Guided Deep Metric Learning

Jorge Gonzalez-Zapata[1], Ivan Reyes-Amezcua[1], Daniel Flores-Araiza[2], Mauricio Mendez-Ruiz[2],
Gilberto Ochoa-Ruiz[2], Andres Mendez-Vazquez[1]

[1]CINVESTAV Unidad Guadalajara, Mexico
[2]Tecnológico de Monterrey, School of Engineering and Sciences, Mexico
jorge.gonzalezzapata@cinvestav.mx, gilberto.ochoa@tec.mx, andres.mendez@cinvestav.mx

Abstract

Deep Metric Learning (DML) methods have been proven relevant for visual similarity learning. However, they sometimes lack generalization properties because they are trained often using an inappropriate sample selection strategy or due to the difficulty of the dataset caused by a distributional shift in the data. These represent a significant drawback when attempting to learn the underlying data manifold. Therefore, there is a pressing need to develop better ways of obtaining generalization and representation of the underlying manifold. In this paper, we propose a novel approach to DML that we call Guided Deep Metric Learning, a novel architecture oriented to learning more compact clusters, improving generalization under distributional shifts in DML. This novel architecture consists of two independent models: A multi-branch master model, inspired from a Few-Shot Learning (FSL) perspective, generates a reduced hypothesis space based on prior knowledge from labeled data, which guides or regularizes the decision boundary of a student model during training under an offline knowledge distillation scheme. Experiments have shown that the proposed method is capable of a better manifold generalization and representation to up to 40% improvement (Recall@1, CIFAR10), using guidelines suggested by Musgrave et al. to perform a more fair and realistic comparison, which is currently absent in the literature.

1. Introduction

DML has proven to be a relevant topic given its strategy of acting directly on the resulting embedding distances to capture the semantic similarity of the data, using the robustness of deep learning models. Over time, DML methods have been integrated to task such as zero-shot [23, 29], few-shot [35, 36] and self-supervised learning [16, 22, 28].

Baseline DML methods have a variety of proposals with different loss functions [8, 17]. However, pair or triplet samples imply high complexity time in the training process. Thus, were introduced sample mining strategies [13, 17]. Still, these strategies do not generalize well to all architectures and can be counterproductive depending on the nature of the data or architecture [25, 40]. In addition, the experiments realized in [30] and [25] indicate overall flaws in the experimental setups in DML that lead to unfair comparison.

Most Machine Learning (ML) models usually assume that the train and test data are drawn from the same distribution (i.i.d. assumption). However, in a realistic scenario, distributional shifts between the train and test set can occur, where the test distribution is unknown and diverges from the train distribution. Precisely, Out-Of-Distribution (OOD) generalization address this problem [31].

While there is still a vague definition of OOD in the literature and characterization of distributional shift is still an open problem [31], we have opted to use these concepts using several references [19, 23, 31, 42] that remain congruent in certain properties. For example, the distributional shifts can be caused by *semantic shift* (or label shift), in which the OOD samples belong to new classes, or by *covariate shift*, where the distribution of the data changes between the train and test scenarios while keeping the same labels. Our proposal focuses on solving for covariate shift.

Among the causes of covariate shift are included problems related to *domain generalization*, i.e., when the train and test domains are disjoint but still share the same labels. As well as by problems related to *sub-population shift*, i.e., when the train and test domains are the same, but their proportions are different. Ideally, a DML model learns an embedding space that generalizes well enough within the train data distribution, known as In-Distribution (ID), avoiding vulnerabilities to data difficulty (unspecific covariate shift). Some methods make use of diverse concepts such as *knowledge distillation* [10, 18, 29] and consider a probable distributional shift in the data [9, 23] to improve generalization.

For example, [23] show through exhaustive experiments how data splitting into train and test involves different distribution shifts that modify the difficulty of the data and proposes few-shot DML to improve generalization upon unknown shifts in test.

This paper presents an approach that brings together adaptations of both Few-Shot Learning (FSL) and knowledge distillation that avoids the restrictions of classification layers. The proposal consists of two independent models: The first one is a multi-branch master model, called GEMINI, that exploits local and global information to generate a reduced hypothesis space based on prior knowledge from the source labeled dataset in the form of triplet samples. This model represents a fast-convergence compact model, with negligible train time cost and no change in test time cost, avoiding problems of large teacher networks and time consumption [10]. Its architecture is an analogy to strategies based on parameter sharing in FSL [39]. The second model, a deep learning model, is a student model that learns an adequate embedding function guided by the mentioned reduced hypothesis space by using a similarity function $s(\cdot,\cdot)$, following the teacher-student concept from knowledge distillation [10, 14]. Exploiting the tractability of the features space low-dimensionality to regularize the student decision bounds (Figure 1).

