frontiers | Frontiers in Robotics and AI

TYPE Original Research
PUBLISHED 14 September 2022
DOI 10.3389/frobt.2022.969380

OPEN ACCESS

EDITED BY
Dimitrios Kanoulas,
University College London,
United Kingdom

REVIEWED BY
Matteo Parigi Polverini,
Agility Robotics, United States
Siddhant Gangapurwala,
University of Oxford, United Kingdom

*CORRESPONDENCE
Maani Ghaffari,
maanigj@umich.edu

SPECIALTY SECTION
This article was submitted to Field Robotics,
a section of the journal
Frontiers in Robotics and AI

RECEIVED 14 June 2022
ACCEPTED 02 August 2022
PUBLISHED 14 September 2022

CITATION
Ghaffari M, Zhang R, Zhu M, Lin CE, Lin T-Y, Teng S, Li T, Liu T and Song J (2022), Progress in symmetry preserving robot perception and control through geometry and learning.
Front. Robot. AI 9:969380.
doi: 10.3389/frobt.2022.969380

Progress in symmetry preserving robot perception and control through geometry and learning

Maani Ghaffari*, Ray Zhang, Minghan Zhu, Chien Erh Lin, Tzu-Yuan Lin, Sangli Teng, Tingjun Li, Tianyi Liu and Jingwei Song

Computational Autonomy and Robotics Laboratory (CURLY), University of Michigan, Ann Arbor, MI, United States

This article reports on recent progress in robot perception and control methods developed by taking the symmetry of the problem into account. Inspired by existing mathematical tools for studying the symmetry structures of geometric spaces, geometric sensor registration, state estimator, and control methods provide indispensable insights into the problem formulations and generalization of robotics algorithms to challenging unknown environments. When combined with computational methods for learning hard-to-measure quantities, symmetry-preserving methods unleash tremendous performance. The article supports this claim by showcasing experimental results of robot perception, state estimation, and control in real-world scenarios.

KEYWORDS

robot perception, robot control, geometric control, invariant extended Kalman filter, Lie groups, equivariant models, equivariant representation learning, deep learning

1 Introduction

Understanding the underlying principles of intelligence is at the heart of Artificial Intelligence (AI) and its applications for robotics, i.e., embodied AI, towards building a fully adaptive autonomous system capable of operating in the real world. Computational mathematics and intelligence have become a pivot for these fields, given the current advances in hardware. By combining, unifying, and expanding our mathematical and data-driven understanding of these areas of science and research, one can push the boundaries towards a unifying cognitive model that.

1. is robust to challenging environments and behavior modes;
2. takes into account hierarchical semantic knowledge of the scene such as objects and affordances as well as the geometry;
3. possesses sufficient mathematical and computational structures to be exploited for developing efficient and generalizable algorithms;

4. follows compositional principles to assemble integrated models that can produce outcomes bigger than the sum of individual modules.

This work provides an overview of our recent efforts for robot perception and control methods that can leverage structures such as symmetry and data simultaneously. Roughly speaking, symmetry of an object is a motion that leaves it unchanged (Tapp, 2021). For example, consider the sphere $S^2 = \{(x_1, x_2, x_3) \in \mathbb{R}^3 \mid x_1^2 + x_2^2 + x_3^2 = 1\}$. Its symmetry group is the three-dimensional orthogonal group O(3), i.e., the disjoint union of all 3D rotations and reflections. No matter how we rotate the sphere, its shape remains the same. More generally, Lie groups model the continuous symmetry of geometric spaces and are equipped with a natural coordinates system called exponential coordinates. An important consequence of this observation is that we can formulate problems more naturally where the Lie group action commutes with the (data-driven) functional representation of data (Section 2), the state estimation and control error dynamics become independent of the current operating point, and only depend on the desired relative motion (Sections 3 and Section 4), and we can lift multimodal signals, including images and point clouds, to some Lie algebras via equivariant networks (Section 5).

Section 2 presents a nonparametric analytical framework that models semantically labeled point clouds for solving the sensor registration problem (Ghaffari et al., 2019; Clark et al., 2021; Zhang et al., 2021). The framework lifts the data into a Reproducing Kernel Hilbert Space (RKHS), where the inner product structure captures the cross-correlation between two labeled point clouds as functions. This framework is an example of an equivariant model for modeling data where a Lie group transformation acts on these functions to align them.

Section 3 presents a robot state estimation framework using an invariant Kalman filtering (Barrau and Bonnabel, 2017; Barrau and Bonnabel, 2018; Hartley et al., 2020) and deep learning for estimating contact events from multi-modal proprioceptive sensory data (Lin et al., 2022). The novel combination of a geometric filter on Lie groups with deep learning to provide learned contact events without physical sensors show a promising direction on how to integrate real-time deep learning in high-frequency robot state estimation tasks.

Section 4 provides an overview of the error-state Model Predictive Control (MPC) on Lie groups and the stability analysis by a Lyapunov function expressed in the Lie algebra (Teng et al., 2022a; Teng et al., 2022b). We derive the linearized configuration error dynamics and equations of motion in the Lie algebra (tangent space at the identity) that, given an initial condition, are globally valid and independent of the system trajectory. This approach leads to a convex MPC algorithm for the tracking control problem using the linearized error dynamics, which can be solved efficiently using Quadratic Programming (QP) solvers. The proposed controller is validated in experiments on quadrupedal robot pose control and locomotion.

Section 5 presents recent frameworks for equivariant feature learning and their applications in registration and place recognition tasks (Zhu et al., 2022b). We learn an embedding for each input in a feature space that preserves the equivariance property, enabled by recent developments in symmetry-preserving neural networks. Symmetry (or equivariance) in a neural network enables efficient learning (by removing the need for data augmentation), generalization, and a clear connection between the changes in the input and output spaces, i.e., explainability.

Finally, Section 6 provides closing remarks by summarizing our new findings and their impacts on robot perception and control. We also discuss future opportunities enabled by the presented results in this article.

2 RKHS registration for spatial-semantic perception

Point clouds obtained by modern sensors such as RGB-D cameras, stereo cameras, and LIDARs contain up to 300, 000 points per scan at 10–60Hz and rich color and intensity (reflectivity of a material sensed by an active light beam) measurements besides the geometric information. In addition, *deep learning* (LeCun et al., 2015) can provide semantic attributes of the scene as measurements (Long et al., 2015; Chen et al., 2017; Zhu et al., 2019).

Illustrated in Figure 1, the following formulation provides a general framework for lifting semantically labeled point clouds into a function space to solve a registration problem (Ghaffari et al., 2019; Clark et al., 2021; Zhang et al., 2021). Consider two (finite) collections of points, $X = \{x_i\}$, $Z = \{z_j\} \subset \mathbb{R}^3$. We want to determine which element $h \in \mathrm{SE}(3)$, aligns the two point clouds X and $hZ = \{hz_j\}$ the "best." To assist with this, we will assume that each point contains information described by a point in an inner product space, $(\mathcal{I}, \langle\cdot,\cdot\rangle_{\mathcal{I}})$. To this end, we will introduce two labeling functions, $\ell_X\colon X \to \mathcal{I}$ and $\ell_Z\colon Z \to \mathcal{I}$. To measure their alignment, we turn the point clouds, X and Z, into functions $f_X, f_Z\colon \mathbb{R}^3 \to \mathcal{I}$ that live in some RKHS, $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$. The action, $\mathrm{SE}(3) \curvearrowright \mathbb{R}^3$ induces an action $\mathrm{SE}(3) \curvearrowright \mathcal{H}$ by h. $f(x) := f(h^{-1}x)$. Inspired by this observation, we will set $h.f_Z := f_{h^{-1}Z}$.

Problem 1. *The problem of aligning the point clouds can now be rephrased as maximizing the scalar products of f_X and $h.f_Z$, i.e., we want to solve*

$$\arg\max_{h \in \mathrm{SE}(3)} F(h), \quad F(h) := \langle f_X, f_{h^{-1}Z} \rangle_{\mathcal{H}}. \tag{1}$$

2.1 Constructing the functions

For the kernel of our RKHS, $\mathcal{H}$, we first choose the squared exponential kernel k: $\mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$:

$$k(x,z) = \sigma^2 \exp\left(\frac{-\|x-z\|_3^2}{2\ell^2}\right), \tag{2}$$

for some fixed real parameters (hyperparameters) σ and ℓ (the *lengthscale*), and $\|\cdot\|_3$ is the standard Euclidean norm on $\mathbb{R}^3$. This allows us to turn the point clouds to functions via $f_X(\cdot) \coloneqq \sum_{x_i \in X} \ell_X(x_i) k(\cdot, x_i)$ and $f_{h^{-1}Z}(\cdot) \coloneqq \sum_{z_j \in Z} \ell_Z(z_j) k(\cdot, h^{-1} z_j)$. Here $\ell_X(x_i)$ encodes the semantic information, for example LIDAR intensity and image pixel color. $k(\cdot, x_i)$ encodes the geometric information. We can now obtain the inner product of f_X and f_Z as

$$\langle f_X, f_{h^{-1}Z} \rangle_{\mathcal{H}} \coloneqq \sum_{x_i \in X, z_j \in Z} \langle \ell_X(x_i), \ell_Z(z_j) \rangle_{\mathcal{I}} \cdot k(x_i, h^{-1} z_j) \tag{3}$$

We use the kernel trick (Murphy, 2012) to substitute the inner products in (3) with the semantic kernel as $\langle f_X, f_{h^{-1}Z} \rangle_{\mathcal{H}} = \sum_{x_i \in X, z_j \in Z} k_c(\ell_X(x_i), \ell_Z(z_j)) \cdot k(x_i, h^{-1} z_j)$. We choose k_c to be the squared exponential kernel with real hyperparameters σ_c and ℓ_c that are set independently.

2.2 Feature embedding via tensor product representation

We now extend the feature space to a hierarchical distributed representation to incorporate the full geometric and hierarchical semantic relationship between the two point clouds. Let (V_1, V_2, ...) be different inner product spaces describing different types of non geometric features of a point, such as color, intensity, and semantics. Their tensor product, $V_1 \otimes V_2 \otimes \ldots$ is also an inner product space. For any $x \in X$, $z \in Z$ with features $\ell_X(x) = (u_1, u_2, \ldots)$ and $\ell_Z(z) = (v_1, v_2, \ldots)$, with $u_1, v_1 \in V_1$, $u_2, v_2 \in V_2, \ldots$, we have

$$\begin{aligned} \langle \ell_X(x), \ell_Z(z) \rangle_{\mathcal{I}} &= \langle u_1 \otimes u_2 \otimes \ldots, v_1 \otimes v_2 \otimes \ldots \rangle \\ &= \langle u_1, v_1 \rangle \cdot \langle u_2, v_2 \rangle \cdot \ldots. \end{aligned} \tag{4}$$

By substituting (4) into (3), we obtain $\langle f_X, f_{h^{-1}Z} \rangle_{\mathcal{H}} = \sum_{x_i \in X, z_j \in Z} \langle u_{1i}, v_{1j} \rangle \cdot \langle u_{2i}, v_{2j} \rangle \ldots k(x_i, h^{-1} z_j)$. After applying the kernel trick we arrive at

$$\begin{aligned} \langle f_X, f_{h^{-1}Z} \rangle_{\mathcal{H}} = &\sum_{x_i \in X, z_j \in Z} k(x_i, h^{-1} z_j) \\ &\cdot \prod_k k_{V_k}(u_{ki}, v_{kj}) \coloneqq \sum_{x_i \in X, z_j \in Z} k(x_i, h^{-1} z_j) \cdot c_{ij}. \end{aligned} \tag{5}$$

Each c_{ij} does not depend on the relative transformation. It is worth noting that, when choosing the squared exponential kernel and when the input point clouds have only geometric information, c_{ij} will be identity, and (5) has the same formulation as Kernel Correlation (Tsin and Kanade, 2004).

2.3 Equivariance property

If instead of working with the inverse of the transformation acting on the function basis we work with the function input, then the equivariance property becomes evident. Let $\mathcal{C}(\mathbb{R}^3)$ be the set of point clouds on $\mathbb{R}^3$ and $\mathcal{H}$ be the RKHS. Let $f: \mathcal{C}(\mathbb{R}^3) \to \mathcal{H}$ be our map which assigns a function to a point cloud. Consider the space of smooth functions on $\mathbb{R}^3$, $C^\infty(\mathbb{R}^3)$, and let the group $\mathcal{G}$ act on $\mathbb{R}^3$. The action lifts to an action on $C^\infty(\mathbb{R}^3)$ via $g.f(x) = f(g^{-1}x)$, $g \in \mathcal{G}$. This inverse is needed to make the action a *group* action:

$$\begin{aligned} (hg).f(x) &= h.f(g^{-1}x) = f(g^{-1}h^{-1}x) \\ &= f\left((hg)^{-1}x\right), \quad h, g \in \mathcal{G}. \end{aligned}$$

Now let Z be a point cloud and f_Z be its associated function. If $\mathcal{G}$ acts on $\mathbb{R}^3$ via isometries, then $k(gx, gz) = k(x, z)$ and we have

$$\begin{aligned} g.f_Z(x) = f_Z(g^{-1}x) &= \sum_j \ell_Z(z_j) \cdot k(g^{-1}x, z_j) \\ &= \sum_j \ell_j \cdot k(x, g z_j) = f_{gZ}(x). \end{aligned}$$

2.4 Experimental results

We present the point cloud registration experiments on real world outdoor and indoor datasets: KITTI (Geiger et al., 2012) odometry and TUM RGB-D data set (Sturm et al., 2012), with the following setup: All experiments are performed in a frame-to-frame manner without skipping images. The first frame's transformation is initialized with identity, and all later frames start with the previous frames' results. The same hyperparameter values such as lengthscale of the kernels in (2) are used for the proposed registration methods within one data set. All the baselines except Robust-ICP (Zhang et al., 2022) use all the pixels without downsampling because they do not provide an optimal point selection scheme. Fast-Robust-ICP and the proposed methods select a subset of pixels via OpenCV's FAST (Rosten and Drummond, 2006) feature detector to reduce the frame-wise running time.

