Generalized Knowledge Distillation via Relationship Matching

Han-Jia Ye, Su Lu, De-Chuan Zhan

Abstract—The knowledge of a well-trained deep neural network (a.k.a. the "teacher") is valuable for learning similar tasks. Knowledge distillation extracts knowledge from the teacher and integrates it with the target model (a.k.a. the "student"), which expands the student's knowledge and improves its learning efficacy. Instead of enforcing the teacher to work on the same task as the student, we borrow the knowledge from a *teacher trained from a general label space* — in this "Generalized Knowledge Distillation (GKD)", the classes of the teacher and the student *may be the same, completely different, or partially overlapped*. We claim that the comparison ability between instances acts as an essential factor threading knowledge across tasks, and propose the **RE**lationship **Fa**cI**l**itated **L**ocal c**L**assifi**E**r **D**istillation (REFILLED) approach, which decouples the GKD flow of the embedding and the top-layer classifier. In particular, different from reconciling the instance-label confidence between models, REFILLED requires the teacher to *reweight* the hard tuples pushed forward by the student and then matches the similarity comparison levels between instances. An embedding-induced classifier based on the teacher model supervises the student's classification confidence and adaptively emphasizes the most related supervision from the teacher. REFILLED demonstrates strong discriminative ability when the classes of the teacher vary from the same to a fully non-overlapped set w.r.t. the student. It also achieves state-of-the-art performance on standard knowledge distillation, one-step incremental learning, and few-shot learning tasks.

Index Terms—Knowledge Distillation, Generalized Knowledge Distillation, Cross-Task, Model Reuse, Representation Learning

✦

1 INTRODUCTION

SUPERVISED deep learning has demonstrated success in a variety of fields [1]. Given the instances and corresponding annotations from the target task, we train a deep neural network to minimize the discrepancy between the model predictions and the ground-truth labels. Knowledge distillation (KD) [2], [3], [4] facilitates the learning efficiency of a deep neural network via taking advantage of the "dark knowledge" from another well-trained model. In detail, a strong classifier, *e.g.*, a neural network trained with deeper architectures [5], high-quality images [6], or precise optimization strategies [7], [8], acts as a "teacher" and guides the training of a "student" model by richer supervision, so that the learning experience from a related task is reused in the current task. KD improves the discriminative ability of the target student model [9], [10], relieves the burden of model storage [3], [5], [4], [7], [11], [12] and enables the training of a deep neural network in low-resource environments [13], [14]. Applications of KD have been witnessed in a wide range of domains such as model/dataset compression [15], [16], [17], [18], [19], [20], multi-task learning [21], [22], and incremental image classification [23], [24].

The teacher's class posterior probability over an instance is the most common dark knowledge, as it indicates the teacher's estimation of how similar an instance is to candidate categories. Besides the extreme "black or white" supervision, the student is asked to align its posterior with the teacher during its training progress. Although prediction matching allows knowledge to be transferred across different architectures [3], [17], its dependence on
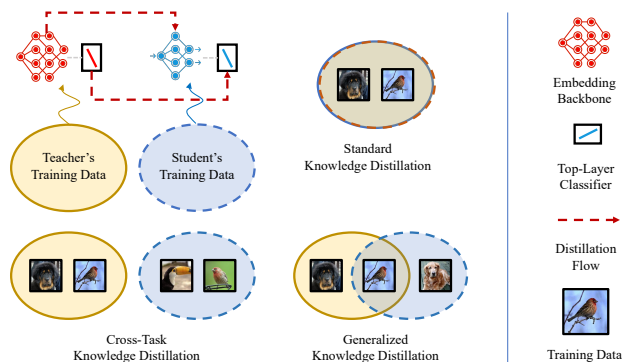

Figure 1: An illustration of strengthening a student model on the target task via distilling the knowledge from a teacher model. In standard Knowledge Distillation (KD), teacher and student share the same set of classes. In cross-task KD, a teacher is learned from images with fully non-overlapping classes, while its learning experience is distilled to facilitate the training of the student model. Generalized KD is a more general case of the previous two, where the student could have *the same, different, or partially overlapped* classes w.r.t. the teacher. The red dotted line indicates the distillation flow for the embedding backbone and the top-layer classifier, respectively, in our proposed model.

instance-label relationship restricts both teacher and student to the same label space.

We emphasize the necessity to use a general teacher and extend KD to more practical applications. In other words, the related teacher should not be limited to having classes that are completely same as the target task. For example, it is intuitive to reuse a well-trained model classifying animals to help the training of a student model on fine-grained birds. In *cross-task KD*, a student distills the knowledge from a

• *H.-J. Ye, S. Lu, and D.-C. Zhan are with State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210023, China. E-mail: {yehj,lus,zhandc}@lamda.nju.edu.cn*

teacher trained on related but non-overlapping label spaces, where the label discrepancy between teacher and student impedes the learning experience transition [25]. *Generalized Knowledge Distillation* (GKD) is a general case of standard KD and cross-task KD, where a teacher could have *the same, fully different, or partially overlapped* classes w.r.t. the student. Figure 1 illustrates the notion of KD variants.

The *comparison ability* — measuring similarity level between two instances based on their embeddings — captures a kind of invariant nature of the model [26] and is *free from the label constraint* [27], [28], [25]. For a teacher and a student discerning 'Husky vs. Birman" and "Poodle vs. Persian" respectively, the teacher's discriminative embedding encoding the "dog-cat" related characteristics can compare Poodle/Persian in the student's task and should be helpful for student's training. We expect the student to benefit from the teacher's knowledge if they are related, *i.e.*, the teacher's comparison ability fits the student's task. Otherwise, the student will perform as well as the one trained without a teacher. Thus, we bridge the knowledge transfer in GKD with the *instance-wise relationship* and thread the knowledge reuse for both embedding and top-layer classifier by taking advantage of the teacher's comparison ability.

To this end, we propose a 2-stage approach **RE**lationship **Fac**Ilitated **L**ocal c**L**assifi**E**r **D**istillation (REFILLED) based on current task's data and a well-trained teacher. First, the discriminative ability of embedding is stressed. For those hard similarity tuples determined by the student's embedding, how the teacher compares them acts as additional supervision. In other words, the teacher promotes the discriminative ability of the student's embedding by specifying *how much* dissimilar an object's impostors should be far away from its target nearest neighbor. The instance-wise knowledge from the teacher is distilled via matching comparisons. The teacher then constructs soft supervision for classifying each instance based on the similarity between the instance and an embedding center, eliminating the restrictions between label spaces. Specifically, the classification confidences of the student are aligned with the embedding-induced "instance-label" predictions from the teacher. The strength of the supervision is weighted by the relatedness between teacher and student automatically.

Empirical results verify that REFILLED effectively transfers the classification ability from various configurations of teachers to a student, including teachers with the same, different, and partially overlapped label spaces. REFILLED also outperforms recent methods in one-step incremental learning, few-shot learning, and middle-shot learning problems. In summary, our contributions are

- We investigate GKD to enhance the training efficiency of a deep neural network by reusing the knowledge from a well-trained teacher *without label restriction*.
- We propose REFILLED which aligns the high-order comparison between models locally and weights the most helpful supervision from the teacher adaptively.
- REFILLED works well in generalized KD, incremental learning, and few-shot learning benchmarks.

After the related literature and the preliminary in Section 2 and Section 3, we formalize our REFILLED approach in Section 4. Finally are experiments and conclusion.

2 RELATED WORK

Knowledge Distillation (KD). Rich supervision plays a crucial role in building a machine learning or visual recognition system, where taking advantage of the learning experience from related pre-trained models becomes a shortcut to facilitate the model training in the current task [29]. Different from fine-tuning [30] or weights matching [31], [32], [33], [34], [35], [36] that regularize the model from the "parameters" perspective, we can reuse the dark knowledge/privileged information [37], [38], [39] to explain or assist the training process of the model from the "data" aspect [40], [3]. Denote a fixed well-trained model from a related task and the model in the current task as the "teacher" and the "student", respectively. KD matches the behaviors of two models on the current task's data [41], [42], [19]. The teacher could be a high-capacity deep neural network trained on the same task [13], [9], [43] or a previous generation of the model along the training progress [44], [7], [8]. The dark knowledge in KD can be implemented as the soft label, *i.e.*, the posterior probability of an instance [3], [45], [17], hidden layer activation [5], [46], [47], [48], parameter flows [4], transformations [49], and pathwise statistics [50]. Distilling the knowledge from one model to another has been investigated for model compression [2], [11], [12], [51] and incremental learning [52], [9], [43].