To test our proposal, we have followed some of the guidelines proposed by [25] to design an experimental protocol that allows us to compare different models under equal conditions as much as possible. Our proposal demostrate the following key contributions:

- The experiments empirically show that the quality of the GEMINI embedding space is reliable given the consistency of the results with a random sample selection, reducing the dependence on the not-so-reliable sample mining strategies.
- Following [23], we further show that FSL adaptations in DML improve generalization with learning circumstances hindered by distributional shifts.
- The evaluation metrics suggest that our proposal can obtain better-delimited embedding spaces with compact clusters than with the compared models, giving less uncertainty between ID and OOD data.
- The performance in low-dimensional (two-dimensional) embeddings positions the proposed model as useful for data visualization.

2. Related works

Deep Metric Learning. Usually, there are two fundamental DML models considered in the literature: The former is the *siamese network* (contrastive loss) [7, 12], a method based on pairwise samples, which encourages small positive pairwise distances and negative pairwise distances above a certain margin. The latter is the *triplet network* (triplet loss) [15] that considers three types of samples: positive, negative, and an anchor. In this case, the distance between the anchor-negative samples should be greater than the distance between the anchor-positive samples by at least a margin. The triplet network improves over the siamese network by using intra-class and inter-class relationships [25], allowing a better fit to the variance differences between classes and making the model less restrictive. In this way, using these same fundamental architectures, there are a variety of proposed loss functions, e.g., Angular loss [37], Mixed loss [4], Margin Loss [40], Multi-similarity loss [38], N-Pairs loss [33], among others [8, 17].

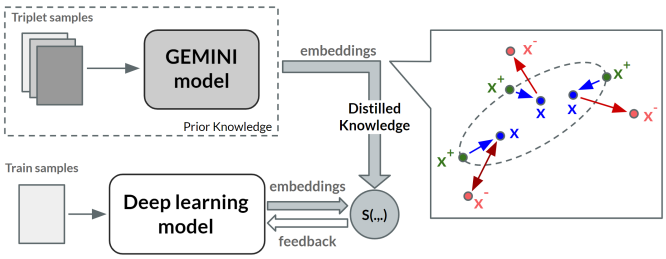

Figure 1. Block diagram of our proposal. The reduced hypothesis space generated a priori by the GEMINI model is used to guide the training of the deep learning model.

However, pairwise or triplet samples can involve high time complexity in the training process. Hence, *sample mining* strategies to identify the most informative examples capable of increasing performance, as well as the training speed, can be used. Nevertheless, for instance, in the case of hard-negative mining, the siamese network generally converges faster. However, the case of the triplet network often leads to problems where all samples have the same embedding and produce noisy gradients [25, 40]. Meanwhile, semi-hard negative mining is recommended for triplet network over hard-negative mining to avoid the risk of *overfitting*. However, in some cases, it might converge quickly at first, but as the number of negative examples within the margin runs out, it drastically slows down its progress [40]. These indicate that choosing an appropriate sampling strategy could be a difficult decision, as it seems to be sensitive to the properties of the underlying dataset or when architecture changes occur.

Meanwhile, new approaches have extended DML methods to other topics and have shown improvements in leveraging data relationships. Such is the case of Zero-Shot Learning (ZSL) and Few-Shot Learning (FSL), paradigms that address applications hindered by a limited number of samples and where it uses prior knowledge to generalize quickly [23, 39].

The ZSL approach, where test and train classes are distinct, intends to learn representation spaces that capture and transfer visual similarity to unseen classes [3, 23, 27, 29]. Faces the challenge of constructing a priori unknown test distribution with an unspecified distributional shift from the train distribution. However, arbitrarily large distributional shifts may cause the captured knowledge from the training data to be less significant to the test data [23], i.e., ill-posed learned representations. In the case of FSL, where at least a few samples of the test distribution are available during training, improve the quality of embeddings or prototype representations [28,32,34,36]. Specifically, in [23] has been proven that adaptations of FSL can improve the generalization capability of DML since even the minor additional domain knowledge provided helps to adjust the learned representation space to achieve better OOD generalization (commonly referring to covariate shift).

Also related to our line of research, there are some approaches following FSL methods that learn by constraining the hypothesis space through prior knowledge using a *parameter sharing* strategy [39]. Such is the case of fine-grained image classification [44], domain adaptation [24], and cross-domain translation [2], where some layers are for capturing global information and others for local information.

Knowledge Distillation. Originally introduced as model compression and acceleration techniques, *knowledge distillation* refers to an approach with teacher-student architecture [11, 14]. The main idea is to learn a student model (small network) from a teacher model (large network). As a result, it's obtained a small network trained to learn to replicate the behavior of the original network. This idea evolved beyond the goal of model compression and was subsequently used to improve performance in computer vision and language modeling tasks [10]. Specifically, in the context of DML, the purpose is to take advantage of the knowledge captured by the master model to learn better embedding functions [5, 35, 36].

In addition, there are variants within this approach with more specific configurations, such as *Self-distillation* [29, 43], where the same network acts as both teacher and student models. As well as *self-supervised learning* [22, 28], oriented to help initialize a network where there is a lack of labeled data. In self-supervised, an initial (pretext) task is learned by a master model using a general content dataset (e.g., ImageNet), which provides the "supervision" to the student model to perform the actual (downstream) task. A common approach in self-supervised learning uses *contrastive learning* to act over the similarity between the embeddings of pair samples [16].

