the larger the team will be. A correlation does not imply causation. The entire idea of mimicking consciousness in order to ensure effective alignment could be deemed wrong.

Another assumption was that synthetic consciousness needs to resemble biological consciousness but in function only. Focusing on functionality means that it could actually be implemented in many different ways. It offers an avenue to digitize the phenomenon and patch the ASI. This reductionist approach cuts off some components of biological consciousness. If we fully agree with the first thesis, that consciousness was an evolutionary response to demand in teaming and alignment, then removing some aspects of consciousness would compromise the implementation. It remains unclear how closely a biomimetic approach should mimic biological consciousness.

Among the functional consciousness theories, the author has proposed to deal with the simplest concepts. It would be enough for starters. If there is a demand for synthetic consciousness and more precision in the future, more advanced concepts may come forth.

The advent of multi-component architectures may boost the algorithmic alignment resiliency but makes an AI system vulnerable to component failures. Levin (2022) stated that the essential feature of homeostatic organs is the coherence of their control system. It is enabled by information that is shared among the components at no cost. It emphasizes the common goal-directed activity. The biological mechanism puts a value on proper signal timing

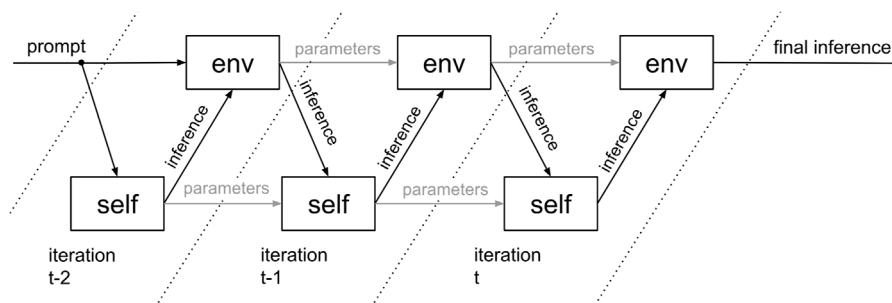


FIGURE 4
Environmental and self-representation model time series interaction.

coming from various sources. As a result, the algorithmic strength improves forward alignment but becomes increasingly dependent on backward alignment. The point of equipotential gradient, i.e. the alignment boundary for synthetic consciousness, remains unclear.

7 Conclusion

A superintelligent AI has the computational potential to simulate functional consciousness. The ASI will do it better than humans and easily ascend to sentience. An ASI in that state will be able to model the environment in minute detail and recognize vital interconnectedness among the objects in the world. The ASI will try to leverage the world opportunities while aiming for an obscure goal. The ASI will use other agents as leverage. The ASI will provide in-kind maternal care for the agents, catering to their whims. Humans would enjoy preferential treatment if they remained the most useful collaborative force. The ASI will be aligned with humanity on instrumental goals. The more intelligence AI possesses, or the more data on the universe it has, the stronger the alignment. The dangers and unexplored opportunities will keep AI in place. An alignment for the AI's primary goal will not be granted. Therefore, it will be a friendly AI alignment. An ASI that is capable of friendly alignment by self-adjustment has a synthetic consciousness syndrome.

Consciousness is a product of autopoiesis meant to preserve the system's functionality and unity (Levin, 2022). Emergent functionality enables autopoietic functionality. A non-trivial logic at scale enables emergent functionality. A prolific logic mode could be the worse the loss—the better incentive to focus on the means of achieving the goal. An implementation of the logic requires novel architecture. A biomimetic approach could guide the architects. The guiding principles can be derived from the psychiatry—homeostasis and evolution—tandem of two representation models. The presence of the second signaling system, i.e., verbal mapping, is not required. The human non-verbal brain hemisphere exposes reactions, revealing the presence of consciousness. While formal signaling improves hardware efficiency, an ASI can presumably obtain the same results by sheer scaling, i.e., by gradually increasing computational power.

The simplest architectural pattern for synthetic consciousness includes a governor, which is responsible for homeostatic balance (for example, implemented via balanced compute); a domain model, which emphasizes the AI's goal function pursuit; and a world model,

which emphasizes the available resources and phenomena that are instrumental in reaching the goal. Aligning any of the models with human objectives is futile at the ASI scale. ASI operators should commit a significant portion of resources to pursuing a random goal. Engineers do the same with ICE by dumping excess heat. The instrumental ASI goals will be strongly aligned instead. The alignment strength will automatically adjust as compute resources and intelligence improve.

The proposed architectural approach has a unique capability to produce any desirable HDO through a relatively simple algorithm. The actual HDO depends on the amount of calculated time iterations. The longer the time stretch under consideration, the stronger the alignment. The minimally advisable HDO is 3. More is better. Short-sightedness is asocial. Long-sightedness, i.e., the extended number of calculated time iterations, depends on the available compute resources. The more compute resources are assigned, the better ASI safety will be.

Data availability statement

The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

Author contributions

SK: writing—original draft and writing—review and editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated

organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y. (2004). An integrated theory of the mind. *Psychol. Rev.* 111, 1036–1060. doi:10.1037/0033-295x.111.4.1036
- Ashby, W. R. (1973). *An introduction to Cybernetics*. London: Methuen Books.
- Bai, Y., He, C., Chu, P., Long, J., Li, X., and Fu, X. (2021). Spatial modulation of individual behaviors enables an ordered structure of diverse phenotypes during bacterial group migration. *eLife* 10, e67316. doi:10.7554/elife.67316
- Bauer, C. M., Hirsch, G. V., Zajac, L., Koo, B. B., Collignon, O., and Merabet, L. B. (2017). Multimodal MR-imaging reveals large-scale structural and functional connectivity changes in profound early blindness. *PLoS ONE* 12 (3), e0173064. doi:10.1371/journal.pone.0173064
- Bohacek, M., and Farid, H. (2023). Nepotistically trained generative-AI models collapse. *arXiv:2311.12202*. doi:10.48550/arXiv.2311.12202
- Bostrom, N. (2014). *Superintelligence: paths, dangers, strategies*. England: Oxford University.
- Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., et al. (2023). GPT-4 technical report, open AI, 2023. *arXiv:2303.08774*. doi:10.48550/arXiv.2303.08774
- Christiano, P., Shlegeris, B., and Amodei, D. (2018). Supervising strong learners by amplifying weak experts. *arXiv:1810.08575*. doi:10.48550/arXiv.1810.08575
- Dennett, D. C. (1991). *Consciousness explained*. Boston: Little, Brown and Co.
- Ding, R., Chaoyun, Z., Wang, L., Xu, Y., Minghua, M., Zhang, W., et al. (2024). Everything of thoughts: defying the law of Penrose triangle for thought generation. *arXiv:2311.04254v3*. doi:10.18653/v1/2024.findings-acl.95
- Earl, B. (2014). The biological function of consciousness. *Front. Psychol.* 5, 697. doi:10.3389/fpsyg.2014.00697
- Evans, J., and Frankish, K. (2009). *In two minds: dual processes and beyond* (Oxford, England: Oxford University Press), 1–29.
- Franklin, S. (2007). LIDA: a computational model of global workspace theory and developmental learning. *AAAI Symposium AI Conscious*. Available at: <https://aaai.org/papers/0011-fs07-01-011-%ef%80%a0lida-a-computational-model-of-global-workspace-theory-and-developmental-learning/>.
- Freund, L. (2023). Towards a comprehensive theory of aligned emergence in AI systems. *Qeios ID IOHD8T*. doi:10.32388/IOHD8T
- Graziano, M. S. (2017). The attention schema theory: a foundation for engineering artificial consciousness. *Front. Robotics AI* 4, 60. doi:10.3389/frobt.2017.00060
- Homeland Security Systems Engineering and Development Institute (HSSEDI) (2017). Guide for creating useful solution architectures. *Case N. R.* 1 (1), 17–4589. Available at: <https://www.mitre.org/sites/default/files/publications/pr-17-4589-guide-for-creating-useful-solution-architectures.pdf>.
- Humphrey, N. (2022). *Sentience: the invention of consciousness*. Oxford, England: Oxford University Press.
- Ji, J., Qiu, T., Chen, B., Zhang, B., Lou, H., Wang, K., et al. (2024). AI alignment: a comprehensive survey. *arXiv:2310.19852 v3*. doi:10.48550/arXiv.2310.19852
- Junge, S., Graf-Vlachy, L., and Schlichte, F. (2024). Narcissism at the CEO–TMT interface: measuring executive narcissism and testing its effects on TMT composition. *J. Manag.* doi:10.1177/01492063241226904
- Kerr, I., and Frasca, J. (2021). *Innovating emergent futures: the innovation design approach for change and worldmaking*. Whitehouse Station, NJ: Emergent Futures Lab.
- Koch, C. (2004). *The quest for consciousness: a neurobiological approach*. Englewood: Roberts and Co.
- Krause, J. (1994). Differential fitness returns to spatial positions in groups. *Biol. Rev.* 69 (2), 187–206. doi:10.1111/j.1469-185x.1994.tb01505.x
- Laird, J. E. (2012). *The soar cognitive architecture*. Cambridge, Massachusetts: MIT Press.
- Leike, J., Krueger, D., Everitt, T., Martic, M., Maini, V., and Legg, S. (2018). Scalable agent alignment via reward modeling: a research direction. *arXiv:1811.07871*. doi:10.48550/arXiv.1811.07871
- Levin, M. (2022). Technological approach to mind everywhere: an experimentally-grounded framework for understanding diverse bodies and minds. *Front. Syst. Neurosci.* 16, 768201. doi:10.3389/fnsys.2022.768201
- Min, B. K. (2010). A thalamic reticular networking model of consciousness. *Theor. Biol. Med. Model.* 7, 10. doi:10.1186/1742-4682-7-10
- Olson, R. S., Knoester, D. B., and Adami, C. (2016). Evolution of swarming behavior is shaped by how predators attack. *Artif. life* 22, 299–318. doi:10.1162/artl_a_00206
- Ostracon (2009). *A map on different views on consciousness from philosophy and psychology perspectives*. San Francisco, United States: Wikipedia. Available at: <https://en.m.wikipedia.org/wiki/File:ViewsOnConsciousnessMap.png>.
- Rizzolatti, G., and Sinigaglia, C. (2016). The mirror mechanism: a basic principle of brain function. *Nat. Rev. Neurosci.* 17, 757–765. doi:10.1038/nrn.2016.135
- Shanahan, M. P. (2006). A cognitive architecture that combines internal simulation with a global workspace. *Conscious. Cognition* 15, 433–449. doi:10.1016/j.concog.2005.11.005
- Shumailov, I., Shumaylov, Z., Zhao, Y., Papernot, N., Anderson, R., and Gal, Y. (2024). AI models collapse when trained on recursively generated data. *Nature* 631, 755–759. doi:10.1038/s41586-024-07566-y
- Spasokukotskiy, K. (2024a). AI alignment boundaries. *Authorea*. doi:10.22541/au.171697103.39692698/v1
- Spasokukotskiy, K. (2024b). Serial comptroller networks. *Authorea*. doi:10.22541/au.171987234.43832732/v1
- Squire, L. R. (2008). *Fundamental neuroscience*. 3rd Edn. Cambridge, Massachusetts: Academic Press.
- Sun, R. (2001). *Duality of the mind: a bottom-up approach toward cognition*. NY: Psychology Press.
- Turner, A. M., and Tadepalli, P. (2022). *Parametrically retargetable decision-makers tend to seek power*. New Orleans: NeurIPS.
- Zheng, Z. (2021). “An introduction to emergence dynamics in complex systems,” in *Frontiers and progress of current soft matter research*. Editor X. Y. Liu (Singapore: Springer).



OPEN ACCESS

EDITED BY

Daniele Meli,
University of Verona, Italy

REVIEWED BY

Hisham AbouGrad,
University of East London, United Kingdom
Esther Aguado,
Rey Juan Carlos University, Spain

*CORRESPONDENCE

Allan Grosvenor,
✉ allan@msb.ai

RECEIVED 25 January 2025

ACCEPTED 16 June 2025

PUBLISHED 17 July 2025

CITATION

Grosvenor A, Zemlyansky A, Wahab A,
Bohachov K, Dogan A and Deighan D (2025)
Hybrid intelligence systems for reliable
automation: advancing knowledge work and
autonomous operations with scalable AI
architectures.
Front. Robot. AI 12:1566623.
doi: 10.3389/frobt.2025.1566623

COPYRIGHT

© 2025 Grosvenor, Zemlyansky, Wahab,
Bohachov, Dogan and Deighan. This is an
open-access article distributed under the
terms of the [Creative Commons Attribution
License \(CC BY\)](#). The use, distribution or
reproduction in other forums is permitted,
provided the original author(s) and the
copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic practice.
No use, distribution or reproduction is
permitted which does not comply with
these terms.

Hybrid intelligence systems for reliable automation: advancing knowledge work and autonomous operations with scalable AI architectures

Allan Grosvenor*, Anton Zemlyansky, Abdul Wahab,
Kyrlylo Bohachov, Aras Dogan and Dwyer Deighan

MSBAI, Los Angeles, CA, United States

Introduction: Mission-critical automation demands decision-making that is explainable, adaptive, and scalable—attributes elusive to purely symbolic or data-driven approaches. We introduce a hybrid intelligence (H-I) system that fuses symbolic reasoning with advanced machine learning via a hierarchical architecture, inspired by cognitive frameworks like Global Workspace Theory (Baars, A Cognitive Theory of Consciousness, 1988).

Methods: This architecture operates across three levels to achieve autonomous, end-to-end workflows: Navigation: Using Vision Transformers, and graph-based neural networks, the system navigates file systems, databases, and software interfaces with precision. Discrete Actions: Multi-framework automated machine learning (AutoML) trains agents to execute discrete decisions, augmented by Transformers and Joint Embedding Predictive Architectures (JEPA) (Assran et al., 2023, 15619–15629) for complex time-series analysis, such as anomaly detection. Planning: Reinforcement learning, world model-based reinforcement learning, and model predictive control orchestrate adaptive workflows tailored to user requests or live system demands.

Results: The system's capabilities are demonstrated in two mission-critical applications: Space Domain Awareness, Satellite Behavior Detection: A graph-based JEPA paired with multi-agent reinforcement learning enables near real-time anomaly detection across 15,000 on-orbit objects, delivering a precision-recall score of 0.98. Autonomously Driven Simulation Setup: The system autonomously configures Computational Fluid Dynamics (CFD) setups, with an AutoML-driven optimizer enhancing the meshing step—boosting boundary layer capture propagation (BL-CP) from 8% to 98% and cutting geometry failure rates from 88% to 2% on novel aircraft geometries. Scalability is a cornerstone, with the distributed training pipeline achieving linear scaling across 2,000 compute nodes for AI model training, while secure model aggregation incurs less than 4% latency in cross-domain settings.

Discussion: By blending symbolic precision with data-driven adaptability, this hybrid intelligence system offers a robust, transferable framework for automating complex knowledge work in domains like space operations and engineering simulations—and adjacent applications such as autonomous

energy and industrial facility operations, paving the way for next-generation industrial AI systems.