Cross-task knowledge transfer. In practical applications, the knowledge from teachers of other tasks, *i.e.*, teachers trained on non-overlapping sets of labels, also assists the training of a target task student. Heterogeneous transfer learning updates both student and teacher on the current and related domains (resp. tasks) to close distribution (resp. label) divergence [22], [53]. Heterogeneous model reuse takes advantage of the teacher from a related task, which relieves the burden of data storage so as to decrease the risk of privacy leaking [35], [54]. Meta-learning has also been utilized to transfer knowledge across different label spaces, *e.g.*, few-shot learning [55], [56], [57], [58], [59], [60]. These approaches usually require special training strategies of the teacher. A *fixed* well-trained teacher is provided in KD, but since KD usually relies on the correspondence between classifier and categories, it is challenging to reuse the classification knowledge from a cross-task teacher. In REFILLED, we bridge the label divergence via comparison matching.

Class incremental learning also takes advantage of KD in a cross-task scenario, where non-overlapped sets of classes arrive sequentially. The classifier on previously seen classes is the teacher, which is incorporated in training the current stage's student without storing historical data [24], [9], [43], [61]. KD helps avoid catastrophic forgetting by matching the student's predictions over previous classes with the teacher [62], [52]. The goal of REFILLED is not to avoid forgetting but transferring the knowledge of the teacher to improve the target classifier. Furthermore, REFILLED utilizes teachers with general label spaces and transfers the classification ability *across different architectures*.

Embedding learning for KD. Embedding learning improves the feature representation by pulling similar instances together and pushing dissimilar ones away [63], [64], [28], [65], [66], [67]. Benefited from kinds of side-information [68], [63], embeddings are learned to explain the given instance-wise relationships [69], [70], [71]. Instead of matching the

instance-label predictions between models, matching the embedding [72], [16], [73], [74], [75], pairwise distance [76], [77], and similarity graph [78], [79], [50], [80], [81] have been investigated. Then "downstream" cross-task clustering and representation learning tasks could be improved [25], [58], [6]. For example, RKD [82] constructs angels over triplets and matches the angels by regression. We emphasize the relationship matching in distilling from a *general teacher* trained from possible *in-task and cross-task* classes w.r.t. the target student. In our embedding distillation stage, we organize instances in a *tuple*, which captures high-order local comparisons efficiently and provides richer supervision from the teacher. The superiority of REFILLED is validated in experiments.

Some concurrent Self-Supervised Learning (SSL) methods distill the relationship in mini-batches [83], [84], [85], [86]. A stronger model or previous generations during the training progress becomes the teacher to compress the model or improve the embedding quality. Different from constructing comparisons with class semantics, in SSL, augmented views of an instance are treated as similar ones while different instances are dissimilar. REFILLED utilizes the characteristic of embeddings to bridge the class gap in GKD, and further emphasize the transfer of classification ability. We investigate our differences with SSL w.r.t. both the distillation strategy and the similarity measure in experiments.

3 KNOWLEDGE REUSE VIA DISTILLATION

We briefly introduce the standard Knowledge Distillation (KD) via matching the soft labels at first. Then we describe the concrete settings of cross-task knowledge distillation and Generalized Knowledge Distillation (GKD).

3.1 Background and Notations

For a C-class classification task, we denote the training data with N examples as $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^D$ and $\mathbf{y}_i \in \{0,1\}^C$ are the instance and the corresponding one-hot label, respectively. Index of 1 in $\mathbf{y}_i$ indicates the class of $\mathbf{x}_i$. Denote the class set as $\mathcal{C}$, where $|\mathcal{C}| = C$. A classifier $f(\mathbf{x})\colon \mathbb{R}^D \mapsto \{0,1\}^C$ (*e.g.*, a deep neural network) predicts the label for an instance $\mathbf{x}$, which could be represented as $f(\mathbf{x}) = W^\top \phi(\mathbf{x})$.[1] There are two components in f, the feature extractor $\phi\colon \mathbb{R}^D \mapsto \mathbb{R}^d$ mapping the raw input to a d dimensional latent space, and a linear classifier $W = [\mathbf{w}_1, \ldots, \mathbf{w}_C] \in \mathbb{R}^{d \times C}$ based on the extracted features. The objective minimizes the discrepancy between the prediction and the true label over all instances in $\mathcal{D}$:

$$\min_f \sum_{i=1}^N \ell\left(f(\mathbf{x}_i),\, \mathbf{y}_i\right) \ . \tag{1}$$

ℓ is the loss such as the cross-entropy. We denote optimizing Eq. 1 from scratch as the vanilla supervised deep learning.

3.2 Standard Knowledge Distillation

Given a well-trained C-class classifier f_T for the class set $\mathcal{C}'$ with $\mathcal{C}' = \mathcal{C}$, it is an effective manner to distill the "dark knowledge" from the *fixed* f_T to help the training

1. We omit the bias term for discussion simplicity.

progress of the target model f. Subscript "T" denotes the model/parameters of the teacher. To improve the training efficacy of f, Hinton, Vinyals, and Dean [3] suggest to align the soft targets of two models besides the vanilla objective:

$$\min_f \sum_{i=1}^N \ell(f(\mathbf{x}_i), \mathbf{y}_i) + \lambda \mathcal{R}(\mathbf{s}_\tau(f_T(\mathbf{x}_i)), \mathbf{s}_\tau(f(\mathbf{x}_i))) \ . \tag{2}$$

$\lambda > 0$ is a trade-off parameter. $\mathbf{s}_\tau$ transforms the confidence for all classes into a C-class posterior probability. $\mathbf{s}_\tau(f(\mathbf{x}_i)) = \mathbf{softmax}(\frac{f(\mathbf{x}_i)}{\tau})$. τ is a non-negative temperature, the larger the value of τ, the smoother the output. $\mathcal{R}(\cdot)$ measures the difference between two distributions, *e.g.*, the Kullback-Leibler (KL) divergence. In Eq. 2, the student not only minimizes the mapping f from an instance to its label over $\mathcal{D}$, but also aligns its predictions with the teacher on the same set of instances. Note that the student and the teacher could use different temperatures.

In standard KD, both teacher f_T and student f target the same C classes. Given the training data and f_T, richer supervision like the soft labels is incorporated when training f, which encodes the relationship between an instance and C candidate classes. f_T could be a deep neural network with a larger capacity [3], [72], [17], which makes f compact and discriminative. f_T could also be a certain generation of f along with the whole training progress. Such self-distillation reduces the training cost and simultaneously enables sufficient training w.r.t. the vanilla strategy [7], [8].

3.3 Cross-Task Knowledge Distillation

The standard KD in Eq. 2 requires the student to be trained for the same labels $\mathcal{C}$ as the teacher, so that their classification results on the same instance could be matched. In a general scenario, it is necessary to borrow the learning experience from a *cross-task* teacher, *i.e.*, f_T has *a non-overlapping class set* $\mathcal{C}'$ and $\mathcal{C}' \cap \mathcal{C} = \emptyset$. Relaxing the requirement of the teacher enables KD in more applications.

3.4 Generalized Knowledge Distillation

GKD takes a further step given a teacher trained on a *general* class set $\mathcal{C}'$ — we do not restrict $\mathcal{C}'$ to be non-intersected with the target classes $\mathcal{C}$. In other words, it could be either $\mathcal{C}' = \mathcal{C}$ (as in standard KD), $\mathcal{C}' \cap \mathcal{C} = \emptyset$ (as in cross-task KD), or even $\mathcal{C}' \cap \mathcal{C} \neq \emptyset$. In the third case, only parts of the target task's classes are related to classes f_T trained for. The teacher provides additional supervision with f_T in the training progress of f. However, due to the fact that the student is *agnostic* of which part of the teacher's supervision is related, *e.g.*, the classes indexes of teacher's predictions that are overlapped with the target classes, it should identify and extract the helpful supervision from the teacher as much as possible instead of treating teacher's supervision uniformly. For example, if f_T is trained on generic animals, it could provide helpful supervision on animal-like instances in the current task but not man-made objects. If all classes in $\mathcal{C}'$ are distant from those in $\mathcal{C}$, we expect f to perform as well as the one trained in the vanilla case without a teacher. GKD helps build a visual recognition system efficiently. Specifically, we can leverage well-trained models from any tasks with different architectures to improve the discerning ability of the system without accessing their training data.