3. Method

In this section, we propose a novel architecture for the task of deep metric learning, called *Guided deep metric learning*. This architecture has key concepts based on both FSL and Knowledge distillation. This architecture consists of two independent models (embedding functions). The first is called *GEMINI*, a fast-convergence compact model, which is in charge of generating a reduced hypothesis space based on prior knowledge. The second one is a deep learning architecture (specifically, a ResNet-18) used to learn the embedding function guided by the hypothesis space generated by the GEMINI model.

3.1. GEMINI Model

Analogous to parameter sharing strategies found on FSL, the GEMINI model consists of two main parts (Figure 2). The first component $f_k(\cdot)$ exploits the local information of each class, one stream layer [1,6] per class. Then, the global fully connected component $g(\cdot)$ tries to exploit the global information of local representations by sharing some parameters between the different classes, producing a unique data representation and, thus, avoiding the strong restrictions of a classification layer, e.g., cross-entropy.

The model uses a triplet dataset X generated from a training dataset D_{train} with c classes. We will use the notation $\boldsymbol{x}_{(k)}$ and $\boldsymbol{x}^{+}_{(k)}$ for the anchor and positive samples of class k and $\boldsymbol{x}^{-}_{(l)}$ for the negative sample of class l, where $k \neq l$. Each sample is input to the network through its respective stream, depending on its class. The outputs are intermediate representations denoted by $f_i(\boldsymbol{x}^{*}_{(i)})$ for each class $i = 1 \dots c$, regardless of the sample type (*).

$$f_k\left(\boldsymbol{x}^{*}_{(k)}\right) = h^{L}_{(k)} \circ h^{L-1}_{(k)} \circ \cdots \circ h^{1}_{(k)}\left(\boldsymbol{x}^{*}_{(k)}\right) \qquad (1)$$

The equation (Eq. 1) represents the layer composition performed at each stream. Here, the triplets $(\boldsymbol{x}_{(k)}, \boldsymbol{x}^{+}_{(k)}, \boldsymbol{x}^{-}_{(l)})$ are selected in two steps: First, a permutation of length 2 (without replacement) is randomly selected from the classes, thus selecting two respective streams. Second, two samples of the first class $(\boldsymbol{x}_{(k)}, \boldsymbol{x}^{+}_{(k)})$ and one of the second class $(\boldsymbol{x}^{-}_{(l)})$ are extracted from the selected permutation.

Each mini-batch contain a certain number of triplet samples, given by a hyper-parameter, of one permutation of the classes, this is denoted by X_b where $b = 1, \dots, \frac{|c|!}{(|c|-2)!}$. This means that, given that there are only two different classes in each mini-batch, only two stream are used (activated) in each mini-batch. Thus, the activation of the streams are control by the given indicator function $\mathbf{1}_{ij}$, where i and j are the stream index and the sample class

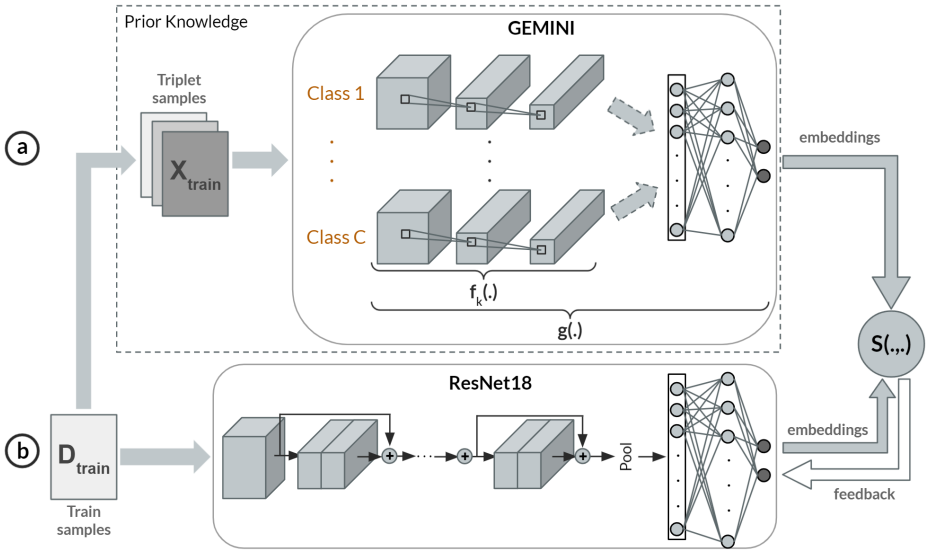

Figure 2. Complete Architecture. First, **a)** The GEMINI model is trained using the triplet data samples X_{train} generated from the original train dataset D_{train}. Then, we acquire the embeddings of the respective train dataset D_{train} from the reduced hypothesis space generated. Second, **b)** The GEMINI's low-dimensional embeddings guide the training of the ResNet through a similarity measure $S(\cdot,\cdot)$ in this low-dimensional space, comparing the distance between the respective embeddings. The discrepancy provides feedback for the ResNet parameters.