The qualitative and quantitative results on KITTI Stereo is provided in Figure 2, Figure 3, and Table 1, respectively. The baselines are GICP (Segal et al., 2009), Multichannel-ICP (Servos and Waslander, 2014), 3D-NDT (Magnusson et al., 2007), and Robust-ICP (Zhang et al., 2022). GICP and NDT are compared with our geometric registration method (*Geometric CVO*, i.e., $\ell_X(x_i) = \ell_Z(z_j) = 1$). Multichannel-ICP competes with our color-assisted registration method (*Color CVO*). GICP and 3D-NDT implementation are from PCL (Rusu and Cousins, 2011). The Robust-ICP implementation is from its open source Github repositiory. The Multichannel-ICP implementation is from (Parkison et al., 2019). The semantic predictions of the images

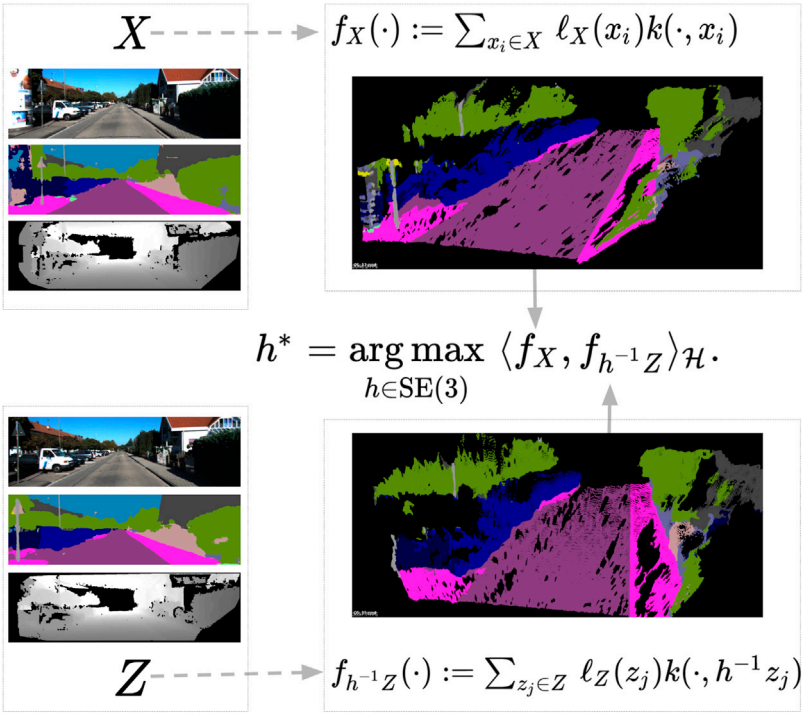

FIGURE 1
Point clouds X and Z are represented by two continuous functions f_X, f_Z in an RKHS. Each point x_i has its own semantic labels, $\ell_X(x_i)$, encoded in the corresponding function representation via a tensor product representation (Zhang et al., 2021). The registration is formulated as maximizing the inner product between two point cloud functions.

come from Nvidia's pre-trained neural network (Zhu et al., 2019), which was trained on 200 labeled images on KITTI. The depth values of the stereo images are generated with ELAS (Geiger et al., 2010). All the baselines and the proposed methods remove the first 100 rows of image pixels that mainly include sky pixels, as well as points that are more than 55 m away. Averaged over sequence 00 to 10, our geometric method has a lower translational error (4.55%) comparing to the GICP (11.23%), NDT (8.50%), and Robust-ICP (11.02%). Our color version has a lower average translational drift (3.69%) than Multichannel-ICP (14.10%). If we add semantic information the error is further reduced (3.64%). In addition, excluding the image I/O and point cloud generation operations, the proposed implementations takes on average 1.4 s per frame on GPU when registering less than 15k downsampled points. Fast-Robust-ICP also takes downsampled point clouds and takes 0.3 s per frame on CPU. GICP, NDT, and Multichannel-ICP on CPU use full point clouds (150k-350k points), and take 6.3, 6.6, and 57 s per frame, respectively.

The qualitative and quantitative results on TUM RGB-D is provided in Figure 2 and Table 2, respectively. We evaluated our method on the `fr1` sequences, which are recorded in an office environment, and `fr3` sequences, which contain image sequences in structured/nostructured and texture/notextured environments. We use the same baselines for geometric registration as KITTI. We compare Color CVO with Dense Visual Odometry (DVO) (Kerl et al., 2013) and Color ICP (Park et al., 2017). We reproduced DVO results with the code from (Pizenberg, 2019). The Color ICP implementation is taken from Open3D (Zhou et al., 2018). From Table 2, the proposed geometric registration outperforms the geometric baselines and achieves a similar performance to DVO and Color ICP. Moreover, with color information, the average error of the proposed registration decreases.

2.5 Discussions and limitations

Results in Section 2.4 demonstrate that embedding features like color and semantics in function representations provide finer data associations. Specifically, in (5), the extra appearance information c_{ij} encodes the similarity in color or semantics between the two associated points. It eliminates pairwise associations whose color or semantic appearances do not agree. Moreover, each point $x_i \in X$ is matched to multiple points $z_j \in Z$. The proposed color registration significantly improves over geometric-only methods in both KITTI Stereo and TUM RGB-D datasets.

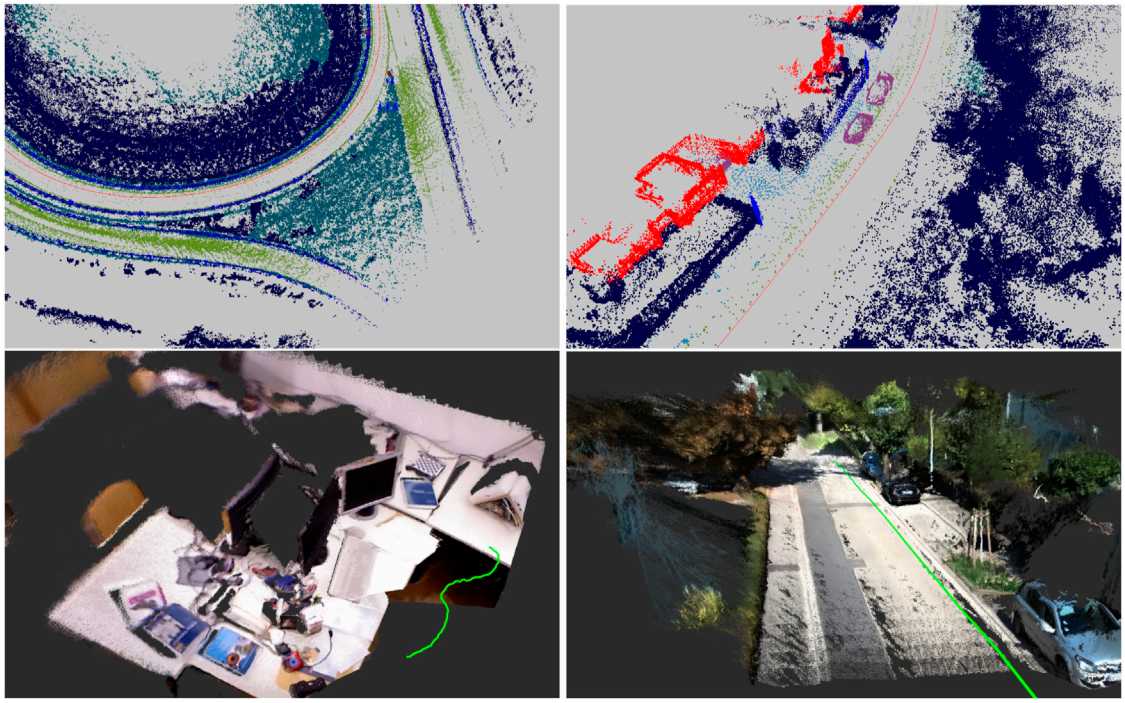

FIGURE 2
Stacked semantic and color point clouds based on frame-to-frame registration results using KITTI (Geiger et al., 2012) LiDAR, TUM RGB-D (Sturm et al., 2012) and KITTI Stereo sensors.

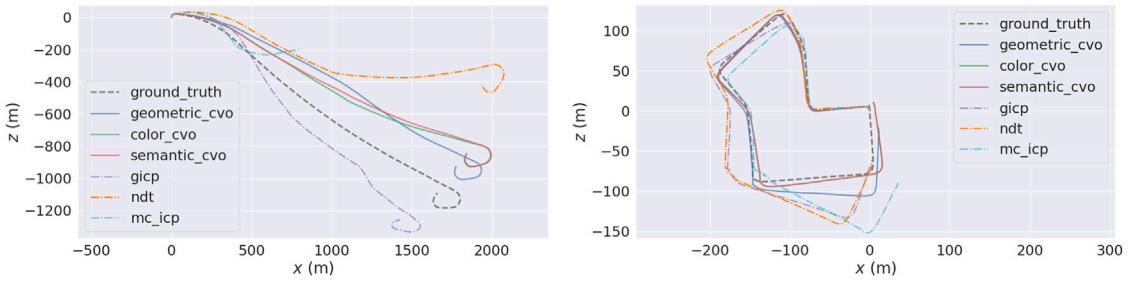

FIGURE 3
An illustration of the proposed registration methods on KITTI Stereo (Geiger et al., 2012) sequence 01 (left) and 07 (right) versus the baselines. The black dashed trajectory is the ground truth. The dot-dashed trajectories are the baselines. Plotted with EVO (Grupp, 2017).

One limitation of the proposed method is the computational complexity introduced by the double sum in (5). However, the double sum is sparse because a point $x_i \in X$ is far away from the majority of the points $z_j \in Z$, either in the spatial (geometry) space or one of the feature (semantic) spaces. But this similarity still has to be calculated with the help of GPU implementations or K-nearest-neighbor search (Blanco and Rai, 2014). In practice, an efficient point selection mechanism like FAST (Rosten and Drummond, 2006) corner selector or DSO's (Engel et al., 2017) image gradient-based pixel selector can reduce the computation time. Alternatively, representation learning can be a way to reduce the number of input points while providing richer features.

3 Learning-aided invariant robot sate estimation

Matrix Lie groups (Chirikjian, 2011; Hall, 2015; Barfoot, 2017) provide natural (exponential) coordinates that exploits

TABLE 1 Results of the proposed frame-to-frame method using the KITTI (Geiger et al., 2012) stereo odometry benchmark averaged over Sequence 00–10. The table lists the average drift in translation, as a percentage (%), and rotation, in degrees per meter($^\circ/m$). The drifts are calculated for all possible subsequences of 100, 200 . . .., 800 m.

	t (%)	r ($^\circ/m$)
Geometric Registration (Proposed)	4.55	0.0236
GICP Segal et al. (2009)	11.23	0.0452
3D-NDT Magnusson et al. (2007)	8.50	0.0396
Robust-ICP Zhang et al. (2022)	11.02	0.0256
Color Registration (Proposed)	3.69	0.0159
MC-ICP Servos and Waslander, (2014)	14.10	0.0488
Semantic Registration (Proposed)	**3.64**	**0.0155**

symmetries of the space (Long et al., 2013; Barfoot and Furgale, 2014; Forster et al., 2016; Mangelson et al., 2020; Mahony and Trumpf, 2021; Brossard et al., 2022). State estimation is the problem of determining a robot's position, orientation, and velocity that are vital for robot control (Barfoot, 2017). An interesting class of state estimators that can be run at high frequency, e.g., 2 kHz, are based on Invariant Extended Kalman Filter (InEKF) (Barrau, 2015; Barrau and Bonnabel, 2017; Barrau and Bonnabel, 2018). The theory of invariant observer design is based on the estimation error being invariant under the action of a matrix Lie group. The fundamental result is that by correct parametrization of the error variable, a wide range of nonlinear problems can lead to (log) linear error dynamics (Bonnabel et al., 2009; Barrau, 2015; Barrau and Bonnabel, 2017).

Proprioceptive state estimators often combine data from an Inertial Measurement Unit (IMU) with signals such as body velocity, kinematics information, and contact events. A successful method in this domain for legged robots is the contact-aided InEKF (Hartley et al., 2020). This approach is attractive because the odometry estimate only depends on inertial, contact, and kinematic data, which barring sensor failure, always exist. Furthermore, the independence from any vision systems make the state estimator robust to perceptually degraded situations (Hartley et al., 2018; Lin et al., 2022). Many existing perception and navigation methods can work well, given a correct though uncertain initial condition; hence, such an accurate dead reckoning can enable higher levels of autonomy for existing systems.

The invariant observer design provides us with a framework with better convergence properties. However, sensory data input likewise plays a crucial role in state estimation tasks. Noisy and biased measurements can hinder the performance of the observer. On the other hand, sensor failures can lead to catastrophic results in state estimation. Recent deep learning methods allow one to address these challenges by estimating the bias or inferring the information that traditional sensors cannot obtain (Liu et al., 2018; Wellhausen et al., 2019). By combining learning with the symmetry-preserving observer design, the performance and robustness of a state estimator can be greatly improved (Brossard et al., 2019; Brossard et al., 2020).

This section reports our recent developments on deep-learning-aided invariant state estimator (Lin et al., 2022). In this work, a deep contact estimator is designed to estimate the foot contact events for legged robots. The learned foot contacts are then used to enforce the non-slip constraint in an InEKF. Although the complete state estimation pipeline is purely proprioceptive, it can achieve a similar performance to a state-of-the-art visual SLAM system. In addition, the program, including the deep contact estimator, runs in real-time (500 Hz) on an MIT Mini Cheetah robot. We also report our new results on developing the InEKF for wheeled platforms in Section 3.4. The data sets and software are available for download .

TABLE 2 The RMSE of Relative Pose Error (RPE) averaged over TUM RGB-D (Sturm et al., 2012) fr1 and fr3 structure v.s texture sequences. The t columns show the RMSE of the translational drift in m/sec and the r columns show the RMSE of the rotational error in deg /sec. The RMSE is averaged over all sequences.

	`fr1`		`fr3 structure v.s texture`	
	t (m/sec)	r (deg/sec)	t (m/sec)	r (deg/sec)
Geometric Registration (Proposed)	0.0730	**2.3805**	0.0794	2.8536
GICP Segal et al. (2009)	0.4034	15.8838	0.2116	5.2979
3D-NDT Magnusson et al. (2007)	0.2290	14.0311	0.2487	6.9860
Robust-ICP Zhang et al. (2022)	0.1487	6.6911	0.2091	5.4168
Color Registration (Proposed)	**0.0545**	2.4333	**0.0754**	2.6651
DVO Kerl et al. (2013)	0.0623	2.6943	0.1386	4.9843
Color ICP Park et al. (2017)	0.1353	5.8985	0.0820	**2.2041**

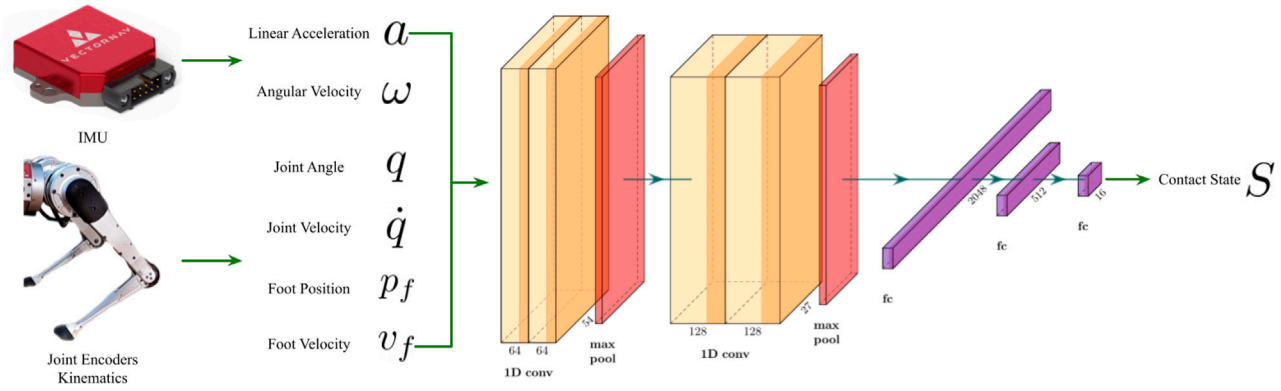

FIGURE 4
The architecture of the proposed contact estimator (Lin et al., 2022). The inputs include linear accelerations and angular velocities from an IMU, joint angles and joint velocities from encoders, and foot positions and velocities from kinematics.

