frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 13 August 2018
doi: 10.3389/fnbot.2018.00049

Knowledge Transfer Between Artificial Intelligence Systems

Ivan Y. Tyukin[1,2]*, *Alexander N. Gorban*[1,2], *Konstantin I. Sofeykov*[1,3] and *Ilya Romanenko*[4]

[1] Department of Mathematics, University of Leicestger, Leicester, United Kingdom, [2] Laboratory of Advanced Methods for High-Dimensional Data Analysis, Lobachevsky State University of Nizhny Novgorod, Nizhny Novgorod, Russia, [3] Imaging and Vision Group, ARM Holdings, Loughborough, United Kingdom, [4] Spectral Edge Ltd, Cambridge, United Kingdom

We consider the fundamental question: how a legacy "student" Artificial Intelligent (AI) system could learn from a legacy "teacher" AI system or a human expert without re-training and, most importantly, without requiring significant computational resources. Here "learning" is broadly understood as an ability of one system to mimic responses of the other to an incoming stimulation and vice-versa. We call such learning an Artificial Intelligence knowledge transfer. We show that if internal variables of the "student" Artificial Intelligent system have the structure of an n-dimensional topological vector space and n is sufficiently high then, with probability close to one, the required knowledge transfer can be implemented by simple cascades of linear functionals. In particular, for n sufficiently large, with probability close to one, the "student" system can successfully and non-iteratively learn $k \ll n$ new examples from the "teacher" (or correct the same number of mistakes) at the cost of two additional inner products. The concept is illustrated with an example of knowledge transfer from one pre-trained convolutional neural network to another.

Keywords: stochastic separation theorems, concentration of measure, knowledge transfer in artificial intelligence systems, error correction, supervised learning, neural networks

OPEN ACCESS

Edited by:
Feihu Zhang,
Northwestern Polytechnical University, China

Reviewed by:
Xiaosu Hu,
University of Michigan, United States
Hong Zhang,
Indiana University, Purdue University Indianapolis, United States

***Correspondence:**
Ivan Y. Tyukin
i.tyukin@le.ac.uk

Received: *02 April 2018*
Accepted: *11 July 2018*
Published: *13 August 2018*

Citation:
Tyukin IY, Gorban AN, Sofeykov KI and Romanenko I (2018) Knowledge Transfer Between Artificial Intelligence Systems. Front. Neurorobot. 12:49. doi: 10.3389/fnbot.2018.00049

1. INTRODUCTION

Explosive development of neuroinformatics and Artificial Intelligence (AI) in recent years gives rise to new fundamental scientific and societal challenges. Developing technologies, professions, vocations, and corresponding educational environments for sustained generation of evergrowing number of AI Systems is currently recognized as amongst the most crucial of these (Hall and Pesenti, 2017). Nurturing and growing of relevant human expertise is considered as a way to address the challenge. The next step, however, is to develop technologies whereby one or several AI systems produce a training environment for the other leading to fully automated passage of knowledge and experience between otherwise independent AI agents.

Knowledge transfer between Artificial Intelligent systems has been the subject of extensive discussion in the literature for more than two decades (Gilev et al., 1991; Jacobs et al., 1991; Pratt, 1992; Schultz and Rivest, 2000; Buchtala and Sick, 2007) (see also a comprehensive review Pan and Yang, 2010). Several technical ideas to achieve AI knowledge transfer have been explored to date. Using or salvaging, parts of the "teacher" AI system in the "student" AI followed by re-training of the "student" has been proposed and extensively tested in Yosinski et al. (2014) and Chen et al. (2015). Alternatives to AI salvaging include model compression (Bucila et al., 2006), knowledge *distillation* (Hinton et al., 2015), and *privileged information* (Vapnik and Izmailov, 2017). These approaches demonstrated substantial success in improving generalization capabilities of AIs as well as in reducing computational overheads (Iandola et al., 2016), in cases of knowledge transfer from

larger AI to the smaller one. Notwithstanding, however, which of the above strategies is followed, their computational implementation, even for the case of transferring or learning just a handful of new examples, often requires either significant resources including access to large training sets and power needed for training, or availability of privileged information that may not necessarily be available to end-users. This contrasts sharply with natural intelligence too as recent empirical evidence reveals that single neurons in human brain are capable of rapid learning of new stimuli (Ison et al., 2015). Thus new frameworks and approaches are needed.

In this contribution we provide new framework for automated, fast, and non-destructive process of knowledge spreading across AI systems of varying architectures. In this framework, knowledge transfer is accomplished by means of Knowledge Transfer Units comprising of mere linear functionals and/or their small cascades. Main mathematical ideas are rooted in measure concentration (Gibbs, 1902; Lévy, 1951; Gromov, 1999, 2003; Gorban, 2007) and stochastic separation theorems (Gorban and Tyukin, 2017, 2018) revealing peculiar properties of random sets in high dimensions. We generalize some of the latter results here and show how these generalizations can be employed to build simple one-shot Knowledge Transfer algorithms between heterogeneous AI systems whose state may be represented by elements of linear vector space of sufficiently high dimension. Once knowledge has been transferred from one AI to another, the approach also allows to "unlearn" new knowledge without the need to store a complete copy of the "student" AI is created prior to learning. We expect that the proposed framework may pave way for fully functional new phenomenon—Nursery of AI systems in which AIs quickly learn from each other whilst keeping their pre-existing skills largely intact.

The paper is organized as follows. In section 2 we introduce a general framework for computationally efficient non-iterative AI Knowledge Transfer and present two algorithms for transferring knowledge between a pair of AI systems in which one operates as a teacher and the other functions as a student. These results are based on Stochastic Separation Theorems (Gorban and Tyukin, 2017) of which the relevant versions are provided here as mathematical background justifying the approach. Section 3 illustrates the approach with examples, and section 4 concludes the paper.

2. NON-ITERATIVE AI KNOWLEDGE TRANSFER FRAMEWORK

2.1. General Setup

Consider two AI systems, a student AI, denoted as AI_s, and a teacher AI, denoted as AI_t. These legacy AI systems process some *input* signals, produce *internal* representations of the input and return some *outputs*. We further assume that some *relevant* information about the input, internal signals, and outputs of AI_s can be combined into a common object, $\boldsymbol{x}$, representing, but not necessarily defining, the *state* of AI_s. The objects $\boldsymbol{x}$ are assumed to be elements of $\mathbb{R}^n$.

Over a period of activity system AI_s generates a set $\mathcal{S}$ of objects $\boldsymbol{x}$. Exact composition of the set $\mathcal{S}$ could depend on a task at hand. For example, if AI_s is an image classifier, we may be interested only in a particular subset of AI_s input-output data related to images of a certain known class. Relevant inputs and outputs of AI_s corresponding to objects in $\mathcal{S}$ are then evaluated by the teacher, AI_t. If AI_s outputs differ to that of AI_t for the same input then an error is registered in the system. Objects $\boldsymbol{x} \in \mathcal{S}$ associated with errors are combined into the set $\mathcal{Y}$. The procedure gives rise to two disjoint sets:

$$\mathcal{M} = \mathcal{S} \setminus \mathcal{Y},\ \mathcal{M} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$$

and

$$\mathcal{Y} = \{\boldsymbol{x}_{M+1}, \ldots, \boldsymbol{x}_{M+k}\}.$$

A diagram schematically representing the process is shown in **Figure 1**. The knowledge transfer task is to "teach" AI_s so that with

a) AI_s does not make such errors
b) existing competencies of AI_s on the set of inputs corresponding to internal states $\boldsymbol{x} \in \mathcal{M}$ are retained, and
c) knowledge transfer from AI_t to AI_s is reversible in the sense that AI_s can "unlearn" new knowledge by modifying just a fraction of its parameters, if required.

Before proceeding with a proposed solution to the above AI Knowledge Transfer problem, understanding basic yet fundamental properties of the sets $\mathcal{Y}$ and $\mathcal{M}$ is needed. These properties are summarized and illustrated with Theorems 1, 2, and 3 below.

2.2. Stochastic Separation Theorems for Non-iterative AI Knowledge Transfer

Let the set

$$\mathcal{M} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$$

be an i.i.d. sample from a distribution in $\mathbb{R}^n$. Pick another set

$$\mathcal{Y} = \{\boldsymbol{x}_{M+1}, \ldots, \boldsymbol{x}_{M+k}\}$$

from the same distribution at random. What is the probability that there is a linear functional separating $\mathcal{Y}$ from $\mathcal{M}$, and, most importantly, if there is a computationally simple and efficient way to determine these?

Below we provide three k-tuple separation theorems: for an equidistribution in the unit ball $B_n(1)$ (Theorems 1 and 2) and for a product probability measure with bounded support (Theorem 3). These two special cases cover or, indeed, approximate a broad range of practically relevant situations including e.g., Gaussian distributions (reduce asymptotically to the equidistribution in $B_n(1)$ for n large enough) and data vectors in which each attribute is a numerical and independent random variable. The computational complexity for determining the separating functionals, as specified by the theorems and their proofs, can be

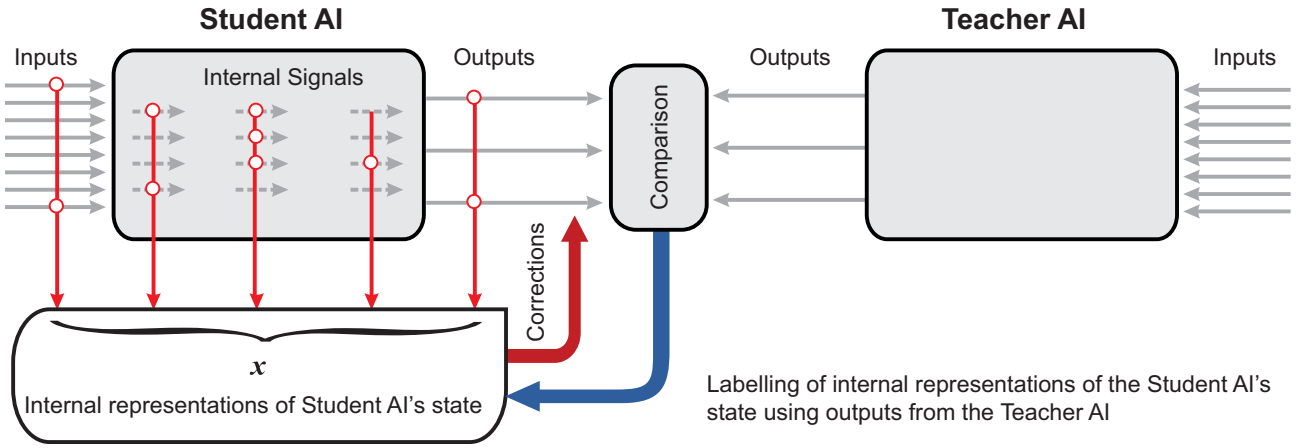

FIGURE 1 | AI Knowledge transfer diagram. AI_S produces a set of its state representations, $\mathcal{S}$. The representations are labeled by AI_t into the set of correct responses, $\mathcal{M}$, and the set of errors, $\mathcal{Y}$. The student system, AI_S, is then augmented by an additional "corrector" eliminating these errors.

remarkably low. If no pre-processing is involved, then deriving the functionals stemming from Theorems 1 and 2 requires merely $k = |\mathcal{Y}|$ vector additions and, possibly, an approximate solution of a constrained optimization problem in two dimensions. For large data sets, this is a significant advantage over support vector machines whose worst-case computational complexity is $O((k+M)^3)$ (Bordes et al., 2005; Chapelle, 2007).