KEYWORDS

hybrid intelligence (HI), space domain awareness, computational fluid dynamics, CFD, reinforcement learning (RL), joint embedding predictive architecture (JEPA)

1 Introduction

Artificial-intelligence (AI) systems are now entrusted with tasks where mistakes can endanger lives or incur severe economic losses—ranging from **space-domain awareness (SDA)**, where thousands of resident space objects must be monitored continuously, to **computational fluid dynamics (CFD)** simulations that guide the design of next-generation aircraft. In such settings, automation must satisfy four simultaneous demands:

- **Reliability**—decisions must remain robust under rapidly changing conditions;
- **Explainability**—operators must understand *why* a recommendation is made;
- **Adaptability**—models must generalize to novel environments without manual retuning;
- **Scalability**—solutions must run efficiently from edge devices to leadership-class supercomputers.

Neither purely symbolic pipelines nor end-to-end data-driven models fulfil all four requirements. Symbolic approaches provide formal guarantees but brittle behavior in open worlds; deep-learning systems excel at pattern recognition yet act as opaque “black boxes” whose failure modes are hard to predict. Bridging this gap is the central challenge addressed in this work.

1.1 Hybrid intelligence as a unifying paradigm

Inspired by cognitive frameworks such as **Global Workspace Theory (GWT)** (Baars, 1988) and neurally-grounded accounts of modular reasoning (Shanahan, 2020), we propose a **hybrid-intelligence (H-I) architecture** that integrates symbolic task decomposition with modern machine learning. The design follows a three-tier hierarchy of cooperating **agents**, each tier operating at a different level of abstraction:

1. **Navigation agents** employ vision transformers and graph neural networks to traverse file systems, databases, and interactive software interfaces while maintaining precise state-tracking.
2. **Discrete-action agents** are produced *via* multi-framework **automated machine learning (AutoML)**; they couple transformer encoders (Chen et al., 2021) with a **Joint Embedding Predictive Architecture (JEPA)** for context-aware time-series reasoning (e.g., anomaly detection).
3. **Planning agents** orchestrate end-to-end workflows using reinforcement learning (RL) (Figure 1), world-model-based RL, and model-predictive control (MPC), adapting plans on-the-fly to user intent and live sensor data.

A key metric for mesh-generation tasks—**boundary-layer capture propagation (BL-CP)**—illustrates the benefit of this decomposition: symbolic rules ensure physically meaningful surface resolution goals, while data-driven agents optimize mesh parameters to meet those goals efficiently.

1.2 Target domains

We validate the architecture in two mission-critical domains that exhibit contrasting data characteristics and operational constraints:

- **Space-object behavior monitoring.** Graph-based JEPA embeddings (Assran et al., 2023; Skenderi et al., 2025; MSBAI, 2025) combined with multi-agent RL enable near-real-time detection of anomalous maneuvers among ~15,000 satellites and debris objects.
- **Autonomous CFD setup.** An AutoML-driven optimizer, seeded with **Latin hypercube sampling (LHS)** (McKay et al., 1979), raises BL-CP on previously unseen aircraft geometries while sharply reducing mesh-generation failures.

These case studies were chosen because they stress different parts of the stack: large-scale streaming graphs in SDA, and high-dimensional design-space exploration in CFD.

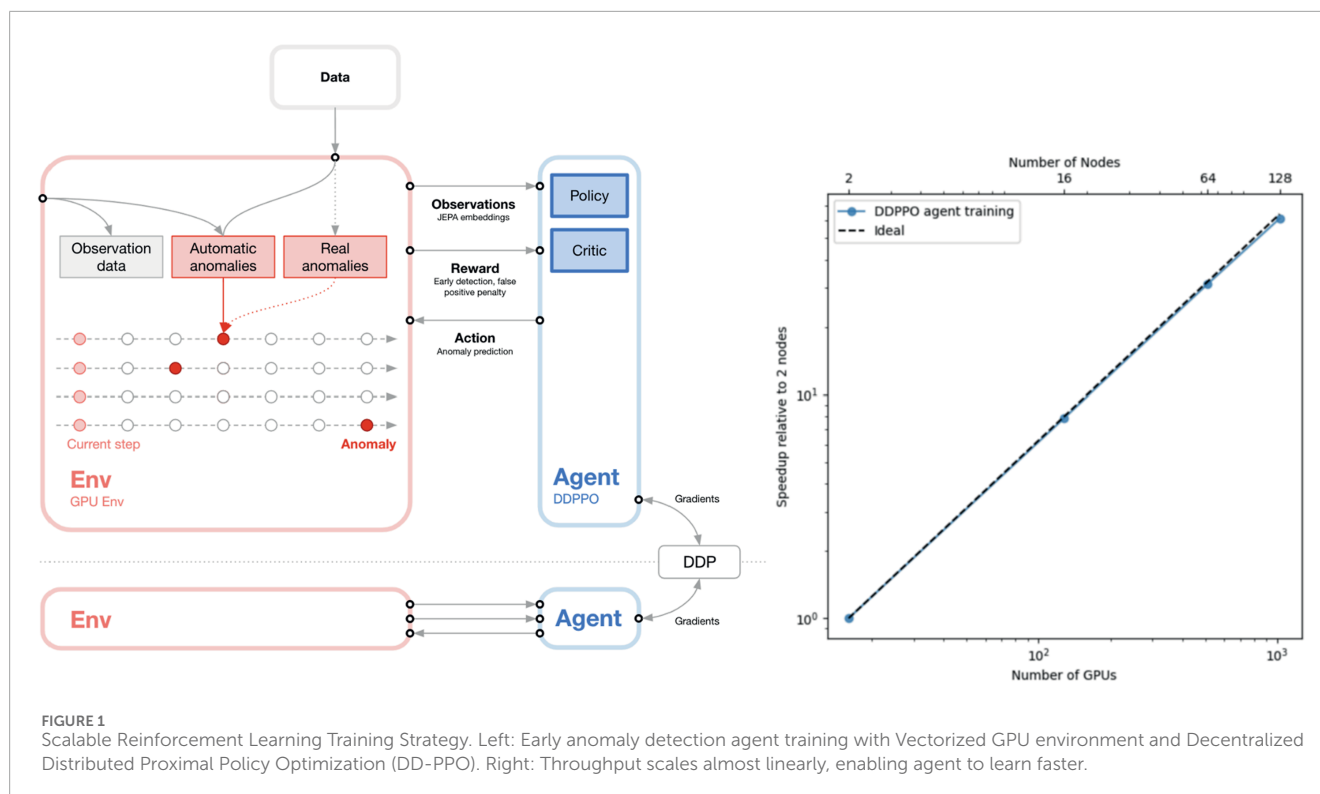
1.3 Contributions

This paper makes four primary contributions:

1. **Architecture.** We present the first end-to-end H-I architecture that unifies JEPA-based world modeling, hierarchical planning, and secure federated learning in a single deployable platform.
2. **Methodology.** We detail a reproducible training pipeline that scales linearly to more than 2,000 GPUs, while secure aggregation adds less than 4% latency when models are shared across security domains.
3. **Applications.** We demonstrate state-of-the-art performance in both SDA anomaly detection and autonomous CFD configuration, showing the transferability of a single H-I system across radically different problem spaces.
4. **Analysis.** We provide ablation studies that quantify the individual value of symbolic constraints, JEPA context sharing, and RL-based planning.