4 REFILLED FOR GENERALIZED KD

We introduce the main idea of **RE**lationship **FacI**litated **L**ocal c**L**assifi**E**r **D**istillation (REFILLED) approach, followed by analyses and discussions of its two stages.

4.1 Decoupling the Distillation via REFILLED

The two components in the target model f, *i.e.*, the embedding ϕ and the top-layer classifier W, capture the correlations between instances and classes, respectively. In GKD, REFILLED makes use of the characteristic of each part in f and transfers the rich supervision from teacher to student by distilling knowledge for corresponding components. The teacher's *comparison ability* does *not depend on concrete labels* and bridges the possible label gap between two models. Matching the relationship makes the distillation in REFILLED be agnostic of classes. As in Figure 2, we align instance-wise comparisons measured by ϕ with the teacher. The embedding distillation specifies how similar two objects are when training ϕ, and makes ϕ as discriminative as the teacher. After that, we construct a similarity-based classifier for target classes based on the teacher's embedding. We also derive a confidence-based criterion to identify helpful instance-wise supervision from the teacher adaptively, which extracts helpful supervision when training f in GKD.

4.2 Distill the Embedding

Instance embedding $\phi(\mathbf{x})$, the penultimate layer output of a deep neural network, encodes discriminative property of objects [87], [88], [26] without a direct dependence on labels [63], [89], [64], [27], [28], [25]. The instance-wise similarity computed based on their embeddings reveals whether two objects are similar or not, and how much they are similar. Therefore, similar instances are close to each other (with smaller distances) and dissimilar ones are far away. To distill the knowledge from a teacher with possible cross-task classes, we first focus on the transferable embedding, making the student's embedding as discriminative as the teacher.

4.2.1 Comparison Matching

Based on class semantics, denote two instances are similar if they come from the same class, and they are dissimilar if they have different labels. The distance between a pair of instances $(\mathbf{x}_i, \mathbf{x}_j)$ based on the embedding function ϕ is $\mathbf{D}_\phi(\mathbf{x}_i, \mathbf{x}_j) = ||\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)||_2$. A good embedding makes embedding-based distances small for similar pairs and large for dissimilar ones. We formulate the similarity relationship between $\mathbf{x}_i$ and others into a *tuple*, *i.e.*,

$$(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{P}}, \mathbf{x}_{i1}^{\mathcal{N}}, \ldots, \mathbf{x}_{iK}^{\mathcal{N}}), \qquad (3)$$

which contains one similar positive neighbor $\mathbf{x}_i^{\mathcal{P}}$ w.r.t. the anchor $\mathbf{x}_i$ and K dissimilar negative impostors $\{\mathbf{x}_{i1}^{\mathcal{N}}, \ldots, \mathbf{x}_{iK}^{\mathcal{N}}\}$.[2] We transform the similarity in the tuple into a probability with softmax operator, which reveals how

2. We assume the same K for different instances for simplicity, and K could vary across tuples automatically. See Section 4.2.2 for details.

much the anchor is close to its target neighbor than those impostors:

$$p_i(\phi) = \mathbf{s}_\tau \left(\left[-\mathbf{D}_\phi\left(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{P}}\right), \right.\right. \qquad (4)$$
$$\left.\left. -\mathbf{D}_\phi\left(\mathbf{x}_i, \mathbf{x}_{i1}^{\mathcal{N}}\right), \ldots, -\mathbf{D}_\phi\left(\mathbf{x}_i, \mathbf{x}_{iK}^{\mathcal{N}}\right)\right]\right).$$

Eq. 4 measures the relative instance-wise similarities. The closer a neighbor or an impostor with $\mathbf{x}_i$, the larger the corresponding element in $p_i(\phi)$. For example, if the target neighbor $\mathbf{x}_i^{\mathcal{P}}$ has a very small distance with $\mathbf{x}_i$, then $p_i(\phi)$ becomes a one-hot label with only the first element equals 1. In the vanilla scenario, we minimize the distance between the anchor with the target neighbor and push all impostors away based on the "similar or not" binary supervision [63], [64], [27], [28], which is the same as the case to minimize the discrepancy between $p_i(\phi)$ with the $K+1$ dimensional ground-truth probability $[1, 0, 0, \ldots, 0]$.

In REFILLED, we take advantage of the "dark knowledge" in $p_i(\phi)$ with richer similarity comparison information — the similar degree of an instance $\mathbf{x}_i$ with a positive and multiple negative candidates are characterized in detail. We improve the discriminative ability of the student's embedding ϕ by distilling the tuple comparison knowledge from the teacher, *i.e.*, we minimize the KL-divergence over all tuples:[3]

$$\min_\phi \sum_i \mathbf{KL}\left(p_i(\phi_T) \,\|\, p_i(\phi)\right). \qquad (5)$$

Eq. 4 describes the fine-grained differences measured by the teacher inside tuples. By aligning comparisons in Eq. 5, the student is expected to be able to compare instances as well as the teacher. For example, the student treats a flying "black tern" and a "red-winged blackbird" as dissimilar in the vanilla scenario and pushes their embeddings apart. The teacher's tuple similarity vector $p_i(\phi_T)$ may indicate that a flying "red-winged blackbird" is more similar to a flying "black tern" than a "black tern" sipping the water, then the student can benefit from the richer supervision through Eq. 4. In experiments, we show that if the comparison ability of the teacher matches the student's task, *e.g.*, both on fine-grained animals, comparison matching makes ϕ discriminative even with limited examples. Additionally, the heterogeneity between teacher and student, *e.g.*, different scales or dimensions between ϕ_T and ϕ, will not influence the matching in Eq. 5. Therefore, it facilitates knowledge transfer across different architectures.

We can rethink Eq. 4 from a retrieval perspective. Given the query instance $\mathbf{x}_i$, we'd like to find its most similar neighbor in $(\mathbf{x}_i^{\mathcal{P}}, \mathbf{x}_i^{\mathcal{N}}, \ldots, \mathbf{x}_{iK}^{\mathcal{N}})$ with the smallest distance. Eq. 4 encodes the probability to query each candidate instance. Eq. 5 matches a *soft* version of the anchor query probability between teacher and student, which promotes the retrieval ability of the student as effective as the teacher.

4.2.2 Construction of the comparisons

Tuples in the form of Eq. 3 contain instances from the target task, and we construct "semi-hard" tuples for comparison matching. In particular, for each instance $\mathbf{x}_i$ in the mini-batch,

3. One instance $\mathbf{x}_i$ may have multiple comparison tuples with different target neighbors. We use one $p_i(\phi)$ to denote them for notation simplicity in the summation. Details are in Section 4.2.2.
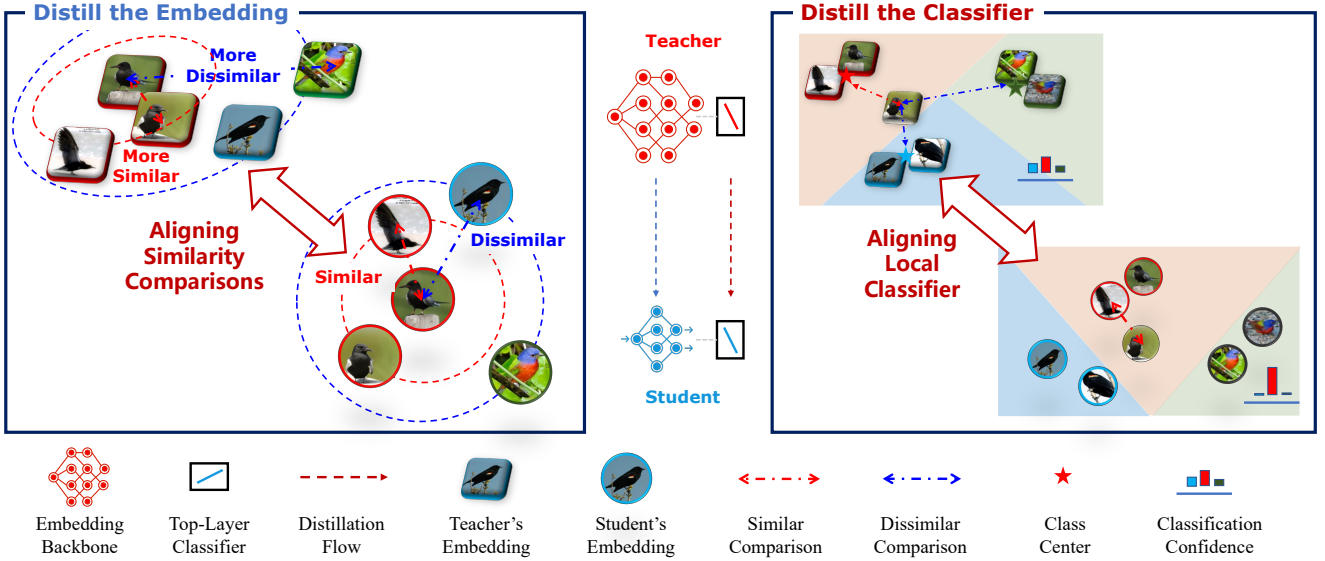
Figure 2: An illustration of the proposed **RE**lationship **F**ac**I**litated **L**ocal c**L**assifi**E**r **D**istillation (REFILLED) approach. The embedding and the top-layer classifier of the student distill the dark knowledge from the teacher's corresponding components, respectively. First, the student organizes instances into tuples and aligns their similarity comparisons with the teacher (left plot), introducing richer supervision such as the similar and dissimilar levels of relationships. REFILLED then matches classification confidences between two models over instances in the target task. We construct an embedding-based classifier with teacher's class prototypes (denoted by stars), which provides the posterior classification probability over both in-task and cross-task categories.