Consider the case when the underlying probability distribution is an equidistribution in the unit ball $B_n(1)$, and suppose that $\mathcal{M} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ and $\mathcal{Y} = \{\boldsymbol{x}_{M+1}, \ldots, \boldsymbol{x}_{M+k}\}$ are i.i.d. samples from this distribution. We are interested in determining the probability $\mathcal{P}_1(\mathcal{M}, \mathcal{Y})$ that there exists a linear functional separating $\mathcal{M}$ and $\mathcal{Y}$. An estimate of this probability is provided in the following theorem.

Theorem 1. *Let $\mathcal{M} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ and $\mathcal{Y} = \{\boldsymbol{x}_{M+1}, \ldots, \boldsymbol{x}_{M+k}\}$ be i.i.d. samples from the equidisribution in $B_n(1)$. Then*

$$\begin{aligned}\mathcal{P}_1(\mathcal{M}, \mathcal{Y}) &\geq \max_{\delta,\varepsilon} 1 - k(1-\varepsilon)^n - \frac{(k-1)k}{2}(1-\delta^2)^{\frac{n}{2}} \\ &\quad - \frac{M}{2}\Delta(\varepsilon, \delta, k)^{\frac{n}{2}} \\ \Delta(\varepsilon, \delta, k) &= 1 - \frac{1}{k}\left[\frac{(1-\varepsilon)^2 - \frac{k-1}{1-\varepsilon}\delta}{\sqrt{1 + \frac{k-1}{1-\varepsilon}\delta}}\right]^2 \qquad (1) \\ &\quad \text{Subject to :} \\ &\quad \delta, \varepsilon \in (0,1) \\ &\quad (k-1)\delta \leq (1-\varepsilon)^3.\end{aligned}$$

If the pair (δ, ε) is a solution of the nonlinear optimization program in (1) then the corresponding separating hyperplane is:

$$\ell_0(\boldsymbol{x}) = 0,\ \ell_0(\boldsymbol{x}) = \left\langle \frac{\bar{\boldsymbol{x}}}{\|\bar{\boldsymbol{x}}\|}, \boldsymbol{x} \right\rangle - \frac{1}{\sqrt{k}} \frac{(1-\varepsilon)^2 - \frac{k-1}{1-\varepsilon}\delta}{\sqrt{1 + \frac{k-1}{1-\varepsilon}\delta}},$$

$$\bar{\boldsymbol{x}} = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{x}_{M+i}.$$

The proof of the theorem is provided in the **Appendix**.

Figure 2 shows how estimate (1) of the probability $\mathcal{P}_1(\mathcal{M}, \mathcal{Y})$ behaves, as a function of $|\mathcal{Y}|$ for fixed M and n. As one can see from this figure, when k exceeds some critical value ($k = 9$ in this specific case), the lower bound estimate (1) of the probability $\mathcal{P}_1(\mathcal{M}, \mathcal{Y})$ drops. This is not surprising since the bound (1) is (A) based on conservative estimates, and (B) these estimates are derived for just one class of separating hyperplanes $\ell_0(\boldsymbol{x})$. Furthermore, no prior pre-processing and/or clustering was assumed for the $\mathcal{Y}$. An alternative estimate that allows us to account for possible clustering in the set $\mathcal{Y}$ is presented in Theorem 2.

Theorem 2. *Let $\mathcal{M} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ and $\mathcal{Y} = \{\boldsymbol{x}_{M+1}, \ldots, \boldsymbol{x}_{M+k}\}$ be i.i.d. samples from the equidisribution in $B_n(1)$. Let $\mathcal{Y}_c = \{\boldsymbol{x}_{M+r_1}, \ldots, \boldsymbol{x}_{M+r_m}\}$ be a subset of m elements from $\mathcal{Y}$ such that*

$$\beta_2 (m-1) \leq \sum_{r_j, r_j \neq r_i} \langle \boldsymbol{x}_{M+r_i}, \boldsymbol{x}_{M+r_j} \rangle \leq \beta_1 (m-1) \text{ for all } i = 1, \ldots, m. \qquad (2)$$

Then

$$\begin{aligned}\mathcal{P}_1(\mathcal{M}, \mathcal{Y}_c) &\geq \max_{\varepsilon} (1-(1-\varepsilon)^n)^k \left(1 - \frac{\Delta(\varepsilon, m)^{\frac{n}{2}}}{2}\right)^M \\ \Delta(\varepsilon, m) &= 1 - \frac{1}{m}\left(\frac{(1-\varepsilon)^2 + \beta_2(m-1)}{\sqrt{1 + (m-1)\beta_1}}\right)^2 \qquad (3) \\ &\quad \text{Subject to :} \\ &\quad (1-\varepsilon)^2 + \beta_2(m-1) > 0 \\ &\quad 1 + (m-1)\beta_1 > 0.\end{aligned}$$

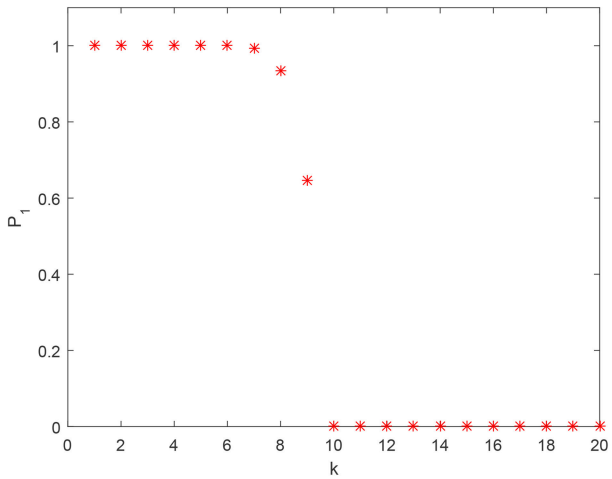

FIGURE 2 | Estimate (1) of $\mathcal{P}_1(\mathcal{M}, \mathcal{Y})$ as a function of k for $n = 2,000$ and $M = 10^5$.

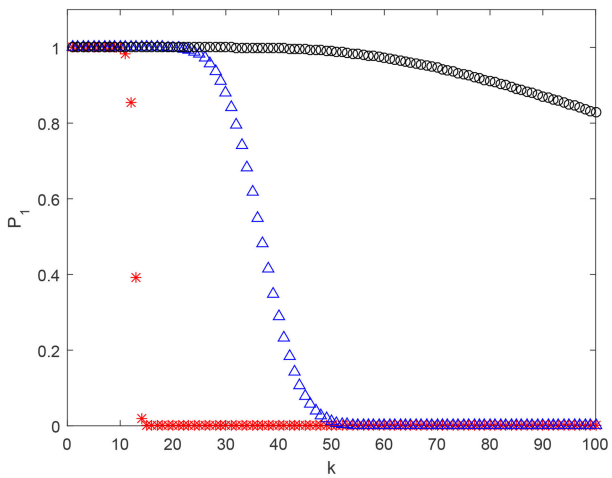

FIGURE 3 | Estimate (3) of $\mathcal{P}_1(\mathcal{M}, \mathcal{Y})$ as a function of k for $n = 2,000$ and $M = 10^5$. Red stars correspond to $\beta_1 = 0.5$, $\beta_2 = 0$. Blue triangles stand for $\beta_1 = 0.5$, $\beta_2 = 0.05$, and black circles stand for $\beta_1 = 0.5$, $\beta_2 = 0.07$.

If the pair (δ, ε) is a solution of the nonlinear optimization program in (3) then the corresponding separating hyperplane is:

$$\ell_0(\boldsymbol{x}) = 0,\ \ell_0(\boldsymbol{x}) = \left\langle \frac{\bar{\boldsymbol{y}}}{\|\bar{\boldsymbol{y}}\|}, \boldsymbol{x} \right\rangle - \frac{1}{\sqrt{m}} \left(\frac{(1-\varepsilon)^2 + \beta_2(m-1)}{\sqrt{1+(m-1)\beta_1}} \right),$$

$$\bar{\boldsymbol{y}} = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{x}_{M+r_i}.$$

The proof of the theorem is provided in **Appendix**.

Examples of estimates (3) for various parameter settings are shown in **Figure 3**. As one can see, in absence of pair-wise strictly positive correlation assumption, $\beta_2 = 0$, the estimate's behavior, as a function of k, is similar to that of (1). However, presence of moderate pair-wise positive correlation results in significant boosts to the values of $\mathcal{P}_1$.

Remark 1. Estimates (1), (3) for the probability $\mathcal{P}_1(\mathcal{M}, \mathcal{Y})$ that follow from Theorems 1, 2 assume that the underlying probability distribution is an equidistribution in $B_n(1)$. They can, however, be generalized to equidistributions in ellipsoids and Gaussian distributions (cf. Gorban et al., 2016a,b). Tighter probability bounds could also be derived if the upper-bound estimates of the volumes of the corresponding spherical caps in the proofs of Theorems 1, 2 are replaced with their exact values (see e.g., Li, 2011).

Remark 2. Note that not only Theorems 1, 2 provide estimates from below of the probability that two random i.i.d. drawn samples from $B_n(1)$ are linearly separable, but also they explicitly present the separating hyperplanes. The latter hyperplanes are similar to Fisher linear discriminants in that the discriminating direction (normal to the hyperplane) is the difference between the centroids.

Whilst having explicit separation functionals as well as thresholds is an obvious advantage from practical view point, the estimates that are associated with such functionals do not account for more flexible alternatives. In what follows we present a generalization of the above results that accounts for such a possibility as well as extends applicability of the approach to samples from product distributions. The results are provided in Theorem 3.

Theorem 3. *Consider the linear space $E = \mathrm{span}\{\boldsymbol{x}_j - \boldsymbol{x}_{M+1} \mid j = M+2, \ldots, M+k\}$, let the cardinality $|\mathcal{Y}| = k$ of the set $\mathcal{Y}$ be smaller than n. Consider the quotient space $\mathbb{R}^n/E$. Let $Q(\boldsymbol{x})$ be a representation of $\boldsymbol{x} \in \mathbb{R}^n$ in $\mathbb{R}^n/E$, and let the coordinates of $Q(\boldsymbol{x}_i)$, $i = 1, \ldots, M+1$ be independent random variables i.i.d. sampled from a product distribution in a unit cube with variances $\sigma_j > \sigma_0 > 0$, $1 \le j \le n-k+1$. Then for*

$$M \le \frac{\vartheta}{3} \exp\left(\frac{(n-k+1)\sigma_0^4}{2} \right) - 1$$

with probability $p > 1 - \vartheta$ there is a linear functional separating $\mathcal{Y}$ and $\mathcal{M}$.

The proof of the theorem is provided in **Appendix**.

Having introduced Theorems 1–3, we are now ready to formulate our main results–algorithms for non-iterative AI Knowledge Transfer.

