Towards Learning Image Similarity from General Triplet Labels

Radu Dondera
Greenfield Vision
info@greenfieldvision.com

Abstract

Metric learning for images has so far focused overwhelmingly on a class-based definition of similarity: two images are similar if they belong to the same class and dissimilar otherwise. Impressive results were achieved on datasets for fine grained categorization, but performance is nearing saturation. Recent work in neuroscience and psychology produced datasets with other types of similarity labels, e.g. the outlier in a group of three, but traditional metric learning methods are ill-suited to such data because of low density of labels. To overcome this difficulty, we propose a novel approach in the teacher-student learning paradigm. Multiple teacher models learn to embed images based only on relations with other images, and then a student model learns to embed images based on both content and dense relations provided by the teachers. We show significant improvement over existing triplet based metric learning methods, both in result quality and in training efficiency. Additionally, through experiments on class based datasets, we show the generality of approaching metric learning via knowledge transfer. Code is available at `github.com/greenfieldvision/tdml`.

1. Introduction

Metric learning is the problem of learning a mapping from a perceptual space to a latent space that reflects semantic similarity. The problem has been studied extensively in computer vision, as many application domains stand to benefit from good visual embeddings: face verification [31], image retrieval [32] and few-shot learning [37] are just a few examples. The advent of deep learning made it possible for metric learning methods to achieve impressive results on three widely used datasets, CUB-200 [40], CARS-196 [19] and SOP [33]. Numerous publications have targeted the datasets and state of the art has been nearing 90% recall at 1 on all three. However, this also indicates saturation and raises the question of using qualitatively different data.

The three datasets, which we will refer to as standard, define similarity via classes: two images are similar if they belong to the same class and dissimilar if they belong to different classes. In this work, we study the problem of metric learning with similarity defined via triplet labels - groups of three images in which the first is more similar to the second than to the third. Datasets in the psychology, neuroscience and human-computer interaction literature contain relative similarity judgments that are not determined by classes and that can be transformed without loss of information into triplet labels. For the THINGS [8] dataset, the task for human annotators was to choose the image least similar to the others from a group of three; for the Imagenet-HSJ [27] and Yummly [41] datasets, the task was to select two images from a group of eight that are most similar to a query image. In the triplet odd-one-out task, the images belong to exactly three different classes, while in the 8-rank-2 task, the images can be from one to nine different classes. The new datasets capture similarity nuances beyond what can be expressed via classes, see figure 1.

Figure 1. Commonly used metric learning datasets such as CUB-200 and CARS-196 (top row) define similarity via class labels and miss some nuances expressed via triplet labels in the THINGS and Imagenet-HSJ datasets (bottom row). According to class labels, all three birds are different, despite the first two having the same color pattern. Again according to class labels, the first car is similar to the second but different from the third, despite the first and third having roughly the same viewing angle. Triplet labels allow stating that the vise and hydrant are more similar to each other than to the hyena, and the grey cats are more similar to each other than to the orange cat. Border colors indicate classes and the order of images in triplets is anchor, positive, negative.

Datasets with triplet labels pose two major challenges. First, the number of triplets grows cubically with the num-

ber of images, so the labels collected given a reasonable budget can only cover a very small fraction of the possible triplets (e.g. 1.46M labels are only 0.14% of total for THINGS [8]). Second, batches of B images have $O(B)$ labeled triplets, considerably lower than the $O(B^2)$ or $O(B^3)$ needed by the existing triplet based metric learning methods [32] [39] [31] [23] for the purpose of batch mining. The methods cannot select the most informative triplets in a batch and must use small and arbitrary sets of triplets, which leads to suboptimal results and inefficient training.

We propose a new knowledge transfer approach for metric learning on triplet labeled datasets. Multiple teacher models are trained without using image features, and then the knowledge of the ensemble is transferred to a student model that does use image features. A **teacher** model maps **image indexes** to the embedding space, so its batches have many more triplets than if its input domain were images (e.g. thousands vs a hundred). The **student** model maps **images** to the embedding space and in each of its batches of size B the teacher ensemble predicts all the $O(B^3)$ triplet labels. By decoupling the task of learning object relations from the task of learning the visual embedding, our approach makes the supervision 100% dense for the latter task, addressing the challenge of low label density in batches. We show that our approach outperforms existing triplet based metric learning methods on the THINGS dataset, on a variant of Imagenet-HSJ and on Yummly.

Further, we apply our knowledge transfer approach to class based datasets. Knowledge distillation losses used with one-hot class encodings match state-of-the-art metric learning losses, suggesting that our approach viably generalizes metric learning on standard datasets.

2. Related Work

2.1. Metric Learning

We categorize metric learning papers by their stance towards class information.

Class-agnostic methods. The methods that do not assume the existence of classes can be roughly split according to the type of loss they use: pairwise, triplet or a combination thereof. Pairwise losses [5] operate on pairs of images, pulling together the similar ones and pushing apart the dissimilar ones. Triplet losses [31] operate on anchor-positive-negative image triplets, pulling together the anchor and the positive and pushing apart the anchor and the negative. The N-pair loss [32] operates on image tuplets of the form $anchor\text{-}positive\text{-}negative_1\text{-}...\text{-}negative_{N-1}$, pulling the anchor and positive closer together than the anchor and closest negative. Randomly selected pairs and triplets are typically not informative (i.e. have 0 loss) and occasionally noisy (i.e. have wrong label), so a central concern in class-agnostic methods is batch sampling and mining [31] [23] [6] [38]. Many methods actually make use of class labels for this purpose, despite not explicitly incorporating them in loss expressions. For example, the authors of [23] choose five images per class, form all the pairs involving them and finally complete triplets with negatives from the other classes in the batch. Optimizing sampling without class information requires knowing all the triplet labels, which is infeasible beyond small datasets (100s of images).