we first look for every $\mathbf{x}_i^{\mathcal{P}}$ from the same class as its neighbors. Then for each pair of $(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{P}})$, we enumerate those impostors $\mathbf{x}_i^{\mathcal{N}}$ with different labels to $\mathbf{x}_i$ but have larger distances than $\mathbf{D}_\phi\left(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{P}}\right)$ [64]. The impostors increase the difficulty of comparing instances and avoid embedding collapse. The distance is measured based on the ℓ_2-normalized embeddings with the current ϕ during the optimization progress of Eq. 5. The impostor number K in Eq. 3 is determined by the "semi-hard" impostors in a mini-batch, which varies for $(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{P}})$. We abbreviate multiple tuples for $\mathbf{x}_i$ with one $p_i(\phi)$ in Eq. 5. In summary, if the student finds tuples are hard to evaluate, it will ask the teacher for help about the concrete measures of the similarity levels.

4.3 Distill the Classifier Adaptively

Benefited from the distilled embedding for instance-instance comparisons, REFILLED further distills the knowledge from the teacher for instance-label classification. A similarity-based classifier is constructed with the help of the teacher's embedding to guide the update of the student's classifier. An adaptive weight is derived from the teacher's confidence to filter out its unhelpful supervision for GKD.

4.3.1 A General Approach for Classifier Distillation

We propose a general approach to distill the discerning ability of the top-layer classifier. A Nearest Class Mean (NCM) classifier [90], [56], [59] is constructed based on the teacher's embeddings ϕ_T, which captures the instance-label relationship for categories in both previous and target tasks without requiring the teacher to share the same label space.

Embedding-based classifier for GKD. With the teacher's embeddings $\phi_T(X) \in \mathbb{R}^{N \times d}$ on X, we compute the embedding center of all C classes in the target task by

$$P = \text{diag}(\mathbf{1} \oslash (Y^\top \mathbf{1})) Y^\top \phi_T(X) \in \mathbb{R}^{C \times d} . \quad (6)$$

$\oslash$ denotes the element-wise division. Each row $\mathbf{p}_c \in \mathbb{R}^d$ of P corresponds to the center of the c-th class. For any instance $\mathbf{x}$ in the target task, we can determine its label based on its similarity with the C centers:

$$p_T(c \mid \mathbf{x}) = \frac{\left(\left(\phi_T(\mathbf{x})^\top \mathbf{p}_c\right) / \|\mathbf{p}_c\|_2\right)}{\sum_{c'=1}^{C} \left(\left(\phi_T(\mathbf{x})^\top \mathbf{p}_{c'}\right) / \|\mathbf{p}_{c'}\|_2\right)} . \quad (7)$$

The larger the cosine similarity between an instance embedding $\phi_T(\mathbf{x}_i)$ to the c-th class center $\mathbf{p}_c$ in the teacher's embedding space, the larger the posterior probability $p_T(c \mid \mathbf{x})$ of class c. The similarity-based classifier takes advantage of the discriminative embedding of the teacher and could be applied for the target task even ϕ_T is trained from non-overlapped label spaces. Thus, we use Eq. 7 to bridge the possible cross-task label gap in GKD, which reveals the relationship between an instance and multiple classes.

Local Knowledge Distillation (LKD). We incorporate the instance-label relationship indicated by the teacher's embedding in Eq. 7 into the training progress of the student's top-layer classifier W. In particular, we match the student's prediction with the teacher through a *local* KD term. If the set of classes in the sampled mini-batch is $\mathcal{S}$ and $\mathcal{S} \subseteq \mathcal{C}$, then only the posterior over $\mathcal{S}$ are considered.

$$\min_f \sum_{i=1}^{N} \ell\left(f(\mathbf{x}_i), \mathbf{y}_i\right) + \lambda \mathbf{KL}\left(\hat{p}_T(\mathcal{S} \mid \mathbf{x}_i) \| \hat{\mathbf{s}}_\tau(f(\mathbf{x}_i))\right) \quad (8)$$

$$\hat{p}_T(\mathcal{S} \mid \mathbf{x}_i) = \mathbf{softmax}(\{p_T(s|\mathbf{x})\}_{s \in \mathcal{S}})$$
$$\hat{\mathbf{s}}_\tau(f(\mathbf{x}_i)) = \mathbf{softmax}(\{\mathbf{w}_s^\top \phi(\mathbf{x}) / \tau\}_{s \in \mathcal{S}}) .$$

In the second term of Eq. 8, rather than aligning two model's confidences of all target classes $\mathcal{C}$, only *partial posteriors* of classes in $\mathcal{S}$ are matched. LKD is not only efficient but also effective in emphasizing the difference between classes when the class number C is large (analyses are in section 4.4.2).

4.3.2 Adaptively Weighted LKD

In GKD, a teacher may be trained from labels only partially overlapped with the target classes, so uniformly matching the student's predictions with the teacher's predictions via Eq. 8 is not optimal. If a target class $c \in \mathcal{C}$ is related to a particular class in $\mathcal{C}'$ that the teacher trained on, the teacher may provide more precise estimations on $p(c \mid \mathbf{x})$ since it is more familiar with instances in class c. If all classes in $\mathcal{C}'$ are not related to c, the teacher may hesitate to provide a confident and helpful instance-label relationship estimation using Eq. 7. Thus, we provide an *adaptive* weight for LKD to emphasize the teacher's guidance over more related instances, which makes the student extract helpful supervision from the teacher. In other words, the student relies on the teacher's supervision over instances from classes that overlapped with the teacher's label space while weakening the teacher's guide for those "novel" classes w.r.t. the teacher.

Particularly, we measure the helpfulness of a teacher based on its prediction confidence. Define the "pseudo" label of an instance as the class index of the maximum confidence, *i.e.*, $\hat{s} = \arg\max_s \left(\{p_T(s|\mathbf{x})\}_{s \in \mathcal{S}} \right)$, the discrepancy between the teacher's prediction $\mathbf{softmax}(\{p_T(s|\mathbf{x})\}_{s \in \mathcal{S}})$ with the "pseudo" label measures how much the teacher is confident of its supervision [91]. We set

$$\lambda_i = 2\lambda \times \sigma\left(-\ell\left(\mathbf{softmax}(\{p_T(s|\mathbf{x})\}_{s \in \mathcal{S}}),\ \hat{s}\right)\right), \quad (9)$$

which is an instance-specific weight for the distillation term. We use the logistic function to transform the range of λ_i to $[0, \lambda]$. Finally, we have the following objective:

$$\min_f \sum_{i=1}^N \ell\left(f(\mathbf{x}_i),\ \mathbf{y}_i\right) + \lambda_i \mathbf{KL}\left(\hat{p}_T(\mathcal{S} \mid \mathbf{x}_i) \parallel \hat{\mathbf{s}}_\tau(f(\mathbf{x}_i))\right). \quad (10)$$

We detach the gradient of λ_i during the optimization. λ_i becomes larger when it is applied to those instances the teacher is familiar with and confident (with smaller loss values in Eq. 9), and has a smaller value when the teacher cannot provide strong supervision. In section 5.1, we verify such adaptive weight is able to differentiate the teacher's classes overlapped or different from the target ones. Another re-weight strategy based on the discrepancy between student's and teacher's predictions is also investigated in experiments.