3.1 Deep contact estimator

The goal of the deep contact estimator is to accurately estimate the foot contact events where the robot's foot maintain zero velocity in the world frame. We model the contact as binary events on each leg $l \in \{RF, LF, RH, LH\}$. The overall contact states of the robot becomes a collection of binary values $C = [\, c_{RF} \;\; c_{LF} \;\; c_{RH} \;\; c_{LH} \,]$, where $c_l \in \{0, 1\}$ with 0 indicates no contact, and 1 denotes a firm contact. For a quadruped robot, there exist 16 different combinations of the contact states. We formulate our approach as a classification task[1].

The contact estimator takes sensor measurements from an IMU, joint encoders, and kinematics as input. To allow the network to extract information from the time domain, a fixed number of past data is concatenated together before inputting into the network. Figure 4 lists the input data along with the network architecture. The linear block contains 3 fully-connected layers that convert the deep features into the 16 classes. Dropout mechanisms are also added to the first 2 fully-connected layers to prevent the network from overfitting. Finally, we employ the cross-entropy loss for the classification task.

3.2 Contact data sets

We create open-sourced contact data sets using an MIT Mini Cheetah robot (Katz et al., 2019). The data sets are collected using an MIT controller (Kim et al., 2019) across 8 different terrains (shown in Figure 5). We record proprioceptive measurements such as joint encoders data, foot positions and velocities, IMU measurements, and estimated joint torques from the controller. The IMU measurements are received at 1000Hz, while other data are recorded at 500Hz. We upsample all measurements to match the IMU frequency after recording the data. In addition to the proprioceptive measurements, we also record RGB-D images with an Intel D455 camera mounted on top of the robot. These RGB-D images are used in a state-of-the-art visual SLAM algorithm, ORB SLAM2 (Mur-Artal and Tardós, 2017). For the grass data sets, we obtain ground truth trajectories from a motion capture system. However, for the rest of the data sets, we use the trajectory from ORB SLAM2 as an approximation to ground truth. In total, around 1,000,000 data points were collected on 8 different terrains. We also include some examples of the robot walking in the air to provide the network with negative examples by holding the robot up and applying the same controller commands. The detailed number of data collection is listed in Table 3. The labels of the ground truth contacts are generated automatically with an offline pre-processing algorithm (self-supervised learning). Detailed of the algorithm can be found in the work of (Lin et al., 2022).

3.3 Experimental results

We evaluate the accuracy, false positive rate, and false negative rate of the proposed contact estimator using the Mini Cheetah robot, as shown in Table 4. We compare our method with a model-based approach (Focchi et al., 2013; Fakoorian et al., 2016; Fink and Semini, 2020), denoted GRF Thresholding, and a fixed gait cycle assumption which assume the pre-determined gait cycle is precisely followed by the controller. Our method performs the best across all three sequences. It is

1 https://github.com/UMich-CURLY/deep-contact-estimator; https://github.com/UMich-CURLY/cheetah_inekf_realtime; https://github.com/UMich-CURLY/husky_inekf.

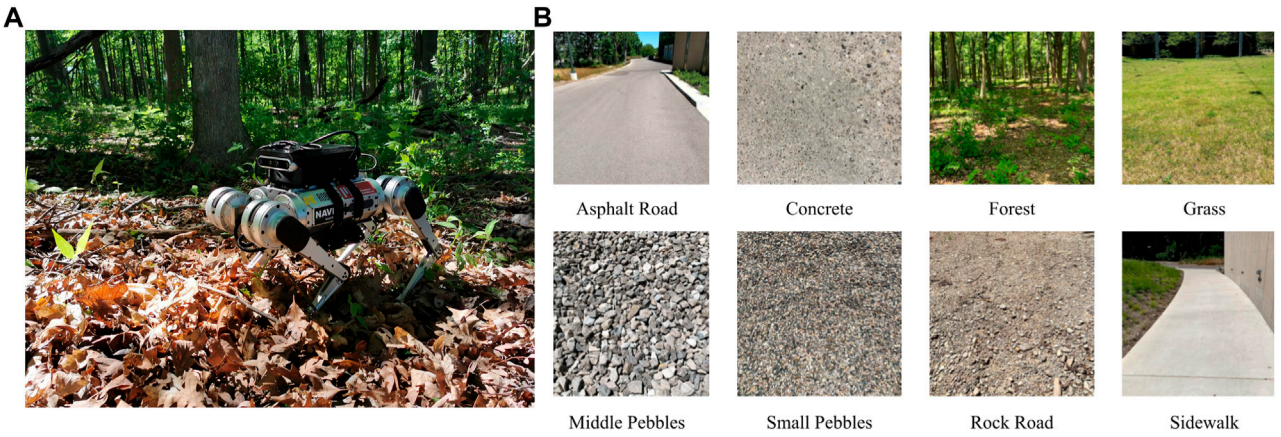

FIGURE 5
(A) Setup of an MIT Mini Cheetah with the perception suite used in the data collection. **(B)** Different ground types in the contact data set.

TABLE 3 Number of data of each terrain in the contact data sets.

Terrain Type										
overall	**air trotting**	**air pronking**	**asphalt road**	**concrete**	**forest**	**grass**	**middle pebble**	**small pebble**	**rock road**	**sidewalk**
1,013,441	44,386	48,972	94,615	465,144	72,144	103,392	44,442	52,669	45,819	58,115

worth noticing that the proposed contact estimator has the lowest false positive rate, which is crucial for state estimation tasks as the violation of the non-slip condition could lead to severe drift in the estimation.

We integrated our contact estimator into the contact-aided InEKF. The entire state estimation pipeline, including our deep contact estimator, runs in real-time at 500 *Hz* on an NVIDIA Jetson AGX Xavier. Figure 6 shows the trajectory generated by the InEKF using different contact sources on a concrete loop sequence. We also run the filter using the ground truth contact data to serve as a reference. Qualitatively compared to the baseline contact detectors, the resulting trajectory with the proposed contact estimation has smaller drifts from the trajectory with ground truth contacts, especially in the height (Y) axis. Furthermore, compared to the baseline contact estimators, the proposed method also yields a smoother trajectory.

3.4 Invariant EKF with body velocity measurements

In addition to legged robots, we also develop state estimation software for wheeled robots using the InEKF. Instead of using the foot contact, here we use the body velocity as measurements in the correction step. Although the implementation is not restricted to a specific platform, we evaluate the performance of the filter on a differential-drive wheeled robot, Husky, from Clearpath robotics. We obtain the body velocity measurements from wheel encoders using a simple differential-drive model, $v_{\text{body}} = \frac{r(\omega_l+\omega_r)}{2}$, where ω_l and ω_r are wheel angular velocities measured by the wheel encoders and r is the wheel radius. Moreover, we also use pseudo velocity measurements by assuming zero velocities on the Y and Z axis (Dissanayake et al., 2001). However, this estimation can be noisy and inaccurate due to slip or bumping on the wheels. In order to know the full potential of this framework, we also record several sequences in a motion capture facility and use the velocity from the motion capture system to correct the estimated state. Figure 7 shows the resulting trajectories. Using the wheel velocity and pseudo velocity measurements, the state estimator can produce a good estimation of the robot pose. If the accuracy of the velocity is improved, then the drift can be further reduced.

Although this section does not discuss the incorporation of learning into the InEKF state estimator, as done previously for the legged robot, the following lessons from our experiments are noteworthy.

- Body velocity measurements provide a generic correction model that can work on any robotic platform. However, accurate body velocity measurement is not readily available. Specifically, the filer requires the ground

TABLE 4 Accuracy comparison against baselines. The proposed method achieves the highest accuracy on all sequences. Although the gait cycle method has an accuracy closer to the proposed method, it does not remove false positives when gait cycle is violated.

Sequence	Method	% Accuracy					% False Positive Rate	% False Negative Rate
		Leg RF	Leg LF	Leg RH	Leg LH	Leg Avg	Leg Avg	Leg Avg
Concrete Short Loop	GRF Thresholding	73.43	70.02	71.69	70.04	71.30	37.07	13.24
	Gait Cycle	85.66	84.98	84.68	85.11	85.11	22.95	**0.00**
	Proposed Method	**98.34**	**97.87**	**97.95**	**98.56**	**98.18**	**1.45**	2.51
Grass Test Sequence	GRF Thresholding	82.55	78.93	84.62	82.48	82.14	26.87	**0.63**
	Gait Cycle	92.41	92.38	91.04	90.55	91.59	10.95	3.53
	Proposed Method	**98.08**	**97.57**	**97.73**	**97.73**	**97.78**	**2.35**	1.98
Forest Test Sequence	GRF Thresholding	80.99	80.09	82.75	83.24	81.77	26.54	1.84
	Gait Cycle	83.03	82.56	84.44	84.28	83.58	24.71	**0.08**
	Proposed Method	**97.05**	**96.62**	**97.24**	**97.40**	**97.08**	**2.82**	3.12

Bold values in tables show the best performance.

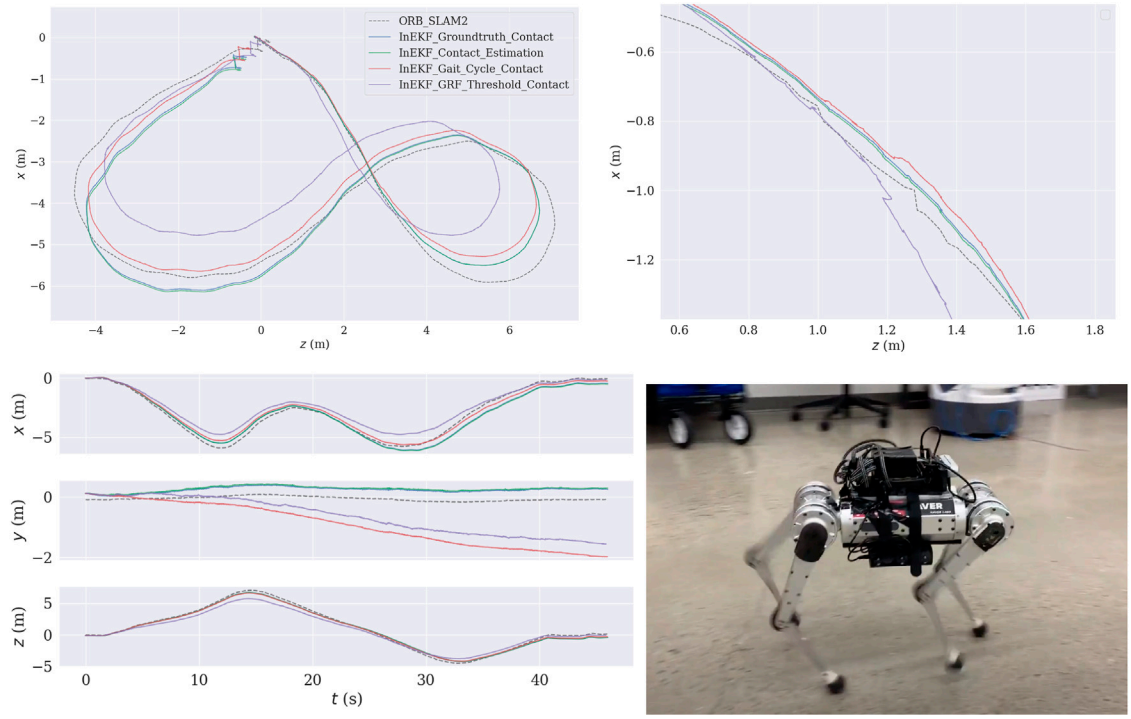

FIGURE 6
Concrete short loop test sequence. Top Left: The bird's-eye view of the trajectories. The estimated trajectory is mapped to the camera frame (*Y* pointing downward, and *Z* pointing forward). Top Right: Zoomed-in of the bird's-eye view. Bottom Left: This figure shows that the gait cycle and GRF thresholding methods produce a significant height (Y) drift. Bottom Right: Robot configuration.

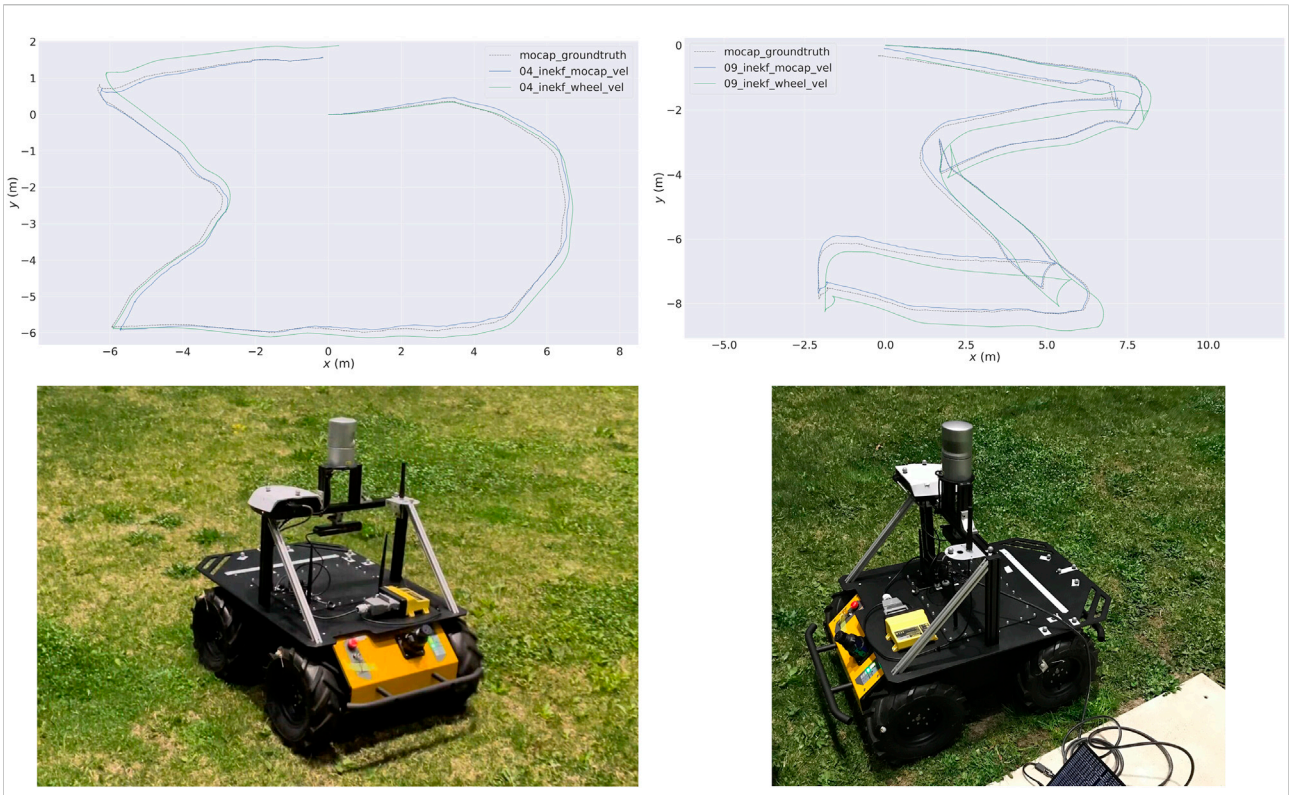

FIGURE 7
Top: Two sequences of trajectories recorded at the University of Michigan MAir motion capture facility. The green lines are the InEKF estimated trajectories using velocity estimated from the wheel encoders, and the blue lines are the InEKF results using velocity from the motion capture system. Bottom: The robot setup. The ground of the facility is planted with natural grass.

referenced body velocity (Teng et al., 2021b; Potokar et al., 2021).

