LEARNING SEMANTIC SIMILARITIES FOR PROTOTYPICAL CLASSIFIERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent metric learning approaches parametrize semantic similarity measures through the use of an encoder trained along with a similarity model, which operates over pairs of representations. We extend such a setting and enable its use in tasks including multi-class classification in order to tackle known issues observed in standard classifiers such as their lack of robustness to out-of-distribution data. We do so by further learning a set of class prototypes, each one representing a particular class. Training is carried out so that each encoded example is pushed towards the prototype corresponding to its class, and test instances are assigned to the class corresponding to the prototype they are closest to. We thus provide empirical evidence showing the proposed setting is able to match object recognition performance of standard classifiers on common benchmarks, while presenting much improved robustness to adversarial examples and distribution shifts. We further show such a model is effective for tasks other than classification, including those requiring pairwise comparisons such as verification and retrieval. Finally, we discuss a simple scheme for few-shot learning of new classes where only the set of prototypes needs to be updated, yielding competitive performance.

1 INTRODUCTION

Despite the performance boost observed by multi-class classifiers based on neural networks compared to alternative approaches, as evidenced since (Krizhevsky et al., 2012), it is now well-known that such a modeling framework suffers from shortcomings that limit its potential deployment in real-world applications. We highlight below some of such limitations:

- A worth mentioning threat regarding the use of current classifiers is the existence of *adversarial examples*, as discussed originally by Szegedy et al. (2013) and Goodfellow et al. (2014). In fact, it is a known property of neural networks that it is possible to impose large variations in their outputs by slightly changing their inputs. Attackers might then exploit such property to fool deployed models into making certain decisions they might benefit from. Several methods have been proposed in recent literature in order to fool state-of-the-art classifiers with changes to the input that are imperceptible to humans.
- The lack of robustness to distribution shifts across train and test data is a further issue that appears in practice and is known to affect performance of current classifiers. For example, an object recognizer trained on natural images will likely observe a performance degradation once test data consists of drawings from the same classes, for instance. Such a shift across train and test data sources is a direct violation of the i.i.d. assumption on top of which most of the supervised learning generalization guarantees are built within the empirical risk minimization framework. Recent literature in domain adaptation has introduced more general settings relaxing the i.i.d. assumption to some extent to help coping with situations found in practice. However, there's still much room for improvement, as most approaches require data from a particular target data distribution. This requirement is still unpractical given that a large number of possible unseen test conditions might appear for a deployed model (Albuquerque et al., 2019).
- Yet another limitation is the case of small data samples since large classifiers in terms of parameter count require large amounts of data so as to achieve high performance. In

several practical situations, however, collecting (and labeling) large datasets is prohibitively costly. Moreover, standard classifiers are bounded to the label set they were presented to during training, while in practice one would ideally be able to extend a trained classifier to predict new classes observed after training. Transfer learning schemes thus appeared as a natural strategy to overcome both issues by enabling fast adaptation once data from novel sources is made available so that: **(i)**-one can leverage a pretrained model on large datasets and adapt it to the data of interest which is scarce; **(ii)**-a classifier does not need to be trained from scratch whenever new classes are taken into account. As such, devising approaches enabling *inexpensive* adaptation of trained classifiers to new data became a relevant research direction, often referred to as *few-shot learning*, yielding approaches such as meta-learning or learning to learn (Schmidhuber, 1987; Bengio et al., 1992; Ravi & Larochelle, 2016; Finn et al., 2017), as well as geometric methods (Koch et al., 2015; Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018).

In this contribution, our main goal is then to develop classification strategies that address to some extent the issues discussed above. As such, the research question we pose is whether one can define multi-class classification approaches which are more robust against adversaries and distribution shifts while supporting adaptation to novel classes, observed after training in small samples. We thus tackle such problem using approaches which leverage both the set of methods commonly grouped under the term *metric learning*, as well as the geometric approaches discussed above for few-shot-classification; *prototypical networks* in particular (Snell et al., 2017). In further detail, we focus on metric learning settings where both an encoder and a similarity or distance models are trained jointly (Koch et al., 2015; Garcia & Vogiatzis, 2019; Monteiro et al., 2020), but augment such setting with a set of class prototypes used in order to assign points to classes.

Our method thus comprises three main components: **(i)**-an encoder that embeds data into a lower dimensional space; **(ii)**-a similarity model which maps a pair of concatenated representations into a similarity score, and finally **(iii)**-a list of class prototypes in which case each one summarizes a whole class into a vector in the embedding space. Based on said components, we can then devise different inference mechanisms depending on the task at hand. For the case of multi-class classification, for instance, at test time, one can predict the class of a particular test instance through measuring its similarity against each prototype and assigning it to the class whose prototype it is most similar to. Similarly, tasks relying on pairwise comparisons can be performed such as verification, i.e. comparing two data instances and determining whether they belong to the same class, or retrieval, i.e. comparing a test instance against a gallery and determining the k elements in the gallery the considered test instance is most similar to. Moreover, each time new classes appear, adapting the model consists of updating the list of prototypes only, while keeping the encoder and similarity unchanged, thus enabling fast adaptation and avoiding issues such as forgetting past classes or overfitting to the new ones. In summary, our contributions are as follows:

1. We introduce an alternative multi-class classification approach based on metric learning methods, which can match the performance of standard classifiers, while offering improved robustness against adversaries and distribution shifts with respect to observed train data.
2. The proposed setting is further shown to perform well in tasks involving pairwise comparison such as verification and image retrieval, thus providing an approach that allows a single model to be used across multiple tasks.
3. The proposed approach supports the inclusion of new classes appearing posterior to training, which we do by simply repartitioning the space using small data samples. We observed doing so yields a simple yet competitive mechanism for few-shot classification.