Class-aware methods. These methods assume the existence of prototypes in the data (which roughly correspond to classes, but the mapping is not necessarily 1:1) and learn their positions in the latent space together with those of the images assigned to them. A seminal paper [24] essentially performs classification and then discards the prototype/class information, keeping only the embedding from images to the latent space. Further improvements have employed angular margins [2] and a sophisticated way to represent data-to-data relations, pushing prototypes apart based on the relative hardness of examples [12]. [43] computed soft data-to-prototype assignments with an optimal transport layer and thus captured more nuanced data-to-data relations. [3] mapped embeddings obtained with vision transformers to the hyperbolic space, while [15] used an unsupervised clustering loss term to better reflect the latent data hierarchy. The class-aware methods have consistently ranked among the best on CUB-200, CARS-196 and SOP, not surprising given that the three standard datasets are class oriented.

Class-skeptical methods. A number of papers identified shortcomings inherent to the use of class labels and highlighted the importance of investigating similarity across class boundaries. [4] built a hierarchy of classes to guide the search for hard examples for a triplet loss. [45] used hierachical labels to learn embeddings at progressively coarse semantic levels. [11] explored a setting in which the similarity of two images is continuous rather than binary and employed a triplet loss that preserves similarity ratios. [20] also recognized the need to represent degrees of similarity and focused on batch decorrelation within an active learning framework for triplets. Two works noticed a compression effect in which images from the same class cluster tightly in the embedding space and the representation loses detail [29] [21]. The solutions involved adapting the sampling procedure and the loss respectively, but it is not clear that they fully resolved the large intra-class variation of some classes in the three standard datasets. Finally, [30] accounted for class similarities by incorporating language knowledge.

2.2. Psychology and Neuroscience

The psychology and neuroscience literatures offer rich perspectives on visual similarity. A well known early paper defined similarity in terms of measures over sets of qualitative properties [34] and highlighted the role of context. Another early paper hypothesized object recognition through a

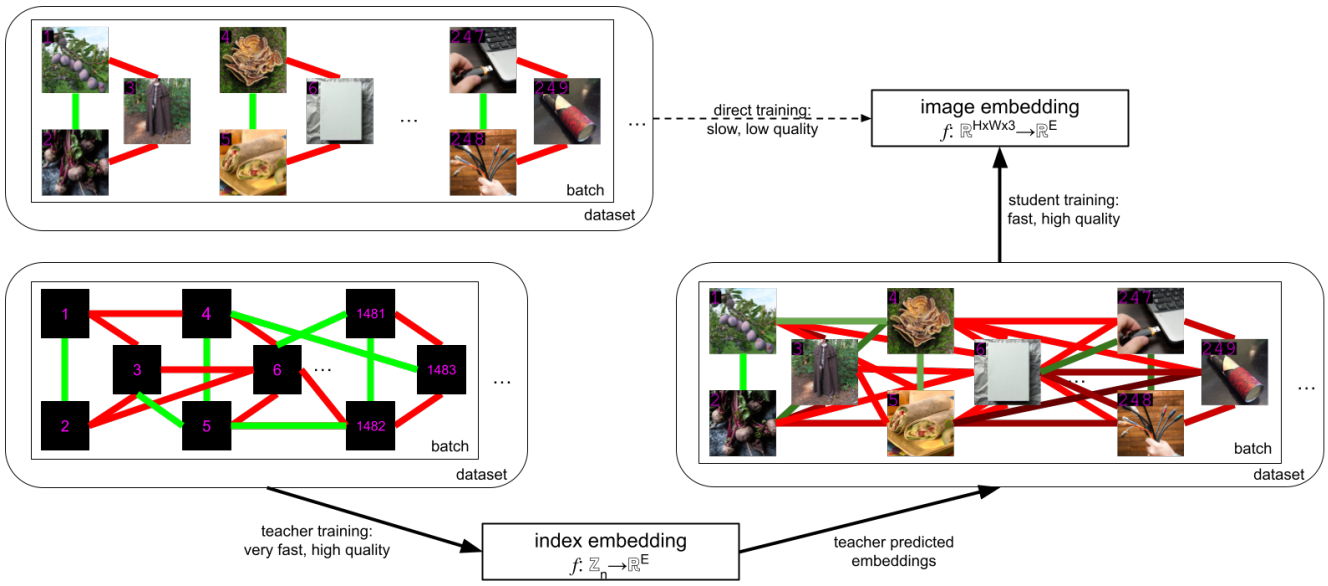


Figure 2. Direct training with existing triplet based methods (dashed line) vs training in our approach (continuous lines). Direct training sees very few informative triplets per batch because the dataset has low labeled triplet density, so it is slow and its results have low quality. Our approach first trains teachers whose input domain is integers instead of images, which enables using batches with very many labeled triplets. Then, a student is trained with batches with all triplets labeled (by the teachers), which is fast and has high quality results.

fixed set of components, which can also serve to define similarity [1]. Recent research has been data-driven: large sets of similarity judgments have been collected and have served to infer high level similarity properties using machine learning [8] and to evaluate state-of-the-art visual representation models [27]. Both datasets are essentially triplet based: in the first, the labeling task is to pick the odd image out from a group of three, while in the second, it is to pick the two most similar images to a query from a group of eight reference images (interestingly, the latter setup had been verified to be more cost effective from a metric learning point of view [41]). It is worth noting the absence of large-scale pairwise datasets; compared to triplet labels, pair labels lack context [8] and have higher levels of noise [10].