- The robot's nonholonomic constraints (i.e., velocity constraints that cannot be integrated) can provide pseudo observations that can significantly improve the performance. However, these constraints are assumptions and detached from the robot's behavior. Learning such constraints provides a way to use sensory inputs instead of assumptions (Brossard et al., 2019; Brossard et al., 2020).
- Moreover, the nonholonomic constraints are violated when the robot drifts. Slip detection and friction estimation are challenging and necessary tasks for future learning-aided robot estimation modules.

4 Symmetry-preserving geometric robot control

The geometry of the configuration space of a robotics system can naturally be modeled using matrix Lie (continuous) groups (Bloch, 2015; Lynch and Park, 2017). For example, the centroidal dynamics of legged robots can be approximated by a single rigid body, whose motion is on SE(3).

The Euler angle based convex Model Predictive Control (MPC) (Di Carlo et al., 2018) has been proposed for locomotion planning on the quadrupedal robot. Zero roll and pitch angle assumptions are validated by assuming a flat ground, which may fail when such assumptions no longer hold. To avoid the problem, the geometric MPC that utilize the symmetry of the Lie group has been proposed. A local control law has been proposed by Kalabić et al. (2016); Kalabić et al. (2017), where the linearized dynamics are defined by a local diffeomorphism from the SE(3) manifold to $\mathbb{R}^n$ space. However, such a diffeomorphism is not unique and too abstract for controller design.

The Variational Based Linearization (VBL) technique (Wu and Sreenath, 2015) are applied to linearize the Lagrangian to obtain the discrete-time equation of motion and applied to robot pose control (Chignoli and Wensing, 2020). A VBL based MPC is proposed by Agrawal et al. (2021) for locomotion on discrete terrain using a gait library. The result suggests that the VBL based linearization can preserve the energy, thus making the system more stable. However,

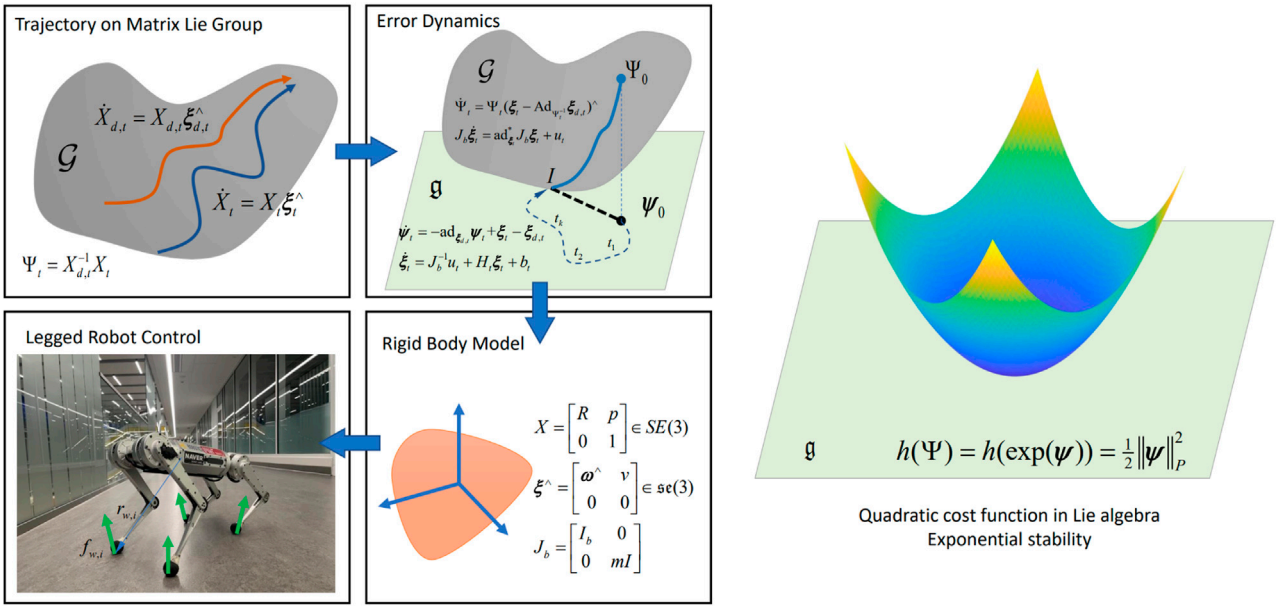

FIGURE 8
The proposed error-state MPC framework by Teng et al. (2022a). The tracking error is defined on a matrix Lie group and linearized in the Lie algebra. A convex MPC algorithm is derived via the linearized dynamics for tracking control. The proposed algorithm is applied to a single rigid body system and verified on a quadrupedal robot MIT Mini Cheetah. A quadratic cost function in the Lie algebra can verify the stability of the proposed MPC.

the VBL method linearized the system at the reference trajectory, which may result in unstable motion (Ding et al., 2021). Other than linearizing at the reference trajectory, the work of Ding et al. (2021) linearized the system at the current operating point to obtain the QP problem for tracking of legged robot trajectory. However, the linearized state matrix of Ding et al. (2021) depends on the orientation, which can be avoided by exploiting the symmetry of the system as done by Teng et al. (2022a,b). The proposed framework is illustrated in Figure 8.

4.1 Error-state convex MPC

For tracking control on Lie group $\mathcal{G}$, we define the desired trajectory as $X_{d,t} \in \mathcal{G}$ and the actual state as $X_t \in \mathcal{G}$, both as function of time t. Given the twists ξ_t and desired twists $\xi_{d,t}$ and the reconstruction equation, we have $\frac{d}{dt}X_t = X_t\xi_t^\wedge$, $\frac{d}{dt}X_{d,t} = X_{d,t}\xi_{d,t}^\wedge$. Similar to the left or right error defined in (Bullo and Murray, 1999), we define the error between X_t^d and X_t as

$$\Psi_t = X_{d,t}^{-1}X_t \in \mathcal{G}. \tag{6}$$

For the tracking problem, our goal is to drive the error from the initial condition Ψ_0 to the identity $I \in \mathcal{G}$. Taking derivative on both sides of (6), we have

$$\begin{aligned}\frac{d}{dt}\Psi_t = \dot{\Psi}_t &= \frac{d}{dt}\left(X_{d,t}^{-1}\right)X_t + X_{d,t}^{-1}\frac{d}{dt}X_t = X_{d,t}^{-1}\frac{d}{dt}X_t - X_{d,t}^{-1}\frac{d}{dt}\left(X_{d,t}\right)X_{d,t}^{-1}X_t\\ &= X_{d,t}^{-1}X_t\xi_t^\wedge - X_{d,t}^{-1}X_{d,t}\xi_{d,t}^\wedge X_{d,t}^{-1}X_t = \Psi_t\xi_t^\wedge - \xi_{d,t}^\wedge\Psi_t.\\ \dot{\Psi}_t &= \Psi_t\left(\xi_t - \Psi_t^{-1}\xi_{d,t}\Psi_t\right)^\wedge = \Psi_t\left(\xi_t - \mathrm{Ad}_{\Psi_t^{-1}}\xi_{d,t}\right)^\wedge.\end{aligned} \tag{7}$$

We define $\psi_t^\wedge$ as an element of the Lie Algebra that corresponds to Ψ_t. Thus by the exponential map, we have $\Psi_t = \exp(\psi_t)$, $\Psi_t \in \mathcal{G}$, $\psi_t^\wedge \in \mathfrak{g}$. Given the first-order approximation of the exponential map, $\Psi_t = \exp(\psi_t) \approx I + \psi_t^\wedge$, and a first-order approximation of the adjoint map $\mathrm{Ad}_{\Psi_t} \approx \mathrm{Ad}_{I+\psi_t^\wedge}$, we can linearize (7) by only keeping the first order term of ψ_t and $\xi_t - \xi_{d,t}$ as:

$$\dot{\Psi}_t \approx \left(I + \dot{\psi}_t^\wedge\right) \approx \left(I + \psi_t^\wedge\right)\left(\xi_t - \mathrm{Ad}_{\left(I-\psi_t^\wedge\right)}\xi_{d,t}\right)^\wedge, \tag{8}$$

$$\dot{\psi}_t = -\mathrm{ad}_{\xi_{d,t}}\psi_t + \xi_t - \xi_{d,t}. \tag{9}$$

Eq. 9 is the linearized velocity error in the Lie algebra.

The dynamics of ξ_t is described by the *forced* Euler-Poincaré equations (Bloch et al., 1996; Bloch, 2015) as $J_b\dot{\xi} = \mathrm{ad}_\xi^* J_b\xi + u$, where $u \in \mathfrak{g}^*$ is the generalized control input force applied to the body fixed principal axes, ad* is the co-adjoint action, and $\mathfrak{g}^*$ is the cotangent space. This model is nonlinear. To compute a locally linear approximation of the nonlinear term, we adopt the Jacobian linearization around the operating point $\bar{\xi}$ as $J_b\dot{\xi} \approx \mathrm{ad}_{\bar{\xi}}^* J_b\bar{\xi} + \frac{\partial \mathrm{ad}_\xi^* J_b\xi}{\partial\xi}\big|_{\bar{\xi}}(\xi - \bar{\xi}) + u$. Thus, we have the linearized dynamics in the following form $\dot{\xi} = H_t\xi + J_b^{-1}u + b_t$, We define

the system states as $x_t \coloneqq \begin{bmatrix} \psi_t \\ \xi_t \end{bmatrix}$. Then, the linearized dynamics becomes $\dot{x}_t = A_t x_t + B_t u_t + h_t$, where

$$A_t \coloneqq \begin{bmatrix} -\mathrm{ad}_{\xi_{d,t}} & I \\ 0 & H_t \end{bmatrix}, B_t \coloneqq \begin{bmatrix} 0 \\ J_b^{-1} \end{bmatrix}, h_t \coloneqq \begin{bmatrix} -\xi_{d,t}, \\ b_t \end{bmatrix}.$$

4.2 Convex MPC design

On Lie groups, our cost function is designed to regulate the tracking error ψ_t and its derivative $\dot{\psi}_t$ rather than the difference between $\xi_{d,t}$ and ξ_t. Thus, our tracking error can be designed as:

$$y_t \coloneqq \begin{bmatrix} \psi_t \\ \dot{\psi}_t \end{bmatrix} = \begin{bmatrix} I & 0 \\ -\mathrm{ad}_{\xi_{d,t}} & I \end{bmatrix} x_t - \begin{bmatrix} 0 \\ \xi_{d,t} \end{bmatrix}. \tag{10}$$

Given some semi-positive definite weights P, Q, and R, we can now write the quadratic cost function as

$$N\left(y_{t_f}\right) = y_{t_f}^{\mathsf{T}} P y_{t_f}, \; L\left(y_t, u_t\right) = y_t^{\mathsf{T}} Q y_t + u_t^{\mathsf{T}} R u_t. \tag{11}$$

Given the future twists $\xi_{d,t}$, initial error state ψ_0 and twist ξ_0, we can define all the matrices. Discretizing the system at time steps $\{t_k\}_{k=1}^{N}$, we can design the MPC as follows.

Problem 2. *Find* $u_k \in \mathfrak{g}^*$ *such that*

$$\begin{aligned} \min_{u_k} \quad & y_N^{\mathsf{T}} P y_N + \sum_{k=1}^{N-1} y_k^{\mathsf{T}} Q y_k + u_k^{\mathsf{T}} R u_k \\ \text{s.t.} \quad & x_{k+1} = A_k x_k + B_k u_k + h_k, \; u_k \in \mathcal{U}_k, x_0 = x(0). \end{aligned}$$

In Problem 2, A_k, B_k, and h_k can be obtained by zero-order hold or Euler first-order integration. Problem 2 is a QP problem that can be solved efficiently, e.g., using OSQP (Stellato et al., 2020).

4.3 Stability analysis

The stability of the proposed controller can be verified by a quadratic Lyapunov cost function in Lie algebra. First, we introduce the left invariant inner product. Then, we can derive the gradients of the quadratic cost function in the tangent space.

Definition 1. *Given* $\phi_1, \phi_2 \in \mathbb{R}^{\dim \mathfrak{g}}$ *and* $\phi_1^\wedge, \phi_2^\wedge \in \mathfrak{g}$, *we define the inner product* $\langle \phi_1^\wedge, \phi_2^\wedge \rangle_{\mathfrak{g}} = \phi_1^{\mathsf{T}} P \phi_2$, *where P is a positive definite matrix. This inner product is left-invariant. To see this, suppose* $X\phi_1^\wedge, X\phi_2^\wedge \in T_X\mathcal{G}, \quad \forall X \in \mathcal{G}, \quad$ *then* $\langle X\phi_1^\wedge, X\phi_2^\wedge \rangle_X = \langle (\ell_{X^{-1}})_* X\phi_1^\wedge, (\ell_{X^{-1}})_* X\phi_2^\wedge \rangle_{\mathfrak{g}} = \langle \phi_1^\wedge, \phi_2^\wedge \rangle_{\mathfrak{g}}$, *where* $(\ell_{X^{-1}})_* = X^{-1}$: $T_X\mathcal{G} \to \mathfrak{g}$ *is the pushforward map.*

Theorem 1. *Consider the state* $X \in \mathcal{G}$, $\phi \in \mathbb{R}^{\dim \mathfrak{g}}$, *and* $X = \exp(\phi)$. *We consider the metric in Definition* 1. *The function* $h = \frac{1}{2}\|\phi\|_P^2$ *is a candidate Lyapunov function and the gradient of h with respect to X is* $\nabla h = X\phi^\wedge$.