respectively,

$$\mathbf{1}_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{Otherwise} \end{cases} \tag{2}$$

In order that, during the training process, the following condition is enforced,

$$\mathbf{1}_{ij} \cdot ||f_i(\boldsymbol{x}_{(j)}) - f_i(\boldsymbol{x}^+_{(j)})|| < \epsilon, \quad \forall \left(\boldsymbol{x}_{(j)}, \boldsymbol{x}^+_{(j)}\right) \in X_b \tag{3}$$

$$\mathbf{1}_{il} \cdot f_i(\boldsymbol{x}^-_{(l)}), \quad \forall \boldsymbol{x}^-_{(l)} \in X_b \tag{4}$$

where ϵ is a small positive number. This is the expected behaviour in each update of the model parameters after each mini-batch, in accordance with the proposed cost function (Eq. 5). However, once they passed through the local component $f_k(\cdot)$, they all pass through the global component $g(\cdot)$ where the samples in the mini-batch share the layer parameters. The embedded representation of the network is denoted by $g(\boldsymbol{x}^*_{(i)}) = g(f_i(\boldsymbol{x}^*_{(i)}))$. Thus, once the intermediate and final representations are obtained for each sample in the triplet sample, both components of the network are coupled through the proposed cost function (Eq. 5).

$$\begin{aligned} L\left(f, g, d, \beta, M\right) = \frac{1}{|X_b|} \sum_{\left(\boldsymbol{x}_{(k)}, \boldsymbol{x}^+_{(k)}, \boldsymbol{x}^-_{(l)}\right) \in X_b} \\ \beta \cdot d\left(f_k\left(\boldsymbol{x}_{(k)}\right), f_k\left(\boldsymbol{x}^+_{(k)}\right)\right) + \ldots \\ (1-\beta)\left[M - d\left(g\left(\boldsymbol{x}_{(k)}\right), g\left(\boldsymbol{x}^-_{(l)}\right)\right)\right]_+ \end{aligned} \tag{5}$$

where the term $\beta \in [0, 1]$ is a weighting parameter, the term $[x]_+ = max(0, x)$ is the hinge loss, and M is a margin. The first term of the cost function evaluates the closeness of the similar samples, using the local information of the class to emphasize the closeness of the intermediate representations.

Note that using this first term alone may certainly lead to a trivial solution, i.e., a mapping of all points to a single point in the embedded space. To prevent this type of solution, a second term has been added which sets away points belonging to different classes. Thus, as the training progress, only the pairs that satisfy $d(g(\boldsymbol{x}_{(k)}), g(\boldsymbol{x}^-_{(l)})) < M$ will produce a cost value. In this way, the first term minimizes intra-class distances, and the second term prevents trivial solutions by maximizing the distances between classes, inter-class distances.

The term M can be defined as $M = d(g(\boldsymbol{x}_{(k)}), g(\boldsymbol{x}^+_{(k)})) + m$, where m is a margin value.

This makes the second term of the cost function resembles the triplet loss function, but with an added term that keeps the similar samples together using the local information. To balance both effects in the loss function there is a β term which, experimentally, its value is usually small (≈ 0.005), but significant to improve the resulting reduced hypothesis space.

Algorithm 1 GEMINI Model

Input: original training set $D_{train} = \{(x_i, y_i)\}_{i=1}^{N_{train}}$
batch size N_{batch}
number of classes C
output dimension size $OutputSize$
number of epochs N_{epochs}
weighting parameter β
margin value parameter m
Output: reduced hypothesis space embeddings $\hat{Z}$

$X_{train} \leftarrow MakeTripletSamples\,(D_{train}, N_{batch})$
$f \leftarrow [f_1, \cdots, f_C]$
$g \leftarrow FullyConnected(OutputSize)$
for $i \leftarrow 1$ to N_{epochs} **do**
  for each $\{(x_{(k)}, x^+_{(k)}, x^-_{(l)}), (k, l)\}_{i=1}^{N_{batch}} \in X_{train}$ **do**
    $a \leftarrow f_k(x_{(k)})$
    $p \leftarrow f_k(x^+_{(k)})$
    $anchor \leftarrow g(a)$
    $positive \leftarrow g(p)$
    $negative \leftarrow g(f_l(x^-_{(l)}))$
    $loss \leftarrow \beta \cdot ||a - p|| + (1 - \beta)(||anchor - positive|| + m - ||anchor - negative||)$
    GEMINI.$gradient.step(loss)$
$\hat{Z} \leftarrow$ GEMINI(D_{train})

The model works by decreasing the distances between anchor and positive samples and, simultaneously, increasing the distance between anchor and negative samples (Figure 1). Rather than engaging with sample mining techniques, which, as mentioned before, have been shown not to work well for all possible scenarios, restricting the generality. An alternative approach arises when the cost function (Eq. 5) is more generalized with a small change in the second term (Eq. 6).

$$\left[M - d\left(g\left(\boldsymbol{x}_{(k)}\right), \boldsymbol{x}^-_{(l)}\right)\right]_+ \tag{6}$$

In this rewriting of the second term, the transformation $g(\boldsymbol{x}^-_{(l)})$ is changed by $\boldsymbol{x}^-_{(l)}$. This implies that the input $\boldsymbol{x}^-_{(l)}$, in this generalization, can be interpreted as a simple vector that imposes a limitation or restriction on the system. These constraints can be samples from another class, proposed by the user or even coming from another model. Thus, it can be considered that the model can shape the distribution of each class according to these specific points.