Eq. 10 helps standard KD as well, where f_T and f have the same class set. Usually, f_T is confident over most instances and almost all λ_i values will be close to 1. Then Eq. 10 degenerates to Eq. 8. Otherwise, if there exist instances f_T is unconfident, those instances may be noisy labeled or corrupted ones. In this case, a smaller λ_i helps the student avoid being negatively affected by the teacher.

4.4 Summary and Discussions of REFILLED

We decouple the distillation flow into two stages for embedding and the top-layer classifier, respectively. First, REFILLED improves the discerning ability of the student's embedding by comparison matching in Eq. 5. Then the classification confidence between teacher and student is aligned in a local manner in Eq. 5, where helpful supervision from the teacher is emphasized with adaptively weighted distillation for GKD. The main flow of REFILLED is summarized in Alg. 1.

Algorithm 1 The Flow of REFILLED.

Input Pre-trained Teacher's Embedding ϕ_T.
Output The target model $f = W \circ \phi$.

Distill the Embedding:
for all Iter = 1,...,MaxIter **do**
  Sample a mini-batch $\{(\mathbf{x}_i, \mathbf{y}_i)\}$.
  Generate tuples $\{(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{P}}, \mathbf{x}_{i1}^{\mathcal{N}}, \dots, \mathbf{x}_{iK}^{\mathcal{N}})\}$ with $\{\phi(\mathbf{x})\}$.
  Compute probability of tuples $p_i(\phi_T)$ as Eq. 4.
  Optimizing ϕ by aligning comparisons in Eq. 5.
end for
Distill the Classifier:
Initialize f with ϕ.
Compute the instance-specific weight λ_i.
Optimizing f with Eq. 10.

4.4.1 Discussions on Embedding Distillation

Why not a direct embedding matching? One intuitive way to match the instance-wise relationship between teacher and student is to align their embeddings directly, *e.g.*, minimizing the loss $\|\phi(\mathbf{x}) - \phi_T(\mathbf{x})\|_2^2$ over all instances in the current task [72], [92], [48]. This constraint requires both models to have the same size of embeddings, which is too strong to satisfy, especially there exists an architecture gap between the two models. An asymmetric map is learned to match embeddings with different dimensions in [73], whose distillation quality is influenced by the dimension difference between teacher and student. [78], [82], [76], [77], [50] keep the embedding-based pairwise relationship (*e.g.*, distances) between teacher and student have similar values. If two models measure similarity at different scales, regularizing teacher and student have same distances between pairs still has drawbacks. Even though the student already obtains the right similarity relationship, the teacher could wrongly adjust it due to their scale differences. Therefore in REFILLED, we ask the teacher to provide its estimation about *relative comparisons* among instances in the form of *tuples* and require the student to align such relative similarity measure for discriminative embeddings. The high order comparisons between instances are utilized.

Another perspective on comparison matching. We illustrate the effect of comparison matching in Eq. 5 in a special case with one negative impostor ($K = 1$). More analyses for $K > 1$ are in the supplementary. With only one impostor in the tuple, we can simplify the term $p_i(\phi)$ in Eq. 4 as $\sigma(\mathbf{D}_\phi\left(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{N}}\right) - \mathbf{D}_\phi\left(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{P}}\right))$, where $\sigma(x) = 1/(1 + e^{-x})$ is the logistic function squashing the input into $[0, 1]$.

Define $\rho_i = 1 - p_i(\phi_T)$ and the logistic loss $\iota(x) = \ln(1 + \exp(-x))$, we can reformulate Eq. 5 as

$$\begin{aligned} \mathbf{KL}\left(p_i(\phi_T) \parallel p_i(\phi)\right) &\cong\ \iota\left(\mathbf{D}_\phi(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{N}}) - \mathbf{D}_\phi(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{P}})\right) \\ &+\ \rho_i\left(\mathbf{D}_\phi(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{N}}) - \mathbf{D}_\phi(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{P}})\right). \end{aligned} \quad (11)$$

$\cong$ neglects constants in the derivation. The first term in Eq. 11 with loss ι forces the distance $\mathbf{D}_\phi(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{N}})$ between dissimilar instances larger than the distance $\mathbf{D}_\phi(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{P}})$ between similar ones, so that the distance with ϕ is matched with the similarity relationship indicated by the tuple. Minimizing the first term equals vanilla embedding learning, making same-class instances close and different-class ones apart.
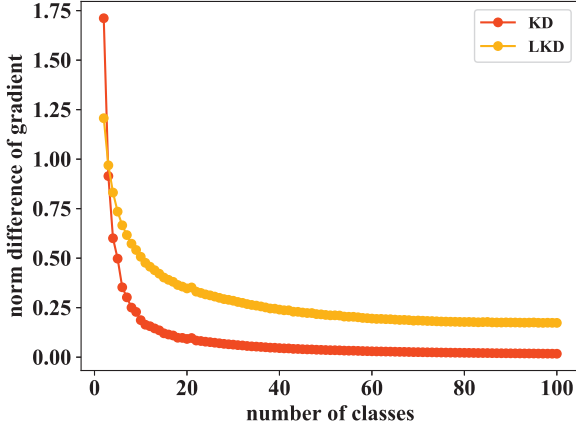
Figure 3: The averaged norm differences between the vanilla cross-entropy loss in Eq. 1 and the KD objective in Eq. 2 over the gradient of all top-layer classifiers, *i.e.*, $\mathbf{mean}_c \| \frac{\partial O_{\mathbf{ce}}}{\partial \mathbf{w}_c} - \frac{\partial O_{\mathbf{kd}}}{\partial \mathbf{w}_c} \|_2$ is shown in red. Its LKD counterpart of the average gradient between Eq. 1 and Eq. 10 is shown in yellow. When the number of classes in the target task grows, the norm difference based on the KD objective decreases fast, which indicates weaker additional supervision introduced by the distillation term. However, such a decrease is mitigated with LKD loss.

An additional term in Eq. 11 is introduced in comparison matching, which makes the anchor's distance between neighbor and impostor close. The strength ρ_i on the second term indicates the teacher's confidence over the given triplet. If $\mathbf{D}_{\phi_T}(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{P}})$ is smaller than $\mathbf{D}_{\phi_T}(\mathbf{x}_i, \mathbf{x}_i^{\mathcal{N}})$, *i.e.*, the teacher measures the target neighbor much similar w.r.t. the anchor than the impostor, then $p_i(\phi_T)$ is large and ρ_i is small. In this case, minimizing Eq. 11 emphasizes the first term, which makes the comparison matching the same as the vanilla embedding learning. In contrast, if the similarity relationship provided by the tuple is not very consistent with the one measured by the teacher, then ρ_i becomes large. For example, given (a) cat, (b) tiger, (c) bear, we will treat both (a, b) and (a, c) as dissimilar ones since they come from different classes. A well-trained ϕ_T may discover the relatedness between (a) and (b), so it would use less force to push (a, b) apart than that for (a, c). Therefore, large ρ_i weakens ι with an opposite objective in the additional term.

In summary, different from the binary label ("similar" or "dissimilar") indicated by the tuple, comparison matching *rectifies the strength* when minimizing (resp. maximizing) the distance between similar (resp. dissimilar) pairs based on the teacher's estimation with ϕ_T.

4.4.2 Discussions on LKD

We analyze the effectiveness of KD by its gradient over the top-layer classifier $W \in \mathbb{R}^{d \times C}$. Without loss of generality, we take the gradient of $\mathbf{w}_c$ over *one single* instance $\mathbf{x}$ as an example, whose target label is c. Denote p_c and q_c as the c-th element in the student's and teacher's normalized prediction $\mathbf{softmax}(f(\mathbf{x}))$ and $\mathbf{softmax}(f_T(\mathbf{x}))$ (the posterior probability of the c-th class given the student's and the teacher's embedding of the instance), respectively. In vanilla learning scenario in Eq. 1 with objective $O_{\mathbf{ce}}$, the gradient w.r.t. $\mathbf{w}_c$ is

$$\frac{\partial O_{\mathbf{ce}}}{\partial \mathbf{w}_c} = \left[-p_c(1 - p_c) \right] \phi(\mathbf{x}) . \quad (12)$$
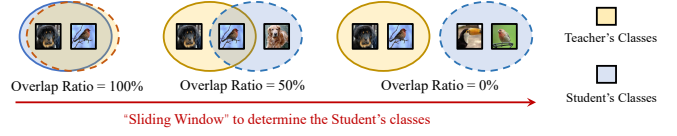

Figure 4: An illustration of the "sliding windows" to generate different configurations of teacher's and student's classes.