2.3. Knowledge Transfer Algorithms

Our first algorithm, Algorithm 1, considers cases when *Auxiliary Knowledge Transfer Units*, i.e. functional additions to existing student AI$_s$, are single linear functionals. The second algorithm, Algorithm 2, extends Auxiliary Knowledge Transfer Units to two-layer cascades of linear functionals.

The algorithms comprise of two general stages, pre-processing stage and knowledge transfer stage. The purpose of the pre-processing stage is to regularize and "sphere"

Algorithm 1 Single-functional AI Knowledge Transfer

1. **Pre-processing**

(a) *Centering*. For the given set $\mathcal{S}$, determine the set average, $\bar{\boldsymbol{x}}(\mathcal{S})$, and generate sets $\mathcal{S}_c$

$$\begin{aligned}\mathcal{S}_c &= \{\boldsymbol{x} \in \mathbb{R}^n \,|\boldsymbol{x} = \boldsymbol{\xi} - \bar{\boldsymbol{x}}(\mathcal{S}),\ \boldsymbol{\xi} \in \mathcal{S}\},\\ \mathcal{Y}_c &= \{\boldsymbol{x} \in \mathbb{R}^n \,|\boldsymbol{x} = \boldsymbol{\xi} - \bar{\boldsymbol{x}}(\mathcal{S}),\ \boldsymbol{\xi} \in \mathcal{Y}\}.\end{aligned}$$

(b) *Regularization*. Determine covariance matrices $\mathrm{Cov}(\mathcal{S}_c)$, $\mathrm{Cov}(\mathcal{S}_c \setminus \mathcal{Y}_c)$ of the sets $\mathcal{S}_c$ and $\mathcal{S}_c \setminus \mathcal{Y}_c$. Let $\lambda_i(\mathrm{Cov}(\mathcal{S}_c))$, $\lambda_i(\mathrm{Cov}(\mathcal{S}_c \setminus \mathcal{Y}_c))$ be their corresponding eigenvalues, and $h_1, \ldots, h_n$ be the eigenvectors of $\mathrm{Cov}(\mathcal{S}_c)$. If some of $\lambda_i(\mathrm{Cov}(\mathcal{S}_c))$, $\lambda_i(\mathrm{Cov}(\mathcal{S}_c \setminus \mathcal{Y}_c))$ are zero or if the ratio $\frac{\max_i\{\lambda_i(\Sigma(\mathcal{S}_c))\}}{\min_i\{\lambda_i(\Sigma(\mathcal{S}_c))\}}$ is too large, project $\mathcal{S}_c$ and $\mathcal{Y}_c$ onto appropriately chosen set of $m < n$ eigenvectors, $h_{n-m+1}, \ldots, h_n$:

$$\begin{aligned}\mathcal{S}_r &= \{\boldsymbol{x} \in \mathbb{R}^n \,|\boldsymbol{x} = H^T\boldsymbol{\xi},\ \boldsymbol{\xi} \in \mathcal{S}_c\},\\ \mathcal{Y}_r &= \{\boldsymbol{x} \in \mathbb{R}^n \,|\boldsymbol{x} = H^T\boldsymbol{\xi},\ \boldsymbol{\xi} \in \mathcal{Y}_c\},\end{aligned}$$

where $H = \left(h_{n-m+1} \cdots h_n\right)$ is the matrix comprising of m significant principal components of $\mathcal{S}_c$.

(c) *Whitening*. For the centered and regularized dataset $\mathcal{S}_r$, derive its covariance matrix, $\mathrm{Cov}(\mathcal{S}_r)$, and generate whitened sets

$$\begin{aligned}\mathcal{S}_w &= \{\boldsymbol{x} \in \mathbb{R}^m \,|\boldsymbol{x} = \mathrm{Cov}(\mathcal{S}_r)^{-\frac{1}{2}}\boldsymbol{\xi},\ \boldsymbol{\xi} \in \mathcal{S}_r\},\\ \mathcal{Y}_w &= \{\boldsymbol{x} \in \mathbb{R}^m \,|\boldsymbol{x} = \mathrm{Cov}(\mathcal{S}_r)^{-\frac{1}{2}}\boldsymbol{\xi},\ \boldsymbol{\xi} \in \mathcal{Y}_r\},\end{aligned}$$

2. **Knowledge transfer**

(a) *Clustering*. Pick $p \geq 1$, $p \leq k$, $p \in \mathbb{N}$, and partition the set $\mathcal{Y}_w$ into p clusters $\mathcal{Y}_{w,1}, \ldots \mathcal{Y}_{w,p}$ so that elements of these clusters are, on average, pairwise positively correlated. That is there are $\beta_1 \geq \beta_2 > 0$ such that:

$$\beta_2(|\mathcal{Y}_{w,i}| - 1) \leq \sum_{\xi \in \mathcal{Y}_{w,i}\setminus\{\boldsymbol{x}\}} \langle \boldsymbol{\xi}, \boldsymbol{x}\rangle \leq \beta_1(|\mathcal{Y}_{w,i}| - 1) \text{ for any } \boldsymbol{x} \in \mathcal{Y}_{w,i}$$

(b) *Construction of Auxiliary Knowledge Units*. For each cluster $\mathcal{Y}_{w,i}$, $i = 1, \ldots, p$, construct separating functionals ℓ_i:

$$\begin{aligned}\ell_i(\boldsymbol{x}) &= \left\langle \frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}, \boldsymbol{x}\right\rangle - c_i,\\ \boldsymbol{w}_i &= \left(\mathrm{Cov}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}) + \mathrm{Cov}(\mathcal{Y}_{w,i})\right)^{-1}\left(\bar{\boldsymbol{x}}(\mathcal{Y}_{w,i}) - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i})\right)\end{aligned}$$

where $\bar{\boldsymbol{x}}(\mathcal{Y}_{w,i})$, $\bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i})$ are the averages of $\mathcal{Y}_{w,i}$ and $\mathcal{S}_w \setminus \mathcal{Y}_{w,i}$, respectively, and c_i is chosen as $c_i = \min_{\boldsymbol{\xi} \in \mathcal{Y}_{w,i}} \left\langle \frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}, \boldsymbol{\xi}\right\rangle$.

(c) *Integration*. Integrate Auxiliary Knowledge Units into decision-making pathways of AI_s. If, for an $\boldsymbol{x}$ generated by an input to AI_s, any of $\ell_i(\boldsymbol{x}) \geq 0$ then report $\boldsymbol{x}$ accordingly (swap labels, report as an error etc.)

the data. This operation brings the setup close to the one considered in statements of Theorems 1, 2. The knowledge transfer stage constructs Auxiliary Knowledge Transfer Units in a way that is very similar to the argument presented in the proofs of Theorems 1 and 2. Indeed, if $|\mathcal{Y}_{w,i}| \ll |\mathcal{S}_w \setminus \mathcal{Y}_{w,i}|$ then the term $\left(\mathrm{Cov}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}) + \mathrm{Cov}(\mathcal{Y}_{w,i})\right)^{-1}$ is close to identity matrix, and the functionals ℓ_i are good approximations of (10). In this setting, one might expect that performance of the knowledge transfer stage would be also closely aligned with the corresponding estimates (1), (3).

Remark 3. Note that the regularization step in the pre-processing stage ensures that the matrix $\mathrm{Cov}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}) + \mathrm{Cov}(\mathcal{Y}_{w,i})$ is non-singular. Indeed, consider

$$\begin{aligned}&\mathrm{Cov}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}) = \tfrac{1}{|\mathcal{S}_w\setminus\mathcal{Y}_{w,i}|}\textstyle\sum_{\boldsymbol{x}\in\mathcal{S}_w\setminus\mathcal{Y}_{w,i}}(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}))\\ &(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}))^T = \tfrac{1}{|\mathcal{S}_w\setminus\mathcal{Y}_{w,i}|}\Big(\textstyle\sum_{\boldsymbol{x}\in\mathcal{S}_w\setminus\mathcal{Y}_w}(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}))\\ &(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}))^T + \textstyle\sum_{\boldsymbol{x}\in\mathcal{Y}_w\setminus\mathcal{Y}_{w,i}}(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}))\\ &(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}))^T\Big).\end{aligned}$$

Denoting $d = \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}) - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_w)$ and rearranging the sum below as

$$\begin{aligned}&\textstyle\sum_{\boldsymbol{x}\in\mathcal{S}_w\setminus\mathcal{Y}_w}(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}))(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}))^T =\\ &\textstyle\sum_{\boldsymbol{x}\in\mathcal{S}_w\setminus\mathcal{Y}_w}(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_w) + d)(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_w) + d)^T =\\ &\textstyle\sum_{\boldsymbol{x}\in\mathcal{S}_w\setminus\mathcal{Y}_w}(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_w))(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_w))^T +\\ &2d\textstyle\sum_{\boldsymbol{x}\in\mathcal{S}_w\setminus\mathcal{Y}_w}(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_w))^T + |\mathcal{S}_w \setminus \mathcal{Y}_w|dd^T\\ &= \textstyle\sum_{\boldsymbol{x}\in\mathcal{S}_w\setminus\mathcal{Y}_w}(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_w))(\boldsymbol{x} - \bar{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_w))^T\\ &+|\mathcal{S}_w \setminus \mathcal{Y}_w|dd^T\end{aligned}$$

we obtain that $\mathrm{Cov}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i})$ is non-singular as long as the sum $\sum_{\boldsymbol{x}\in\mathcal{S}_w\setminus\mathcal{Y}_w}(\boldsymbol{x}-\bar{\boldsymbol{x}}(\mathcal{S}_w\setminus\mathcal{Y}_w))(\boldsymbol{x}-\bar{\boldsymbol{x}}(\mathcal{S}_w\setminus\mathcal{Y}_w))^T$ is non-singular. The latter property, however, is guaranteed by the regularization step in Algorithm 1.

Remark 4. Clustering at Step 2.a can be achieved by classical k-means algorithms (Lloyd, 1982) or any other method (see e.g., Duda et al., 2000) that would group elements of $\mathcal{Y}_w$ into clusters according to spatial proximity.

Remark 5. Auxiliary Knowledge Transfer Units in Step 2.b of Algorithm 1 are derived in accordance with standard Fisher linear discriminant formalism. This, however, need not be the case, and other methods, e.g., support vector machines (Vapnik, 2000), could be employed for this purpose there. It is worth mentioning, however, that support vector machines might be prone to overfitting (Han, 2014) and their training often involves iterative procedures such as sequential quadratic minimization (Platt, 1999).

Furthermore, instead of the sets $\mathcal{Y}_{w,i}$, $\mathcal{S}_w \setminus \mathcal{Y}_{w,i}$ one could use a somewhat more aggressive division: $\mathcal{Y}_{w,i}$ and $\mathcal{S}_w \setminus \mathcal{Y}_w$, respectively.