3.2. Complete Architecture

The complete architecture follows the offline knowledge distillation guidelines. In our proposal, the student network is a deep learning model of choice; in this case, we opted for the PyTorch ResNet-18 implementation. We say that this model is "guided" to replicate the behavior of the master network (GEMINI model).

This approach has the advantage that GEMINI has already searched the space and arrived at a hypothesis space constrained by prior knowledge. With such a reduced hypothesis space, the ResNet model is expected to need fewer samples to converge to a suitable hypothesis, closer to the optimum, and have a lower risk of overfitting. Both models are coupled by a similarity function, $s(\cdot, \cdot)$, which measures the deviation of the ResNet hypothesis from the reduced hypothesis space (Figure 2).

The complete architecture is described in two steps: First, acquire the low-dimensionality embeddings of the reduced hypothesis space $\hat{z}_i \in Z \subseteq \mathbb{R}^m$ corresponding to each sample in the original training dataset $x_i \in D_{train} \subseteq \mathbb{R}^d$ where $m \ll d$, from a previously trained GEMINI model. Second, the ResNet model takes as input the original training dataset D_{train}, directly embedding each sample $x_i \in D_{train}$ into a lower-dimensional space $z_i \in Z$ without a classification layer. The resulting embeddings z_i are compared with the ones obtained from the GEMINI model $\hat{z}_i$ through a similarity measure $s(z_i, \hat{z}_i)$ in the low-dimensionality space $Z \subseteq \mathbb{R}^m$. Our chosen similarity measure is the l_2-distance. The discrepancy is then used as feedback for updating the ResNet parameters.

Algorithm 2 Knowledge Distillation General Model

Input: original training set $D_{train} = \{(x_i, y_i)\}_{i=1}^{N_{train}}$
GEMINI's reduced hypothesis space embeddings $\hat{Z}$
similarity function S$(\cdot,\cdot)$
number of epochs N_{epochs}
batch size N_{batch}
Output: low-dimensionality embeddings Z

$X_{train} \leftarrow \left(D_{train}, \hat{Z}\right)$
$Z \leftarrow \emptyset$
for $i \leftarrow 1$ to N_{epochs} **do**
  for each $\{(x, \hat{z})\}_{i=1}^{N_{batch}} \in X_{train}$ **do**
    $z \leftarrow$ ResNet.$forward(x)$
    $loss \leftarrow s(z, \hat{z})$
    ResNet.$gradient.step(loss)$
$Z \leftarrow ResNet(X_{train})$

4. Results

This section evaluates, analyzes, and compares the performance of our proposed method with other approaches. The evaluation of the performance follows a few guidelines suggested by [25] to perform a more fair and realistic comparison. To claim that an algorithm outperforms other methods, the conditions to which the models compare must remain as similar as possible. The rationale for this approach is to ensure that it is the algorithm (not any external design choices) the one improving performance. Therefore, the designed experimental setup keeps consistency in parameter choices and avoids some commonly used techniques that could interfere with the results. For example, the use of different pre-trained network architectures (leading to differences in initial accuracies), the choice of the data augmentation strategy, and choice of the optimizer (e.g., SGD, Adam, RMSprop), among other design choices that remain inconsistent (variable) throughout the literature.

4.1. Experimental setup

All of the experiments have been implemented using Python 3.8 and Pytorch 1.6 on an NVIDIA RTX 3060 super 12 GB. In addition, no data augmentation nor pre-processing (besides global normalization to zero mean and unit variance) were applied. For all the networks, the dimensionality of the output embedding space is two, gradient clipping with a factor of 0.1 and a weight decay with a decay factor of 0.0001. Further, the backbone of all networks (including our proposed method) is a ResNet-18, where the batches were constructed randomly (assuming a uniform distribution) without any sample miner. We used the repository by [26] for some losses, reducers, and DML metrics.