With KD in Eq. 2 (denote its objective as $O_{\mathbf{kd}}$), the gradient over the classifier $\mathbf{w}_c$ of the c-th class is:

$$\frac{\partial O_{\mathbf{kd}}}{\partial \mathbf{w}_c} = \sum_{\mathbf{x}} \left[-p_c + \sum_{c'=1}^{C} p_{c'} q_c \right] \phi(\mathbf{x}) . \quad (13)$$

When considering the soft supervision from the teacher in KD, not only the instance from the target class but also those from helpful related classes (the ones with large $p_{c'}$) will be incorporated to guide the update of the classifier. Since the summation in Eq. 13 is computed over all C classes, the normalized class posterior q_c becomes small if C is large, so that the helpful class instance will not be stressed obviously. Therefore, we consider a *local* version of the knowledge distillation term LKD in Eq. 8, where only the class set $\mathcal{S}$ in the current mini-batch are considered, *i.e.*, the influence of a helpful related class selected by the teacher will be better emphasized in the update of $\mathbf{w}_c$.

We empirically verify this claim on CIFAR-100. We compute the gradient difference between the vanilla Cross-Entropy (CE) loss and the KD variants w.r.t. all top-layer classifiers, and further use the averaged norm over randomly sampled instances to measure the additional supervision introduced with KD variants. The smaller the norm difference, the weaker the additional supervision signal provided by the teacher. Figure 3 plots the change of the norm difference when we increase the number of randomly sampled classes in a task from 2 to 100, and all the gradients are measured during the initial optimization of the model. We find when the number of classes becomes larger, the norm of gradient difference between vanilla KD loss and CE loss decreases faster than that between LKD loss and CE loss. Thus the supervision made by the vanilla KD teacher is weakened more than the supervision made by the LKD counterpart.

5 EXPERIMENTS

We verify REFILLED on a variety of classification tasks, namely GKD, standard KD, one-step incremental learning, and few/middle-shot learning. We analyze the results first, followed by ablation studies and visualizations in each part. The code is available at https://github.com/njulus/GKD.

5.1 Generalized Knowledge Distillation

REFILLED can distill knowledge from a general teacher no matter how its label space overlapped or not w.r.t. the target classes. We evaluate REFILLED for both cases.

5.1.1 Setups

Datasets. Caltech-UCSD Birds-200-2011 (CUB) [93] is a fine-grained dataset with 200 different species of birds. We split
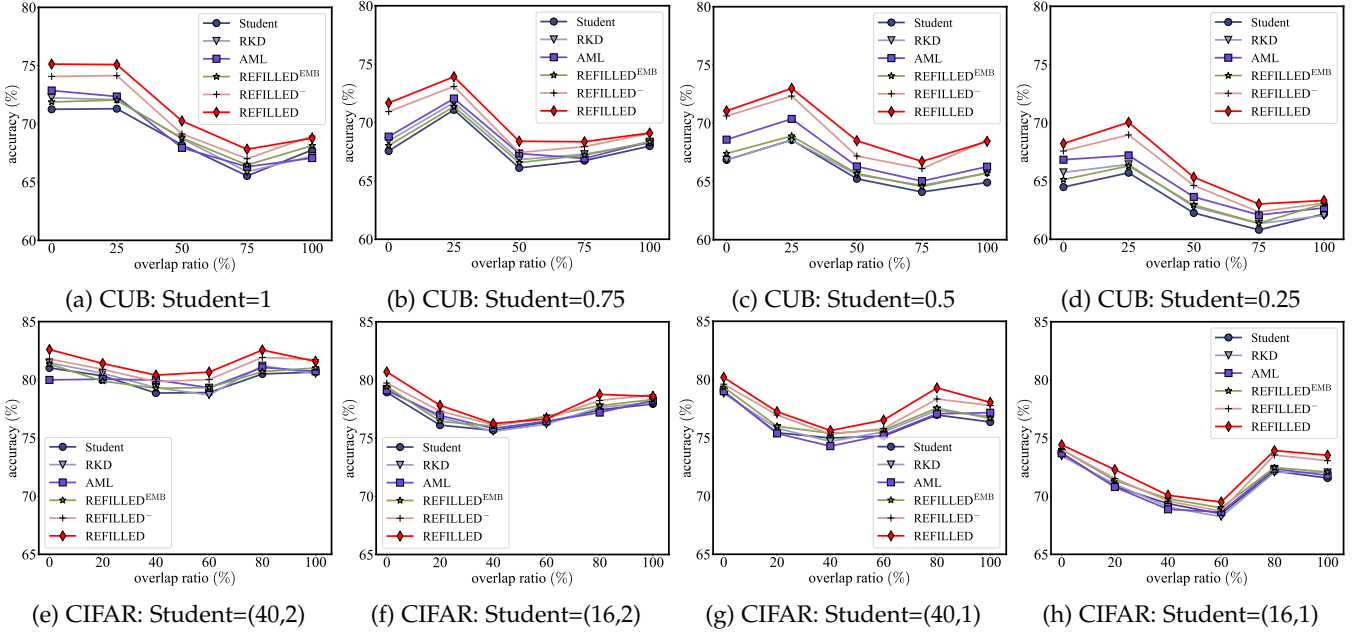
Figure 5: The mean accuracy on GKD tasks upon CUB (upper) and CIFAR-100 (lower). The overlap ratio of student's label space w.r.t. teacher's changes from 0% (cross-task KD) to 100% (standard KD). The architecture of teacher is MobileNet-1.0 and WRN-(40,2) for CUB and CIFAR, respectively. We vary the architecture of the student. REFILLED$^{\mathrm{EMB}}$ only distills the embedding from the teacher, and REFILLED^{-} does not utilizes the adaptive weights to select the helpful knowledge from the teacher.

two sets of the 100 classes as candidate class sets based on the given class indexes. Since classes in CUB are sorted in alphabetic order and classes with numerically close indexes are more similar, there is a relatively large semantic gap between these two sets. We use the first 100 classes to train the teacher. For the student, we change its target classes over 100 classes with a "sliding window" from one candidate set to another (illustrated in Figure 4). In other words, the student shares the same 100 classes with the teacher at first, and targets 100 non-overlapped classes at last. The student and the teacher have overlapped but not the same set of classes in the intermediate cases — we investigate the cases when the student has a class overlap ratio $\{0\%, 25\%, 50\%, 75\%, 100\%\}$ with the teacher. In each 100-way classification task, both teacher and student use randomly sampled 70% data in each class for training and the remaining instances for test. We do not use the attribute information of CUB. As a pre-processing, we crop all images based on the provided bounding boxes. We also consider **CIFAR-100** [94], which contains 100 classes with 600 32×32 images per class. In each class, there are 500 images for training and 100 images for test. We use a similar strategy as [95] to split two 50-class sets, and the overlap ratio of the student's label space w.r.t. teacher's changes from $\{0\%, 20\%, 40\%, 60\%, 80\%, 100\%\}$.

Evaluations. We use the classification accuracy over the student's 100 classes on CUB (50 classes on CIFAR-100) as the criterion and evaluate whether the learned teacher can successfully help the student *no matter how many classes they share*. The average value over three random trials is reported.

Implementation details. Through minimizing the cross-entropy objective as Eq. 1, we train a teacher model based on its corresponding training set with MobileNets [96] (width multiplier is 1.0) and Wide ResNets [97] (WRN, width is 2 and depth is 40) for CUB and CIFAR-100, respectively. We

use the Stochastic Gradient Descent (SGD) as the default optimizer, where the momentum is 0.9, batch-size is 256, maximum epoch number is 200, initial learning rate is 0.1, and we time the learning rate by 0.2 after 50 epochs. We hold out a part of examples from the training set for validation, from which the best set of hyper-parameters are selected. With the best-selected hyper-parameters, we re-train the teacher model on the whole training set. During the training, we use the random crop together with the horizontal flip as the data augmentation. This is the same when training the student. For CUB, we use different configurations of the MobileNets and adjust the model complexity with different width multipliers (complicated models have larger multipliers). While for CIFAR-100, we change the (depth, width) pair of the WRN. There are two stages for the student. In both stages, the temperature τ of the teacher's model is set to 2, and we do not smooth the logits of the student. When distilling the embedding, we set momentum to 0.9, batch-size to 256, maximum epoch to 200, initial learning rate to 0.1, and we time the learning rate by 0.2 after 50 epochs. While in the second stage, we use the same hyper-parameters. We tune λ from the hold out validation set. We find the best λ is close to 2 and the performance of REFILLED is not very sensitive to λ. Note that we construct an instance-specific weight (λ_i in Eq. 10) based on this λ value.