Depending on configuration of samples $\mathcal{S}$ and $\mathcal{Y}$, Algorithm 1 may occasionally create Knowledge Transfer Units, ℓ_i, that are "filtering" errors too aggressively. That is, some $\boldsymbol{x} \in \mathcal{S}_w \setminus \mathcal{Y}_w$ may accidentally trigger non-negative response, $\ell_i(\boldsymbol{x}) \geq 0$, and as a result of this, their corresponding inputs to A_s could be ignored or mishandled. To mitigate this, one can increase the number of clusters and Knowledge Transfer Units, respectively. This will increase the probability of successful separation and hence alleviate the issue. An alternative practical strategy to limit the number of Knowledge Transfer Units, when the system is evolving in time, is to retain only most relevant ones taking into account acceptable rates of performance and size of the "relevant" set $\mathcal{S}$. The link between these is provided in Theorems 1, 2. On the other hand, if increasing the number of Knowledge Transfer Units or dismissing less relevant ones is not desirable

for some reason, then two-functional units could be a feasible remedy. Algorithm 2 presents a procedure for such an improved AI Knowledge Transfer.

Algorithm 2 Two-functional AI Knowledge Transfer

1. **Pre-processing**. Do as in Step 1 in Algorithm 1
2. **Knowledge Transfer**

(a) *Clustering*. Do as in Step 2.a in Algorithm 1

(b) *Construction of Auxiliary Knowledge Units*.

1: Do as in Step 2.b in Algorithm 1. At the end of this step *first-stage* functionals ℓ_i, $i = 1, \ldots, p$ will be derived.

2: For each set $\mathcal{Y}_{w,i}$, $i = 1, \ldots, p$, evaluate the functionals ℓ_i for all $\boldsymbol{x} \in \mathcal{S}_w \setminus \mathcal{Y}_{w,i}$ and identify elements $\boldsymbol{x}$ such that $\ell_i(\boldsymbol{x}) \geq 0$ and $\boldsymbol{x} \in \mathcal{S}_w \setminus \mathcal{Y}_w$ (incorrect error assignment). Let $\mathcal{Y}_{e,i}$ be the set containing such elements $\boldsymbol{x}$.

3: **If** (there is an $i \in \{1, \ldots, p\}$ such that $|\mathcal{Y}_{e,i}| + |\mathcal{Y}_{w,i}| > m$) **then** increment the value of p: $p \leftarrow p + 1$, and return to Step 2.a.

4: **If** (all sets $\mathcal{Y}_{e,i}$ are empty) **then** proceed to Step 2.c.

5: For each pair of ℓ_i and $\mathcal{Y}_{w,i} \cup \mathcal{Y}_{e,i}$ with $\mathcal{Y}_{e,i}$ not empty, project orthogonally sets $\mathcal{Y}_{w,i}$ and $\mathcal{Y}_{e,i}$ onto the hyperplane $\ell_i(\boldsymbol{x}) = 0$ and form the sets $\mathcal{L}_i(\mathcal{Y}_{w,i})$ and $\mathcal{L}_i(\mathcal{Y}_{e,i})$:

$$\mathcal{L}_i(\mathcal{Y}_{w,i}) = \left\{ \boldsymbol{x} \in \mathbb{R}^m \mid \boldsymbol{x} = \left(I_m - \frac{\boldsymbol{w}_i \boldsymbol{w}_i^T}{\|\boldsymbol{w}_i\|^2} \right) \boldsymbol{\xi} + \frac{c_i \boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}, \ \boldsymbol{\xi} \in \mathcal{Y}_{w,i} \right\},$$
$$\mathcal{L}_i(\mathcal{Y}_{e,i}) = \left\{ \boldsymbol{x} \in \mathbb{R}^m \mid \boldsymbol{x} = \left(I_m - \frac{\boldsymbol{w}_i \boldsymbol{w}_i^T}{\|\boldsymbol{w}_i\|^2} \right) \boldsymbol{\xi} + \frac{c_i \boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}, \ \boldsymbol{\xi} \in \mathcal{Y}_{e,i} \right\}.$$

6: Construct a functional $\ell_{2,i}$ separating $\mathcal{L}_i(\mathcal{Y}_{w,i})$ from $\mathcal{L}_i(\mathcal{Y}_{e,i})$ so that $\ell_{2,i}(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in \mathcal{Y}_{w,i}$ and $\ell_{2,i}(\boldsymbol{x}) < 0$ for all $\boldsymbol{x} \in \mathcal{Y}_{e,i}$.

(c) *Integration*. Integrate Auxiliary Knowledge Units into decision-making pathways of AI$_s$. If, for an $\boldsymbol{x}$ generated by an input to AI$_s$, any of the predicates $(\ell_i(\boldsymbol{x}) \geq 0) \wedge (\ell_{2,i}(\boldsymbol{x}) \geq 0)$ hold true then report $\boldsymbol{x}$ accordingly (swap labels, report as an error etc.).

In what follows we illustrate the approach as well as the application of the proposed Knowledge Transfer algorithms in a relevant problem of a computer vision system design for pedestrian detection in live video streams.

3. EXAMPLE

Let AI_s and AI_t be two systems developed, e.g., for the purposes of pedestrian detection in live video streams. Technological progress in embedded systems and availability of platforms such as Nvidia Jetson TX2 made hardware deployment of such AI systems at the edge of computer vision processing pipelines feasible. These platforms, however, lack computational power that would enable to run state-of-the-art large scale object detection solutions like ResNet (He et al., 2016) in real-time. Smaller-size convolutional neural networks such as SqueezeNet (Iandola et al., 2016) could be a way to move forward. Still, however, these latter systems have hundreds of thousands trainable parameters which is typically several orders of magnitude larger than in e.g., Histograms of Oriented Gradients (HOG) based systems (Dalal and Triggs, 2005). Moreover, training these networks requires substantial computational resources and data.

In this section we illustrate application of the proposed AI Knowledge Transfer technology and demonstrate that this technology can be successfully employed to compensate for the lack of power of an edge-based device. In particular, we suggest that the edge-based system is "taught" by the state-of-the-art teacher in a non-iterative and near-real time way. Since our building blocks are linear functionals, such learning will not lead to significant computational overheads. At the same time, as we will show later, the proposed AI Knowledge Transfer will result in a major boost to the system's performance in the conditions of the experiment.

3.1. Definition of AI_s and AI_t and Rationale

In our experiments, the teacher AI, AI_t, was modeled by an adaptation of SqueezeNet (Iandola et al., 2016) with circa 725 K trainable parameters. The network was trained on a "teacher" dataset comprised of 554 K non-pedestrian (negatives), and 56 K pedestrian (positives) images. Positives have then been subjected to standard augmentation accounting for various geometric and color perturbations. The network was trained for 100 epochs, which took approximately 16 h on Nvidia Titan Xp to complete. The student AI, AI_s, was modeled by a linear classifier with HOG features (Dalal and Triggs, 2005) and 2016 trainable parameters. The values of these parameters were the result of AI_s training on a "student" dataset, a sub-sample of the "teacher" dataset comprising of 16 K positives (55 K after augmentation) and 130 K negatives, respectively. The choice of AI_s and AI_t systems enabled us to emulate interaction between low-power edge-based AIs and their more powerful counterparts that could be deployed on a higher-spec embedded system or, possibly, on a server or in a computational cloud.

We note that for both AI_t and AI_s the set of negatives is several times larger than the set of positives. This makes the datasets somewhat unbalanced. Unbalanced datasets are not uncommon in object detection tasks. There are several reasons why such unbalanced datasets may emerge in practice. Every candidate for inclusion in the set of positives is typically subjected to thorough human inspection. This makes the process time-consuming and expensive, and as a result imposes limitations on the achievable size of the set. Negatives are generally easier to generate. Note also that the set of negatives in our experiments is essentially the set of all objects that are not pedestrians. This latter set has significantly broader spectrum of variations than the set of positives. Accounting for this larger variability without imposing any further prior assumptions or knowledge could be achieved via larger samples. This was the strategy we have adopted here.

In order to make the experiment more realistic, we assumed that internal states of both systems are inaccessible for direct observation. To generate sets $\mathcal{S}$ and $\mathcal{Y}$ required in Algorithms 1 and 2 we augmented system AI_s with an external generator of HOG features of the same dimension. We assumed, however, that positives and negatives from the "student" dataset are available for the purposes of knowledge transfer. A diagram representing this setup is shown in **Figure 4**. A candidate image is evaluated by two systems simultaneously as well as by a HOG features generator. The latter generates 2016 dimensional vectors of HOGs and stores these vectors in the set $\mathcal{S}$. If outputs of AI_s and

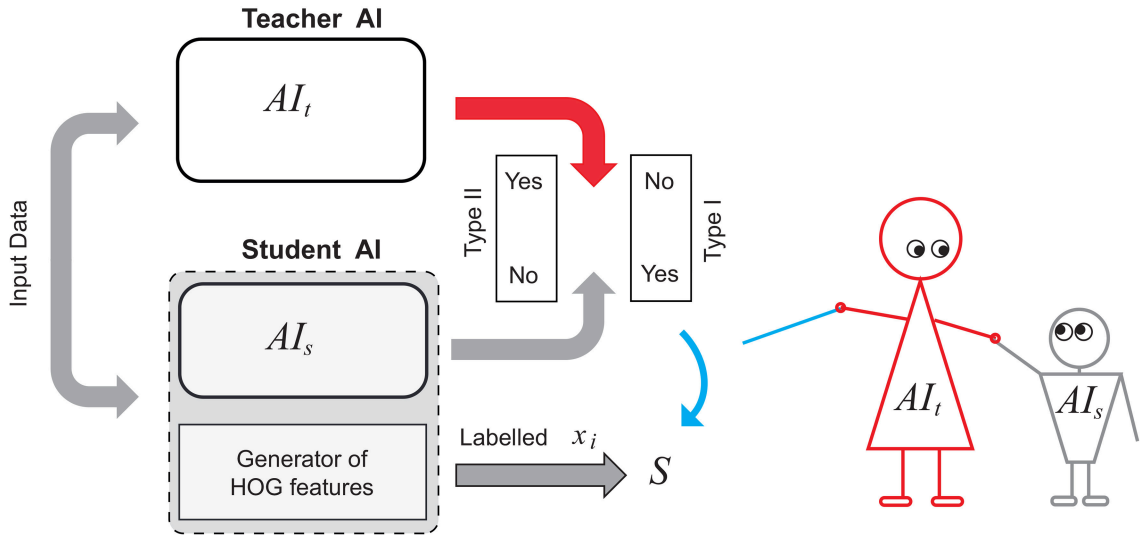

FIGURE 4 | Knowledge transfer diagram between a teacher AI and a student AI augmented with HOG-based feature generator.

AI_t do not match then the corresponding feature vector is added to the set $\mathcal{Y}$.

3.2. Error Types Addressed

In this experiment we consider and address two types of errors: false positives (original Type I errors) and false negatives (original Type II errors). The error types were determined as follows. An error is deemed as *false positive* (for the original data) if AI_s reported presence of a correctly sized full-figure image of pedestrian in a given image patch whereas no such object was there. Similarly, an error is deemed as *false negative* (for the original data) if a pedestrian was present in the given image patch but AI_s did not report it there.