The networks to be compared have been trained with a learning range of 0.1-0.5, using an SGD optimizer and a batch size range of 32-512 samples. In the specific case of our proposed model, the GEMINI model has been trained at a learning rate of 0.001, margin value of 3.0, SGD optimizer with a batch size range of 32-64 triplet samples, and takes between 10 to 15 epochs to converge to a satisfactory solution. The complementary deep learning model (ResNet-18) has been trained at a learning rate of 0.1, SGD optimizer, and a batch size range of 32-128 samples.[1]

For base performance comparison purposes, we considered the baseline models: Siamese and Triplet network, described in section 2. Additionally, recent models such as Multi-Similarity Loss [38] and Margin Loss [40] were considered. The metrics to evaluate the performance of all architectures were the following: Recall@K, F1-score, and Normalized Mutual Information (NMI). In addition, we have used the metrics proposed by [25]: R-Precision (RP) and Mean Average Precision at R (MAP@R). All these metrics were obtained using the kNN classifier ($k = 1$) of scikit-learn in the test embedding space (specifically, ResNet-18 output, the student model in our proposal).

The experiments were on well-known datasets: MNIST [21], Fashion-MNIST [41], and CIFAR10 [20]. We chose these because they are easily comparable benchmark datasets. Tables 1-3 show the mean performance across training runs with a 95% confidence interval. The bold type represents the best result.

Model	Recall@1	F-score	NMI	RP	MAP@1
Siamese	98.27 ± 0.10	98.27 ± 0.10	$\mathbf{95.16 \pm 1.83}$	97.88 ± 0.16	97.73 ± 0.17
Triplet	97.71 ± 0.10	97.71 ± 0.10	90.98 ± 1.40	96.6 ± 0.10	96.02 ± 0.12
Margin	98.81 ± 0.25	98.81 ± 0.25	91.56 ± 1.90	95.08 ± 2.88	96.17 ± 1.87
MultiSim	98.58 ± 0.19	98.58 ± 0.19	90.54 ± 1.96	91.28 ± 1.19	90.12 ± 1.35
Ours	$\mathbf{99.00 \pm 0.18}$	$\mathbf{99.00 \pm 0.18}$	92.08 ± 1.80	$\mathbf{98.56 \pm 0.80}$	$\mathbf{98.38 \pm 1.00}$

Table 1. Performance on MNIST

Model	Recall@1	F-score	NMI	RP	MAP@1
Siamese	84.03 ± 0.58	84.03 ± 5.12	75.42 ± 0.54	80.55 ± 0.98	75.00 ± 1.29
Triplet	83.86 ± 0.20	83.90 ± 0.21	75.64 ± 1.05	78.69 ± 0.23	72.07 ± 0.28
Margin	87.75 ± 0.77	87.76 ± 0.77	79.28 ± 1.26	83.40 ± 2.66	79.91 ± 3.59
MultiSim	90.33 ± 0.29	90.35 ± 0.28	77.30 ± 1.24	80.58 ± 1.56	77.27 ± 1.93
Ours	$\mathbf{91.93 \pm 0.18}$	$\mathbf{91.91 \pm 0.17}$	$\mathbf{79.42 \pm 1.66}$	$\mathbf{89.06 \pm 0.64}$	$\mathbf{87.93 \pm 0.66}$

Table 2. Performance on Fashion-MNIST

Model	Recall@1	F-score	NMI	RP	MAP@1
Siamese	25.32 ± 1.53	25.96 ± 1.02	25.17 ± 0.99	24.22 ± 1.07	11.76 ± 4.73
Triplet	41.93 ± 0.93	42.12 ± 0.95	39.43 ± 0.71	36.65 ± 0.88	20.03 ± 1.02
Margin	55.71 ± 5.81	55.86 ± 5.84	45.15 ± 2.65	40.90 ± 3.00	24.73 ± 3.40
MultiSim	56.65 ± 6.15	57.04 ± 6.22	47.51 ± 4.98	47.72 ± 5.96	36.04 ± 7.31
Ours	$\mathbf{80.11 \pm 0.72}$	$\mathbf{80.09 \pm 0.72}$	$\mathbf{61.36 \pm 2.31}$	$\mathbf{75.98 \pm 0.98}$	$\mathbf{73.93 \pm 1.03}$

Table 3. Performance on CIFAR10

Given an average performance of 10 experiments for the proposed method and the other methods. As shown, in general, our proposal is superior to the others. MNIST (Table 1) shows on average a marginal improvement. Compared to Margin loss our proposal improves by 0.19% in Recall@1, and up to 1.3% with respect to the Triplet network. In Fashion-MNIST (Table 2), compared to Multi-similarity loss our proposal improves by 1.7% and up to 9.6% in Recall@1 compared to the Triplet network. Finally, in CIFAR10 (Table 3) there is a clear significant improvement. Compared to Multi-similarity loss, there is an improvement of about 40% in Recall@1 and about 3 times better compared to the Siamese network.

In addition to better performance, our proposal consistently demonstrates stable performances (narrow confidence intervals) during the different datasets. Meanwhile, the other models showed more unstable performances, especially in the most difficult dataset (Table 3). Thus, our proposal can be considered, in general, less sensitive to datasets with certain difficulty (distributional shift).

[1] Our source code is available at https://github.com/G-DML.