2 Background and related work

Metric learning approaches are concerned with representing data in a metric space where semantic properties can be inferred using distances. The literature in this field can be classified in terms of two main groups of approaches trying to do so under two distinct settings, and we will refer to those as *distance metric learning*, introduced originally by Xing et al. (2003), and *deep metric learning* represented most notably by siamese networks (Bromley et al., 1994; Chopra et al., 2005; Hadsell et al., 2006). In the case of the former, one learns a so-called Mahalanobis distance which, given

$x, y \in \mathbb{R}^d$, will have the form: $\sqrt{(x-y)^\intercal W(x-y)}$, where $W \in \mathbb{R}^{d \times d}$ is positive semidefinite. Learning is designed so that $\sqrt{(x-y)^\intercal W(x-y)}$ will be small for semantically close x and y. Several extensions of the setting introduced by Xing et al. (2003) were proposed (Shalev-Shwartz et al., 2004; Globerson & Roweis, 2006; Weinberger & Saul, 2009; Ying & Li, 2012). For the case of deep metric learning, on the other hand, one's interest is to learn a non-linear encoder $\mathcal{E} : \mathbb{R}^D \mapsto \mathbb{R}^d$, and often $D \gg d$, so that some *standard distance* such as the one based on the L_2 norm $||\mathcal{E}(x) - \mathcal{E}(y)||_2$ will be small for semantically close x and y. While the two discussed settings are equivalent in their final goal, *their focus is not the same*: distance metric learning approaches focus on learning a semantically meaningful distance measure while deep metric learning methods focus on projecting the data onto a space where standard distances are meaningful.

A relatively recent research direction consists in the combination of the above described settings, i.e. jointly training an encoder along with a distance/similarity model. This is the case in (Koch et al., 2015) where a symmetric model was used to map the absolute difference of a pair of representations into a similarity score. In (Garcia & Vogiatzis, 2019), training of a distance/similarity model is done by imitation learning of cosine similarities measured between representations, which authors claim to simplify training compared to the direct use of cosine scores. In (Pitis et al., 2020), authors focus on distance models supporting asymmetric properties of the data, while still satisfying the triangle inequality. Learned Bregman divergences were evaluated by Cilingir et al. (2020), and completely unconstrained similarity models, in the sense that any property such as symmetry is imposed in the learned distance, were proposed in (Monteiro et al., 2020) for verification tasks. Learnable similarities parametrized by neural networks were further employed in (Wenliang et al., 2019; Liu et al., 2020) for the implementation of learned kernels, used to perform MMD-based (Gretton et al., 2012) 2-sample tests. For the case of few-shot learning under geometric approaches, so-called *prototypical networks* (Snell et al., 2017) follow a similar idea to that of metric learning in the sense that training consists of building a metric space where distances are indicative of properties of the data. However, that setting introduced the idea of partitioning the said space using class prototypes, i.e. a set of vectors representing each class, thus enabling its use to classification tasks since one can assign a test instance to the class corresponding to its closest prototype. That approach was also used under few-shot classification settings, in which case a new partitioning of the space is computed once small samples corresponding to new classes are presented to the model. We thus propose a strategy to extend the setting in (Monteiro et al., 2020) and include a partitioning with prototypes in the learned *pseudo* metric space so that the final model can be used to perform tasks such as multi-class and few-shot classification, while still supporting tasks involving pairwise comparisons.

3 SOLVING DIFFERENT TASKS THROUGH LEARNED SIMILARITIES

Assume (x, y) represents instances from $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^D$ is the data space while $\mathcal{Y}$ is the space of labels, which will be always a discrete set in the cases considered herein. We thus consider the setting where the following components are to be learned: an *encoder* $\mathcal{E} : \mathbb{R}^D \mapsto \mathbb{R}^d$, a *similarity* model $\mathcal{S} : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$, and a set of *prototypes* $\mathcal{C} \in \mathbb{R}^{|\mathcal{Y}| \times d}$. As it will be further discussed, such three components can be then used to perform different types of inference regarding properties of underlying data, and thus solve different tasks.

3.1 TRAINING

Training is carried out to enforce the following properties: **i**-the similarity as measured between a particular example and the prototype corresponding to its class labels should be high; **ii**-the similarities measured between examples from the same class should be high, while examples from different classes should yield a low similarity score. We thus design training objectives aimed at enforcing such properties. For the first property, we consider a training sample of size m and employ the standard multi-class cross-entropy criterion, but *use the similarity measured between a training instance and each prototype as the set of logits* as opposed to output layers defined by an affine transformation, commonly employed in standard classifiers, i.e. we perform maximum likelihood estimation on the categorical conditional distribution defined by:

$$P(\mathcal{Y}|x') = \text{softmax}(\mathcal{S}(\mathcal{E}(x'), \mathcal{C}_{1:|\mathcal{Y}|})), \tag{1}$$

where $\mathcal{C}_i$, $i \in [|\mathcal{Y}|]$, indicates the prototype corresponding to class i. The corresponding training loss, denoted $\mathcal{L}_{class}$, will be then given by:

$$\mathcal{L}_{class} = -\frac{1}{m}\sum_{i=1}^{m} \log \frac{e^{\mathcal{S}(\mathcal{E}(x_i),\mathcal{C}_{y_i})}}{\sum_{j=1}^{|\mathcal{Y}|} e^{\mathcal{S}(\mathcal{E}(x_i),\mathcal{C}_j)}}, \tag{2}$$

where x_i and y_i indicate the i-th training example. In order for the learned similarity to be *meaningful* for pairwise comparisons (i.e. the second property highlighted in the previous discussion), we make use of the following binary classification objective, used before in the settings discussed in (Auckenthaler et al., 2000) and (Monteiro et al., 2020), aimed at discriminating pairs of examples from the same and from different classes:

$$\mathcal{L}_{pair} = -\frac{1}{|T^+|}\sum_{x^+ \in T^+} \log(\sigma(\mathcal{S}(\mathcal{E}(x^+)))) - \frac{1}{|T^-|}\sum_{x^- \in T^-} \log(1-\sigma(\mathcal{S}(\mathcal{E}(x^-)))), \tag{3}$$

where σ stands for the logistic function, and x^+ and x^- indicate pairs of examples denominated *trials* and denoted by T, i.e. $T = \{x', x''\}$. The sums are taken over the set of positive or target trials T^+ obtained from the training sample, i.e. those for which x' and x'' belong to the same class, and the set of negative or non-target trials T^-. We further define the application of the encoder over a trial as $\mathcal{E}(T) = \{\mathcal{E}(x'), \mathcal{E}(x'')\}$

Initializing and updating the list of prototypes: $\mathcal{C}$ is initialized randomly such that its entries are i.i.d. sampled from a standard Gaussian distribution. We thus update $\mathcal{C}$ every iteration through a moving average given at iteration t by: $\mathcal{C}_t = \lambda\mathcal{C}_{t-1} + (1-\lambda)\mathcal{C}^*_{t-1}$, where $\lambda \in [0,1]$ is a hyperparameter, and $\mathcal{C}^*_{t-1}$ is a copy of $\mathcal{C}_{t-1}$ where the rows corresponding to classes observed in the current minibatch are substituted by the average representations of each such classes.

Practical details: We now describe the design choices made so as to implement the discussed components and minimize $\mathcal{L}$. Both $\mathcal{E}$ and $\mathcal{S}$ are implemented as neural networks, while $\mathcal{C}$ is a matrix where each row represents a prototype for a particular class.

Training is carried out with stochastic gradient descent with gradients estimated over minibatches of training data. In order to compute $\mathcal{L}_{pair}$, each minibatch has to contain multiple examples from the same class otherwise T^+ will be empty. We thus sample minibatches ensuring that is the case (c.f. appendix for further implementation details). We empirically observed including a standard classification loss accelerates convergence across all evaluations performed. We thus include a dense output layer to allow for computation of such a loss, which we denote by $\mathcal{L}_{aux}$. Training is thus carried out to minimize the total loss $\mathcal{L} = \mathcal{L}_{class} + \mathcal{L}_{pair} + \mathcal{L}_{aux}$. A high-level training procedure is depicted in Algorithm 1 and illustrated in Figure 3 in the appendix.

Algorithm 1 Training procedure.

$\mathcal{E}, \mathcal{S} = InitializeModels()$
$\mathcal{C} = InitializePrototypes()$
repeat
  $x, y = SampleMinibatch()$
  $z = \mathcal{E}(x)$
  $\mathcal{C} = UpdatePrototypes(z, y, \mathcal{C})$
  $z^+ = GetPositivePairs(z, y)$
  $z^- = GetNegativePairs(z, y)$
  $y' = DenseLayer(z)$
  $\mathcal{L} = \mathcal{L}_{pair} + \mathcal{L}_{class} + \mathcal{L}_{aux}$
  $\mathcal{E}, \mathcal{S} = UpdateRule(\mathcal{E}, \mathcal{S}, \mathcal{L})$
until Maximum number of iterations reached
return $\mathcal{E}, \mathcal{S}, \mathcal{C}$

3.2 TESTING

We now define the set of tasks one can tackle using trained $\mathcal{E}$, $\mathcal{S}$, and $\mathcal{C}$ along with the inference mechanisms employed for each such task.

Multi-class classification: For the case where one is given a test instance x' and desires to determine its class label y', it will be given by the following classifier:

$$\arg\max_{i \in [|\mathcal{Y}|]} \mathcal{S}(\mathcal{E}(x'), \mathcal{C}_i). \tag{4}$$

Few-shot classification: If new classes are considered after training, repartitioning can be performed with few data points by creating a new set of prototypes $\mathcal{C}'$ defined such that each entry corresponds to the average representation of each new class. Inference is thus performed following the schemed defined for multi-class classification.

Verification: Now assume one is given a trial $\{x', x''\}$ and desires to determine whether their respective labels are such that $y' = y''$. One can then do the following:

$$\mathbb{1}_{\mathcal{S}(\mathcal{E}(x'),\mathcal{E}(x''))>\tau}, \tag{5}$$

where τ is an user-defined decision threshold.

Retrieval: Given a test instance x' and a gallery of instances denoted by $X = \{x_1, x_2, ..., x_n\} : x_i \in \mathcal{X} \; \forall \; i \in [n]$, determine k elements in X such that their labels match the underlying label y' of x'. The result will thus be:

$$\underset{x'' \in X}{\text{k-arg max}} \, \mathcal{S}(\mathcal{E}(x'), \mathcal{E}(x'')), \tag{6}$$

where the operator k-arg max denotes repeating the $\arg\max$ operator k times, removing the current result each time prior to the next $\arg\max$ operation.