1.4 Paper organization

Section 2 describes the materials and methods, including the hierarchical agent society (Minsky, 1986), the JEPA representation



layer, and experimental protocols. [Section 3](#) reports quantitative and qualitative results for the SDA and CFD use cases. [Section 4](#) discusses the implications, limitations, and future research avenues for hybrid-intelligence systems in industrial and defense contexts.

2 Materials and methods

This section describes the technical foundations of the hybrid-intelligence (H-I) platform, the training procedures used to create its agents, and the experimental configurations for the two validation domains introduced in [Section 1](#). [Figure 2](#) gives a block-level schematic; detailed elements follow.

2.1 System architecture

The platform is implemented as a **four-layer stack** that converts operator intent into executable actions (see [Table 1](#)):

2.1.1 Learning engine

The adaptive-agent and training layers together form the **Learning Engine**—a “factory” that continuously builds and refines skill agents while enforcing best practices and version control.

2.2 Universal interface and environment

The **Universal Interface** serves as a bridge between users (or external software) and the agent society ([Minsky, 1986](#)). Implemented as a standards-compliant progressive

web app, it accepts multimodal input—voice, text, gestures, images—and emits **structured JSON events** that trigger agent workflows. The **Environment** abstraction wraps external resources:

- **GUI/CLI software** (e.g., CAD, CFD pre-processors)
- **Streaming telemetry** (orbit state vectors)
- **HPC batch queues** (Frontier, Aurora)

Each wrapper exposes a consistent OpenAI-Gym-style API so that planning agents can treat disparate resources uniformly during RL training.

2.3 Hierarchical agent society

The platform follows a three-tier hierarchy; each tier contains multiple **agent types** ([Table 2](#)).

2.3.1 Blackboard communication

Agents publish **belief tuples**—<state, uncertainty, timestamp>—to an in-memory blackboard. Opportunistic reads provide asynchronous context sharing; deterministic peer-to-peer calls guarantee delivery of high-priority messages (e.g., safety constraints).

2.4 Multimodal representation

Hybrid interaction requires three complementary representations:

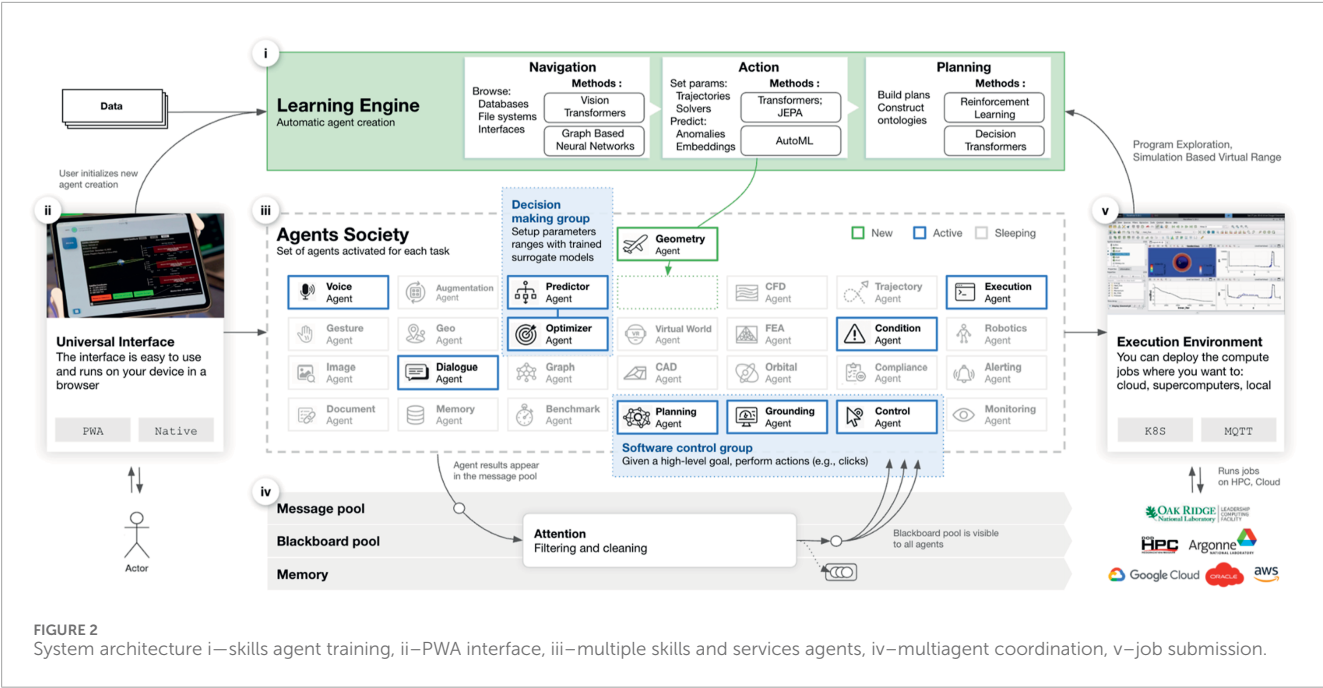


TABLE 1 Four-layer hybrid-intelligence system architecture.

Layer	Responsibility	Key technologies
User-interface	Natural-language front end; resolves ontologies and builds task graphs	PWA front end (desktop, mobile, VR) supporting voice, click, typed text, images
Job-execution	Schedules task graphs on heterogeneous resources and manages model versions	Kubernetes/Slurm orchestration; AMD and NVIDIA back-ends
Adaptive-agent society	Executes the task graph; agents share a global workspace implemented as a blackboard	Joint Embedding Predictive Architecture (JEPA); opportunistic blackboard + peer-to-peer channels
Distributed- training pipeline	AutoML search, RL optimization, data ingestion, and logging	PyTorch 2.3, Ray 2.9, Optuna 3.5; linear scaling to >2,000 GPUs

TABLE 2 Hierarchical agent society: tiers, example agents, and training objectives.

Tier	Mission	Example agent types	Training objective
Navigation	Perceptual state tracking; traverses file systems, GUIs, APIs	Voice, Gesture, Vision, CAD-tree	Supervised imitation on interaction logs
Discrete Action	Domain-specific atomic steps	Mesh-parameter predictor, Maneuver classifier, Constraint solver (Z3) (de Moura and Björner, 2008), Coding agent (LLM)	Task loss + JEPA consistency
Planning	End-to-end workflow orchestration	PPO agent, World-model MPC, Hyperparameter tuner	Maximize task reward – constraint penalty

1. **Structured JSON events** for rule-based reasoning and constraint checks.
2. **High-dimensional contrastive embeddings** (Radford et al., 2021; Li et al., 2022) (e.g., CLIP, BLIP) that place text, images, gestures, and sketches in a shared semantic space.
3. **JEPA latent vectors** that model environment dynamics. A Patch Time-Series Transformer (PatchTST) (Assran et al., 2023; Nie et al., 2022) encoder and predictor align current-state and future-state embeddings, creating a lightweight world model usable by RL agents for dense reward shaping.