Our main focus was to replicate a deployment scenario in which AI_t is capable of evaluating only small image patches at once in a given processing quanta. At the same time AI_s is supposed to be able to process whole frame in reasonable time, but its accuracy is lower. This makes assessment of all relevant images by AI_t not viable computationally and, in addition, rules out automated detection of Type II errors (original false negatives) when AI_s is scanning the image at thousands of positions per frame. On the other hand, the number of positive responses of AI_s is limited by several dozens of smaller size patches per frame, which is assumed to be well within the processing capabilities of AI_t. In view of these considerations, we therefore focused mainly on errors of Type I (original false positives). Nevertheless, in section 3.4 we discuss possible ways to handle Type II errors in the original system and provide an illustrative example of how this might be done.

It is worthwhile to mention that output labels of the chosen teacher AI, AI_t, do not always match ground truth labels. AI_t may make an occasional error too, and examples of such errors are provided in **Figure 5**. Regardless of these, performance of AI_t was markedly superior to that of AI_s and hence for the sake of testing the concept, these rare occasional errors of AI_t have been discarded in the experiments.

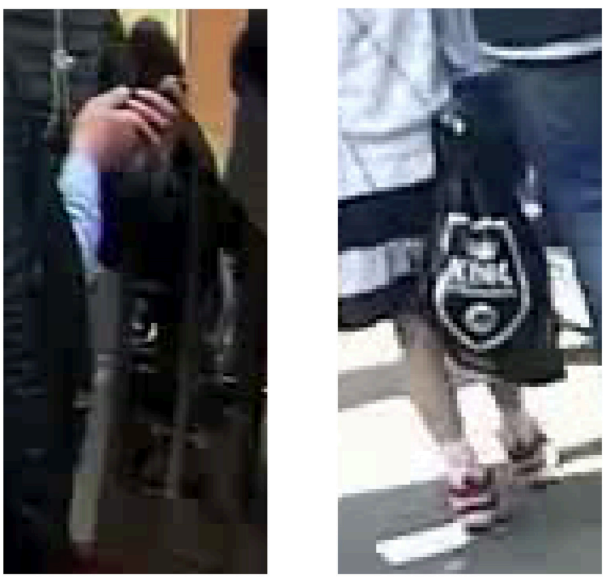

FIGURE 5 | Examples of False positives generated by the teacher AI, AI_t, for NOTTINGHAM video (Burton, 2016).

3.3. Datasets Used in Experiments and Experimental Error Types

To test the approach we used NOTTINGHAM video (Burton, 2016) containing 435 frames of live footage taken with an action camera. The video, as per manual inspection, contains 4,039 full-figure images of pedestrians.

For the purposes of training and testing Knowledge Transfer Units, the video has been passed through AI_s, and AI_s returned detects of pedestrian shapes. These detects were assessed by AI_t and labeled accordingly (see the diagram in **Figure 4**). At this stage, decision-making threshold in AI_s was varying from -0.3 to 2 to capture a reasonably large sample of false positives whose scores are near the decision boundary. This labeled set of feature vectors has been partitioned into two non-overlapping subsets: Set 1 consisting of circa 90% of true positives and 90% of false positives, and Set 2 being its complement. HOG features corresponding to original Type I errors (false positives) in Set 1 as well as all HOG features extracted from 55 K images of positives that have been used to train AI_s were combined into the *training set*. This training set was then used to derive Knowledge Transfer Units for AI_s.

Sets 1 and 2 constitute different *testing sets* in our example. The first testing set (Set 1) enables us to assess how the modified AI_s copes with removing "seen" original Type I errors (false positives) in presence of "unseen" true positives. The second testing set (Set 2) will be used to assess generalization capabilities of the final system.

We note that labeling of false positives involved outputs of AI_t rather than ground truth labels. Visual inspection of AI_t labels revealed that they contain few dozens of false positives too. This number, however, is negligibly small as compared to the overall number of true positives (circa 2,000) and false positives (circa 800) of student AI, AI_s.

Finally, to quantify performance of the proposed knowledge transfer approach, it is important to distinguish between definitions of error types (Type I and Type II) for the original system and error types characterizing *performance of the Knowledge Transfer Units* themselves. The corresponding definitions are provided in **Table 1**. Note that true negatives (marked by star, $*$, in the table) do not occur in the experiments. If what follows and unless stated otherwise we shall refer to these definitions.

Results of the application of Algorithms 1, 2 as well as the analysis of their performance on the testing sets are provided below.

TABLE 1 | Definition of the error types in knowledge transfer experiments.

Response of AI_t	Response of AI_S after knowledge transfer	Error type
Yes	Yes	True positive
	No	False negative
No	Yes	False positive
	No	True negative*

3.4. Results

We generated 10 different realizations of Sets 1 and 2. This resulted in 10 different samples of the training and testing sets. The algorithms have been applied to all these different combinations. Single run of the preprocessing step, Step 1, took, on average, 23.5 s to complete on an Apple laptop with 3.5 GHz A7 processor. After the pre-processing step only 164 principal components have been retained. This resulted in significant reduction of dimensionality of the feature vectors. In our experiments pre-processing also included normalization of the whitened vectors so that their L_2 norm was always one. This brings the data onto the unit sphere which is somewhat more aligned with Theorems 1 and 2. Steps 2 in Algorithms 1, 2 took 1 and 24 ms, respectively. This is a major speed-up in comparison to complete re-training of AI_s (several minutes) or AI_t (hours). Note also that complete re-training does not offer any guarantees that the errors are going to be mitigated either.

Prior to running Steps 2 of the algorithms we checked if the feature vectors corresponding to errors (false positives) in

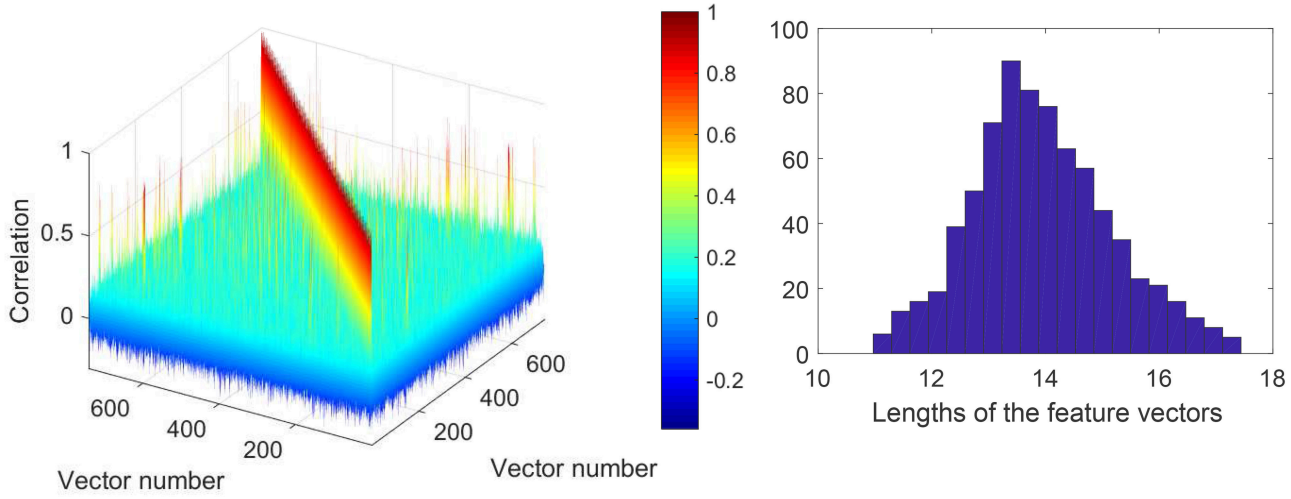

FIGURE 6 | (Left): Correlation diagram between elements of the set $\mathcal{Y}$ (elements to be learned away by AI_S). **(Right)**: Histogram of lengths of the feature vectors in the training set after pre-processing.

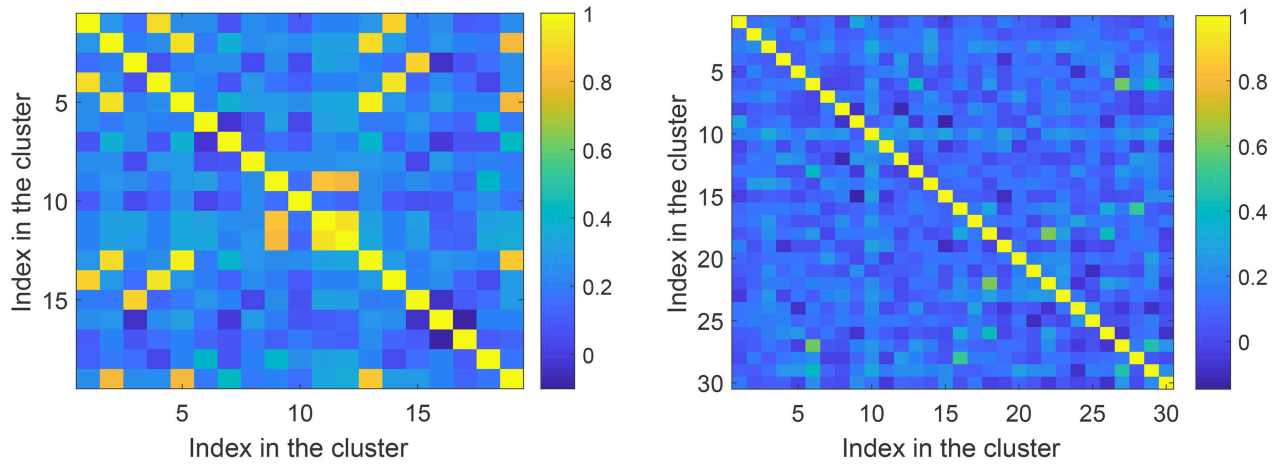

FIGURE 7 | Correlations within clusters after Step 2. **Left panel** corresponds to a cluster with large threshold $\simeq 0.76$. **Right panel** corresponds to a cluster with a lower threshold $\simeq 0.43$.