Finally, we show that a linear feedback in Lie algebra can regulate the state to the identity exponentially.

Theorem 2. *Consider the state in Theorem* 1 *as a trajectory. Let* $\xi^\wedge \in \mathfrak{g}$. *The system* $\dot{X} = X\xi^\wedge$ *can be exponentially stabilized to* $X = I$ *by linear feedback* $\xi = K\phi$, *where K is a gain matrix that is Hurwitz.*

The detailed proof of the theorems are presented in the work of Teng et al. (2022b). For the proposed MPC, we can follow the same steps and estimate the region of attraction. For the unconstrained case, the resulting LQR problem will lead to a linear feedback that can be verified by Theorem 2.

4.4 Validation on quadrupedal robot

We conduct two experiments on the quadrupedal robot Mini Cheetah (Katz et al., 2019) to evaluate the proposed MPC. Both experiments use a single rigid body model to approximate the torso motion. We apply MIT controller (Di Carlo et al., 2018) with the proposed MPC to plan the Ground Reaction Force (GRF).

4.4.1 Robot pose tracking

In this experiment, a mixture of roll and yaw reference angle is applied for tracking. The reference signals and snapshots of robot motion are presented in Figure 9. Each controller is implemented three times. The details of the responses are presented in Figure 10. It can be seen that as no feedforward force at the equilibrium is provided, all controllers have steady-state error. However, the geometric-based controller, i.e., proposed and the VBL based MPC, has a smaller steady-state error than the Euler angle-based one. As the VBL based MPC does not conserve the scale of the error, the convergence rate is much lower than our controller, especially when the opposite Euler angle signal is applied at the middle of the reference profile.

4.4.2 Robot trotting

We also apply our controller to robot locomotion. Ours and baseline controllers are deployed to plan the robot's GRF given command twists. Then the GRF is applied to the Whole Body Impulse Control (WBIC) (Kim et al., 2019) to obtain the joint torques. Unlike the conventional whole-body controller, WBIC prioritizes the GRF generation by penalizing the deviation of GRF from the planned GRF. We increase the penalty for the GRF by 1e4 times in the original WBIC, so the GRF merely deviates from the planned one.

We first apply a step signal in yaw rate. Then we add a step signal in x motion in the robot frame, and the yaw rate becomes a sinusoidal signal. The reference is presented in Figure 12 and the snapshots of the experiments are in Figure 11. We find that ours and the VBL-MPC can better track the yaw rate than the Euler angles-based MPC, as expected. As the orientation and position tracking errors are small because every step is integrated from the current

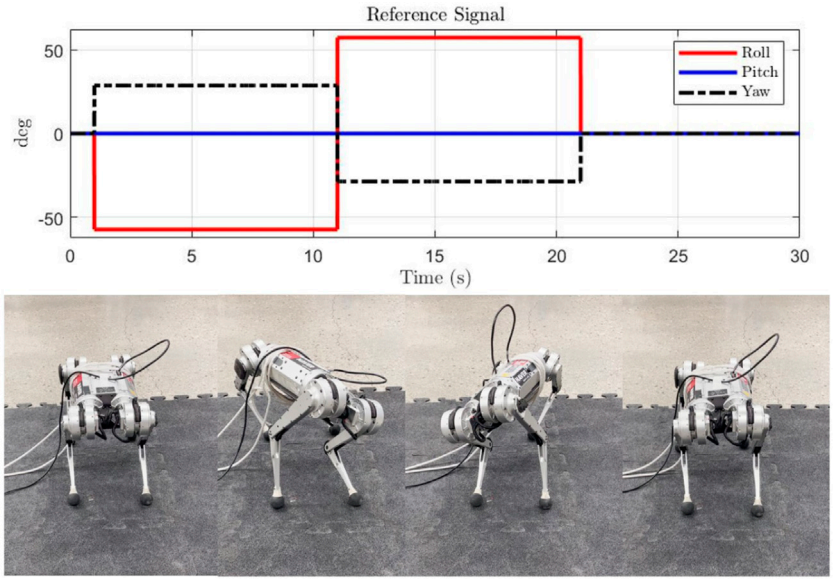

FIGURE 9
Reference signal for roll and yaw angle tracking. From 1 to 11 s, the robot roll changes from 0 to -57.3° and yaw changes from 0 to 28.5°. Then the robot leans to the opposite side for 10 s.

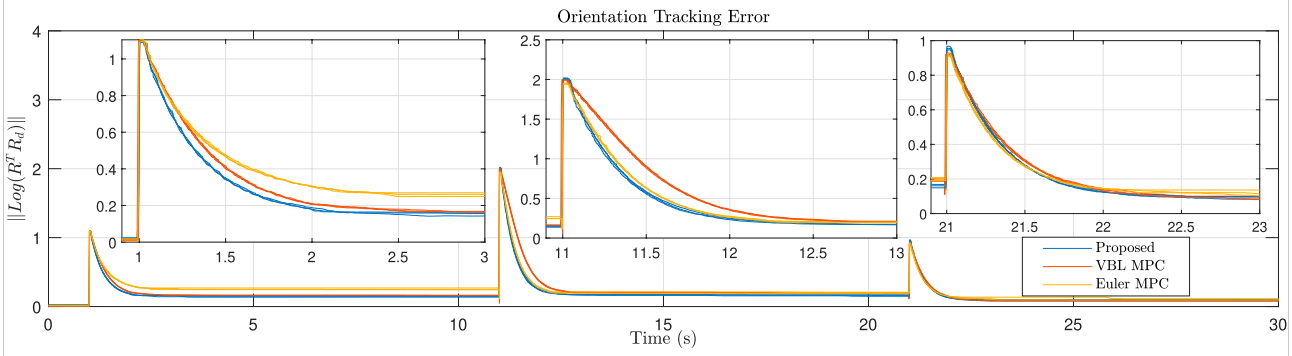

FIGURE 10
Error convergence for roll and yaw tracking. When a new step signal is applied, our controller converges faster than the baseline methods and has a smaller steady-state error. The Euler angle-based MPC has a larger steady-state error as both roll and yaw signals are applied.

state, it is reasonable that all controllers perform well in position tracking. The result can be seen in Figure 12.

5 Equivariant representation learning: Augmenting geometry with learning

Learning equivariant representation of geometric data can provide efficiency and generalizability in challenging robot perception tasks. Loosely speaking, *equivariance* is a property for a map such that given a transformation in the input, the output changes in a predictable way determined by the input transformation. Mathematically the equivariance is represented as commutativity: a function $f: X \rightarrow X$ is equivariant to a set of transformations G, if for any $g \in G$, $g \cdot f(x) = f(g \cdot x)$, $\forall x \in X$. For example, applying a translation on a 2D image and then going through a convolution layer is identical to processing the original image with a convolution layer and then shifting the output feature map. Therefore convolution layers are translation-equivariant.

An equivariant network captures the inherent symmetry of data, disentangling the information dependent on and

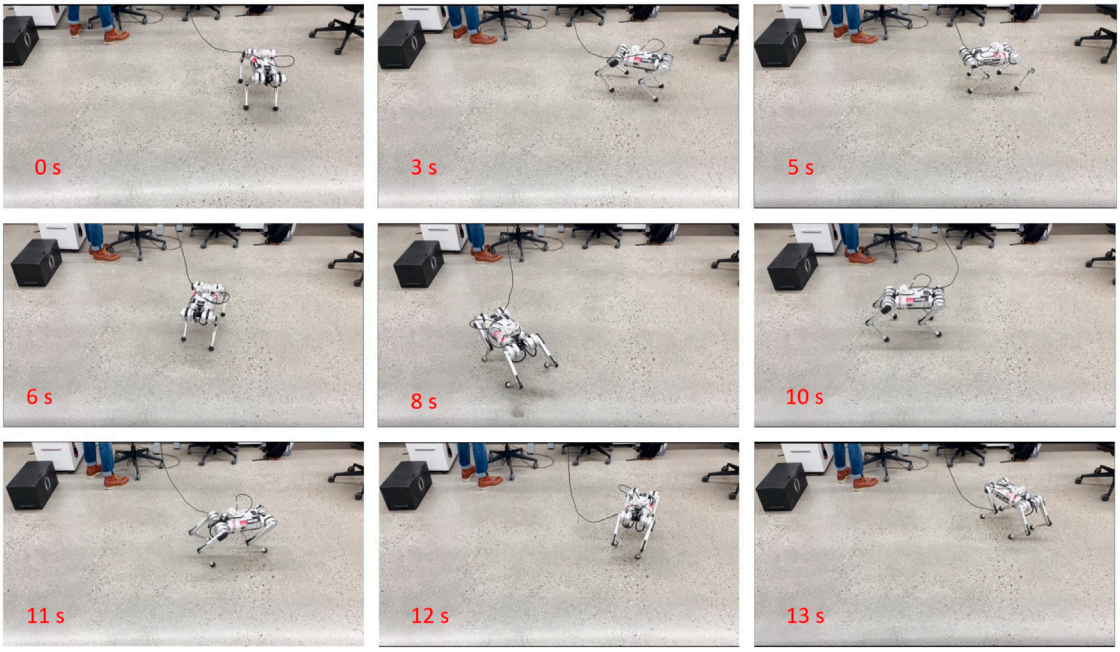

FIGURE 11
Snapshots of the experiments on reference tracking in Mini Cheetah trotting. The time corresponds to the reference signal in Figure 12.

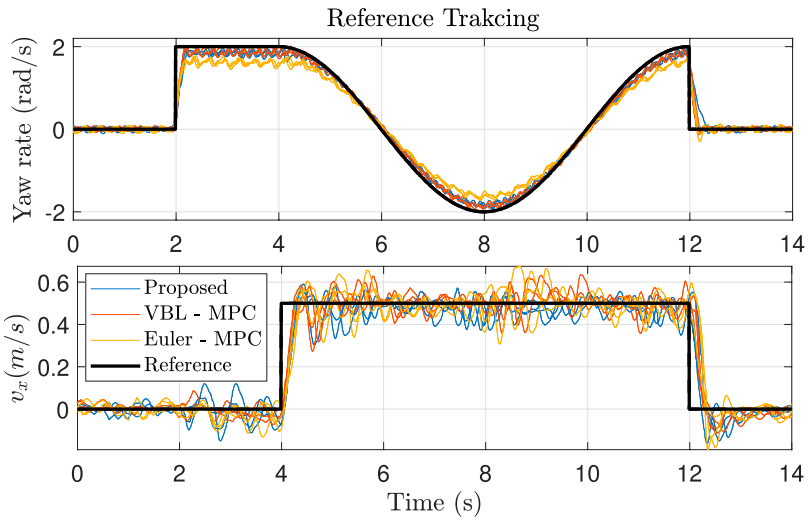

FIGURE 12
Reference tracking for quadrupedal robot trotting. Each controller is tested three times. The responses are too noisy; thus, the results are smoothed using the moving average filter.

independent of the transformations. As an analogy, this is akin to the notion of coordinates-free calculations on manifolds in modern mathematrics. In a coordinates-free setup, one can distinguish the intrinsic properties of the problem from those of a particular choice of coordinates (Tu, 2011). We mainly focus on the rigid body transformations, decoupling the poses and the

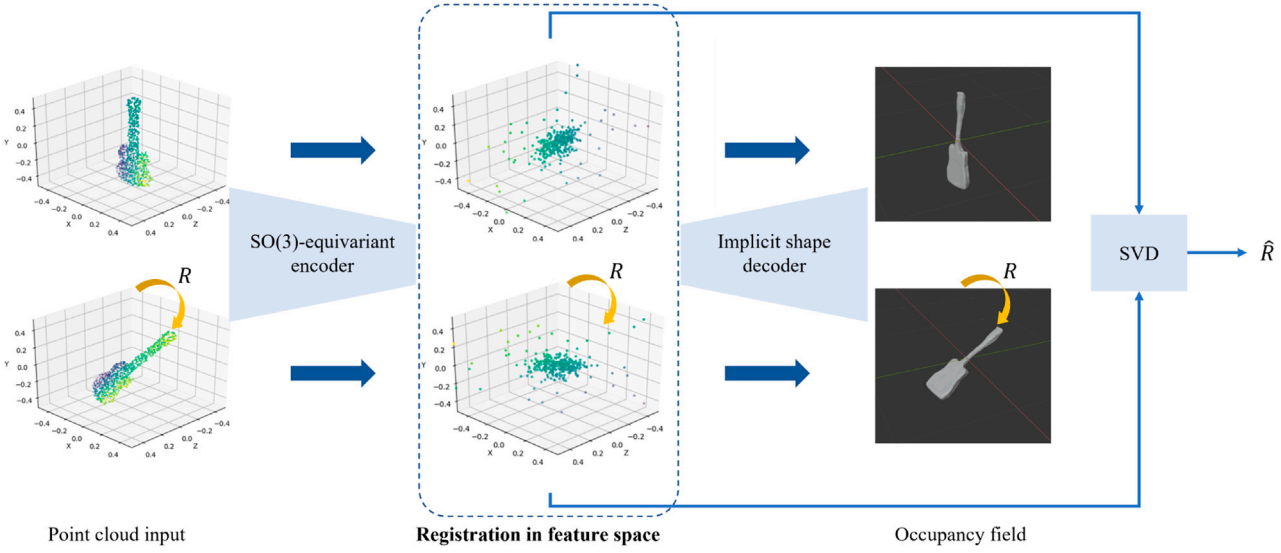

FIGURE 13
Overview of the SO(3)-equivariant registration network (Zhu et al., 2022b). The point cloud input is of shape $\mathbb{R}^{N\times 3}$, and the encoded feature is of shape $\mathbb{R}^{C\times 3}$. N is the number of points, and C is the dimension of features. Occupancy field is a function $v(p) \mapsto [0,1], p \in \mathbb{R}^3$ mapping any 3D coordinate to an occupancy value. The rotation is estimated by aligning the features using Horn's method (Horn et al., 1988).

pose-independent information, e.g., shapes and semantics, from the geometric data by leveraging equivariant feature learning.

5.1 Point cloud registration with SO(3)-equivariant implicit shape representations

We proposed an initialization-independent rotation registration method for point clouds by leveraging a SO(3)-equivariant feature learner (Zhu et al., 2022b). An overview of the network structure is depicted in Figure 13. A point cloud is mapped to a feature space equipped with SO(3) rotations represented as 3 × 3 matrix multiplications, consistent with the input Euclidean space. Therefore, the rotational registration can be approached by solving the Orthogonal Procrustes problem in the feature space. Our method achieved accurate rotation registration regardless of initial estimation error. It also implies that our method falls in the correspondence-free category, where the step of data association, i.e., matching corresponding points in two point clouds, is not needed.