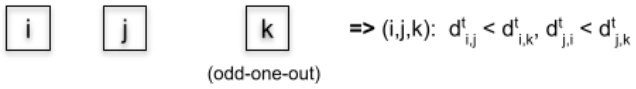
2.3. Knowledge Distillation

Our approach follows the teacher-student learning paradigm, as it trains teacher models and then transfers their knowledge to a student model. It is not pure knowledge distillation because the teachers have a different input domain than the student, but parallels can be drawn with knowledge distillation methods. An early work [28] distilled a wide and deep teacher into a thin and deep student by linearly predicting the teacher network's intermediate layers. This is easily adaptable to metric learning by moving the constraints to the embedding layer, but methods that transferred instance relations (e.g. pairwise distances) achieved better performance [25] [26]. A disadvantage of these methods is not accounting for sample importance, which is addressed in [13] and extended to unsupervised metric learning [14]. Among the first works to investigate transferring knowledge from multiple teachers, [42] proposed voting on the positive and negative elements given the anchor element in a triplet. The scheme is robust but it ignores the important detail of how close the positive and negative are to the anchor.

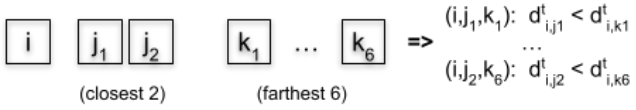
The paper that inspired our work transferred the knowledge in a metric learning model via pairwise similarities [13] for purposes of self-distillation and model compression. The key difference with respect to our approach is that our teacher models are trained on image relations as opposed to image data. While [13] do self-distillation for metric learning on class based datasets, we use a variant of distillation to enable effective and efficient metric learning on triplet labeled datasets. Also, while [13] focus on a specific loss, we investigate knowledge transfer more broadly and reveal its suitability to standard metric learning.

3. Proposed Approach

Our approach divides the process of learning an embedding for images from general triplet labels into two steps. In the first step, it learns an ensemble of teacher models that embed images based on their relations to other images and not their content. In the second step, it learns a student model that embeds images based on both their content and the dense relations predicted by the teacher models. By contrast, the existing triplet based methods learn an embedding for images directly from image triplets, see figure 2.



(a) 1 undirected triplet from each odd-one-out task



(b) 12 directed triplets from each 8-rank-2 task

Figure 3. The two triplet label types. Each loss used to train teacher models is defined in two corresponding versions.

Embedding task. We learn an embedding function f from either integers or images to a latent space, $f : \mathbb{Z}_n \rightarrow \mathbb{R}^E$ or $f : \mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}^E$ where E is the embedding size. We denote the value of the function for an input i with f_i, the dot product of two embedding vectors with $s_{i,j}$ and the Euclidean distance between them with $d_{i,j}$.

Triplet label types. Note that we differentiate between two types of triplet labels. The first type comes from a triplet odd-one-out task, which produces information of the form images *i* and *j* are closer to each other than to image *k* - this is called an undirected triplet. The second type comes from 8-rank-2 tasks, which produce information of the form image *j* is closer to image *i* than image *k* - this is called a directed triplet. See figure 3 for details.

3.1. Teacher Models

The first step in our approach is to train teacher models with labeled triplets of indexes (each image in a labeled triplet of images is replaced with its index). A teacher essentially takes in an integer representing the image index and produces an embedding vector. A single layer network suffices, as it can specify embedding vectors independently for all the images. We train with the STE loss [36]:

$$L(i,j,k) = -\log \frac{e^{\frac{s_{i,j}}{\tau}}}{e^{\frac{s_{i,j}}{\tau}} + e^{\frac{s_{i,k}}{\tau}} + e^{\frac{s_{j,k}}{\tau}}} \qquad (1)$$

where τ is a temperature parameter. This is for undirected triplets: the fraction under the logarithm expresses the probability of the pair (i, j) being the closest of the three pairs in the (i, j, k) triplet. For directed triplets, the loss is

$$L(i,j,k) = -\log \frac{e^{\frac{s_{i,j}}{\tau}}}{e^{\frac{s_{i,j}}{\tau}} + e^{\frac{s_{i,k}}{\tau}}} \qquad (2)$$

as the fraction under the logarithm expresses the probability that i is closer to j than to k.

It is also possible to employ margin based losses for teachers. For undirected triplets,

$$L(i,j,k) = \frac{[m + d_{i,j} - d_{i,k}]_+ + [m + d_{j,i} - d_{j,k}]_+}{2} \qquad (3)$$

while for directed triplets,

$$L(i,j,k) = [m + d_{i,j} - d_{i,k}]_+ \qquad (4)$$

where $[.]_+$ denotes the positive part function and m is a margin parameter. However, these losses produce slightly worse models on two of the three datasets considered in this paper, see section 4.3.