GKD baselines. There are three types of baselines.

- *Classification on the teacher's embedding.* We construct classifiers for the target classes based on teacher's embeddings ϕ_T *only when teacher and student share the same architecture.* Based on this, we apply the nearest neighbor (1NN) classifier and the linear logistic regression (LR). Besides, we fine-tune the teacher's model over instances in the student's split (FT). Since using a large learning rate will make the student obtain the same weights as training from

Width Multiplier	1	0.75	0.5	0.25
Teacher's EMB	1NN: 43.29, LR: 51.66, FT: 61.78			
Student	68.20	66.11	65.23	62.26
REFILLED^{EMB}	68.79	66.56	65.68	62.94
REFILLED^{LKD}	68.64	66.91	66.03	63.55
REFILLED$^-$	69.14	67.43	67.18	64.62
REFILLED	**70.25**	**68.39**	**68.50**	**65.53**

Table 1: Mean accuracy of student models on CUB. The overlap ratio of student's label space w.r.t. teacher's is fixed to 50%. The "Teacher's EMB" baseline only applies when it has the same architecture (same embedding dimension) with the student. We set teacher to MobileNet-1.0 and vary the width multiplier of the student in $\{1, 0.75, 0.5, 0.25\}$.

(depth, width)	(40, 2)	(16, 2)	(40, 1)	(16, 1)
Teacher's EMB	1NN: 64.27, LR: 67.94, FT: 71.13			
Student	78.90	76.37	75.14	68.48
REFILLED^{EMB}	79.33	76.90	75.67	69.00
REFILLED^{LKD}	79.47	76.29	75.30	68.14
REFILLED$^-$	80.02	76.66	75.79	68.72
REFILLED	**80.66**	**76.66**	**76.52**	**69.50**

Table 2: Mean accuracy of student on CIFAR-100. The class overlap ratio between student and teacher is 60%. We set teacher to WRN-(40,2) and vary the (depth, width) of the student.

scratch, we use a small initial learning rate (0.0001) and a fixed number of epochs (50) in our experiments.

- *Variants of cross-task KD*. We compare our method with recent representative embedding-based KD approaches, *i.e.*, the Relational Knowledge Distillation (RKD) [82] and Asymmetric Metric Learning (AML) [73]. We then fine-tune the whole student model with its distilled embedding. Hyper-parameters are tuned in the same way as REFILLED.
- *Variants of* REFILLED. We investigate the importance of different components in REFILLED. We consider fine-tuning the model with cross-entropy based on the embedding distilled by comparison matching in Eq. 5, which is denoted as "REFILLED^{EMB}". "REFILLED^{LKD}" means we train the student with cross-entropy and LKD from scratch without using the distilled embedding. "REFILLED$^-$" denotes the REFILLED variant without instance-adaptive weights λ_i.

5.1.2 Results and Analyses on GKD

The results of GKD in Figure 5 include standard KD (overlap ratio=100%), cross-task KD (overlap ratio=0%), and other general cases. The test accuracy of the student becomes higher when learning the task with more complicated models. Points in Figure 5 denote fully different target datasets since the student possesses diverse subsets of 100 and 50 classes in CUB and CIFAR-100, respectively. The accuracy of the student indicates the difficulty of the target task, where the student gets low accuracy when the overlap ratio nears 75% on CUB and 50% on CIFAR.

In both CUB and CIFAR, we find that the embedding-based distillation methods such as RKD [82] and AML [73] improve over the vanilla training denoted as "student". Our REFILLED greatly improves student's performance, outperforming RKD and AML *no matter how the student's label space and architecture change w.r.t. the teacher's*. Specifically, our re-weighted version REFILLED helps more w.r.t. REFILLED$^-$ without adaptive weights when the overlap ratio is low (especially in cross-task KD). When student and teacher share the same classes in standard KD, REFILLED achieves little improvements than REFILLED$^-$. Detailed results of various methods in Figure 5 are reported in the supplementary.

Next, we investigate the GKD configurations when nearly half of the student's labels are overlapped with the teacher (*overlap ratio equals 50% for CUB and 60% for CIFAR*). **Will all components in REFILLED help?** Given the well-trained teacher, we investigate three variants in Table 1

and Table 2 besides training the student model directly (denoted as "student"). We find REFILLED and its variants improve a lot over the "student", which verifies the effectiveness of each component. In particular, by comparing REFILLED^{EMB} and "Student", we find REFILLED learns more discriminative embeddings after the first stage, fine-tuning upon which leads to better "downstream" classification results. REFILLED^{LKD} applies our LKD together with vanilla objective, whose results indicate LKD itself acts as a useful distillation strategy. The improvements between REFILLED$^-$ over REFILLED^{EMB}/REFILLED^{LKD} show that our comparison matching and LKD facilitate knowledge transfer in an orthogonal manner. The best performance of REFILLED validates that if we adjust the influence of the teacher for different instances during the GKD, the helpful supervision from the teacher will direct the student to generalize better. **Will adaptive weights λ_i differentiate seen and unseen classes?** As demonstrated, the instance-specific weights λ_i in Eq. 9 is a key component for GKD. Since there simultaneously exist cross-task instances (denoted as unseen) and same-task instances (denoted as seen) in the student's training set, the teacher may predict with different confidences to them, *e.g.*, higher confidence over those seen instances where the teacher is trained from. We check whether the adaptive λ_i can differentiate seen and unseen classes. Figure 6a illustrates the distribution of λ_i on CIFAR-100. where we set λ to 1 and $\lambda_i \in (0, 1]$. We find weights for instances belonging to the seen classes are observably larger than those for instances belonging to unseen classes. This means our re-weight strategy can successfully recognize seen/unseen classes and put larger weights on those familiar instances automatically. In addition, we also compute the AUC when we use λ_i to differentiate seen and unseen classes in the GKD scenario, whose value is 0.959. **Other re-weight strategies of λ_i.** In Eq. 9, REFILLED re-weights the distillation term with higher weights on a target task instance if the teacher's prediction is close to its Pseudo Label (denoted as "PL-Weight"). There are some other re-weight strategies [98]. For example, [99] constructs a probabilistic model and utilizes another branch to estimate the variance in distillation, which requires more parameters. We also investigate another re-weight implementation inspired from [99], which incorporates both student's prediction $p_S(s|\mathbf{x})$ and teacher's prediction $p_T(s|\mathbf{x})$ into account without additional branches. In detail, we set λ_i based on the divergence between two predictions, *i.e.*,

$$\lambda_i = 2\lambda \times \sigma \left(-\mathbf{KL}\left(\{p_T(s|\mathbf{x})\}_{s \in \mathcal{S}}, \{p_S(s|\mathbf{x})\}_{s \in \mathcal{S}}\right)\right),$$

(a) Weight distribution.
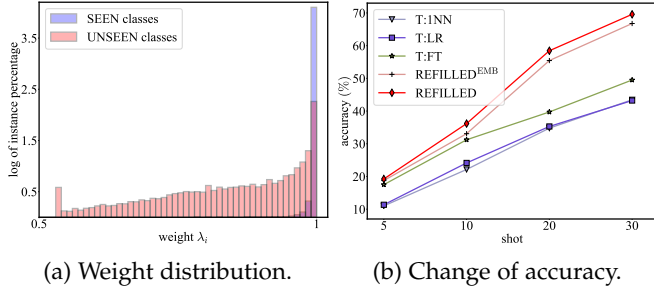
(b) Change of accuracy.

Figure 6: (a) Teacher's weight distribution of instances belonging to both seen classes and unseen classes. For simplicity, λ is set to 1 and $\lambda_i \in (0, 1]$. The weights of instances from seen classes are observably higher than those of unseen classes, and it is easy to recognize unseen classes. (b) The change of accuracy when the number of instances per class (shot) varies. The overlap ratio of student's labels space w.r.t. teacher's is set to 0%. The width multiplier of student is set to 1. T:1NN means the nearest neighbour classifier based on teacher's embedding network. T:LR means the logistic regression classifier trained on teacher's embedding network. T:FT means fine-tuning teacher's embedding network together with a linear classifier.

width multiplier	1	0.75	0.5	0.25
REFILLED$^-$	69.14	67.43	67.18	64.62
Gap Weight	70.13	68.05	68.13	65.47
PL Weight	**70.25**	**68.39**	**68.50**	**65.53**

Table 3: Mean accuracy of student models on CUB. The overlap ratio of student's label space w.r.t. teacher's is fixed to 50%.