The SO(3)-equivariant feature learning is realized through a backbone network called Vector Neuron (Deng et al., 2021). The key idea is to augment the scalar feature in each feature dimension to a vector in $\mathbb{R}^3$. In Vector Neuron networks, the feature matrix with feature dimension C corresponding to a set of N points is $V \in \mathbb{R}^{N\times C\times 3}$. The mapping between layers can be written as $f: \mathbb{R}^{N\times C_l\times 3} \to \mathbb{R}^{N\times C_{l+1}\times 3}$, where l is the layer index. Following this design, the representation of SO(3) rotations in feature space is straightforward: $g(R) \cdot V := VR$, where $g(R)$ denotes the rotation operation in the feature space, parameterized by the 3 × 3 rotation matrix $R \in$ SO(3). Here we ignore the first dimension N of V for simplicity, and the SO(3)-equivariance of the linear layer: $f_{\text{lin}}(V) = WV$, where $W \in \mathbb{R}^{C_{l+1}\times C_l}$, can be easily verified as follows.

$$g(R) \cdot f_{\text{lin}}(V) = WVR = f_{\text{lin}}(g(R) \cdot V). \tag{12}$$

For further discussions beyond the linear layers, see the work of Deng et al. (2021).

We design an encoder-decoder structure to learn the features. We also improve the robustness to noise in sampled points by decoding an implicit shape representation following the Occupancy network (Mescheder et al., 2019). Our method is tested on the synthetic object-wise data set ModelNet40 (Wu et al., 2015), shown in Table 5. For further experiments using real-world indoor RGB-D data set 7Scenes (Shotton et al., 2013), see the work of (Zhu et al., 2022b).

5.2 Efficient SE(3)-equivariant representations learning

Our recent work (Zhu et al., 2022a) extends the SO(3)-equivariance to SE(3)-equivariance to better deal with arbitrary rigid body transformations of 3D point-cloud data. We use Convolutional Neural Networks (CNNs) which inherit translational equivariance. Existing work of equivariant convolutional networks are mainly in two types. First is

TABLE 5 Rotational registration error given rotated copies of point clouds. Tested using ModelNet40 (Wu et al., 2015) official test set. The best are shown in bold. The second best are shown in *italic*. All values are in degrees.

Max initial rotation angle			0	30	60	90	120	150	180
Categories	**Global**	**Methods**	**Rotation error after registration**						
Correspondence-free	N	PCR-Net Sarode et al. (2019)	7.08	9.50	27.38	68.22	109.61	129.29	133.49
	N	FMR Huang et al. (2020)	**0.00**	0.45	5.29	21.95	42.26	66.39	79.43
	Y	Ours Zhu et al. (2022b)	*0.02*	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**
Correspondence-based	N	RPM-Net Yew and Lee, (2020)	0.26	0.27	0.42	1.57	2.85	3.42	4.02
	Y	DeepGMR Yuan et al. (2020)	*0.02*	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**	**0.02**

Bold values in tables show the best performance.

TABLE 6 Experiment result of pose estimation on ModelNet40 data set (Wu et al., 2015) on the plane category. Two numbers are shown for GPU memory consumption and running speed for training/inference separately, given the same input size for two methods. Notice that the numbers are not directly comparable to Table 5 due to different experiment settings.

Methods	Memory (GB) ↓	Speed (fps) ↑	Mean error (°) ↓	Median error (°) ↓	Max error (°) ↓
EPN Chen et al. (2021)	22.2/16.9	1.1/1.6	1.25	1.11	6.63
Ours Zhu et al. (2022a)	**4.3/2.8**	**6.7/11.1**	**1.17**	**1.08**	**5.90**

Bold values in tables show the best performance.

regular G-CNNs (G for *group*) (Cohen and Welling, 2016), which lift the domain of the feature function space from the input Euclidean space to the group of transformations of interest. Second is steerable G-CNNs (Thomas et al., 2018), which leave the domain of the feature function space untouched but design the codomain to be *steerable* with the stabilizer subgroup. More detailed introductions can be found in the work of Cohen et al. (2018). The former strategy consumes much larger memory than a conventional CNN, while the latter usually results in complex design and restrictions on the kernel and convolution structure, both limiting broader applications in practice. We propose a new strategy to lift the domain of feature space to a proper subgroup of SE(3), and to apply a trivial steering representation on the subgroup, which addresses both problems mentioned above. Our proposed point-cloud convolution network learns expressive SE(3)-equivariant features with a much smaller footprint than existing methods. See Table 6 for a comparison between our method and a baseline regular G-CNN method, EPN (Chen et al., 2021).

To be more specific, our convolution structure is built upon KPConv (Thomas et al., 2019). We choose SO(2) as the stabilizer and work with feature maps defined on the domain $\tilde{X} = \mathrm{SE}(3)/\mathrm{SO}(2)$ which is homeomorphic to the Cartesian product $S^2 \times \mathbb{R}^3$. We extend the KPConv from $\mathbb{R}^3$ to $S^2 \times \mathbb{R}^3$. We discretize SO(3) into the icosahedral rotation group $\mathcal{I}$ with 60 elements, following EPN by Chen et al. (2021), containing all rotational symmetries of an icosahedron. SO(2) is discretized as the group of multiples of 72° planar rotations, which is a cyclic group of degree 5. Then we obtain a discretization of the sphere $\overline{S^2} = \overline{\mathrm{SO}(3)}/\mathrm{SO}(2)$ of size 12 corresponding to the vertices of an icosahedron, where $\bar{\cdot}$ (a top bar) denotes the discretized space. As a result, the domain of feature maps in our network is $\overline{S^2} \times \mathbb{R}^3$. It turns out that we can design an SE(3)-equivariant convolution in this space in a simple and efficient form while maintaining expressiveness. The full $\overline{\mathrm{SO}(3)}$ information can be recovered from the $\overline{S^2}$ feature maps through a permutation layer. An overview of the network structure is shown in Figure 14.

5.3 Place recognition via SE(3)-invariant representation

Place recognition, also known as loop closure detection, enables a robot to determine if it has seen a place before and provides loop closure candidates for SLAM algorithms to eliminate accumulated error. The widely used sensors include RGB, Stereo, Thermal, Event-Triggered, and RGB-D, which are in the form of 2D structured images or 3D unstructured points (Barros et al., 2021). For general tasks with 2D images, place recognition tasks suffer less because the training and testing images differ trivially in roll direction during data collecting procedures. Yet, the roll angles deviate significantly in

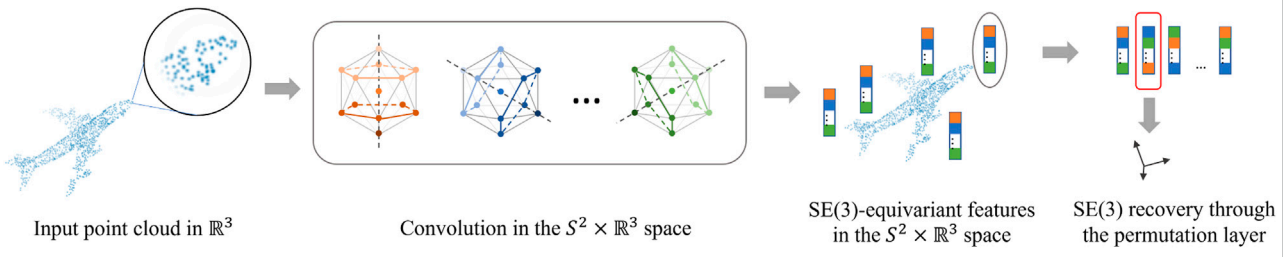

FIGURE 14
A high-level illustration of our efficient SE(3)-equivariant network. We lift the convolution to $S^2 \times \mathbb{R}^3$, which is a rare choice for SE(3)-equivariant feature learning. The different colors represent channels in S^2.

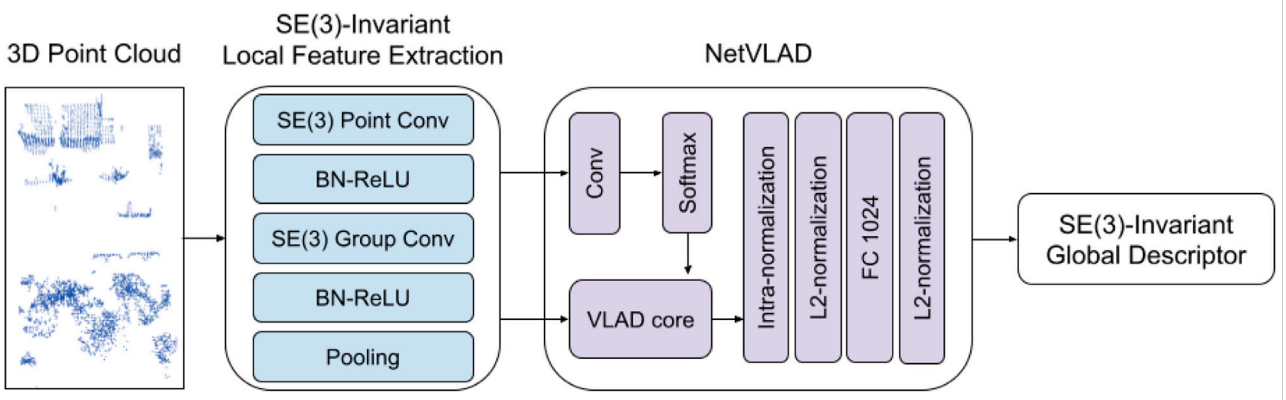

FIGURE 15
Overview of the SE(3)-invariant place recognition network. In this network, SE(3)-invariant features are learned from input point clouds. The local feature extraction consist of SE(3) point convolution, SE(3) group convolution, batch normalization followed by leaky ReLU activation, and pooling layer which makes equivariant features invariant. Global descriptors are computed by aggregating local features using NetVLAD. The output descriptors can perform place recognition tasks.

challenging scenarios like surgery (Song et al., 2021), underwater robot (Li et al., 2015) or special camera setup in general cases. Orientational differences widely exist and pose great difficulty to place recognition with 3D unstructured point cloud perception. Therefore, place recognition methods can benefit from a representation that is robust to arbitrary transformations of 3D point cloud data.

The image-based localization can be categorized as constructing hand-crafted rotation-invariant descriptors in 2D (Cummins and Newman, 2008; Gálvez-López and Tardos, 2012), learning the global descriptor (Kendall et al., 2015; Sünderhauf et al., 2015; Kim et al., 2017) or a combination of both (Tian et al., 2020; Song et al., 2022). Although learning-based methods achieve better accuracy and robustness, Lowry et al. (2015) suggested that place-recognition scenarios with large orientation differences still rely on hand-crafted descriptors which are designed for robust feature matching. This is especially true for 3D point clouds suffering more from orientation differences. Existing point cloud-based place recognition methods improve the transformation robustness by extracting 3D hand-crafted rotation-invariant descriptors (Kim and Kim, 2018; Wang et al., 2019; Yin et al., 2019; Kim et al., 2021) and randomly rotating them during training (Uy and Lee, 2018; Cattaneo et al., 2021). However, hand-crafted features can lose structural information and these methods do not take translation into consideration.

To avoid an exhaustive data augmentation with all possible transformations and improve generalizability, we propose an SE(3)-invariant place recognition representation network for the 3D point cloud. An overview of the network structure is shown in Figure 15. We use EPN (Chen et al., 2021) to extract SE(3)-invariant local features. NetVLAD (Arandjelovic et al., 2016) is applied to aggregate local features and construct SE(3)-invariant global descriptors.

We evaluate the proposed place representation using the Oxford RobotCar (Maddern et al., 2017) benchmark created by

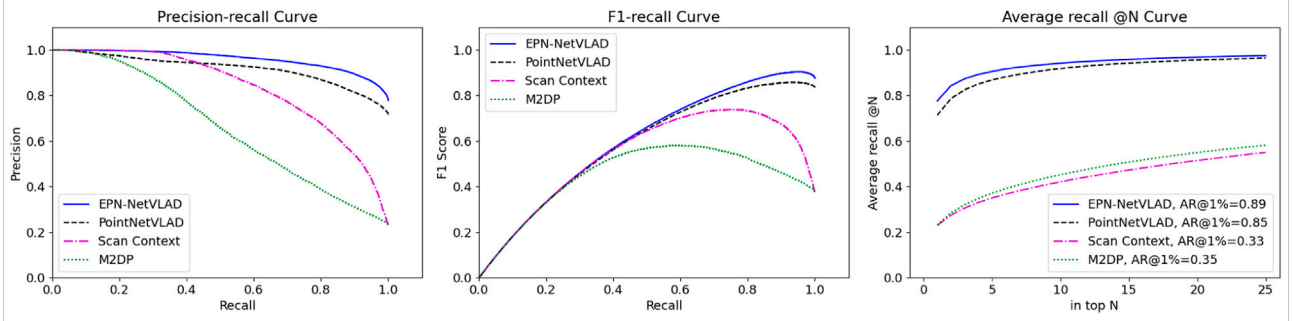

FIGURE 16
Experimental results of proposed method (EPN-NetVLAD, blue line), state-of-the-art approaches PointNetVLAD (Uy and Lee, 2018), Scan Context (Kim and Kim, 2018), and M2DP (He et al., 2016) on Oxford RobotCar benchmark.

TABLE 7 Experiment result showing the average recall (%) at top 1% for each of the models. Both methods are only trained on Oxford (Maddern et al., 2017) and tested on other different data sets (Uy and Lee, 2018).

Datasets	Oxford	U.S.	R.A.	B.D.
PointNetVLAD Uy and Lee, (2018)	84.94%	80.79%	73.86%	69.29%
EPN-NetVLAD (Ours)	**89.17%**	**87.82%**	**81.98%**	**76.91%**

Bold values in tables show the best performance.

Uy and Lee (2018). The precision-recall curves of the proposed method and other baseline methods are shown in Figure 16. The proposed network EPN-NetVLAD outperforms the baselines. To show the generlizability of the proposed method, we experiment with three in-house data sets of a university sector (U.S.), a residential area (R.A.) and a business district (B.D.) (Uy and Lee, 2018). The result is shown in Table 7 and our method performs better in all the data sets that we did not train on.

6 Closing remarks and future opportunities

Autonomy via computational intelligence is a multifaceted research domain that nicely integrates mathematics, computer science, and engineering and can have enormous impacts on our future and improve our quality of life. Robotics plays a unique role by connecting the real world to AI, i.e., embodied AI. Many challenges in robotics are natural problems in AI because they show what it takes to develop an autonomous system capable of operating in the wild. We reviewed some of the recent efforts in symmetry-preserving robot perception and control methods. In particular, by symmetry, we refer to invariance or equivariance properties under a group action enabled by Lie groups or their discrete subgroups.