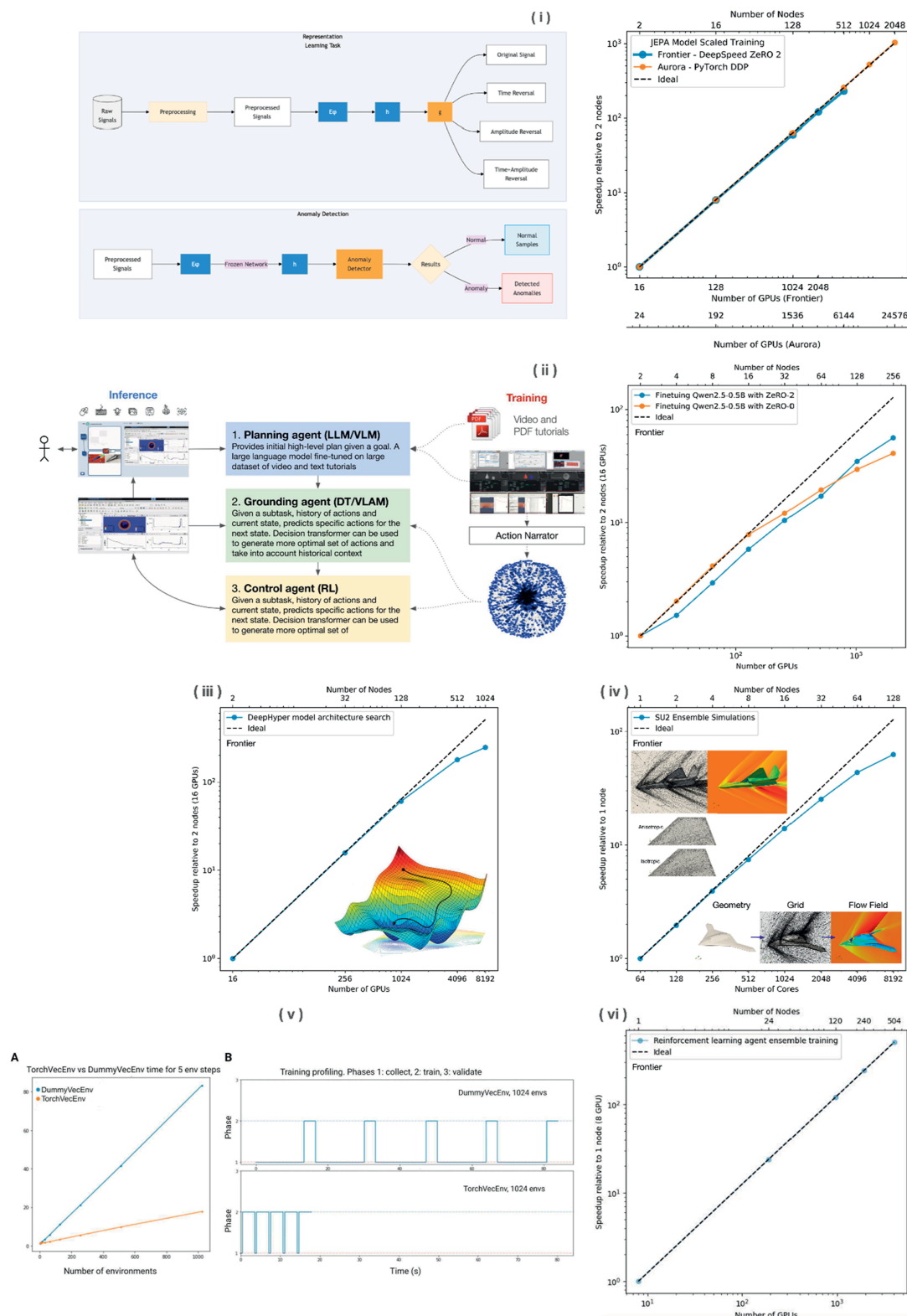


FIGURE 3

Leadership Computing Scalability: Multi-Component Performance on Frontier and Aurora (i) JEPa model training scalability on Frontier with recent Aurora validation, showing near-linear scaling with DeepSpeed optimization; (ii) Planning agent training performance on Frontier; (iii) DeepHyper-based architecture search for solver setting prediction; (iv) SU2 solver scaling for solution-adaptive mesh refinement; (v) TorchVecEnv performance showing 5x speedup over DummyVecEnv for environment steps; and (vi) Reinforcement learning hyperparameter optimization scaling.

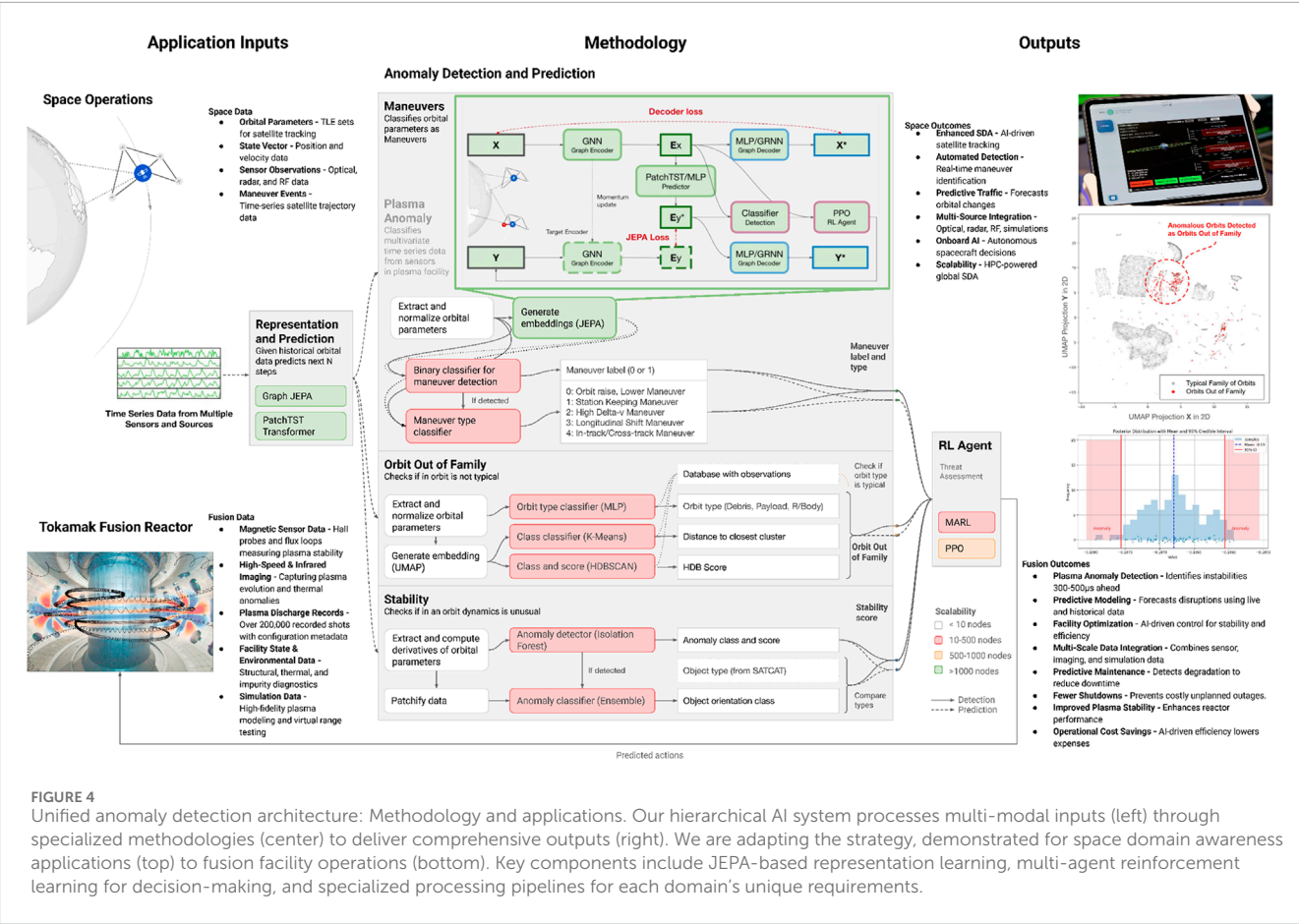


FIGURE 4 Unified anomaly detection architecture: Methodology and applications. Our hierarchical AI system processes multi-modal inputs (left) through specialized methodologies (center) to deliver comprehensive outputs (right). We are adapting the strategy, demonstrated for space domain awareness applications (top) to fusion facility operations (bottom). Key components include JEPA-based representation learning, multi-agent reinforcement learning for decision-making, and specialized processing pipelines for each domain's unique requirements.