Embedding vectors are L_2 normalized. To keep the teacher models as close as possible to psychological theories of human similarity [8], we constrain the embedding coordinates to be positive and use L_1 regularization to encourage as many weights to zero as possible. Unlike [8], we do not postprocess the embeddings after training.

Since [27] argue that the process of learning an embedding from relative similarity labels has high variance, we used a small ensemble of teacher models trained on the same data. We note that in practice the benefit of using multiple teachers is small, see section 4.3.

3.2. Student Model

The second step in our approach is to train a student model with continuous supervision from the teacher ensemble, e.g. the degree of similarity between two images. The student is an image classification backbone [7] with the softmax layer replaced by a simple linear projection layer for the embedding. Embedding vectors are L_2 normalized.

To train the student, our approach can utilize knowledge distillation losses like [25] or relaxed versions of metric learning losses like [13]. Both the relational knowledge distillation loss and the relaxed contrastive loss have pairwise terms, which are not necessarily optimal for transferring triplet based knowledge. Therefore, we investigated relaxations of other popular metric learning losses and also a new knowledge distillation loss tailored to our problem.

In this section we use the term triplet loss to mean any loss defined on three data points, as opposed to the specific expression in equation 4, which we call triplet margin loss. In the definitions that follow, we use $d^t_{i,j}$ and $s^t_{i,j}$ to denote the distance and dot product predicted by teacher t between images i and j, and $\overline{d^t_{i,j}}$ and $\overline{s^t_{i,j}}$ the average of these quantities over the teacher ensemble.

Relaxed Triplet Margin (RTM). A well known loss in the discrete label case is the triplet margin [39]. We define its relaxed version by

$$\begin{aligned} y_{rtm}(i,j,k) &= \sigma((\overline{d^t_{i,k}} - \overline{d^t_{i,j}})/\tau) \\ l_{rtm}(i,j,k) &= [m + d_{i,j} - d_{i,k}]_+ \\ L_{rtm}(i,j,k) &= y_{rtm}(i,j,k)\, l_{rtm}(i,j,k) \end{aligned} \qquad (5)$$

where σ is the logistic sigmoid function and τ and m are temperature and margin parameters respectively. The larger the teacher distance between i and k compared to that between i and j, the harder the student has to push k apart and i and j together, if k is within distance $m + d_{i,j}$ of i.

Relaxed Facenet (RF). To address the issue that hard cases and label noise can corrupt the gradient, [31] allow only semihard triplets (triplets where the negative is at least as far away from the anchor as the positive) to contribute to the loss. We define a relaxed version of the Facenet loss as

$$\begin{aligned} y_{rf}(i,j,k) &= \sigma((\overline{d^t_{i,k}} - \overline{d^t_{i,j}})/\tau) \\ l_{rf}(i,j,k) &= \delta(d_{i,k} \geq d_{i,j})[m + d_{i,j} - d_{i,k}]_+ \\ L_{rf}(i,j,k) &= y_{rf}(i,j,k)\, l_{rf}(i,j,k) \end{aligned} \tag{6}$$

where δ and σ are the indicator and logistic sigmoid functions, and τ and m are temperature and margin parameters. In the original implementation of [31], only the positive pairs with the largest semihard loss contribute. We modify the loss expression accordingly:

$$\begin{aligned} y'_{rf}(i,j) &= e^{-\frac{\overline{d^t_{i,j}}}{\tau}} \\ l'_{rf}(i,j) &= \max_k y_{rf}(i,j,k)\, l_{rf}(i,j,k) \\ L'_{rf}(i,j) &= y'_{rf}(i,j)\, l'_{rf}(i,j) \end{aligned} \tag{7}$$

The closer the teachers place i and j, the harder the student has to push k apart and i and j together, for the k whose distance to i comes closest to the threshold $m + d_{i,j}$ without exceeding it.

Relaxed InfoNCE (RI). While the InfoNCE loss [35] is not a triplet loss, a relaxed version of it can easily be defined and a comparison against it is meaningful. We define the relaxed InfoNCE loss as

$$L_{ri}(i,j) = -\log \frac{\frac{1+\overline{s^t_{i,j}}}{2} e^{\frac{s_{i,j}}{\tau}}}{\frac{1+\overline{s^t_{i,j}}}{2} e^{\frac{s_{i,j}}{\tau}} + \sum_k \frac{1-\overline{s^t_{i,k}}}{2} e^{\frac{s_{i,k}}{\tau}}} \tag{8}$$

where τ is a temperature parameter. The more similar a pair is according to the teachers, the harder the student will pull it together while keeping dissimilar pairs far apart.