The RKHS registration framework presented in Section 2 provides a unified model for registration that jointly integrates geometric and semantic measurements and does not require explicit data association. This framework is intimately connected with deep learning models. The inner product of the functions viewed as cross-correlation can be modeled as a network layer to combine the power of functional modeling with feature and kernel learning. Moreover, since the framework is equivariant, it can be directly combined with equivariant feature learners, e.g., via deep kernel learning (Wilson et al., 2016). An important open problem is a relationship among our framework, discrete-continuous smoothing and mapping (Doherty et al., 2022), dynamic scene graphs (Rosinol et al., 2021), and learning-aided smoothing and mapping (Huang et al., 2021) for robot perception and navigation. These are attractive research directions that we will explore in the future.

The learning-aided state estimation framework, presented in Section 3, can be extended to multi-task networks (Liu et al., 2019; Maninis et al., 2019; Hu and Singh, 2021) for tasks such as slip detection and friction coefficient estimation (Focchi et al., 2018; Romeo and Zollo, 2020), terrain classification (Hoepflinger et al., 2010; Walas et al., 2016; Wu et al., 2016; Ahmadi et al., 2021), covariance estimation (Brossard et al., 2020), sensor calibration and integration (Liu et al., 2020; Brossard et al., 2022; Ji et al., 2022), and motion mode detection (Brossard et al., 2019). A high-frequency implementation of these works on robots can significantly improve their capabilities for navigating challenging environments. Moreover, the work of Hwangbo et al. (2019) designs a learning-based controller using a policy network that maps kinematic observations and the joint state history to the joint position targets. Then an actuator network takes the joint velocity history and joint position error history to learn the joint torque. The success of Hwangbo et al. (2019) suggests that our multimodal approach to learning can improve the controller performance while further optimization of the contact estimation network size is possible.

In Section 4, we developed a new error-state MPC approach on connected matrix Lie groups for robot control.

By exploiting the existing symmetry of the pose control problem on Lie groups, we showed that the linearized tracking error dynamics and equations of motion in the Lie algebra are globally valid and evolve independently of the system trajectory. In addition, we formulated a convex MPC program for solving the problem efficiently using QP solvers. A Lyapunov function expressed in Lie algebra is introduced to verify the exponential stability of the proposed controller. The experimental results confirm that the proposed approach provides faster convergence when rotation and position are controlled simultaneously. Future work will implement the trajectory optimization using this geometric control framework proposed by Teng et al. (2022b) for robot control. Another interesting research direction is to incorporate learning into this framework (Shi et al., 2019; Li et al., 2022; Ma et al., 2022; O'Connell et al., 2022; Power and Berenson, 2022; Rodriguez et al., 2022). In addition, the IIG algorithm (Ghaffari Jadidi et al., 2019), combined with an MPC (Teng et al., 2021a), can provide an integrated kinodynamic planner that takes the robot stability, control constraints, and the value of information from sensory data into account. Gan et al. (2022) show that the value of information can be learned from multimodal sensory input via learning from demonstrations and self-supervised trajectory ranking to deal with sub-optimal demonstrations.

In Section 5, we showed how equivariant neural networks can serve as powerful feature learners to improve data efficiency and generalizability across different tasks. In particular, we provided results on registration and place recognition tasks. We argue that our efficient SE(3)-equivariant network (Zhu et al., 2022a) can be a reliable feature learner for a variety of robot perception and control problems, including those mentioned in this article. Furthermore, this symmetry-preserving representation can be an answer to the long-standing question of a "good" representation for robot mapping.

In addition to the point cloud-based SE(3)-invariant place recognition, it is of great interest to investigate the image-based version in challenging scenarios ranging from unstructured outdoors to endoscopy and colonoscopy (Song et al., 2021). Cohen and Welling (2016), Cohen et al. (2018) provide valuable insights on equipping the existing learning-based algorithms with group-invariant feature extraction ability. Traditional hand-crafted descriptors can be substituted with learnable deep SE(3)-invariant image descriptors. More importantly, we believe a natural future direction for robotics is towards developing structure-preserving and correct-by-construction computational models, such as our SE(3)-equivariant network, to enable efficient and generalizable multimodal learning.

Finally, this article aims to serve as an invitation to developing algorithms that respect the geometry of problems in robotics, preserve structures such as symmetry, and use modern computation methods such as deep learning. We presented methods ranging from purely geometric to end-to-end learning. As such, the central message of this paper is not about outperforming a particular framework, but it lies in the combined power of geometry and learning and the possibility of modeling traditional geometric problems using geometric networks such as equivariant deep networks. The latter will lead to explainable large-scale computational models for robotics and autonomous systems.

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 in the article/supplementary material.

Author contributions

MG developed the main narrative and led manuscript preparation. RZ and TianL developed the RKHS registration framework in Section 2. CL developed the SE(3)-invariant place recognition work in Section 5.3. T-YL and TinL developed the learning-aided InEKF in Section 3. ST developed the MPC on Lie group work in Section 4. JS developed the SE(3)-invariant place recognition work in Section 5.3 and helped with the organization of the paper.

Funding

Toyota Research Institute provided funds to support this work. Funding for M. Ghaffari was in part provided by NSF Award No. 2118818. This work was also supported by MIT Biomimetic Robotics Lab and NAVER LABS.

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

References

Agrawal, A., Chen, S., Rai, A., and Sreenath, K. (2021). Vision-aided dynamic quadrupedal locomotion on discrete terrain using motion libraries. *arXiv preprint arXiv:2110.00891* 13

Ahmadi, A., Nygaard, T., Kottege, N., Howard, D., and Hudson, N. (2021). Semi-supervised gated recurrent neural networks for robotic terrain classification. *IEEE Robot. Autom. Lett.* 6, 1848–1855. doi:10.1109/lra.2021.3060437

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. (2016). "NetVLAD: CNN architecture for weakly supervised place recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5297.

Barfoot, T. D., and Furgale, P. T. (2014). Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Trans. Robot.* 30, 679–693. doi:10.1109/tro.2014.2298059

Barrau, A., and Bonnabel, S. (2018). Invariant kalman filtering. *Annu. Rev. Control Robot. Auton. Syst.* 2, 237–257. doi:10.1146/annurev-control-060117-105010

Barrau, A., and Bonnabel, S. (2017). The invariant extended Kalman filter as a stable observer. *IEEE Trans. Autom. Contr.* 62 (2), 17978–21812. doi:10.1109/tac.2016.2594085

Barrau, A. (2015). *Non-linear state error based extended Kalman filters with applications to navigation*. Ph.D. thesis, 8. Mines Paristech.

Barros, T., Pereira, R., Garrote, L., Premebida, C., and Nunes, U. J. (2021). Place recognition survey: An update on deep learning approaches. *arXiv preprint arXiv: 2106.10458* 20

[Dataset] Blanco, J. L., and Rai, P. K. (2014). Nanoflann: a C++ header-only fork of FLANN. a library for nearest neighbor (NN) with kd-trees. Available at: https://github.com/jlblancoc/nanoflann 7.

Bloch, A., Krishnaprasad, P. S., Marsden, J. E., and Ratiu, T. S. (1996). The Euler-Poincaré equations and double bracket dissipation. *Commun. Math. Phys.* 175, 1–42. doi:10.1007/bf02101622

Bloch, A. M. (2015). *Nonholonomic mechanics and control*. New York, NY: Springer.

Bonnabel, S., Martin, P., and Rouchon, P. (2009). Non-linear symmetry-preserving observers on Lie groups. *IEEE Trans. Autom. Contr.* 54, 1709–1713. doi:10.1109/tac.2009.2020646

Brossard, M., Barrau, A., and Bonnabel, S. (2020). AI-IMU dead-reckoning. *IEEE Trans. Intell. Veh.* 511, 585–595. doi:10.1109/tiv.2020.2980758

Brossard, M., Barrau, A., and Bonnabel, S. (2019). "RINS-W: Robust inertial navigation system on wheels," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE), 22.11

Brossard, M., Barrau, A., Chauchat, P., and Bonnabel, S. (2022). Associating uncertainty to extended poses for on Lie group IMU preintegration with rotating earth. *IEEE Trans. Robot.* 38, 998–1015. doi:10.1109/tro.2021.3100156

Bullo, F., and Murray, R. M. (1999). Tracking for fully actuated mechanical systems: A geometric framework. *Automatica* 35, 17–34. doi:10.1016/s0005-1098(98)00119-8

Cattaneo, D., Vaghi, M., and Valada, A. (2021). LCDNet: Deep loop closure detection for LiDAR SLAM based on unbalanced optimal transport. *arXiv e-print , arXiv*–2103 20

Chen, H., Liu, S., Chen, W., Li, H., and Hill, R. (2021). "Equivariant point network for 3D point cloud analysis," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 21.20

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 834–848. doi:10.1109/tpami.2017.2699184

Chignoli, M., and Wensing, P. M. (2020). Variational-based optimal control of underactuated balancing for dynamic quadrupeds. *IEEE Access* 8, 49785–49797. doi:10.1109/access.2020.2980446

Chirikjian, G. S. (2011). *Stochastic models, information theory, and lie groups, volume 2: Analytic methods and modern applications*, 7. Springer Science & Business Media.

Cohen, T. S., Geiger, M., and Weiler, M. (2018). Intertwiners between induced representations (with applications to the theory of equivariant neural networks). *arXiv preprint arXiv:1803.10743* 19,23

Cohen, T., and Welling, M. (2016). Group equivariant convolutional networks. *Proc. Int. Conf. Mach. Learn.* 19, 23.

Cummins, M., and Newman, P. (2008). Fab-map: Probabilistic localization and mapping in the space of appearance. *Int. J. Rob. Res.* 27, 647–665. doi:10.1177/0278364908090961

Deng, C., Litany, O., Duan, Y., Poulenard, A., Tagliasacchi, A., and Guibas, L. (2021). Vector neurons: A general framework for SO(3)-equivariant networks. *arXiv preprint arXiv:2104.12229* 19

Di Carlo, J., Wensing, P. M., Katz, B., Bledt, G., and Kim, S. (2018). "Dynamic locomotion in the MIT cheetah 3 through convex model-predictive control," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE), 15.

Ding, Y., Pandala, A., Li, C., Shin, Y.-H., and Park, H.-W. (2021). Representation-free model predictive control for dynamic motions in quadrupeds. *IEEE Trans. Robot.* 37, 1154–1171. doi:10.1109/tro.2020.3046415

Dissanayake, G., Sukkarieh, S., Nebot, E., and Durrant-Whyte, H. (2001). The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications. *IEEE Trans. Rob. Autom.* 17, 731–747. doi:10.1109/70.964672

Doherty, K. J., Lu, Z., Singh, K., and Leonard, J. J. (2022). Discrete-continuous smoothing and mapping. *arXiv preprint arXiv:2204.11936* 22

Engel, J., Koltun, V., and Cremers, D. (2017). Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 611–625. doi:10.1109/tpami.2017.2658577

Fakoorian, S. A., Simon, D., Richter, H., and Azimi, V. (2016). "Ground reaction force estimation in prosthetic legs with an extended kalman filter," in 2016 Annual IEEE Systems Conference (SysCon) (IEEE).

Fink, G., and Semini, C. (2020). "Proprioceptive sensor fusion for quadruped robot state estimation," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE).

Focchi, M., Barasuol, V., Frigerio, M., Caldwell, D. G., and Semini, C. (2018). "Slip detection and recovery for quadruped robots," in *Robotics research* (Springer), 185.

Forster, C., Carlone, L., Dellaert, F., and Scaramuzza, D. (2016). On-manifold preintegration for real-time visual–inertial odometry. *IEEE Trans. Robot.* 33, 1–21. doi:10.1109/tro.2016.2597321

Gálvez-López, D., and Tardos, J. D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Trans. Robot.* 28, 1188–1197. doi:10.1109/tro.2012.2197158

Gan, L., Grizzle, J. W., Eustice, R. M., and Ghaffari, M. (2022). Energy-based legged robots terrain traversability modeling via deep inverse reinforcement learning. *IEEE Robot. Autom. Lett.* 7, 8807–8814. doi:10.1109/lra.2022.3188100

Geiger, A., Lenz, P., and Urtasun, R. (2012). "Are we ready for autonomous driving? The kitti vision benchmark suite," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6.5

Geiger, A., Roser, M., and Urtasun, R. (2010). "Efficient large-scale stereo matching," in Proceedings of the Asian Conference on Computer Vision.6

Ghaffari Jadidi, M., Valls Miro, J., and Dissanayake, G. (2019). Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring. *Int. J. Rob. Res.* 38, 658–685. doi:10.1177/0278364919844575

Ghaffari, M., Clark, W., Bloch, A., Eustice, R. M., and Grizzle, J. W. (2019). "Continuous direct sparse visual odometry from RGB-D images," in Proceedings of the Robotics: Science and Systems Conference, 3.2

[Dataset] Grupp, M. (2017). Evo: Python package for the evaluation of odometry and SLAM. Available at: https://github.com/MichaelGrupp/evo 6.

Hall, B. (2015). *Lie groups, lie algebras, and representations: An elementary introduction*, 222. Springer, 7.

Hartley, R., Ghaffari Jadidi, M., Gan, L., Huang, J.-K., Grizzle, J. W., and Eustice, R. M. (2018). "Hybrid contact preintegration for visual-inertial-contact state estimation using factor graphs," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE), 3783.

Hartley, R., Ghaffari, M., Eustice, R. M., and Grizzle, J. W. (2020). Contact-aided invariant extended kalman filtering for robot state estimation. *Int. J. Rob. Res.* 39, 402–430. doi:10.1177/0278364919894385

He, L., Wang, X., and Zhang, H. (2016). "M2dp: A novel 3D point cloud descriptor and its application in loop closure detection," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE), 231.

Hoepflinger, M. A., Remy, C. D., Hutter, M., Spinello, L., and Siegwart, R. (2010). "Haptic terrain classification for legged robots," in Proceedings of the IEEE International Conference on Robotics and Automation (IEEE), 2828.

Horn, B. K. P., Hilden, H. M., and Negahdaripour, S. (1988). Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A* 5, 1127–1135. doi:10.1364/josaa.5.001127

Hu, R., and Singh, A. (2021). "Unit: Multimodal multitask learning with a unified transformer," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1439.

Huang, Q., Pu, C., Fourie, D., Khosoussi, K., How, J. P., and Leonard, J. J. (2021). "NF-iSAM: Incremental smoothing and mapping via normalizing flows," in Proceedings of the IEEE International Conference on Robotics and Automation (IEEE), 1095.