4 EVALUATION

Our goal is to devise a general framework which is able to perform different tasks. As such, the evaluation we decided to carry out consists of testing the proposed approach across a wide variety of tasks and modalities of data. Our evaluations consist of: multi-class classification, in which case we evaluate convolutional classifiers on MNIST (LeCun et al., 2010) and CIFAR-10 (Krizhevsky et al., 2009) and *show improved robust accuracy*. We further perform evaluation on object recognition tasks considering larger resolution images under domain shift. For that, we employ the standard PACS benchmark (Li et al., 2017) where we show that the prototypical classifiers with learned similarities introduced herein *outperform recently introduced alternatives under domain shift*, and mainly do so in the most challenging cases where a notable domain mismatch is observed (e.g. natural images vs. sketches). We then proceed to tasks that rely on pairwise comparisons of test instances and run evaluations on a large scale verification task on audio using the VoxCeleb corpus (Nagrani et al., 2017; Chung et al., 2018), and then evaluate our proposed approach on image retrieval tasks employing popular benchmarks such as CARS196 (Krause et al., 2013) and CUB200-2011 (Wah et al., 2011). Finally, the appendix contains evaluations discussing how to easily repartition the space so that new classes can be evaluated at test time, in which case we report experiments using *mini*ImageNet (Vinyals et al., 2016). Ablations are further reported in the appendix using the full ImageNet (Deng et al., 2009) to show the importance of the use of the auxiliary classification loss.

We remark that the training procedure presented in Algorithm 1 is employed for *training models used for all tasks discussed above*, and no specialization to any task of interest is performed since we seek evidence regarding how effective the proposed approach is in yielding a general enough set of components (i.e. $\mathcal{E}$, $\mathcal{S}$, and $\mathcal{C}$) which perform on pair or better than alternatives.

4.1 MULTI-CLASS CLASSIFICATION

4.1.1 ROBUSTNESS AGAINST ADVERSARIES

We report in Table 1 the accuracy obtained using convolutional classifiers trained on MNIST and CIFAR-10 considering both clean data as well as FGSM (Goodfellow et al., 2014) and PGD (Madry et al., 2017) attacks[1] under L_∞ norm budgets, and considering the white-box access model so that the attacker has full access to the target model. Models trained using Algorithm 1 are compared against previously proposed defense strategies. Specifically, adversarial training (AT) (Madry et al., 2017), adversarial logit pairing (ALP) (Kannan et al., 2018), triplet loss adversarial training (TLA) (Mao et al., 2019), and TRADES (Zhang et al., 2019) are considered for comparison. The results given by an undefended model as reported by Mao et al. (2019) are further included for reference.

[1]Attacks were implemented using *FoolBox*: `https://foolbox.readthedocs.io/en/stable/index.html`

A standard LeNet and the wide residual architecture introduced in (Madry et al., 2017) were employed for the cases of MNIST and CIFAR-10, respectively, and an attack budget of 0.3 and $\frac{8}{255}$ was considered when each such dataset was evaluated. We evaluated our models both with and without adversarial training, and report the results obtained when inference is performed using the scheme represented in expression 4, which we denote by *SIM* so as to indicate that the similarity model $\mathcal{S}$ is used for inference, as well as for the case where the auxiliary output layer used to compute $\mathcal{L}_{aux}$ is used to predict label of test samples, which we indicate by DOL in a reference to *dense output layer*. In order to have a full white-box access model, each such output layer is exposed to the attacker in each evaluation so that attacks are created accounting for the specific inference procedure that will be used for testing. Based on the reported results, we verify that similarity/distance based inference is inherently less affected by small norm adversarial perturbations given that for both the cases of MNIST and CIFAR-10 with our undefended models the robust accuracy across considered attacks was much higher for the *SIM* case as compared to *DOL* as well as the undefended standard classifier. With adversarial training, both inference mechanisms yield higher performance than considered alternatives against several attackers, and more importantly, *without affecting the clean accuracy as much as previous methods*.

Table 1: Adversarial robustness evaluation in term of accuracy (%) considering PGD and FGSM attackers under L_∞ budgets of 0.3 and $\frac{8}{255}$ for the cases of MNIST and CIFAR-10, respectively. The number of steps employed for each attack is represented within parenthesis. We consider evaluations obtained with the similarity classifier as indicated by *SIM* as well as utilizing the auxiliary output layer which we indicate by *DOL*.

MNIST					
	General purpose	Clean	PGD (40)	PGD (100)	FGSM (1)
Undefended	✓	99.20	0.00	0.00	34.48
AT	×	99.24	97.31	96.58	94.82
ALP	×	98.91	97.34	96.62	95.06
TLA	×	99.52	**98.17**	**97.72**	96.96
TRADES ($1/\lambda = 6$)	×	99.48	96.07	-	95.6
Ours - *DOL*	✓	99.31	0.02	0.01	17.18
Ours - *SIM*	✓	**99.36**	23.85	13.61	68.51
Ours - *DOL* + AT	×	98.71	95.04	93.78	97.62
Ours - *SIM* + AT	×	98.79	95.35	93.98	**97.84**
CIFAR-10					
	General purpose	Clean	PGD (7)	PGD (20)	FGSM (1)
Undefended	✓	95.01	0.00	0.00	13.35
AT	×	87.14	55.63	49.79	45.72
ALP	×	89.79	60.29	51.89	48.5
TLA	×	86.21	58.88	53.87	51.59
TRADES ($1/\lambda = 1$)	×	88.64	-	49.14	48.9
TRADES ($1/\lambda = 6$)	×	84.92	-	**56.61**	56.43
Ours - *DOL*	✓	**96.20**	47.39	9.27	57.42
Ours - *SIM*	✓	96.14	65.46	27.87	**65.46**
Ours - *DOL* + AT	×	93.29	**80.73**	54.29	46.85
Ours - *SIM* + AT	×	92.55	80.04	56.13	52.98

4.1.2 Robustness under domain shift

We now assess the performance of the proposed classification strategy once domain shifts across train and test data occur. We do so by making use of the PACS domain-generalization benchmark (Li et al., 2017) consisting of 224x224 RGB images distributed into 7 classes and originated from four different *domains*: Photo (P), Art painting (A), Cartoon (C), and Sketch (S). We thus follow the *leave-one-domain-out* evaluation protocol such that data from three out of the four available domains are used for training while evaluation is carried out on the data from the left out domain. A comparison is carried out with recent methods specifically designed to enable out-of-distribution generalization introduced in (Dou et al., 2019) and (Chattopadhyay et al., 2020), as well as with the results reported in (Gulrajani & Lopez-Paz, 2020) where standard classifiers were evaluated against domain generalization approaches. Experiments were carried out using a ResNet-50 (He et al., 2016) pretrained on ImageNet.