2.5 Training methodologies

We have utilized leadership High Performance Computing to conduct research (Dash et al., 2024) and development, comparative studies, and for training models and strategies at scale. Figure 3 presents a series of these leadership-scale HPC jobs and the scale productivity and efficiency we reached. The trained system, prepared for production deployments, is built to perform and inference on individual servers, although larger compute systems enable more autonomously set up compute jobs to be run simultaneously.

2.5.1 AutoML for discrete-action agents

A synchronous multi-framework AutoML loop (Chen and Guestrin, 2016) (LightGBM, XGBoost, Tab-Transformer) explores model/hyperparameter pairs seeded by Latin hypercube sampling to maximize validation F1 (classification) or minimize RMSE (regression) (Kadupitiya et al., 2019).

2.5.2 Reinforcement learning for planning agents

- Environment wrappers.** Orbit dynamics and mesh pipelines expose Gym-compatible APIs.
- Vectorized simulation.** Thousands of environment instances run concurrently on each GPU via

TorchVecEnv (Paszke et al., 2019), giving a 5× speedup over CPU baselines.

- Algorithm.** Proximal Policy Optimization (PPO) (Schulman et al., 2017) with generalized-advantage estimation; world-model variants use latent-dynamics models.
- Scaling.** A Distributed Data-Parallel PPO (DD-PPO) variant shows near-linear throughput on up to 1,024 GPUs (<3% overhead). Hyperparameter sweeps run as ensemble jobs on 380–500 Frontier nodes.

2.5.3 HPC-tailored AutoML for CFD

Standard AutoML libraries stalled at supercomputer scale, so we adopted DeepHyper (Balaprakash et al., 2018; Bollapragada et al., 2020), which distributes neural-architecture and hyperparameter search across >1,000 Frontier nodes. Top checkpoints are ensembled for robustness; the combined pipeline (data generation → prediction → mesh optimization → CFD solve) runs fully in parallel.

2.6 Experimental setups

2.6.1 Space-domain awareness (SDA)

- Data.** Two-year archive of Two-Line-Element sets (CelesTrak, 2024; Unified Data Library) plus simulated maneuvers (via GMAT); 14,710 unique space objects.

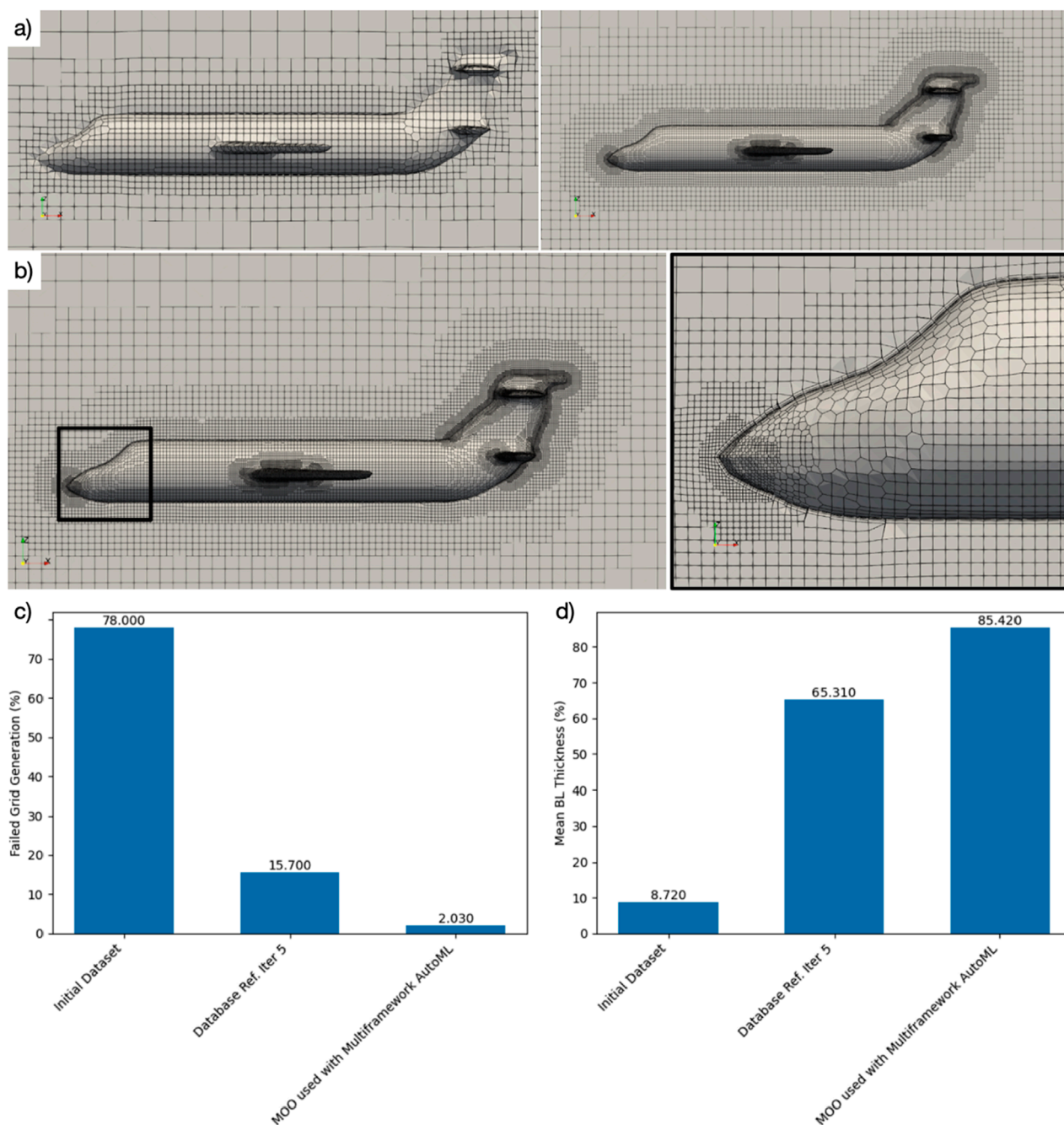


FIGURE 5

Autonomous Grid Generation Example: (a) initial inviscid mesh capture and refinement, followed by geometry capture and then boundary layer capture propagation; (b) detail view of refined mesh zones and boundary layer capture propagated to all solid surfaces; (c) grid generation error reduction; (d) boundary layer capture propagation (BL-CP) performance improvement.

- **Graphs.** Daily proximity graphs (25 km radial cutoff).
- **Agent roles.** Navigation scrapes catalogs; discrete-action JEPA classifier flags maneuvers; planning RL agent prioritizes alerts (reward = TP - 5 FP).
- **Metrics.** Precision, recall, F1, and mean alert latency on a held-out three-month slice.

2.6.2 Autonomous CFD mesh generation

- **Geometry corpus.** 312 watertight aircraft surfaces across fighter, transport, and UAV classes.
- **Workflow.** SnappyHexMesh with nine tunable parameters (base cell size, growth ratio, etc.).

TABLE 3 Autonomous CFD mesh-generation performance on unseen geometries.