Soft Triplet Margin Regression (STMR). The RKD loss [25] attempts to preserve the distances between pairs and the angles formed by the points in triplets. Given our problem setting, it is worth considering another invariant: the difference between the i-k distance and the i-j distance in a triplet (i,j,k). We observe that if the difference in the teacher is very large, the student does not need to match it exactly. It suffices for the student difference to be somewhat large, as this still captures dissimilarity. We define the soft triplet margin regression loss as

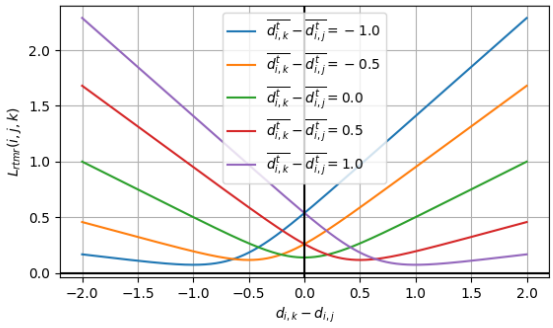


Figure 4. The STMR loss as a function of the student distance difference, for multiple values of the teacher distance difference. STMR is the Huber loss with a more lenient slope for one branch.

$$\begin{aligned} \alpha(i,j,k) &= \sigma((\overline{d^t_{i,k}} - \overline{d^t_{i,j}})/\tau_1) \\ \gamma(i,j,k) &= \log(1/\alpha(i,j,k) - 1)/\tau_2 \\ dd(i,j,k) &= (\overline{d^t_{i,k}} - \overline{d^t_{i,j}}) - (d_{i,k} - d_{i,j}) + \gamma(i,j,k) \\ L_{stmr}(i,j,k) &= \alpha(i,j,k)\zeta_{\tau_2}(dd_{i,j,k}) + \\ &\quad (1 - \alpha(i,j,k))\, \zeta_{\tau_2}(-dd_{i,j,k}) \end{aligned} \tag{9}$$

where ζ is the softplus function and τ_1 and τ_2 are temperature parameters. The distance difference $dd(i,j,k)$ is offset by the factor $\gamma(i,j,k)$ to zero out the gradient of the loss when $\overline{d^t_{i,k}} - \overline{d^t_{i,j}} = d_{i,k} - d_{i,j}$. The student will push points to match the difference of the teacher predicted distances, but not so strongly if both its difference and the teacher difference are large. See figure 4 for an illustration.

4. Results

4.1. Datasets

We tested our approach on three image datasets labeled with triplets: THINGS [8], a variant of ImageNet-HSJ [27] which we call IHSJC, and Yummly [41]. In its raw form, THINGS consists of 26,107 images from 1,854 classes, with 1.46M class level triplets. ImageNet-HSJ consists of 50,000 images from 1,000 classes, with 384K 8-rank-2 trials at image level. Each trial specifies a query, two positives and six negatives, and we converted this information into twelve class level triplets by taking all the combinations with the class of the query, of a positive and of a negative. We filtered out the triplets with less than three classes and those involved in contradictions, eventually arriving at 2.4M class triplets. Yummly consists of 100 food images not associated with classes and has 189K triplets.

We cannot simply split the triplet set of a dataset randomly into training, validation and test because the same image may appear in different subsets, e.g. both in training and test. This is fine for teacher models, but not for directly trained or student models, which use image data. We followed a different protocol for each model type:

1. Teacher protocol. First, we randomly distributed the classes (images for Yummly) into two subsets, training/validation and test, in an 80%-20% ratio. Then, we placed the triplets with all three classes/images from the same subset into the triplet version of that subset, discarding the triplets that did not meet this condition. For example, we put (i, j, k) in the training/validation triplet subset if i,j,k were training/validation classes; we discarded (i, j, k) with i and j training/validation classes and k test class.

2. Direct training (regular) protocol. For Yummly, this was identical to the teacher protocol. For THINGS and IHJC, we transformed each class triplet into image triplets by randomly sampling images from the three classes. We generated a fixed number of image triplets per class triplet, chosen to cover the entirety of the dataset images (2 for THINGS and 3 for IHSJC).

3. Student protocol. We first randomly distributed the classes (images for Yummly) into three subsets, training, validation and test, in a 60%-20%-20% ratio that respected the teacher split. Since the teachers can predict the relation between any three classes/images, we formed all the possible triplets with training classes/images and put them in training, and did the same for validation and test.

The three protocols guarantee correct assessment of model quality, as test images do not appear in the training or validation data. Additionally, the teacher split allows the student model to access supervision on the validation subset. See the supplementary material for dataset statistics.

Unless otherwise stated, all evaluation results below are reported on the test subset of the regular split.

4.2. Model Settings

In the main experiments, we finetuned both the student and directly trained models from a ResNet-50 backbone pretrained on ImageNet [7]. We employed a standard augmentation and preprocessing scheme: flip horizontally with probability 50%, take a random square crop with at least 50% overlap with the original image and resize to 224 x 224. At test time, images were resized to 256 on the shortest edge and center cropped to 224 x 224. All three model types (direct, teacher and student) were trained with the Adam optimizer [16] and used an embedding of size 128.

We trained 5 teacher models on each dataset. On THINGS and Yummly the batch size was set to 3,333 triplets and on IHSJC to 100,000. Each teacher was trained to convergence for 100 epochs with learning rate 10^{-3}.

For student models, the batch size was set to 249 on THINGS and IHSJC, except for the RKD loss [25], for which 172 was the largest value that fit the triplet angle constraints in GPU memory. On Yummly, the batch size was set to 60, the size the training set. Each student was trained to convergence for 100 epochs with learning rate 10^{-6}.

The directly trained models used batch size 249, 10 epochs and learning rate 10^{-4}, ensuring convergence.