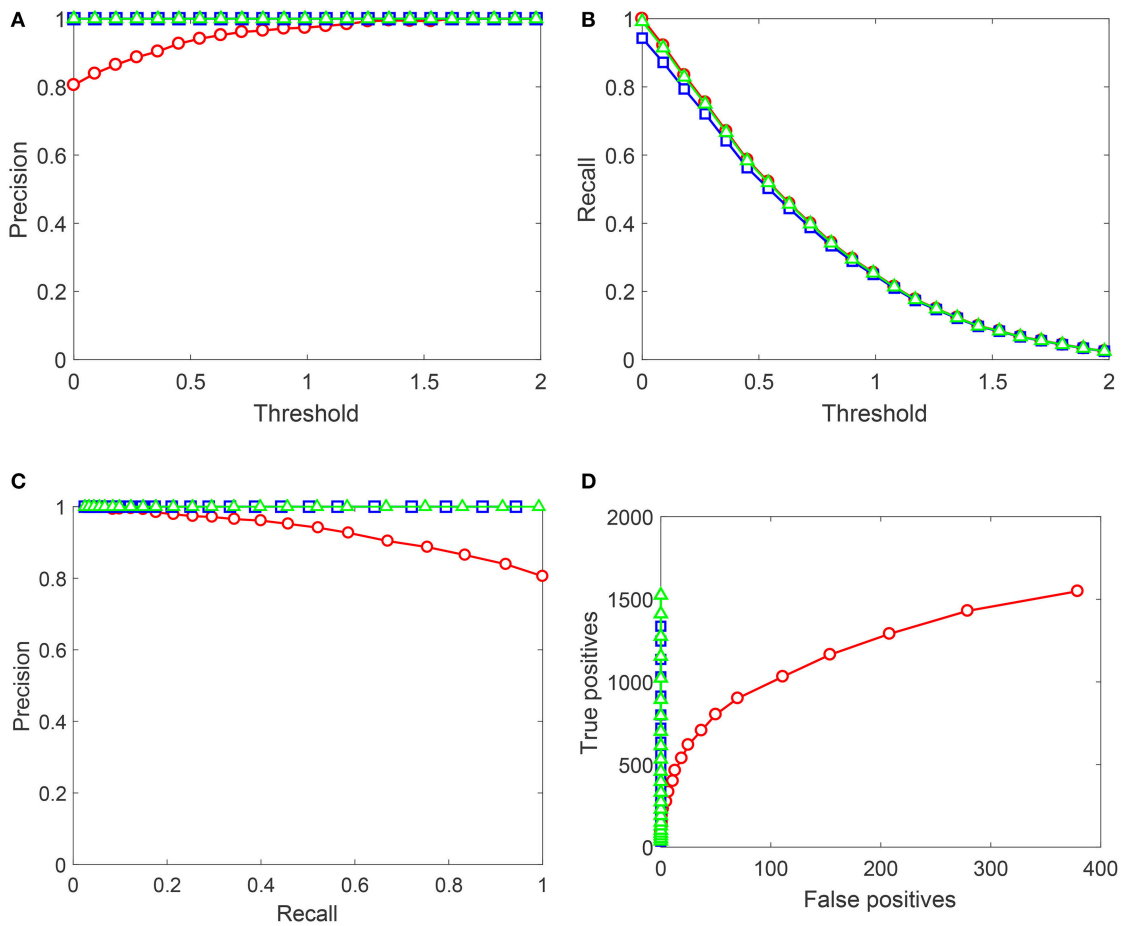

FIGURE 8 | Performance of the up-trained AI_S on Testing set 1 (Set 1). Red circles correspond to the original AI_S without Knowledge Transfer Units. Blue squares correspond to AI_S after Algorithm 1, and green triangles illustrate application of Algorithm 2. **(A)** Shows Precision (averaged over 10 realizations of Set 1) as a function of the decision threshold in AI_S, **(B)** shows Recall (averaged over 10 realizations of Set 1), **(C)** contains the corresponding Precision-Recall data, and **(D)** shows True positives vs. False positives for a single realization of Set 1. Lines between points are shown for the sake of visualization and do not represent any measurements or indicate achievable performance.

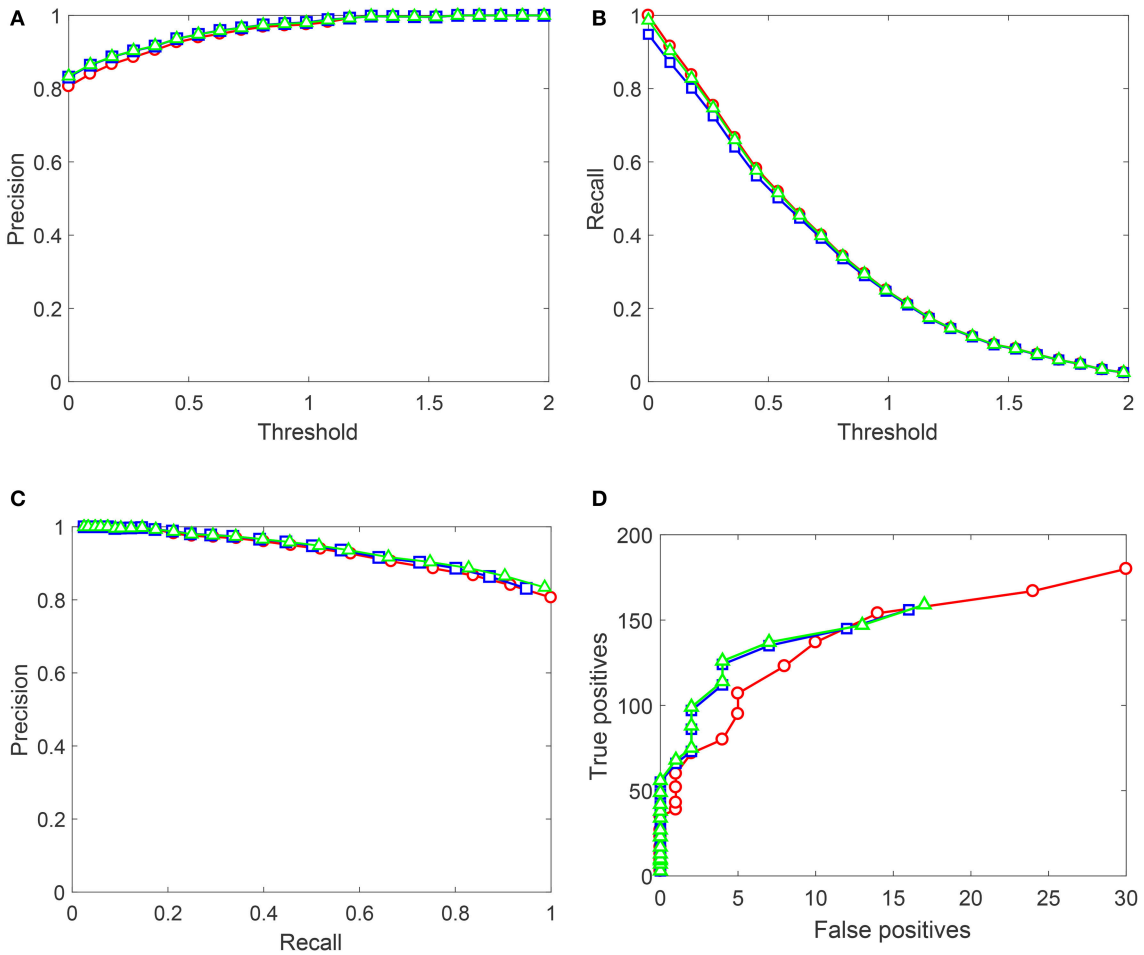

FIGURE 9 | Performance of the up-trained AI_S on Testing set 2 (Set 2). Red circles correspond to the original AI_S without Knowledge Transfer Units. Blue squares correspond to AI_S after Algorithm 1, and green triangles illustrate application of Algorithm 2. **(A)** Shows Precision (averaged over 10 realizations of Set 2) as a function of the decision threshold in AI_S, **(B)** shows Recall (averaged over 10 realizations of Set 2), **(C)** contains the corresponding Precision-Recall data, and **(D)** shows True positives vs. False positives for a single realization of Set 2. Lines between points are shown for the sake of visualization and do not represent any measurements or indicate achievable performance.

the training set are correlated. This allows an informed choice of the number of clusters parameter, p, in Algorithms 1 and 2. A 3D color-coded visualization of correlations, i.e., $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$, between pre-processed (after Step 1) elements in the set $\mathcal{Y}$ is shown in **Figure 6**, left panel. A histogram of lengths of all vectors in the training set is shown in **Figure 6**, right panel. Observe that the lengths concentrate neatly around $\sqrt{164}$, as expected. According to **Figure 6**, elements of the set $\mathcal{Y}$ are mostly uncorrelated. There are, however, few correlated elements which, we expect, will be accounted for in Step 2.a of the algorithms. In absence of noticeably wide-spread correlations we set $p = 30$ which is equivalent to roughly 25 elements per cluster. Examples of correlations within clusters after Step 2 was complete are shown in **Figure 7**. Note that the larger is the threshold the higher is the expected correlation between elements, and the higher are the chances that such Knowledge Transfer Unit would operate successfully (see Theorems 1 and 2).

Performance of Algorithms 1, 2 on the Testing sets generated from NOTTINGHAM video is summarized in **Figures 8**, **9**. In these figures we showed behavior of

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}$$

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}},$$

as functions of decision-making threshold in AI_s, Precision-Recall and True positives vs. False positives charts. Red circles correspond to the original AI_s without Knowledge Transfer Units. Blue squares correspond to AI_s after Algorithm 1, and green triangles illustrate application of Algorithm 2. Note that the maximal number of false positives in **Figure 8** does not exceed 400. This is due to that the threshold was now varied in the operationally feasible interval $[0, 2]$ as opposed to $[-0.3, 2]$ used for gather training data.

TABLE 2 | Recovering Type II errors of the original AI_S.

Number of false negatives converted (out of 307)	Number of false positives remained (out of 410)	Threshold
137	169	0.05
87	86	0.1
45	35	0.15
10	14	0.2
8	2	0.25
1	0	0.3

As **Figure 9** shows, performance of AI_s on Testing set 1 (Set 1) improves drastically after the application of both algorithms. Algorithm 2 outperforms Algorithm 1 by some margin which is most noticeable from the plot in **Figure 8D**. Near-ideal performance in Precision-Recall space can be explained by that the training set contained full information about false positives (but not true positives). It is important to observe that Algorithm 2 did not flag nearly all "unseen" true positives.

As for results shown in **Figure 9**, performance patterns of AI_s after the application of both algorithms change. Algorithm 1 results in a minor drop in Recall figures, and Algorithm 2 recovers this drop to the baseline performance. Precision improves slightly for both algorithms, and True positives vs. False positives curves for AI_s with Knowledge Transfer Units dominate those of plain AI_s. This suggests that not only the proposed Knowledge Transfer framework allows to acquire new knowledge as specified by labeled data but also has a capacity to generalize the knowledge further. The degree of such generalization will obviously depend on statistics of the data. Yet, as **Figure 9** demonstrates, this is a viable possibility.

Our experiments showed how the approach could be used for filtering Type I errors in the original system. The technology, however, could be used to recover Type II errors too (false negatives in the original system), should the data be available. Several strategies might be evoked to obtain this data. The first approach is to use background subtraction to detect a moving object and pass the object through both AI_s and AI_t. The second approach is to enable AI_s to report detects that are classified as negatives but still are reasonably close to the detection boundary. This is the strategy which we adopted here. We validated HOG features in AI_s corresponding to scores in the interval $[-0.3, 0]$ with the teacher AI, AI_t. Overall, 717 HOG vectors have been extracted by this method, of which 307 have been labeled by AI_t as positives, and 410 were considered as negatives. We took one of the HOG feature vectors labeled as True positive, $\boldsymbol{x}_v$, and constructed (after applying the pre-processing transformation from previous experiments) several separating hyperplanes with the weights given by $\boldsymbol{x}_v/\|\boldsymbol{x}_v\|$ and thresholds c_v varying in $[0, 1]$. Results are summarized in **Table 2**. As before, we observe strong concentration of measure effect: the Knowledge Transfer Unit shows extreme selectivity for sufficiently large values of c_v. In this particular case $c_v = 0.25$ provides maximal gain at the lowest risk of expected error in future [see. e.g., the right-hand side of (1) in Theorem 1 at $k = 1$, $\varepsilon = 0.75$ for an estimate].