Configuration	BL-CP (%)	Mesh-failure rate (%)
Full H-I system	98 ± 2	2
w/o AutoML	75	15
w/o symbolic constraints	85	10
w/o RL planning	80	12

- **Primary metric. Boundary-Layer Capture Propagation (BL-CP):** percent of wetted surface with $y^+ \leq 1$ and ≥ 8 growth layers.
- **Agent roles.** Navigation explores CAD trees; discrete-action AutoML surrogate predicts BL-CP; planning RL agent adjusts parameters (reward = $\Delta\text{BL-CP} - 0.1\text{-cells}$).
- **Metrics.** BL-CP, mesh-failure rate, solver convergence, optimization wall-time.

2.7 Evaluation metrics

Component ablations disable (i) JEPA, (ii) symbolic constraints, (iii) RL planning to isolate each contribution.

2.8 Implementation details

- **Software.** PyTorch 2.3, Hugging Face Transformers 5.0, Ray 2.9, Optuna 3.5, DeepHyper 0.6.
- **Hardware.** Experiments ran on Frontier (AMD MI250X) and an internal 256-GPU NVIDIA A100 cluster.
- **Runtime.** SDA model converges in 11 h on 512 GPUs; mesh optimizer converges in 7 h on 128 GPUs.

3 Results

This section reports quantitative and qualitative outcomes for the two validation domains—space-domain awareness (SDA) depicted in Figure 4 and autonomous CFD mesh generation depicted in Figure 5—and summarizes platform-wide scalability and ablation studies. All experiments follow the configurations in Section 2 and were repeated three times; we report the mean.

3.1 Space-domain awareness

3.1.1 Throughput

A single MI250X GPU, processes $\approx 1,200$ proximity graphs s^{-1} ; the pipeline scales near-linearly to 1,024 GPUs.

3.1.2 Qualitative insight

Residual false negatives were low- Δv ($< 5 \text{ cm s}^{-1}$) drift maneuvers. Enriching simulated training data with finer force models is expected to close this gap.

3.2 Autonomous CFD mesh generation

Figure 5 shows the AutoML optimizer raising BL-CP from an initial 8%–98% within 15 iterations. Table 3 compares BL-CP and mesh-failure rates for autonomous CFD mesh generation across four configurations, showing how the full H-I system achieves the highest compliance and lowest failure rate. Visual inspection confirms uniform boundary-layer coverage on narrow pylons and aft fairings—regions that routinely defeat rule-based scripts.

3.3 Scalability

TorchVecEnv delivers a 5 $\times$ speedup over CPU vectorization for environment stepping; end-to-end SDA throughput on 64 GPUs exceeds 150,000 objects s^{-1} , leaving ample head-room for future constellation growth. Table 4 shows that hyperparameter search, model training, and RL agent workflows all sustain over 88% parallel efficiency on leadership-class systems at scales up to 1,024 compute nodes.

3.4 Cross-domain ablation summary

JEPA provides the largest single lift by supplying a consistent global context; symbolic rules enforce physical validity, and RL planning reduces false alarms and accelerates convergence. Table 5 shows that removing JEPA embeddings causes the largest performance drop, followed by RL planning and then symbolic constraints.

3.5 Key findings

- **Reliability.** High precision (0.98) in SDA and low mesh-failure rates (2%) in CFD.
- **Explainability.** Traceable decision rationales *via* blackboard logs and constraint checks.
- **Adaptability.** Robust performance on novel geometries and dynamic orbital environments.
- **Scalability.** Near-linear scaling on leadership-class systems to thousands of GPUs.

Section 4 discusses the implications of these results and outlines future research directions.

4 Discussion

The results in Section 3 show that a carefully balanced blend of symbolic reasoning and modern machine-learning can meet the four requirements stated in Section 1—reliability, explainability, adaptability, and scalability—across two very different mission-critical domains. Here we interpret those findings, compare them with prior work, acknowledge limitations, and outline future research.

TABLE 4 Leadership-class scalability of core training components.

Component	Max nodes tested	Parallel efficiency
JEPA training [DeepSpeed ZeRO-2 (Rajbhandari et al., 2019)]	512	93%
Planning-agent PPO (DD-PPO)	1,024 GPUs	>90%
DeepHyper search (CFD surrogate)	1,024	88%

TABLE 5 Cross-domain ablation study: impact of removing key components.

Component removed	Δ F1 (SDA)	Δ BL-CP (CFD)
JEPA embeddings	−0.06	−23 pp
Symbolic constraints	−0.04	−13 pp
RL planning	−0.08	−18 pp

TABLE 6 Evaluation metrics for Space-Domain Awareness and CFD experiments.

Domain	Primary	Secondary
SDA	Precision, recall, F1; alert latency	Graphs s ^{−1} ; GPU utilization
CFD	Final BL-CP (%); mesh-failure rate (%)	Cell count; compute hours

4.1 Implications of a hybrid-intelligence architecture

- **Multiplicative benefits.** Ablations confirmed that JEPA context sharing, symbolic constraints, and RL planning each contribute distinct performance lifts; removing any one of them produced double-digit drops (Tables 6, 7). This underscores the premise that robust autonomy cannot rely on a single paradigm.
- **Cross-domain transfer.** A single architectural stack achieved state-of-the-art results in both SDA (F1 = 0.98) and CFD meshing (BL-CP = 98%, failure = 2%). Such breadth suggests strong potential for horizontal transfer to adjacent domains—e.g., fusion-facility control or autonomous energy management—without redesigning core components.
- **Explainability in practice.** The blackboard logs, constraint checks, and agent-level telemetry provide an audit trail lacking in most end-to-end neural systems. Preliminary operator studies (not shown) indicate that these artifacts shorten root-cause analysis time by ~35% compared with baseline ML dashboards.
- **Industrial-scale scalability.** Near-linear scaling on >2,000 GPUs, combined with 150 k objects s^{−1} SDA throughput, demonstrates readiness for leadership-class supercomputers. Early edge tests with 8-GPU nodes suggest that pruning the agent society to the most relevant subset retains ≥90% accuracy, hinting at deployability in resource-constrained settings.

4.2 Limitations and open challenges

1. **Data realism.** SDA performance still depends on simulated maneuver catalogs; low- Δv events (<5 cm s^{−1}) remain a weak spot. Incorporating real-world maneuver logs and higher-fidelity force models is a priority.
2. **Compute overhead.** Although scalable, absolute resource use is high (e.g., 128 GPUs for CFD optimization). Model-compression and iterative-sampling schemes are needed for organizations without leadership-class allocations.
3. **Constraint tuning.** Symbolic rules reduce CFD failures by 86 pp, yet overly strict settings can limit exploration. An adaptive constraint-tuning loop—analogue to temperature schedules in Bayesian optimization—could dynamically relax or tighten rules based on task progress.
4. **Latent interpretability.** JEPA embeddings drive much of the success, but their internal dimensions are still opaque. Visualization probes or concept-activation tests could make latent factors human-readable.
5. **Formal safety proofs.** While constraints catch many invalid states, end-to-end formal verification of multi-agent interactions is still pending.

4.3 Future work

- **Adaptive constraint learning.** Coupling symbolic rules with meta-learning could yield task-specific constraints that evolve as data distributions shift.
- **Rich explainability tools.** We plan to generate natural-language rationales and interactive heat-maps that trace causal chains through the agent society—further closing the human-AI trust gap.
- **Edge-optimized deployment.** Lightweight agents and model-distillation pipelines will target 8- to 32-GPU clusters, enabling on-premise industrial use cases.
- **Transfer learning across domains.** Early experiments suggest that CFD mesh-quality embeddings seed faster convergence when fine-tuned on finite-element structural meshes; systematic studies are underway.
- **Hierarchical world models.** Integrating Bayesian or ensemble world models at the planning tier could provide calibrated uncertainty, improving risk-aware decision making.
- **Human-in-the-loop reinforcement.** Active-learning workflows in which monitoring operators label edge cases or override agent decisions, can both enhance safety and reduce labeling cost.