4.3. Quantitative Evaluation

Baseline. We compared our approach against representative triplet based metric learning methods [39] [23] [32] and methods whose input can be derived from labeled triplets [5] [38]. On the new datasets, these methods collapse to simply using a loss on fixed triplets, which we call direct training. Note that the subset of labeled triplets is arbitrary and small compared to the set of all triplets, which prevents mining. Further note that high performing class based methods, e.g. [12] [15] [18], do not apply to the new datasets, as similarity does not follow from class information.

Results. To optimally configure our approach, we combined it with a wide range of losses, both from section 3.2 and from the existing literature. We present results in table 1. The metric used was the fraction of correct triplets (FCT), where a directed triplet (i, j, k) is correct if the model predicted distances follow $d_{i,j} < d_{i,k}$ and an undirected triplet is correct if both $d_{i,j} < d_{i,k}$ and $d_{j,i} < d_{j,k}$. The table shows the FCT average and standard deviation over 5 randomly initialized training runs; since the Yummly dataset only has 100 images, we also used 5 different splits into training, validation and test for it. On the THINGS dataset, the FCT for random guessing is 33.33% and the inter-rater agreement measured by [8] is 67.22%; on both IHSJC and Yummly, the FCT for random guessing is 50% but inter-rater agreement was not estimated. Thus, the FCT values must be interpreted relative to the 33-67 and 50-100 ranges, respectively. Our approach yields higher quality models than direct training with existing triplet based methods: the scaled gains are 8% on THINGS, 12% on IHSJC and 16% on Yummly. The optimal loss for our approach is RKD [25], as it does best on two datasets and is statistically close to best on the third ($p > 0.8$ with Welch's t-test).

It is insightful to compare direct training vs. our approach when they use the discrete and relaxed version of the same loss respectively. The pairs of rows D+Contrastive and TS+RC, D+Triplet and TS+RTM, D+InfoNCE and TS+RI, and D+MS and TS+RMS in table 1 show similar gains for our approach as in the overall comparison: at least 7% on THINGS, 12% on IHSJC and 14% on Yummly. The teachers do not learn perfect embeddings, but the fact that they provide dense supervision has a bigger impact on the quality of the student model.

Efficiency. During training, our approach sees each image once per epoch, while direct training on average sees

Method	THINGS	IHSJC	Yummly
	FCT[%]		
Random guessing	33.33	50.00	50.00
D+Contrastive [5]	43.95	53.68	55.60
D+Triplet [39]	50.67	57.51	63.55
D+Margin [23]	51.28	59.77	59.79
D+InfoNCE [32]	52.20	60.28	62.40
D+MS [38]	47.05	53.56	63.66
TS+RC [13]	55.79 ± 0.08	66.47 ± 0.15	70.90 ± 5.34
TS+RTM	55.85 ± 0.10	66.02 ± 0.07	70.77 ± 4.45
TS+RF	55.80 ± 0.11	66.17 ± 0.08	71.27 ± 5.25
TS+RI	55.69 ± 0.12	66.24 ± 0.13	**72.07** ± 5.98
TS+RMS [13]	56.05 ± 0.07	66.06 ± 0.04	71.32 ± 5.92
TS+MTT [42]	55.80 ± 0.06	65.93 ± 0.09	71.25 ± 5.28
TS+RKD [25]	**56.17** ± 0.18	**66.48** ± 0.08	71.27 ± 4.32
TS+STMR	56.00 ± 0.08	66.31 ± 0.06	71.12 ± 5.73
Inter-rater agreement	67.22	not available	not available

Table 1. Results across datasets, approaches and losses with a ResNet-50 backbone. Our teacher-student approach (the TS+ rows) outperforms direct training with triplet based metric learning losses (the D+ rows). The optimal loss for our approach is RKD, as it does best on two datasets and is statistically indistinguishable from best on the third dataset.

Type of method	THINGS	Imagenet-HSJ	Yummly
	overall time [h] (im. reps. per epoch)		
D+	23.7 (56)	61.4 (48)	1.4 (1,903)
TS+	1.7 (1)	2.6 (1)	0.15 (1)

Table 2. Efficiency data. Direct training repeats each image a few tens of times or more during an epoch, so it takes significantly longer to converge than our approach. In addition to repetitions, the time is affected by triplet directedness and batch size.

Method	THINGS	Imagenet-HSJ	Yummly
	FCT[%]		
STE Teacher	62.81	90.16	81.98
Margin Teacher	62.34	86.27	82.01

Table 3. Evaluation of single teacher models. The values effectively upper bound our approach, as it transfers knowledge from teacher models to a student model. Teacher models in the literature are close in quality or worse, see text.

an image a few tens of times or more, see table 2. The large difference comes from how batches are formed: direct training samples labeled triplets of images, while student model training samples images. There are many more labeled triplets than images (e.g. 1.46M vs 26K for THINGS) and these triplets cannot be packed densely in batches due to their low density. The teacher models train at negligible cost as they are single layer networks with scalar inputs, so overall our approach is much faster than direct training.