Huang, X., Mei, G., and Zhang, J. (2020). "Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 11366.

Hwangbo, J., Lee, J., Dosovitskiy, A., Bellicoso, D., Tsounis, V., Koltun, V., et al. (2019). Learning agile and dynamic motor skills for legged robots. *Sci. Robot.* 4, eaau5872. doi:10.1126/scirobotics.aau5872

Ji, G., Mun, J., Kim, H., and Hwangbo, J. (2022). Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion. *IEEE Robot. Autom. Lett.* 7, 4630–4637. doi:10.1109/lra.2022.3151396

Kalabić, U., Gupta, R., Di Cairano, S., Bloch, A., and Kolmanovsky, I. (2016). "MPC on manifolds with an application to SE(3)," in Proceedings of the American Control Conference (IEEE), 7.

Kalabić, U. V., Gupta, R., Di Cairano, S., Bloch, A. M., and Kolmanovsky, I. V. (2017). MPC on manifolds with an application to the control of spacecraft attitude on SO(3). *Automatica* 76, 293–300. doi:10.1016/j.automatica.2016.10.022

Katz, B., Di Carlo, J., and Kim, S. (2019). "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in Proceedings of the IEEE International Conference on Robotics and Automation (IEEE), 15.

Kendall, A., Grimes, M., and Cipolla, R. (2015). "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in Proceedings of the IEEE International Conference on Computer Vision, 2938.

Kerl, C., Sturm, J., and Cremers, D. (2013). "Dense visual SLAM for RGB-D cameras," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE).

Kim, D., Di Carlo, J., Katz, B., Bledt, G., and Kim, S. (2019). Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. *arXiv preprint arXiv:1909.06586 9,17*

Kim, G., Choi, S., and Kim, A. (2021). Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments. *IEEE Trans. Robot.* 20, 1856–1874. doi:10.1109/tro.2021.3116424

Kim, G., Kim, A., Lim, J. Y., and Baek, H. C. (2018). Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map. *J. Nurs. Educ.* 4809, 21–27. doi:10.3928/01484834-20180102-05

Kim, H. J., Dunn, E., and Frahm, J.-M. (2017). "Learned contextual feature reweighting for image geo-localization," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE), 3251.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature* 521, 436–444. doi:10.1038/nature14539

Li, G., Tunchez, A., and Loianno, G. (2022). *Learning model predictive control for quadrotors. arXiv preprint arXiv:2202.07716* 22.

Li, J., Eustice, R. M., and Johnson-Roberson, M. (2015). "High-level visual features for underwater place recognition," in Proceedings of the IEEE International Conference on Robotics and Automation (IEEE), 3652.

Lin, T.-Y., Zhang, R., Yu, J., and Ghaffari, M. (2022). "Legged robot state estimation using invariant Kalman filtering and learned contact events," in Proceedings of the 5th Conference on Robot Learning.

Liu, K., Ok, K., Vega-Brown, W., and Roy, N. (2018). "Deep inference for covariance estimation: Learning Gaussian noise models for state estimation," in Proceedings of the IEEE International Conference on Robotics and Automation (IEEE), 1436.

Liu, S., Johns, E., and Davison, A. J. (2019). "End-to-end multi-task learning with attention," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Liu, W., Caruso, D., Ilg, E., Dong, J., Mourikis, A. I., Daniilidis, K., et al. (2020). Tlio: Tight learned inertial odometry. *IEEE Robot. Autom. Lett.* 5, 5653–5660. doi:10.1109/lra.2020.3007421

Long, A. W., Wolfe, K. C., Mashner, M. J., and Chirikjian, G. S. (2013). "The banana distribution is Gaussian: A localization study with exponential coordinates," in Proceedings of the Robotics: Science and Systems Conference, 7.265

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431.

Lowry, S., Sünderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P., et al. (2015). Visual place recognition: A survey. *IEEE Trans. Robot.* 32, 1–19. doi:10.1109/tro.2015.2496823

Lynch, K. M., and Park, F. C. (2017). *Modern robotics*. Cambridge University Press, 12.

Ma, H., Zhang, B., Tomizuka, M., and Sreenath, K. (2022). Learning differentiable safety-critical control using control barrier functions for generalization to novel environments. *arXiv preprint arXiv:2201.01347* 22

Maddern, W., Pascoe, G., Linegar, C., and Newman, P. (2017). 1 year, 1000 km: The Oxford robotcar dataset. *Int. J. Rob. Res.* 36, 3–15. doi:10.1177/0278364916679498

Magnusson, M., Lilienthal, A., and Duckett, T. (2007). Scan registration for autonomous mining vehicles using 3D-NDT. *J. Field Robot.* 24 (803–827 6), 803–827. doi:10.1002/rob.20204

Mahony, R., and Trumpf, J. (2021). Equivariant filter design for kinematic systems on lie groups. *IFAC-PapersOnLine* 54, 253–260. doi:10.1016/j.ifacol.2021.06.148

Mangelson, J. G., Ghaffari, M., Vasudevan, R., and Eustice, R. M. (2020). Characterizing the uncertainty of jointly distributed poses in the Lie algebra. *IEEE Trans. Robot.* 36, 1371–1388. doi:10.1109/tro.2020.2994457

Maninis, K.-K., Radosavovic, I., and Kokkinos, I. (2019). "Attentive single-tasking of multiple tasks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1851.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). "Occupancy networks: Learning 3D reconstruction in function space," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4460.

Mur-Artal, R., and Tardós, J. D. (2017). ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robot.* 33, 1255–1262. doi:10.1109/tro.2017.2705103

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. The MIT Press, 4.

O'Connell, M., Shi, G., Shi, X., Azizzadenesheli, K., Anandkumar, A., Yue, Y., et al. (2022). Neural-fly enables rapid learning for agile flight in strong winds. *Sci. Robot.* 7, eabm6597. doi:10.1126/scirobotics.abm6597

Park, J., Zhou, Q.-Y., and Koltun, V. (2017). "Colored point cloud registration revisited," in Proceedings of the IEEE International Conference on Computer Vision, 143.

Parkison, S. A., Ghaffari, M., Gan, L., Zhang, R., Ushani, A. K., and Eustice, R. M. (2019). Boosting shape registration algorithms via reproducing kernel Hilbert space regularizers. *IEEE Robot. Autom. Lett.* 4, 4563–4570. doi:10.1109/lra.2019.2932865

[Dataset] Pizenberg, M. (2019). DVO (without ROS dependency). Available at: https://github.com/mpizenberg/dvo/tree/76f65f0c9b438675997f595471d39863901556a97.

Potokar, E. R., Norman, K., and Mangelson, J. G. (2021). Invariant extended kalman filtering for underwater navigation. *IEEE Robot. Autom. Lett.* 6, 5792–5799. doi:10.1109/lra.2021.3085167

Power, T., and Berenson, D. (2022). Variational inference mpc using normalizing flows and out-of-distribution projection. *arXiv preprint arXiv:2205.04667* 22

Rodriguez, I. D. J., Ames, A. D., and Yue, Y. (2022). LyaNet: A Lyapunov framework for training neural ODEs. *arXiv preprint arXiv:2202.02526* 22

Romeo, R. A., and Zollo, L. (2020). Methods and sensors for slip detection in robotics: A survey. *IEEE Access* 8, 73027–73050. doi:10.1109/access.2020.2987849

Rosinol, A., Violette, A., Abate, M., Hughes, N., Chang, Y., Shi, J., et al. (2021). Kimera: From SLAM to spatial perception with 3D dynamic scene graphs. *Int. J. Rob. Res.* 40, 1510–1546. doi:10.1177/02783649211056674

Rosten, E., and Drummond, T. (2006). "Machine learning for high-speed corner detection," in Proceedings of the European Conference on Computer Vision (Springer).

Rusu, R. B., and Cousins, S. (2011). "3D is here: Point cloud library (PCL)," in Proceedings of the IEEE International Conference on Robotics and Automation.

Sarode, V., Li, X., Goforth, H., Aoki, Y., Srivatsan, R. A., Lucey, S., et al. (2019). PCRNet: Point cloud registration network using PointNet encoding. *ArXiv* abs/1908.07906 18

Segal, A., Haehnel, D., and Thrun, S. (2009). "Generalized-ICP," in Proceedings of the Robotics: Science and Systems Conference (Seattle, WA).

Servos, J., and Waslander, S. L. (2014). "Multi channel generalized-ICP," in Proceedings of the IEEE International Conference on Robotics and Automation (IEEE), 3644.

Shi, G., Shi, X., O'Connell, M., Yu, R., Azizzadenesheli, K., Anandkumar, A., et al. (2019). "Neural lander: Stable drone landing control using learned dynamics," in Proceedings of the IEEE International Conference on Robotics and Automation (IEEE).

Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., and Fitzgibbon, A. (2013). "Scene coordinate regression forests for camera relocalization in RGB-D images," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2930.

Song, J., Patel, M., and Ghaffari, M. (2022). Fusing convolutional neural network and geometric constraint for image-based indoor localization. *IEEE Robot. Autom. Lett.* 20, 1674–1681. doi:10.1109/lra.2022.3140832

Song, J., Patel, M., Girgensohn, A., and Kim, C. (2021). Combining deep learning with geometric features for image-based localization in the Gastrointestinal tract. *Expert Syst. Appl.* 185, 115631. doi:10.1016/j.eswa.2021.115631

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S. (2020). Osqp: An operator splitting solver for quadratic programs. *Math. Program. Comput.* 12, 637–672. doi:10.1007/s12532-020-00179-2

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). "A benchmark for the evaluation of RGB-D SLAM systems," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE).573

Sünderhauf, N., Shirazi, S., Dayoub, F., Upcroft, B., and Milford, M. (2015). "On the performance of ConvNet features for place recognition," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE), 4297.

Tapp, K. (2021). *Symmetry*. Springer, 2.

Teng, S., Chen, D., Clark, W., and Ghaffari, M. (2022a). An error-state model predictive control on connected matrix Lie groups for legged robot control. *arXiv preprint arXiv:2203.08728* 2,13

Teng, S., Clark, W., Bloch, A., Vasudevan, R., and Ghaffari, M. (2022b). Lie algebraic cost function design for control on Lie groups. *arXiv preprint arXiv: 2204.09177* 2, 13, 15, 22

Teng, S., Gong, Y., Grizzle, J. W., and Ghaffari, M. (2021a). Toward safety-aware informative motion planning for legged robots. *arXiv preprint arXiv: 2103.14252* 22

Teng, S., Mueller, M. W., and Sreenath, K. (2021b). "Legged robot state estimation in slippery environments using invariant extended kalman filter with velocity update," in Proceedings of the IEEE International Conference on Robotics and Automation (IEEE), 3104–311011.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., and Guibas, L. J. (2019). "KPConv: Flexible and deformable convolution for point clouds," in Proceedings of the IEEE International Conference on Computer Vision.19

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., et al. (2018). Tensor field networks: Rotation-and translation-equivariant neural networks for 3D point clouds. *arXiv preprint arXiv:1802.08219* 19

Tian, M., Nie, Q., and Shen, H. (2020). "3D scene geometry-aware constraint for camera localization with deep learning," in Proceedings of the IEEE International Conference on Robotics and Automation (IEEE), 4211.

Tsin, Y., and Kanade, T. (2004). "A correlation-based approach to robust point set registration," in European conference on computer vision (Springer), 558–5694.

Tu, L. W. (2011). *An introduction to manifolds*. Springer, 18.

Uy, M. A., and Lee, G. H. (2018). "PointNetVlad: Deep point cloud based retrieval for large-scale place recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 21.

Walas, K., Kanoulas, D., and Kryczka, P. (2016). "Terrain classification and locomotion parameters adaptation for humanoid robots using force/torque sensing," in Proceedings of the International Conference on Humanoid Robots (Humanoids) (IEEE), 133.

Wang, Y., Sun, Z., Xu, C.-Z., Sarma, S., Yang, J., and Kong, H. (2019). LiDAR iris for loop-closure detection. *arXiv preprint arXiv:1912.03825* 20

Wellhausen, L., Dosovitskiy, A., Ranftl, R., Walas, K., Cadena, C., and Hutter, M. (2019). Where should I walk? Predicting terrain properties from images via self-supervised learning. *IEEE Robot. Autom. Lett.* 4, 1509–1516. doi:10.1109/lra.2019.2895390

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. (2016). "Deep kernel learning," in Artificial intelligence and statistics (PMLR), 370.

Wu, G., and Sreenath, K. (2015). Variation-based linearization of nonlinear systems evolving on $SO(3)$ and $\mathbb S{2}$. *IEEE Access* 3, 1592–1604. doi:10.1109/access.2015.2477880

Wu, X. A., Huh, T. M., Mukherjee, R., and Cutkosky, M. (2016). Integrated ground reaction force sensing and terrain classification for small legged robots. *IEEE Robot. Autom. Lett.* 1, 1125–1132. doi:10.1109/lra.2016.2524073

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., et al. (2015). "2D shapenets: A deep representation for volumetric shapes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Yew, Z. J., and Lee, G. H. (2020). "RPM-Net: Robust point matching using learned features," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 11824.

Yin, H., Wang, Y., Ding, X., Tang, L., Huang, S., and Xiong, R. (2019). 3D LiDAR-based global localization using Siamese neural network. *IEEE Trans. Intell. Transp. Syst.* 21, 1380–1392. doi:10.1109/tits.2019.2905046

Yuan, W., Eckart, B., Kim, K., Jampani, V., Fox, D., and Kautz, J. (2020). "DeepGMR: Learning latent Gaussian mixture models for registration," in Proceedings of the European Conference on Computer Vision (Springer), 733.

Zhang, J., Yao, Y., and Deng, B. (2022). Fast and robust iterative closest point. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 3450–3466. doi:10.1109/TPAMI.2021.3054619

Zhang, R., Lin, T.-Y., Lin, C. E., Parkison, S. A., Clark, W., Grizzle, J. W., et al. (2021). "A new framework for registration of semantic point clouds from stereo and RGB-D cameras," in Proceedings of the IEEE International Conference on Robotics and Automation (IEEE).

Zhou, Q.-Y., Park, J., and Koltun, V. (2018). Open3D: A modern library for 3D data processing. *arXiv:1801.09847* 7

Zhu, M., Ghaffari, M., Clark, W. A., and Peng, H. (2022a). E^2PN: Efficient SE(3)-equivariant point network. *arXiv preprint arXiv:2206.05398 19,23*

Zhu, M., Ghaffari, M., and Peng, H. (2022b). "Correspondence-free point cloud registration with SO(3)-equivariant implicit shape representations," in Proceedings of the 5th Conference on Robot Learning, 19.

Zhu, Y., Sapra, K., Reda, F. A., Shih, K. J., Newsam, S., Tao, A., et al. (2019). "Improving semantic segmentation via video propagation and label relaxation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6.