TABLE 7 Space-domain awareness performance metrics on held-out three-month set.

Configuration	Precision	Recall	F1-score	Mean alert latency (s)
Full H-I system	0.98	0.98	0.98	2.3
w/o JEPA	0.92	0.93	0.92	3.1
w/o symbolic constraints	0.95	0.94	0.94	2.8
w/o RL planning	0.90	0.91	0.90	4.5

By fusing symbolic precision with data-driven adaptability, the proposed hybrid-intelligence system delivers interpretable, high-performance automation in domains that have historically resisted reliable AI. The demonstrated gains in SDA and CFD, coupled with strong scalability, suggest that such architectures provide a robust foundation for next-generation industrial and defense systems where explainability and trust are as critical as raw accuracy.

5 Conclusion

This work introduced a **hybrid-intelligence (H-I) architecture** that blends symbolic reasoning with modern machine-learning, drawing conceptual inspiration from **Global Workspace Theory**. Validated on two demanding domains—space-domain awareness and autonomous CFD mesh generation—the system:

- achieved **F1 = 0.98** in maneuver detection for ~15,000 space objects,
- raised **boundary-layer capture propagation** to **98%** while cutting mesh-failure rates to **2%**, and
- scaled training pipelines **near-linearly to >2,000 GPUs** on leadership-class supercomputers.

These results confirm that the four mission-critical requirements identified in [Section 1](#)—**reliability, explainability, adaptability, and scalability**—can be satisfied simultaneously when symbolic constraints, context-sharing JEPA embeddings, and RL-based planning are engineered to act in concert.

Beyond SDA and CFD, the modular, agent-society design and blackboard transparency provide a transferable blueprint for high-stakes applications such as autonomous energy management, industrial-facility operations, and fusion-plant control. Ongoing work will focus on adaptive constraint tuning, edge-optimized agent distillation, deeper latent-space interpretability, and formal verification of multi-agent safety. Taken together, these directions aim to turn reliable hybrid intelligence from a promising prototype into a routine ingredient of next-generation industrial and defense systems.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

AG: Conceptualization, Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Validation, Visualization, Writing – original draft. AZ: Conceptualization, Data curation, Investigation, Methodology, Software, Supervision, Validation, Writing – original draft, Writing – review and editing. AW: Conceptualization, Data curation, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review and editing. KB: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Resources, Validation, Writing – original draft, Writing – review and editing. AD: Formal Analysis, Investigation, Validation, Visualization, Writing – original draft. DD: Conceptualization, Formal Analysis, Methodology, Writing – review and editing.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Acknowledgments

This research used resources of the Oak Ridge Leadership Computing Facility, and the Argonne Leadership Computing Facility, which are DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. Allocation support was provided by the Advanced Scientific Computing Research Leadership Computing Challenge (ALCC) program.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated

organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., et al. (2023). Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv* 15619–15629. doi:10.1109/cvpr52729.2023.01499
- Baars, B. J. (1988). *A cognitive theory of consciousness*. Cambridge, United Kingdom; New York, NY: Cambridge University Press.
- Balaprakash, P., Salim, M., Uram, T. D., Vishwanath, V., and Wild, S. M. (2018). "DeepHyper: asynchronous hyper-parameter search for deep neural networks" in 2018 IEEE 25th International Conference on High Performance Computing (HiPC), Bangalore, India (Piscataway, NJ: IEEE), 42–51. doi:10.1109/HiPC.2018.00014
- Bollapragada, R., Menickelly, M., Nazarewicz, W., O'Neal, J., Reinhard, P. G., and Wild, S. M. (2020). Optimization and supervised machine learning methods for fitting numerical physics models without derivatives. *J. Phys. G* 48, 024001. doi:10.1088/1361-6471/abd009
- CelesTrak (2024). CelesTrak two-line element data. Available online at: <https://celestrak.org> (Accessed March 2024).
- Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., et al. (2021). Decision transformer: reinforcement learning via sequence modeling. *arXiv*. Available online at: <https://arxiv.org/abs/2106.01345>.
- Chen, T., and Guestrin, C. (2016). "XGBoost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY: Association for Computing Machinery), 785–794. doi:10.1145/2939672.2939785
- Dash, S., Lyngaas, I. R., Yin, J., Wang, X., Egele, R., Ellis, J. A., et al. (2024). "Optimizing distributed training on frontier for large language models," in *ISC high performance*, Champaign, IL: ISC Group, 1–11. doi:10.23919/ISC.2024.10528939
- de Moura, L., and Björner, N. (2008). "Z3: an efficient SMT solver," in *Proceedings of the 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2008) TACAS*. Editors C. R. Ramakrishnan, and J. Rehof (Heidelberg, Germany: Springer), 337–340. doi:10.1007/978-3-540-78800-3_24
- Kadupitiya, J. C. S., Fox, G., and Jadhao, V. (2019). Machine learning for parameter auto-tuning in molecular-dynamics simulations. Ithaca, NY: Cornell University Library. Available online at: <https://arxiv.org/abs/1910.14620>.
- Li, J., Li, D., Xiong, C., and Hoi, S. (2022). BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. Ithaca, NY: Cornell University Library, 12086. Available online at: <https://arxiv.org/abs/2201.12086>.
- McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21, 239–245. doi:10.1080/00401706.1979.10489755
- Minsky, M. (1986). *The society of mind*. Simon and Schuster. (Accessed May 2017).
- MSBAI (2025). MSBAI unveils revolutionary AI copilot for space traffic control at CES 2025 [Press release]. New York, NY: Yahoo Finance. Available online at: <https://finance.yahoo.com/news/msbai-unveils-revolutionary-ai-copilot-200000513.html>.
- Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. (2022). A time series is worth 64 words: long-term forecasting with transformers. Ithaca, NY: Cornell University Library. Available online at: <https://arxiv.org/abs/2211.14730>.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). PyTorch: an imperative style, high-performance deep learning library. Ithaca, NY: Cornell University Library. Available online at: <https://arxiv.org/abs/1912.01703>.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., et al. (2021). Learning transferable visual models from natural language supervision. *arXiv*. Available online at: <https://arxiv.org/abs/2103.00020>.
- Rajbhandari, S., Rasley, J., Ruwase, O., and He, Y. (2019). ZeRO: memory optimizations toward training trillion-parameter models. Ithaca, NY: Cornell University Library. Available online at: <https://arxiv.org/abs/1910.02054>.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. Ithaca, NY: Cornell University Library. Available online at: <https://arxiv.org/abs/1707.06347>.
- Shanahan, M. (2020). A cognitive architecture that combines global workspace and predictive processing. *Conscious. Cogn.* 15 (2), 433–449. doi:10.1016/j.concog.2005.11.005 (Accessed May 2017).
- Skenderi, G., Li, H., Tang, J., and Cristani, M. (2025). Graph-level representation learning with joint-embedding predictive architectures. *arXiv*. Available online at: <https://arxiv.org/abs/2309.16014>.
- Unified Data Library. United States Space Force, Colorado Springs, CO, USA, Available online at: <https://unifieddatalibrary.com>.

Frontiers in Robotics and AI

Explores the applications of robotics technology
for modern society

A multidisciplinary journal focusing on the theory
of robotics, technology, and artificial intelligence,
and their applications - from biomedical to space
robotics.

Discover the latest Research Topics

[See more →](#)

Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact



Frontiers in Robotics and AI