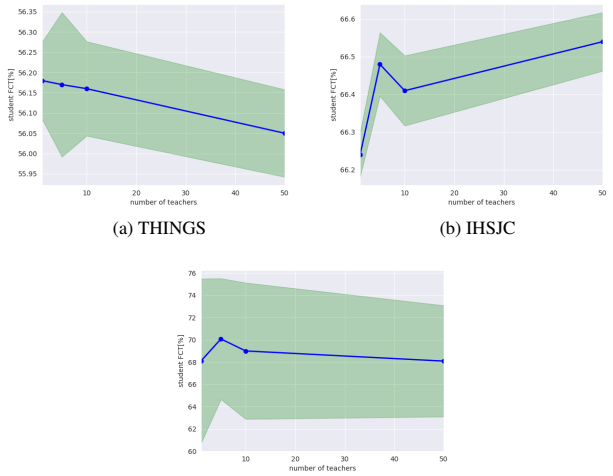
Teachers. Good teachers are crucial to the success of our approach. We report results for teacher models in table 3. Note that the numbers are for the triplets in the validation set, as the instances in the test set do not appear in training. For THINGS, the FCT is close to the value obtained by [8] when training an embedding on a slightly more forgiving dataset split: ours 62.8% vs. theirs 63.7% [44]. We show results on ImageNet-HSJ instead of IHSJC, as a similar evaluation was conducted on the former dataset [27]. The authors did not split the dataset into training, validation and test, but they report a triplet accuracy of 80.7%, clearly lower than our 90.1%. While neither comparison is apples-to-apples, the numbers do prove the effectiveness of our teacher training method.

We show the dependence of the student model quality on the number of teachers in figure 5. Five teachers do better on IHSJC and Yummly, but overall the difference between one and multiple teachers is marginal. See the supplementary material for the dependence on teacher embedding size.

Backbones. A transformer backbone does not affect the ranking of the losses, see the supplementary material.

4.4. Qualitative Evaluation

In figure 6, we compare direct training with the teacher-student paradigm by examining the nearest neighbors of a few query images. The images are from the test subset of THINGS and the two models compared are trained with the best performing losses on that dataset, InfoNCE and RKD. Whenever one model looks consistently better than the other, it finds neighbors with higher semantic similarity with the query, e.g. they are both containers, as opposed



(a) THINGS (b) IHSJC

(c) Yummly

Figure 5. Student model quality w.r.t. the number of teachers. For all datasets, the difference between 1 vs 5/10/50 teachers is marginal, with a slight advantage for 5 teachers.

Method	CUB-200	CARS-196	SOP
	R@1		
HIER^{384} [15]	85.7	88.3	86.1
TS+RKD^{384}	83.7	82.7	80.0
TS+STMR^{384}	84.7	87.7	84.4

Table 4. Knowledge distillation losses on standard datasets. STMR is within 2% of state of the art, showcasing the competitiveness of knowledge transfer as a framework for standard datasets.

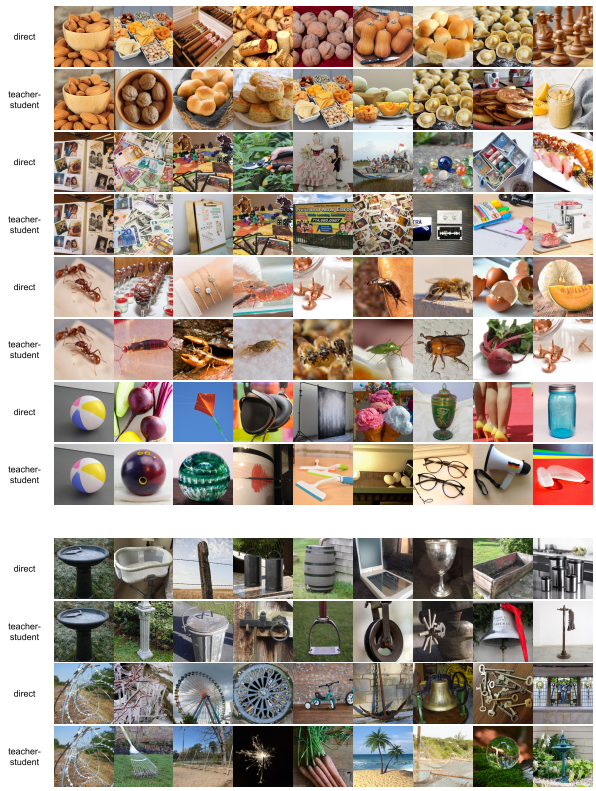


Figure 6. THINGS dataset: queries and search results with the best directly trained model (D+InfoNCE) vs. the best teacher-student trained model (TS+RKD). TS+RKD does better in the first four cases and worse in the last two. The superior model captures semantics better, e.g. soft round object in a room, or crawling insect.

to sharing low level features like shininess. See the supplementary material for an analysis of the embedding spaces.

4.5. Evaluation on Standard Datasets

Our approach is designed for datasets with triplet labels. Standard datasets can be converted to this format by taking all the groups of three images where only two share a class, so our approach can still be applied. Note that it becomes a traditional method if a metric learning loss is used: the teachers converge to rotated one-hot class encodings and this supervision turns the relaxed loss into the original loss. Interestingly, one-hot supervision makes knowledge distillation losses new metric learning losses. Our approach thus generalizes the class based formulation of metric learning, both to more datasets and to more losses.

We computed results on standard datasets for the top two knowledge distillation losses, RKD and STMR. For fair comparison with the state of the art, we used a ViTS backbone [17] and embedding size 384, training with AdamW [22] for 100 epochs with learning rate 10^{-5}. Both RKD and STMR were combined with a triplet mining scheme [23] complementary to the focus of our approach on increasing label density. RKD did not fully benefit from triplet mining, as it operates both on triplets and on pairs, but STMR did and its recall at 1 is within 2% of the state of the art, see table 4. The good results with STMR on both the standard and the new datasets suggest the generality of knowledge transfer as a framework for metric learning.