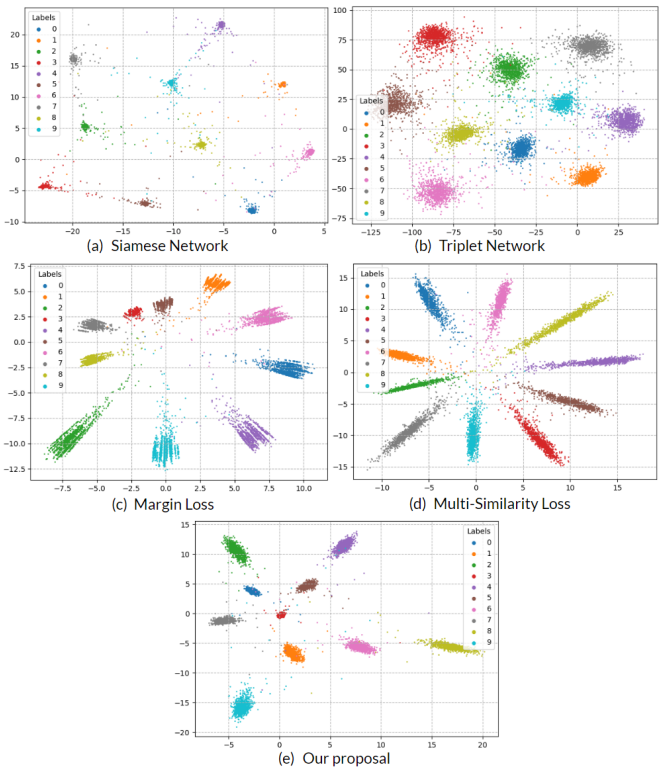

Figure 3. Embedding spaces using the MNIST dataset.

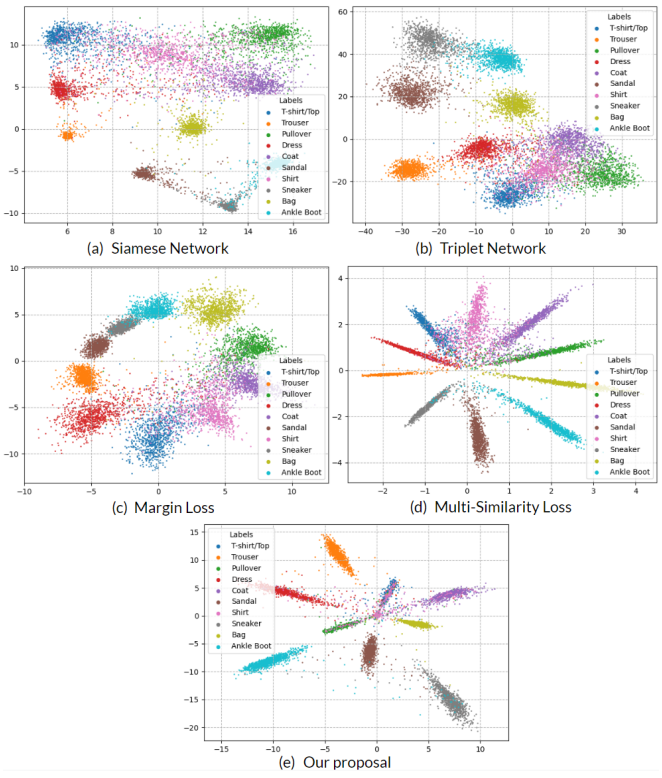

Figure 4. Embedding spaces using the Fashion-MNIST dataset.

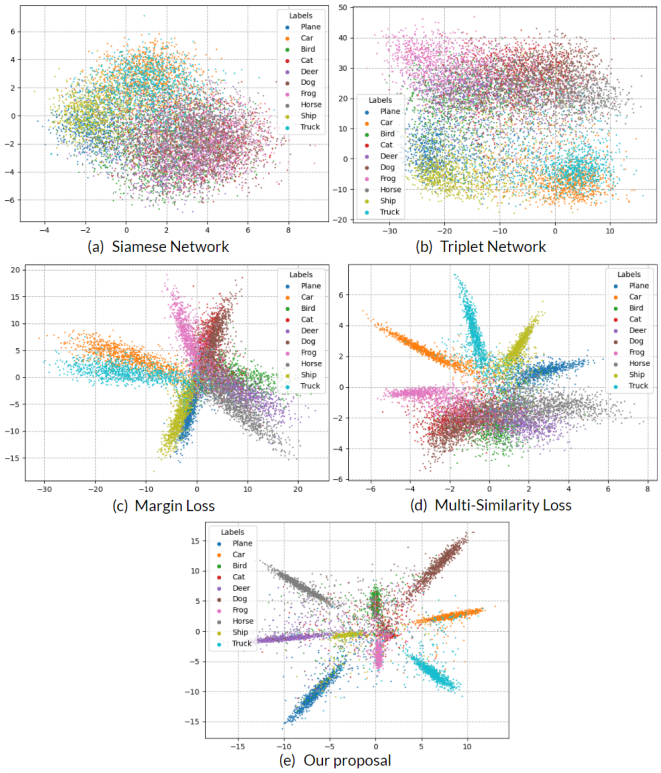

Figure 5. Embedding spaces using the CIFAR10 dataset.