4. CONCLUSION

In this work we proposed a framework for instantaneous knowledge transfer between AI systems whose internal state used for decision-making can be described by elements of a high-dimensional vector space. The framework enables development of non-iterative algorithms for knowledge spreading between legacy AI systems with heterogeneous non-identical architectures and varying computing capabilities. Feasibility of the framework was illustrated with an example of knowledge transfer between two AI systems for automated pedestrian detection in video streams.

In the basis of the proposed knowledge transfer framework are separation theorems (Theorem 1–3) stating peculiar properties of large but finite random samples in high dimension. According to these results, $k < n$ random i.i.d. elements can be separated form $M \gg n$ randomly selected elements i.i.d. sampled from the same distribution by few linear functionals, with high probability. The theorems are proved for equidistributions in a ball and in a cube. The results can be trivially generalized to equidistributions in ellipsoids and Gaussian distributions. Discussing these in detail here is the beyond the scope and vision of this work. Nevertheless, generalizations to other meaningful distributions, relaxation of the independence requirement, and a broader view on how the proposed technology could be used in multiagent AI systems is presented in technical report (Gorban et al., 2018).

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

The work was supported by the Ministry of Education and Science of Russia (Project No. 14.Y26.31.0022) and Innovate UK (Knowledge Transfer Partnership grant KTP010522).

REFERENCES

Bordes, A., Ertekin, S., Weston, J., and Bottou, L. (2005). Fast kernel classifiers with online and active learning. *J. Mach. Learn. Res.* 6, 1579–1619. Available online at: http://www.jmlr.org/papers/volume6/bordes05a/bordes05a.pdf

Buchtala, O., and Sick, B. (2007). "Basic technologies for knowledge transfer in intelligent systems," in *IEEE Symposium on Artificial Life, 2007. ALIFE'07* (Honolulu, HI), 251–258.

Bucila, C., Caruana, R., and Niculescu-Mizil, A. (2006). "Model compression," in *KDD* (Philadelphia, PA: ACM), 535–541.

Burton, R. (2016). Nottingham video. *A Test Video for Pedestrians Detection Taken From the Streets of Nottingham by an Action Camera.* Available online at: https://youtu.be/SJbhOJQCSuQ

Chapelle, O. (2007). Training a support vector machine in the primal. *Neural Comput.* 19, 1155–1178. doi: 10.1162/neco.2007.19.5.1155

Chen, T., Goodfellow, I., and Shlens, J. (2015). Net2net: Accelerating learning via knowledge transfer. *arXiv [preprint]. arXiv:1511.05641.*

Dalal, N., and Triggs, B. (2005). "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (San Diego, CA), 886–893.

Duda, R., Hart, P., and Stork, D. (2000). *Pattern Classification*. New York, NY: Wiley.

Gibbs, J. (1902). *Elementary Principles in Statistical Mechanics, Developed With Especial Reference to the Rational Foundation of Thermodynamics*. New York, NY: Dover Publications.

Gilev, S., Gorban, A., and Mirkes, E. (1991). Small experts and internal conflicts in learning neural networks (malye eksperty i vnutrennie konflikty v obuchaemykh neironnykh setiakh). *Akademiia Nauk SSSR Doklady* 320, 220–223.

Gorban, A. (2007). Order-disorder separation: geometric revision. *Phys. A* 374, 85–102. doi: 10.1016/j.physa.2006.07.034

Gorban, A., Burton, R., Romanenko, I., and I., T. (2016a). One-trial correction of legacy ai systems and stochastic separation theorems. *arXiv [preprint]. arXiv:1610.00494*.

Gorban, A., Golubkov, A., Grechuk, B., Mirkes, E., and Tyukin, I. (2018). Correction of AI systems by linear discriminants: Probabilistic foundations. *Inf. Sci.* 466, 303–322. doi: 10.1016/j.ins.2018.07.040

Gorban, A., and Tyukin, I. (2017). Stochastic separation theorems. *Neural Netw.* 94, 255–259. doi: 10.1016/j.neunet.2017.07.014

Gorban, A., and Tyukin, I. (2018). Blessing of dimensionality: mathematical foundations of the statistical physics of data. *Philos. Trans. R. Soc. A* 376:20170237. doi: 10.1098/rsta.2017.0237

Gorban, A., Tyukin, I., Prokhorov, D., and Sofeikov, K. (2016b). Approximation with random bases: pro et contra. *Inform. Sci.* 364–365, 129–145. doi: 10.1016/j.ins.2015.09.021

Gromov, M. (1999). *Metric Structures for Riemannian and non-Riemannian Spaces. With Appendices by M. Katz, P. Pansu, S. Semmes. Translated from the French by Sean Muchael Bates*. Boston, MA: Birkhauser.

Gromov, M. (2003). Isoperimetry of waists and concentration of maps. *Geomet. Funct. Anal.* 13, 178–215. doi: 10.1007/s000390300004

Hall, W., and Pesenti, J. (2017). *Growing the Artificial Intelligence Industry in the UK.* Department for Digital, Culture, Media & Sport and Department for Business, Energy & Industrial Strategy.

Han, H. (2014). "Analyzing support vector machine overfitting on microarray data," in *Intelligent Computing in Bioinformatics. ICIC 2014. Lecture Notes in Computer Science*, eds D. Huang, K. Han, and M. Gromiha (Cham: Springer), 148–156.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV), 770–778.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv [preprint]. arXiv:1503.02531*.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. (2016). Squeezenet: alexnet-level accuracy with 50x fewer parameters and $<$ 0.5 mb model size. *arXiv [preprint]. arXiv:1602.07360*.

Ison, M., Quian Quiroga, R., and Fried, I. (2015). Rapid encoding of new memories by individual neurons in the human brain. *Neuron* 87, 220–230. doi: 10.1016/j.neuron.2015.06.016

Jacobs, R., Jordan, M., Nowlan, S., and Hinton, G. (1991). Adaptive mixtures of local experts. *Neural Comput.* 3, 79–87.

Lévy, P. (1951). *Problèmes Concrets D'analyse Fonctionnelle, 2nd Edn.* Paris: Gauthier-Villars.

Li, S. (2011). Concise formulas for the area and volume of a hyperspherical cap. *Asian J. Math. Stat.* 4, 66–70. doi: 10.3923/ajms.2011.66.70

Lloyd, S. P. (1982). Least squares quantization in pcm. *IEEE Trans. Inform. Theory* 28, 129–137.

Pan, S. J., and Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22, 1345–1359. doi: 10.1109/TKDE.2009.191

Platt, J. (1999). "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods – Support Vector Learning*, eds B. Schölkopf, C. Burges, and A. Smola (Cambridge: MIT Press), 185–208.

Pratt, L. (1992). "Discriminability-based transfer between neural networks," in *Advances in Neural Information Processing Systems*, Vol. 5 (San Francisco, CA: Morgan Kaufmann Publishers Inc.), 204–211.

Schultz, T., and Rivest, F. (2000). "Knowledge-based cascade correllation," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks* (Como), 641–646.

Vapnik, V. (2000). *The Nature of Statistical Learning Theory*. New York, NY: Springer-Verlag.

Vapnik, V., and Izmailov, R. (2017). Knowledge transfer in SVM and neural networks. *Ann. Math. Artif. Intell.* 81, 1–17. doi: 10.1007/s10472-017-9538-x

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems*, Vol. 2 (Montreal, QC: MIT Press), 3320–3328.

Conflict of Interest Statement: KS was employed by ARM Holding, and IR was employed by Spectral Edge Ltd. At the time of preparing the manuscript IR was employed by ARM Holding. All data used in experiments have been provided by ARM Holding.

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

APPENDIX

Proof of Theorem 1. Consider the set $\mathcal{Y}$. The probability that a single element of $\mathcal{Y}$ belongs to $B_n(1) \setminus B_n(1-\varepsilon)$ is $1-(1-\varepsilon)^n$. Recall that if $A_1, \ldots, A_m$ are arbitrary events then

$$P(A_1 \& A_2 \& \cdots \& A_m) \geq 1 - \sum_{i=1}^{m} (1 - P(A_i)). \tag{4}$$

According to (4), the probability that $\mathcal{Y} \subset B_n(1) \setminus B_n(1-\varepsilon)$ (event E_1) satisfies

$$P(E_1) \geq 1 - k(1-\varepsilon)^n.$$

Pick $\boldsymbol{x}_{M+i} \in \mathcal{Y}$, and consider the largest equator of the ball $B_n(1)$ that is orthogonal to the element $\boldsymbol{x}_{M+i}$. Let $D_\delta(\boldsymbol{x}_{M+i})$ denote the δ-thickening of the disc associated with this equator. Only one such disc exists, and it is uniquely defined by $\boldsymbol{x}_{M+i}$ and δ (see **Figure A1**). Consider the following events:

$\boldsymbol{x}_{M+2} \in D_\delta(\boldsymbol{x}_{M+1})$: event E_2,
$[\boldsymbol{x}_{M+3} \in D_\delta(\boldsymbol{x}_{M+1})] \& [\boldsymbol{x}_{M+3} \in D_\delta(\boldsymbol{x}_{M+2})]$: event E_3,
$\ldots$
$[\boldsymbol{x}_{M+k} \in D_\delta(\boldsymbol{x}_{M+1})] \& [\boldsymbol{x}_{M+k} \in D_\delta(\boldsymbol{x}_{M+2})] \& \ldots \& [\boldsymbol{x}_{M+k} \in D_\delta(\boldsymbol{x}_{M+k-1})]$: event E_k.