5. Conclusion

We presented a novel metric learning approach for triplet labeled datasets that define similarity in a more general way than class based datasets. In a teacher-student learning paradigm, we trained multiple teacher models based only on image relations and then transferred their knowledge to a student model that uses image data. By focusing the teachers on relations and the student on visual features, we were able to use very large batches for the teachers and to provide dense supervision to the student, two important aspects impacting model quality and training efficiency. Our approach showed clear improvement over direct training with existing triplet based metric learning methods; additionally, we highlighted the wide applicability of knowledge transfer as a framework for metric learning.

References

[1] Irving Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94: 115–147, 1987. 3

[2] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 2

[3] Aleksandr Ermolov, Leyla Mirvakhabova, Valentin Khrulkov, Nicu Sebe, and Ivan Oseledets. Hyperbolic vision transformers: Combining improvements in metric learning. In *CVPR*, 2022. 2

[4] Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R. Scott. Deep metric learning with hierarchical triplet loss. In *ECCV*, 2018. 2

[5] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 2, 6, 7

[6] Ben Harwood, B. G. Vijay Kumar, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *ICCV*, 2017. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 6

[8] Martin N. Hebart, Charles Y. Zheng, Francisco Pereira, and Chris I. Baker. Revealing the multidimensional mental representations of natural objects underlying human similarity judgements. *Nature Human Behaviour*, 4, 2020. 1, 2, 3, 4, 5, 6, 7

[9] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[10] Maurice Kendall and Jean D. Gibbons. *Rank Correlation Methods*. Oxford University Press, 5 edition, 1990. 3

[11] S. Kim, M. Seo, I. Laptev, M. Cho, and S. Kwak. Deep metric learning beyond binary supervision. In *CVPR*, 2019. 2

[12] S. Kim, D. Kim, M. Cho, and S. Kwak. Proxy anchor loss for deep metric learning. In *CVPR*, 2020. 2, 6

[13] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Embedding transfer with label relaxation for improved metric learning. In *CVPR*, 2021. 3, 4, 7

[14] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Self-taught metric learning without labels. In *CVPR*, 2022. 3

[15] Sungyeon Kim, Boseung Jeong, and Suha Kwak. Hier: Metric learning beyond class labels via hierarchical regularization. In *CVPR*, 2023. 2, 6, 8

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[17] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 8

[18] Dmytro Kotovenko, Pingchuan Ma, Timo Milbich, and Björn Ommer. Cross-image-attention for conditional embeddings in deep metric learning. In *CVPR*, 2023. 6

[19] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, 2013. 1

[20] Priyadarshini Kumari, Ritesh Goru, Siddhartha Chaudhuri, and Subhasis Chaudhuri. Batch decorrelation for active metric learning. In *IJCAI*, 2020. 2

[21] Elad Levi, Tete Xiao, Xiaolong Wang, and Trevor Darrell. Rethinking preventing class-collapsing in metric learning with margin-based losses. In *ICCV*, 2021. 2

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 8

[23] R. Manmatha, Chao-Yuan Wu, Alexander J. Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. In *ICCV*, 2017. 2, 6, 7, 8

[24] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh. No fuss distance metric learning using proxies. In *ICCV*, 2017. 2

[25] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, 2019. 3, 4, 5, 6, 7

[26] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *ECCV*, 2018. 3

[27] Brett D. Roads and Bradley C. Love. Enriching imagenet with human similarity judgments and psychological embeddings. In *CVPR*, 2020. 1, 3, 4, 5, 7

[28] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015. 3

[29] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Björn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *International Conference on Machine Learning*, 2020. 2

[30] K. Roth, O. Vinyals, and Z. Akata. Integrating language guidance into vision-based deep metric learning. In *CVPR*, 2022. 2

[31] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 1, 2, 5

[32] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Neural Information Processing Systems*, 2016. 1, 2, 6, 7

[33] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016. 1

[34] Amos Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977. 2

[35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *Arxiv preprint*, 2018. 5

[36] Laurens van der Maaten and Kilian Weinberger. Stochastic triplet embedding. In *IEEE International Workshop on Machine Learning for Signal Processing*, 2012. 4

[37] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016. 1

[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 *CVPR*, 2019. 2, 6, 7

[39] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 2009. 2, 4, 6, 7

[40] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 1

[41] Michael Wilber, Iljung Kwak, and Serge Belongie. Cost-effective hits for relative similarity comparisons. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 2(1), 2014. 1, 3, 5

[42] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *International Conference on Knowledge Discovery and Data Mining*, 2017. 3, 7

[43] Yanfu Zhang, Lei Luo, Wenhan Xian, and Heng Huang. Learning better visual data similarities via new grouplet non-euclidean embedding. In *ICCV*, 2021. 2

[44] Charles Y. Zheng, Francisco Pereira, Chris Ian Baker, and Martin N. Hebart. Revealing interpretable object representations from human behavior. In *ICLR*, 2019. 7

[45] Wenzhao Zheng, Yuan Huang, Borui Zhang, Jie Zhou, and Jiwen Lu. Dynamic metric learning with cross-level concept distillation. In *ECCV*, 2022. 2