Considering the average performance once each domain is left out, we once again observe improved robustness once similarity-based classification is employed when compared to both standard classifiers and domain generalization approaches, which indicates such classifiers rely less on domain factors that might correlate with labels on training domains. We hypothesize such a feature comes

Table 2: Evaluation on the PACS benchmark in terms of accuracy (%) for the cases where each of the available domains are left out of training.

	General purpose	P	A	C	S	Average
Dou et al. (2019)	×	95.01	82.89	80.49	72.29	82.67
Gulrajani & Lopez-Paz (2020)	✓	**97.80**	**88.10**	78.00	79.10	85.75
Chattopadhyay et al. (2020)	×	94.49	82.57	78.11	78.32	83.37
Ours - *SIM*	✓	97.07	86.38	**83.66**	**84.63**	**87.93**

from the metric learning framework used to train our models, i.e. domain information is less helpful when trying to minimize the combination of $\mathcal{L}_{class}$ and $\mathcal{L}_{pair}$, which renders resulting classifiers less dependent on the underlying domains used at training time. A gap in performance in our favor can be particularly observed for the evaluation cases where domains corresponding to cartoons and sketches are left out, given that such domains present a large discrepancy compared to the natural images that compose the bulk of training data. In fact, for the case photos on the other hand, in which case the underlying data correspond to natural images, the standard classifier discussed in (Gulrajani & Lopez-Paz, 2020) outperforms our model.

4.2 VERIFICATION

In order to perform evaluations that rely on pairwise comparisons of test instances, we consider the verification setting where trials corresponding to pairs of test instances are presented to the model. Its task is then to decide whether the examples in the trial belong to the same class. We thus make use of the VoxCeleb corpus (Nagrani et al., 2017; Chung et al., 2018) which consists of a large scale set of audio clips collected from videos of interviews available online.

We compared models trained on the second release of the corpus, which is composed of audio recordings from 5994 different speakers, against a set of published results on the three test partitions made available along with that release: (**i**)-*VoxCeleb1 Test set*, which correspond to data obtained from 40 speakers, (**ii**)-*VoxCeleb1-Extended*, which is given by the complete first release of the corpus and contains 1251 speakers, and (**iii**)-*VoxCeleb1-Hard*, which is made up of a subset of the data from *VoxCeleb1-Extended* yielding trials known to be hard to distinguish. We highlight that *the set of speakers represented in all test partitions is disjoint to that relative to train data*. The encoder $\mathcal{E}$ in this case corresponds to the 1-dimensional convolutional model introduced by Snyder et al. (2018). Details on the audio pre-processing and feature extraction are included in the appendix.

Results are reported in terms of equal error rate (EER) in Table 3. EER corresponds to any coordinate of the point in the *detection error tradeoff curve* where false positive and miss detection rates match (the lower the better). In this case, we take advantage of the fact that cosine similarities can be further used to compare representations of test trials, and observed combining it with the scores given by $\mathcal{S}$ by simply summing both similarities yields a further boost in performance.

Table 3: Verification performance on the VoxCeleb test partitions reported in terms of EER (%).

	Scoring	General purpose	EER (%)		
			VoxCeleb1 test set	Extended	Hard
Chung et al. (2018)	Cosine	×	3.95	4.42	7.33
Xie et al. (2019)	Cosine	×	3.22	3.13	5.06
Hajavi & Etemad (2019)	Cosine	×	4.26	-	-
Xiang et al. (2019)	Cosine	×	2.69	2.76	4.73
Monteiro et al. (2020)	PLDA	×	2.51	2.60	4.62
Monteiro et al. (2020)	SIM	×	2.51	2.57	4.73
Ours	Cosine	✓	2.62	2.69	4.48
Ours	*SIM*	✓	2.55	2.75	4.76
Ours	Cosine + *SIM*	✓	**2.45**	**2.55**	**4.39**

4.3 IMAGE RETRIEVAL

We further verified the performance of the proposed approach on another set of tasks which require comparisons of pairs of examples. In this case, we considered the retrieval setting where a test example is compared against a gallery and k "similar" examples need to be selected from that gallery.

We thus make use of the CARS196 (Krause et al., 2013) and CUB200-2011 (Wah et al., 2011) datasets and closely follow the evaluation protocol discussed by Wu et al. (2017).

Results are reported in terms of *Recall@k* (Oh Song et al., 2016) (the higher the better), or *R@k* for short, and summarized in Figures 1 and 2, while the complete set of results is reported in Tables 7 and 8 in the appendix. Compared approaches consist of several metric learning methods designed for the retrieval problem. We use the indicators + and - to refer to the highest and lowest performances amongst the considered baselines. Results were obtained considering a ResNet-50 pretrained on ImageNet and fine tuned on each of the considered datasets. As a reference, we further report the results obtained by the pretrained model prior to fine tuning. We thus claim the proposed approach is competitive in that its performance lies close to the + line (which corresponds to a strong baseline using an ensemble of metrics approach (Sanakoyeu et al., 2019)) and outperforms most of the compared methods while using a much simpler and general training procedure requiring no special mining strategy of hard triplets, and using moderate batch sizes enabling practical training in single GPU hardware, as opposed to complex metric learning pipelines.