According to (4) and **Figure A1**, [cf. Gorban et al., 2016b, proof of Proposition 3 and estimate (26)], it is hence clear that

$$P(E_2) \geq 1 - (1-\delta^2)^{\frac{n}{2}},\ P(E_3) \geq 1 - 2(1-\delta^2)^{\frac{n}{2}}, \ldots,\ P(E_k) \geq 1 - (k-1)(1-\delta^2)^{\frac{n}{2}}.$$

Suppose that event E_1 ($\|\boldsymbol{x}_{M+i}\| \geq 1-\varepsilon$ for all $i = 1, \ldots, k$) and events $E_2, \ldots, E_k$ occur. Then

$$|\cos(\boldsymbol{x}_{M+i}, \boldsymbol{x}_{M+j})| \leq \frac{\delta}{(1-\varepsilon)} \text{ for all } i \neq j,\ i = 1, \ldots, k,$$

and

$$-\frac{\delta}{(1-\varepsilon)} \leq \langle \boldsymbol{x}_{M+i}, \boldsymbol{x}_{M+j} \rangle \leq \frac{\delta}{(1-\varepsilon)} \text{ for all } i \neq j,\ i = 1, \ldots, k. \tag{5}$$

Consider the vector

$$\bar{\boldsymbol{x}} = \frac{1}{k} \sum_{i=1}^{k} \boldsymbol{x}_{M+i}.$$

Equation (5) implies that

$$\frac{1}{k}\left((1-\varepsilon)^2 - \frac{k-1}{1-\varepsilon}\delta\right) \leq \langle \bar{\boldsymbol{x}}, \boldsymbol{x}_{M+i} \rangle \leq \frac{1}{k}\left(1 + \frac{k-1}{1-\varepsilon}\delta\right) \text{ for all } i = 1, \ldots, k \tag{6}$$

and, consequently,

$$\|\bar{\boldsymbol{x}}\|^2 = \langle \bar{\boldsymbol{x}}, \bar{\boldsymbol{x}} \rangle = \frac{1}{k} \sum_{i=1}^{k} \langle \bar{\boldsymbol{x}}, \boldsymbol{x}_{M+i} \rangle \leq \frac{1}{k}\left(1 + \frac{k-1}{1-\varepsilon}\delta\right). \tag{7}$$

Finally, consider

$$\ell_0(\boldsymbol{x}) = \left\langle \frac{\bar{\boldsymbol{x}}}{\|\bar{\boldsymbol{x}}\|}, \boldsymbol{x} \right\rangle - \frac{1}{\sqrt{k}} \frac{(1-\varepsilon)^2 - \frac{k-1}{1-\varepsilon}\delta}{\sqrt{1 + \frac{k-1}{1-\varepsilon}\delta}}. \tag{8}$$

It is clear that if $\|\boldsymbol{x}_{M+i}\| \geq 1-\varepsilon$ and (5) hold then (6), (7) assure that $\ell_0(\boldsymbol{x}_{M+i}) \geq 0$ for all $\boldsymbol{x}_{M+i} \in \mathcal{Y}$. The hyperplane $\ell_0(\boldsymbol{x}) = 0$ partitions the unit ball $B_n(1)$ into the union of two disjoint sets: the spherical cap $\mathcal{C}$

$$\mathcal{C} = \{\boldsymbol{x} \in B_n(1) \,|\, \ell_0(\boldsymbol{x}) \geq 0\} \tag{9}$$

and its complement in $B_n(1)$, $B_n(1) \setminus \mathcal{C}$. The volume $\mathcal{V}$ of the cap $\mathcal{C}$ can be estimated from above as

$$\mathcal{V}(\mathcal{C}) \leq \mathcal{V}(B_n(1)) \frac{\Delta(\varepsilon, \delta, k)^{\frac{n}{2}}}{2},$$

$$\Delta(\varepsilon, \delta, k) = 1 - \left[\frac{1}{\sqrt{k}} \frac{(1-\varepsilon)^2 - \frac{k-1}{1-\varepsilon}\delta}{\sqrt{1 + \frac{k-1}{1-\varepsilon}\delta}}\right]^2.$$

Hence the probability that $\ell_0(\boldsymbol{x}_i) < 0$ for all $\boldsymbol{x}_i \in \mathcal{M}$ (event E_{k+1}) can be estimated from below as

$$P(E_{k+1}) \geq 1 - \frac{M}{2}\Delta(\varepsilon, \delta, k)^{\frac{n}{2}},$$

and

$$P(E_1 \& E_2 \& \cdots \& E_k \& E_{k+1}) \geq 1 - k(1-\varepsilon)^n - \frac{(k-1)k}{2}(1-\delta^2)^{\frac{n}{2}} - \tfrac{M}{2}\Delta(\varepsilon, \delta, k)^{\frac{n}{2}}.$$

This is a lower bound for the probability that the $\mathcal{M}$ can be separated from $\mathcal{Y}$ by the hyperplanes $\ell_0(\boldsymbol{x}) = 0$. Given that this estimate holds for all feasible values of ε, δ, statement (1) follows. □

Proof of Theorem 2. Consider the set $\mathcal{Y}$. Observe that, since elements $\boldsymbol{x}_i$ are drawn independently from $B_n(1)$, $\|\boldsymbol{x}_{M+i}\| \geq 1-\varepsilon$, $\varepsilon \in (0,1)$ for all $i = 1, \ldots, k$, with probability $p = (1-(1-\varepsilon)^n)^k$. Consider now the vector $\bar{\boldsymbol{y}}$

$$\bar{\boldsymbol{y}} = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{x}_{M+r_i},$$

and evaluate the following inner products

$$\left\langle \frac{\bar{\boldsymbol{y}}}{\|\bar{\boldsymbol{y}}\|}, \boldsymbol{x}_{M+r_i} \right\rangle = \frac{1}{m\|\bar{\boldsymbol{y}}\|} \left(\langle \boldsymbol{x}_{M+r_i}, \boldsymbol{x}_{M+r_i} \rangle + \sum_{r_j,\, j \neq i} \langle \boldsymbol{x}_{M+r_i}, \boldsymbol{x}_{M+r_j} \rangle \right), \quad i = 1, \ldots, m.$$

According to assumption (2),

$$\left\langle \frac{\bar{\boldsymbol{y}}}{\|\bar{\boldsymbol{y}}\|}, \boldsymbol{x}_{M+r_i} \right\rangle \geq \frac{1}{m\|\bar{\boldsymbol{y}}\|} \left((1-\varepsilon)^2 + \beta_2(m-1)\right)$$

and, respectively,

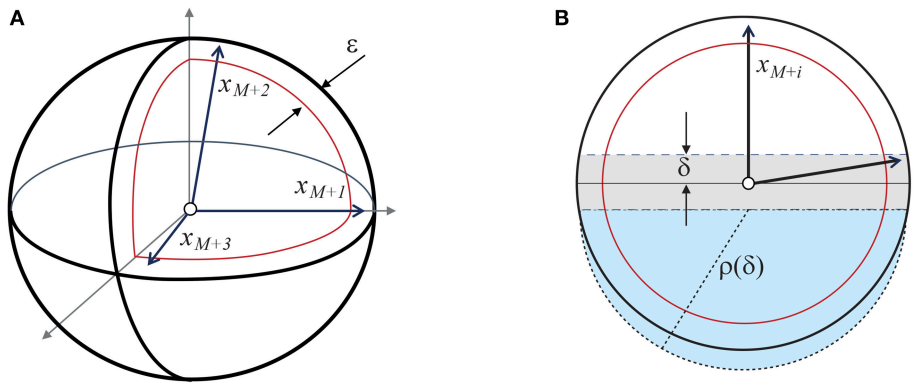

FIGURE A1 | Illustration to the proof of Theorem 1. **(A)** Shows $\boldsymbol{x}_{M+1}$, $\boldsymbol{x}_{M+2}$, and $\boldsymbol{x}_{M+3}$ in the set $B_n(1) \setminus B_n(1-\varepsilon)$. **(B)** shows $\boldsymbol{x}_{M+i}$ and the corresponding disc $D_\delta(\boldsymbol{x}_{M+i})$ (shaded gray area). The combined volume of the spherical caps below and above the disc is bounded from above by the volume of the n-ball with radius $\rho(\delta) = \sqrt{1-\delta^2}$. The corresponding half-ball of radius $\rho(\delta)$ is shown as the blue shaded area.

$$\frac{1}{m}\left(1+(m-1)\beta_1\right) \geq \langle \bar{\boldsymbol{y}}, \bar{\boldsymbol{y}} \rangle \geq \frac{1}{m}\left((1-\varepsilon)^2+\beta_2(m-1)\right)$$

Let $(1-\varepsilon)^2+\beta_2(m-1) > 0$ and $(1-\varepsilon)^2+\beta_1(m-1) > 0$. Consider

$$\ell_0(\boldsymbol{x}) = \left\langle \frac{\bar{\boldsymbol{y}}}{\|\bar{\boldsymbol{y}}\|}, \boldsymbol{x} \right\rangle - \frac{1}{\sqrt{m}}\left(\frac{(1-\varepsilon)^2+\beta_2(m-1)}{\sqrt{1+(m-1)\beta_1}}\right). \quad (10)$$

It is clear that $\ell_0(\boldsymbol{x}_{M+r_i}) \geq 0$ for all $i = 1, \ldots, m$ by the way the functional is constructed. The hyperplane $\ell_0(\boldsymbol{x}) = 0$ partitions the ball $B_n(1)$ into two sets: the set $\mathcal{C}$ defined as in (9) and its complement, $B_n(1) \setminus \mathcal{C}$. The volume $\mathcal{V}$ of the set $\mathcal{C}$ is bounded from above as

$$\mathcal{V}(\mathcal{C}) \leq \mathcal{V}(B_n(1))\frac{\Delta(\varepsilon, m)^{\frac{n}{2}}}{2}$$

where

$$\Delta(\varepsilon, m) = 1 - \frac{1}{m}\left(\frac{(1-\varepsilon)^2+\beta_2(m-1)}{\sqrt{1+\beta_1(m-1)}}\right)^2.$$

Estimate (3) now follows. □

Proof of Theorem 3. Observe that, in the quotient space $\mathbb{R}^n/E$, elements of the set

$$\mathcal{Y} = \{\boldsymbol{x}_{M+1}, \boldsymbol{x}_{M+1} + (\boldsymbol{x}_{M+2} - \boldsymbol{x}_{M+1}), \ldots, \boldsymbol{x}_{M+1} + (\boldsymbol{x}_{M+k} - \boldsymbol{x}_{M+1})\}$$

are vectors whose coordinates coincide with that of the quotient representation of $\boldsymbol{x}_{M+1}$. This means that the quotient representation of $\mathcal{Y}$ consists of a single element, $Q(\boldsymbol{x}_{M+1})$. Furthermore, dimension of $\mathbb{R}^n/E$ is $n-k+1$. Let $R_0^2 = \sum_{i=1}^{n-k+1}\sigma_i^2$ and $\bar{Q}(\boldsymbol{x}) = \mathbb{E}(Q(\boldsymbol{x}))$. According to Theorem 2 and Corollary 2 from (Gorban and Tyukin, 2017), for $\vartheta \in (0,1)$ and M satisfying

$$M \leq \frac{\vartheta}{3}\exp\left(\frac{(n-k+1)\sigma_0^4}{2}\right) - 1,$$

with probability $p > 1-\vartheta$ the following inequalities hold:

$$\frac{1}{2} \leq \frac{\|Q(\boldsymbol{x}_j) - \bar{Q}(\boldsymbol{x})\|^2}{R_0^2} \leq \frac{3}{2}, \left\langle \frac{Q(\boldsymbol{x}_i) - \bar{Q}(\boldsymbol{x})}{R_0}, \frac{Q(\boldsymbol{x}_{M+1}) - \bar{Q}(\boldsymbol{x})}{\|Q(\boldsymbol{x}_{M+1}) - \bar{Q}(\boldsymbol{x})\|} \right\rangle < \frac{1}{\sqrt{2}}$$

for all i, j, $i \neq M+1$. This implies that the hyperplane $\ell_0(\boldsymbol{x}) = 0$, where

$$\ell_0(\boldsymbol{x}) = \left\langle \frac{Q(\boldsymbol{x}) - \bar{Q}(\boldsymbol{x})}{R_0}, \frac{Q(\boldsymbol{x}_{M+1}) - \bar{Q}(\boldsymbol{x})}{\|Q(\boldsymbol{x}_{M+1}) - \bar{Q}(\boldsymbol{x})\|} \right\rangle - \frac{1}{\sqrt{2}}$$

separates $\mathcal{M}$ and $\mathcal{Y}$ with probability $p > 1-\vartheta$. □