The choice of the two-dimensional embedding size was motivated to push the proposed method to a more restrictive condition, such as producing low-dimensional outputs, and consequently, to its potential for tasks such as data visualization or more data insight.

In order to provide more information, figures 3-5 show the test set in the embedding spaces learned by the different methods and data sets. We can generally observe that our proposal achieves better compactness and separation of the different classes on the different datasets. It is appreciated particularly in the most difficult dataset (Figure 5).

5. Conclusions

In this paper, we analyze Deep Metric Learning (DML) models for embedding learning, the relationships and implementations that connect them to Few-Shot Learning (FSL), and how some models found in FSL have an architecture analogous to knowledge distillation with a change in their approach.

We proposed a DML model that integrates FSL and knowledge distillation concepts to develop an architecture that uses local and global information for better manifold generalization and data representation capabilities, providing better performance and stability compared to the compared methods. It also demonstrates that FSL adaptations boost generalization performance in DML models but also

meaningful embeddings can be learned without a strict sample selection phase.

Our approach and the models to which it was compared have been implemented in a careful experimental setup. Seeking the conditions were as similar as possible to avoid unfair comparisons, and we have used metrics that provide a completer picture of the generated embedding space. Though our results were based only on simple datasets, we will perform additional experiments on more closely related datasets with DML or FSL methods and out-of-distribution generalization metrics to further verify the performance.

References

[1] Javad Abbasi Aghamaleki and Vahid Ashkani Chenarlogh. Multi-stream cnn for facial expression recognition in limited training data. *Multimedia Tools and Applications*, 78(16):22861–22882, 2019. 3

[2] Sagie Benaim and Lior Wolf. One-shot unsupervised cross domain translation. *advances in neural information processing systems*, 31, 2018. 3

[3] Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4613–4623, 2020. 3

[4] Long Chen and Yuhang He. Dress fashionably: Learn fashion collocation with deep mixed-category metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2

[5] Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 3

[6] Vahid Ashkani Chenarlogh, Farbod Razzazi, and Najmeh Mohammadyahya. A multi-view human action recognition system in limited data case using multi-stream cnn. In *2019 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pages 1–11. IEEE, 2019. 3

[7] 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, pages 539–546. IEEE, 2005. 2

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 1, 2

[9] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. *arXiv preprint arXiv:2202.01197*, 2022. 1

[10] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018. 1, 2, 3

[11] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021. 3

[12] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 2

[13] Ben Harwood, Vijay Kumar BG, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2821–2829, 2017. 1

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 3

[15] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015. 2

[16] Wenlong Ji, Zhun Deng, Ryumei Nakada, James Zou, and Linjun Zhang. The power of contrast for feature learning: A theoretical analysis. *arXiv preprint arXiv:2110.02473*, 2021. 1, 3

[17] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019. 1, 2

[18] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Embedding transfer with label relaxation for improved metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2021. 1

[19] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021. 1

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 6

[22] Timo Milbich, Karsten Roth, Homanga Bharadhwaj, Samarth Sinha, Yoshua Bengio, Björn Ommer, and Joseph Paul Cohen. Diva: Diverse visual feature aggregation for deep metric learning. In *European Conference on Computer Vision*, pages 590–607. Springer, 2020. 1, 3

[23] Timo Milbich, Karsten Roth, Samarth Sinha, Ludwig Schmidt, Marzyeh Ghassemi, and Björn Ommer. Characterizing generalization under out-of-distribution shifts in deep metric learning. *arXiv preprint arXiv:2107.09562*, 2021. 1, 2, 3

[24] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. *Advances in neural information processing systems*, 30, 2017. 3

[25] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer, 2020. 1, 2, 6

[26] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. Pytorch metric learning, 2020. 6

[27] 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*, pages 4004–4012, 2016. 3

[28] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. Self-supervised knowledge distillation for few-shot learning. *arXiv preprint arXiv:2006.09785*, 2020. 1, 3

[29] Karsten Roth, Timo Milbich, Bjorn Ommer, Joseph Paul Cohen, and Marzyeh Ghassemi. Simultaneous similarity-based self-distillation for deep metric learning. In *International Conference on Machine Learning*, pages 9095–9106. PMLR, 2021. 1, 3

[30] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *International Conference on Machine Learning*, pages 8242–8252. PMLR, 2020. 1

[31] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021. 1

[32] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 3

[33] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pages 1857–1865, 2016. 2

[34] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS 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*, pages 1199–1208, 2018. 3

[35] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019. 1, 3

[36] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *European Conference on Computer Vision*, pages 266–282. Springer, 2020. 1, 3

[37] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2593–2601, 2017. 2

[38] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019. 2, 6

[39] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020. 2, 3

[40] 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*, pages 2840–2848, 2017. 1, 2, 6

[41] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6

[42] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021. 1

[43] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722, 2019. 3

[44] Yabin Zhang, Hui Tang, and Kui Jia. Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data. In *Proceedings of the european conference on computer vision (ECCV)*, pages 233–248, 2018. 3