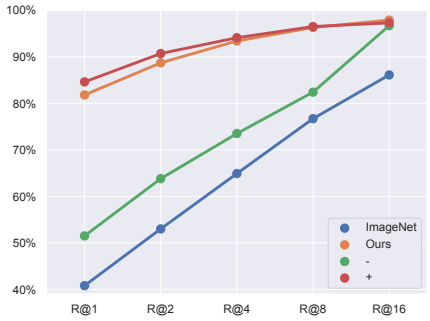

Figure 1: *R@K* evaluation of proposed methods on the CARS196 dataset.

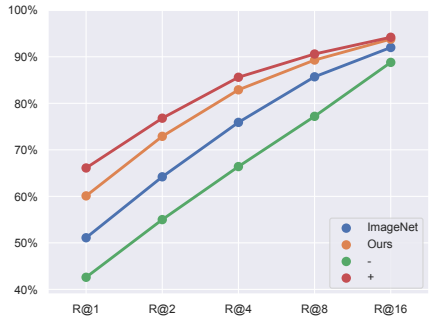

Figure 2: *R@K* evaluation of proposed methods on the CUB200-2011 dataset.

5 Conclusion

We introduced a metric learning scheme which enables different types of tasks to be performed using the same set of models: an encoder responsible for embedding data into a lower dimensional space, a similarity model which outputs a similarity score given a pair of embeddings, and a set of class prototypes where each one represents a class observed during training. We then presented empirical evidence showing such a setting yields improvements in long standing issues such as adversarial robustness, since small perturbations in norm were observed to have a lesser effect on distance-based inference compared to standard classifiers, as well as robustness against distribution shifts, which indicates the posed training strategy is more effective in avoiding models that rely on correlations between training domain factors and labels, since domain information is not as helpful for such training scheme as it can be for the case of maximum likelihood estimation with standard classifiers. Moreover, performance across a set of tasks such as verification and image retrieval further showed classifiers defined under the proposed scheme perform competitively or better than alternatives designed targeting their particular applications, thus representing a step towards defining models that fit in the *one model to solve them all* category.

In terms of future work, we conjecture that the scope of the proposed setting can be further enlarged by utilizing the kernel function given by $\mathcal{K} = \mathcal{S}$, thus defining a strategy to learn kernels tailored to the data of interest, similarly to past work such as (Wenliang et al., 2019) and (Liu et al., 2020). Doing so might then be effective for performing even further tasks such as non-parametric 2-sample tests based on MMD scores (Gretton et al., 2012), as well as outlier/novelty detection, in which case approaches such one-class SVMs (Schölkopf et al., 2000) using the learned kernel can be considered. We further remark that, in order to define a Mercer's kernel using $\mathcal{K}$, one can then consider inducing symmetry through a kernel $\mathcal{K}_S$ defined by $\mathcal{K}_S = f(\mathcal{K}(x', x''), \mathcal{K}(x'', x'))$, where $f : \mathbb{R}^2 \mapsto \mathbb{R}$ is symmetric, e.g. $f(\cdot, \cdot) = \max(\cdot, \cdot)$.

REFERENCES

Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H Falk, and Ioannis Mitliagkas. Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*, 2019.

Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1-3):42–54, 2000.

Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, volume 2. Univ. of Texas, 1992.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a siamese time delay neural network. In *Advances in neural information processing systems*, pp. 737–744, 1994.

Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. *arXiv preprint arXiv:2008.12839*, 2020.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 539–546. IEEE, 2005.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *Proc. Interspeech 2018*, pp. 1086–1090, 2018.

Kubra Cilingir, Rachel Manzelli, and Brian Kulis. Deep divergence learning. In *International Conference on Machine Learning*, 2020.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, pp. 6450–6461, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135, 2017.

Noa Garcia and George Vogiatzis. Learning non-metric visual similarity for image retrieval. *Image and Vision Computing*, 82:18–25, 2019.

Amir Globerson and Sam T Roweis. Metric learning by collapsing classes. In *Advances in neural information processing systems*, pp. 451–458, 2006.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. *ArXiv e-prints*, December 2014.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceeedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1735–1742. IEEE, 2006.

Amirhossein Hajavi and Ali Etemad. A deep neural network for short-segment speaker recognition. *Proc. Interspeech 2019*, pp. 2878–2882, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5220–5224. IEEE, 2017.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, 2019.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Dougal J Sutherland. Learning deep kernels for non-parametric two-sample tests. In *International Conference on Machine Learning*, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. Metric learning for adversarial robustness. In *Advances in Neural Information Processing Systems*, pp. 480–491, 2019.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Joao Monteiro, Isabela Albuquerque, Jahangir Alam, R Devon Hjelm, and Tiago Falk. An end-to-end approach for the verification problem: learning the right distance. In *International Conference on Machine Learning*, 2020.

Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning*, pp. 3664–3673. PMLR, 2018.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: A large-scale speaker identification dataset. *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4004–4012, 2016.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 721–731, 2018.

Silviu Pitis, Harris Chan, Kiarash Jamali, and Jimmy Ba. An inductive bias for distances: Neural nets that respect the triangle inequality. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HJeiDpVFPr`.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Karsten Roth, Biagio Brattoli, and Bjorn Ommer. Mic: Mining interclass characteristics for improved metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8000–8009, 2019.

Artsiom Sanakoyeu, Vadim Tschernezki, Uta Buchler, and Bjorn Ommer. Divide and conquer the embedding space for metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 471–480, 2019.

Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-... hook*. Diplomarbeit, Technische Universität München, München, 1987.

Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *Advances in neural information processing systems*, pp. 582–588, 2000.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Shai Shalev-Shwartz, Yoram Singer, and Andrew Y Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 94. ACM, 2004.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pp. 4077–4087, 2017.

David Snyder, Guoguo Chen, and Daniel Povey. MUSAN: A Music, Speech, and Noise Corpus, 2015. arXiv:1510.08484v1.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust DNN embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333. IEEE, 2018.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pp. 1857–1865, 2016.

Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Learnable structured clustering framework for deep metric learning. *arXiv preprint arXiv:1612.01213*, 1(2):8, 2016.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Evgeniya Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss. In *Advances in Neural Information Processing Systems*, pp. 4170–4178, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pp. 3630–3638, 2016.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

Li Wenliang, Dougal Sutherland, Heiko Strathmann, and Arthur Gretton. Learning deep kernels for exponential family densities. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6737–6746, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL `http://proceedings.mlr.press/v97/wenliang19a.html`.

Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2840–2848, 2017.

Xu Xiang, Shuai Wang, Houjun Huang, Yanmin Qian, and Kai Yu. Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition. *arXiv preprint arXiv:1906.07317*, 2019.

Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Utterance-level aggregation for speaker recognition in the wild. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5791–5795. IEEE, 2019.

Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pp. 521–528, 2003.

Yiming Ying and Peng Li. Distance metric learning with eigenvalue optimization. *Journal of machine Learning research*, 13(Jan):1–26, 2012.

Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. In *Proceedings of the IEEE international conference on computer vision*, pp. 814–823, 2017.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pp. 7472–7482, 2019.

SUPPLEMENTARY MATERIAL

A ABLATION STUDY

We performed ablations using the full ImageNet in order to assess the importance of the use of the auxiliary loss $\mathcal{L}_{aux}$. We thus trained models with and without $\mathcal{L}_{aux}$ in the training partition of ImageNet using Algorithm 1 and performed evaluations in terms of multi-class classification performance and verification. Results for each such case are reported in Tables 4 and 5. In this case, $\mathcal{E}$ was implemented as the convolutional stack of a ResNet-50. For the classification case, performance is assessed in terms of top-1 and top-5 accuracy, while EER and the area above the operation curve (i.e. 1-AUC) are reported for the case of verification. Verification trials were defined by creating all possible pairs of examples out of the test data. In both cases, removing the auxiliary output layer negatively affected performance, more notably so in the case of verification using cosine scores. Since such an auxiliary layer does not add any significant cost of any sort and boosts performance across considered evaluations, its use is justified.

Table 4: Classification performance in terms of accuracy (%).

	Top-1	Top-5
DOL	73.15	91.26
SIM	71.33	90.33
Ablation	70.37	89.85

Table 5: Verification performance in terms of EER (%) and 1-AUC (%).

	Scoring	EER	1-AUC
Complete model	SIM	3.54	0.63
Complete model	Cosine	5.33	1.60
Ablation	SIM	4.02	0.75
Ablation	Cosine	10.06	3.93

B FEW-SHOT CLASSIFICATION

We further evaluate the proposed framework under the few-shot classification setting in which case new classes are presented to the model after training, and small samples from each such novel class are made available. We run evaluations considering the *mini*ImageNet (Vinyals et al., 2016) which consists of a subset of 100 classes from ImageNet containing 600 images for each class. We follow the setting introduced by Ravi & Larochelle (2016), which splits the data so that 64, 16, and 20 disjoint sets of classes are included in the training, validation, and testing partitions, respectively.

Results are reported in terms of average top-1 accuracy along with boundaries of its 95% confidence interval considering a sample of 1000 randomly selected tasks. Each task is randomly created by giving to the model a set containing *N-ways* classes and *K-shots* examples per class, while 15 test examples per class are evaluated in each task. The encoder $\mathcal{E}$ is implemented as the convolutional stack of the ResNet-12 architecture, which is also used in all the compared approaches.

A comparison is carried out with a set of approaches carefully designed for the few-shot classification setting, sometimes including sophisticated adaptation schemes for the novel classes and having the evaluation process simulated at training time with the so-called *episodic training*. For the case of our model on the other hand, we intend to verify how simply training it using Algorithm 1 fares against such specialized methods. We thus train our models on the training partition of *mini*ImageNet, and once data from novel classes is given at test time, we make use of it to build a novel set of class prototypes C' which is then used to define the classifier expressed in 4, i.e. no fine tuning of $\mathcal{E}$ and $\mathcal{S}$ is performed. We thus show such a simple approach yields a performance inline with recent sophisticated approaches. We additionally computed the accuracy yielded by substituting $\mathcal{S}$ by the cosine similarity and observed summing up both scores yielded an accuracy gain for the 1-shot case.

Table 6: 5-way few-shot classification on *mini*ImageNet. Results consist of average top-1 accuracy along with confidence intervals considering 1000 randomly selected tasks. All evaluations consider a ResNet-12 architecture.

	General purpose	1-shot	5-shots
MatchNet (Vinyals et al., 2016)	×	63.08±0.80	75.99±0.60
SNAIL (Mishra et al., 2018)	×	55.71±0.99	68.88±0.92
AdaResNet (Munkhdalai et al., 2018)	×	56.88±0.62	71.94±0.57
TADAM (Oreshkin et al., 2018)	×	58.50±0.30	76.70±0.30
MetaOptNet (Lee et al., 2019)	×	62.64±0.61	78.63±0.46
Prototypical Networks (Snell et al., 2017; Lee et al., 2019)	×	59.25±0.64	75.60±0.48
Ours - Cosine	✓	59.00±0.65	77.52±0.49
Ours - *SIM*	✓	60.28±0.65	75.26±0.52
Ours - Cosine + *SIM*	✓	61.27±0.65	76.83±0.50

C RETRIEVAL RESULTS

The complete set of results summarized in Figures 1 and 2 are reported in Tables 7 and 8 for the cases of CARS196 and CUB200-2011, respectively.

Table 7: *R@K* (%) evaluation of proposed methods on the CARS196 dataset.

	General purpose	R@1	R@2	R@4	R@8	R@16
Schroff et al. (2015)	×	51.5	63.8	73.5	82.4	–
Oh Song et al. (2016)	×	53.0	65.7	76.0	84.3	–
Song et al. (2016)	×	58.1	70.6	80.3	87.8	–
Sohn (2016)	×	71.1	79.7	86.5	91.6	–
Yuan et al. (2017)	×	73.7	83.2	89.5	93.8	96.7
Wu et al. (2017)	×	79.6	86.5	91.9	95.1	97.3
Roth et al. (2019)	×	82.6	89.1	93.2	–	–
ImageNet	✓	40.8	53.0	64.9	76.7	86.1
SIM	✓	81.8	88.7	93.4	96.3	97.9
Sanakoyeu et al. (2019) (Ensemble of metrics)	×	84.6	90.7	94.1	96.5	–

Table 8: *R@K* (%) evaluation of proposed methods on the CUB200-2011 dataset.

	General purpose	R@1	R@2	R@4	R@8	R@16
Ustinova & Lempitsky (2016)	×	52.8	64.4	74.7	83.9	90.4
Ustinova & Lempitsky (2016)	×	50.3	61.9	72.6	82.4	88.8
Schroff et al. (2015)	×	42.6	55.0	66.4	77.2	–
Oh Song et al. (2016)	×	43.6	56.6	68.6	79.6	–
Song et al. (2016)	×	48.2	61.4	71.8	81.9	–
Sohn (2016)	×	51.0	63.3	74.3	83.2	–
Yuan et al. (2017)	×	53.6	65.7	77.0	85.6	91.5
Wu et al. (2017)	×	63.6	74.4	83.1	90.0	94.2
Roth et al. (2019)	×	66.1	76.8	85.6	–	–
ImageNet	✓	51.1	64.6	75.9	85.7	92.0
SIM	✓	60.1	72.9	82.9	89.3	93.8
Sanakoyeu et al. (2019) (Ensemble of metrics)	×	65.9	76.6	84.4	90.6	–

D PRACTICAL DETAILS

An illustration of the proposed model is depicted in Figure 3.

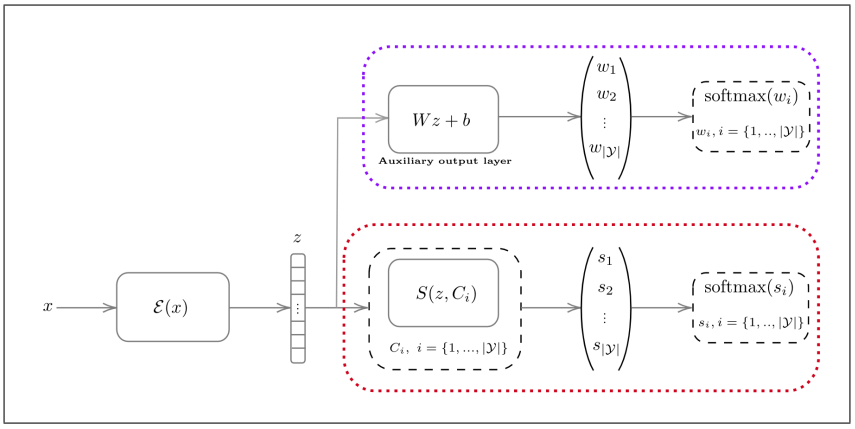

Figure 3: Illustration of the proposed model.

Training details: Training was carried out with SGD with momentum for most of the experiments, except for the case of MNIST, where Adam with default hyperparameters was employed, and retrieval where Adam was employed with an initial learning rate of $1e-5$ reduced by a factor of 0.1 on epochs 20 and 50, and hyperparameters such as β_1 and β_2 were set to 0.5 and 0.999, respectively. Standard schedules for the learning rate were employed for the case of CIFAR-10, consisting of a reduction by a factor of 0.1 every 30 epochs, while a decay every 10 epochs was used for ImageNet experiments, in which case pretrained encoders were employed to speed up convergence. For the specific case of VoxCeleb, we found the additive margin softmax (Wang et al., 2018) yielded higher performance when used as an auxiliary loss, and in that case the learning rate schedule introduced in (Vaswani et al., 2017) was employed. We set $\lambda - 0.9$ across all experiments. Regularization strategies such as weight decay and label smoothing (Szegedy et al., 2016) were further employed.

Minibatch construction for cases corresponding to large label sets: In cases where the size of the label set $|\mathcal{Y}|$ was larger than the batch sizes being employed, which was the case for experiments performed with ImageNet, VoxCeleb, and the retrieval datasets, we needed to defined a sampling strategy which would ensure that multiple examples from observed classes would appear in each minibatch so as to enable the definition of target trials T^+ in order to compute $\mathcal{L}_{pair}$, as per its definition in Equation 3. We thus build minibatches such that 5 examples per class are included for each class observed, and we make sure to update such groups of 5 examples every epoch to allow for diverse minibatches throughout training.

Data preparation for verification experiments on VoxCeleb: The audio data from VoxCeleb is augmented following the procedure discussed by Snyder et al. (2018). We thus add reverberation, using reverberation times within 0.25s - 0.75s, and further add background noise consisting of music samples (SNR within 5-15dB), and babble samples (SNR within 10-20dB). Noise samples were picked from the MUSAN corpus (Snyder et al., 2015) while room impulse responses used to simulate reverberation were picked from (Ko et al., 2017). The data used for distortions of the original audio are available at `https://www.openslr.org/`. Features of audio are extracted such that 30 mel-frequency cepstral coefficients are obtained with a short-time Fourier transform using a 25ms Hamming window with 60% overlap. Audio is downsampled to 16kHz and a simple energy voice activity detector filters out silent frames.