A larger gap indicates that an instance is more dissimilar to the target task, and this strategy will output smaller weight λ_i. We denote the manner setting λ_i above as "Gap-Weight", and compare with our previous "PL-Weight" in Table 3 and Table 4 on CUB and CIFAR, respectively. We find both strategies improve w.r.t. the vanilla version REFILLED$^-$ without additional λ_i, and PL-weight gets slightly better results in most cases.

One-stage vs. two-stage learning. We train REFILLED in a two-stage manner in Alg. 1 as [5], [16], where objectives in two stages could be trained in a joint way. REFILLED$^\gamma$ means training REFILLED in a one-stage manner with additional balancing hyper-parameter γ. From the model design perspective, the two-stage training in REFILLED works well since the distilled discriminative embedding acts as a better initialization hence *improves the discerning ability* of the model. While training with a combined objective *regularizes* the classifier by matching the predictions between student and teacher, which relies on a suitable regularization strength. From the implementation perspective, an important issue for the joint training of the combined objective is to *set the right balance* among the embedding learning (relationship distillation), classification (cross-entropy), and knowledge transition (LKD) losses. In our empirical study, we tune γ on validation set but it is a bit hard to find the optimal balance. In the two-stage training strategy, we can first learn a good embedding till convergence, and then use such embedding to initialize the second stage, where the balance between classification and distillation is solved with an annealing

(depth, width)	(40, 2)	(16, 2)	(40, 1)	(16, 1)
REFILLED$^-$	80.02	76.66	75.79	68.72
Gap Weight	80.53	76.71	76.38	69.61
PL Weight	80.66	76.66	76.52	69.50

Table 4: Mean accuracy of student models on CIFAR-100. The overlap ratio of student's label space w.r.t. teacher's is 60%.

Overlap Ratio = 50%	1	0.75	0.5	0.25
REFILLED	**75.13**	**71.67**	**71.06**	**68.22**
REFILLED$^\gamma$	74.25	70.38	69.94	67.52
Overlap Ratio = 0%	1	0.75	0.5	0.25
REFILLED	**75.13**	**71.67**	**71.06**	**68.22**
REFILLED$^\gamma$	74.25	70.38	69.94	67.52

Table 5: Mean accuracy on CUB using one-stage learning with additional balance hyper-parameter γ (denoted as REFILLED$^\gamma$) and the proposed two-stage learning. The overlap ratio of student's label space w.r.t. teacher's is set to 50% (upper) and 0% (lower) for GKD and cross-task KD, respectively.

strategy. From the results in Table 5, the two-stage training makes REFILLED easier to achieve higher performance.

5.1.3 Results and Analyses on Cross-Task KD

We further consider a more difficult scenario where the student target fully non-overlapped classes w.r.t. the teacher (*overlap ratio equals* 0%).

Results of REFILLED on more teacher's architectures. In previous experiments, we fix the teacher's architecture and vary the complexity of student's model. We also investigate how REFILLED influences the student's discriminative ability when we change the teacher's architecture. We consider GKD on two different sets of 100 classes on CUB, and we set both teacher and student as MobileNet but with different depths. We use a weaker teacher, which has multiplier width 0.75, and we investigate whether such as weaker teacher can improve the student. The results are in Table 6. We find REFILLED improves the classification ability of a student via a cross-task teacher even the teacher's complexity is weaker than the student. The phenomenon is consistent with [100], and we attribute the improvement to the usage of relationship comparisons when training the target student.

Moreover, we set teacher and student to different neural network families on CIFAR-100, *i.e.*, teacher is WRN-(40, 2) and the student is ResNet [101]. We vary the depth (layer number) of the student's ResNet model in $\{50, 32, 14\}$. The results are listed in Table 7. Similarly, REFILLED facilitates the cross-task distillation across different model families, which validates the general knowledge transfer ability of REFILLED.

Will adaptive weights λ_i identify helpful instances? When the overlap ratio of student's label space w.r.t. to teacher's is 0%, all classes in student's training set are unseen to the teacher. Under this situation, which instances will be assigned highest confidence weights? The larger the λ_i, the more confident teacher is on i-th instance. In Figure 7, we can see that new images that are visually similar to the teacher's confident images in seen classes have higher weights λ_i.

Width Multiplier	1	0.75	0.5	0.25
Student	71.25	67.56	66.85	64.48
RKD [82]	70.94	67.95	67.32	65.03
AML [73]	71.32	68.29	67.85	65.55
REFILLED$^{\text{EMB}}$	69.41	68.82	67.11	64.89
REFILLED$^-$	72.58	68.35	68.72	65.46
REFILLED	**73.29**	**69.32**	**69.04**	**66.04**

Table 6: Mean accuracy of student on CUB. The class overlap ratio between student and teacher is 0%. We set teacher as MobileNet-0.75 and vary the width multiplier of the student.

depth	50	32	14
Student	75.46	76.72	73.54
RKD [82]	76.95	76.91	73.88
AML [73]	76.42	76.21	73.89
REFILLED$^{\text{EMB}}$	76.82	77.15	73.90
REFILLED$^-$	77.05	77.41	74.11
REFILLED	**77.43**	**77.92**	**74.35**

Table 7: Mean accuracy of student on CIFAR-100. The overlap ratio of student's label space w.r.t. teacher's is fixed to 0%. We set teacher as a WRN-(40, 2), while varying the depth of the student's ResNet model. REFILLED achieves the best performance among others.

This means our proposed adaptive weights successfully find useful instances to transfer knowledge even though all classes are new.

REFILLED with different sizes of target task data. To test the extreme of the distillation ability of REFILLED, we construct the target classification task with different sizes of training data on CUB. When the number of available training data is small, it is more difficult to train the student model, so that the help from the teacher becomes more important. We vary the number of instances per class (shot) in the student's task from 5 to 30, and the averaged classification accuracies are shown in Figure 6b. When the shot number increases, the student performs better since there are more training data. REFILLED keeps a performance margin with comparison methods in all cases. We find the gap between REFILLED and the vanilla methods becomes larger when there are more shots, which indicates the distillation variants become powerful given more training data. More results on learning with limited target class data could be found in Section 5.4.

Knowledge transfer across distant tasks. At the end of this subsection, we analyze the phenomenon of REFILLED when it transfers the knowledge across distant tasks which are not too related. Particularly, we train the teacher on all 200 classes on CUB for fine-grained birds, and set the target task as a 120-way fine-grained dog classification over the Stanford Dogs Dataset [102]. We follow the standard splits of both datasets. We compare our method with a self-supervised learning method SEED [86], which distills the learned representation from previous generations. Mean accuracy results are listed in Table 8. In this case, REFILLED is hard to improve based
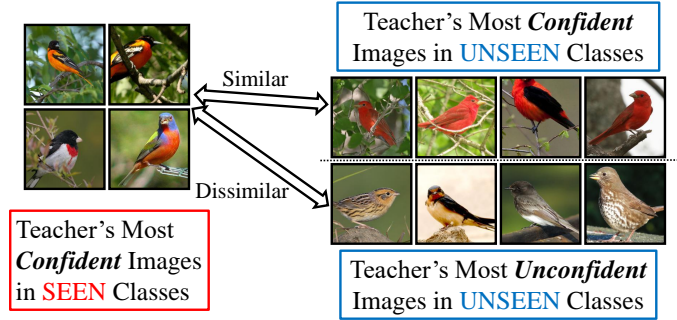

Figure 7: Visualization of teacher's most confident images (largest λ_i values) in SEEN classes along with teacher's most confident/unconfident images in UNSEEN classes. Although classes in the student's training set are unseen to the teacher, our instance-specific weighting strategy assigns higher weights to those images similar to teacher's confident images.

Width Multiplier	1	0.75	0.5	0.25
Student	72.35	70.69	70.11	68.57
SEED [86]	72.78	70.90	70.55	68.32
REFILLED	72.27	70.81	70.35	68.64

Table 8: Mean accuracy of student models on Dogs dataset. The teacher is trained on 200 classes from CUB. We set teacher as MobileNet-1.0 and vary the width multiplier of the student.

on the vanilla student, since the knowledge of the teacher to compare objects (about fine-grained birds) may not fit the target task (about fine-grained dogs). Since the comparison ability learned from data augmentations in SEED does not depend on classes, it transfers slightly better to a distant task than the comparison ability learned from class semantics in REFILLED. But in this case, REFILLED can still get nearly the same performance as the vanilla student. So although the teacher is not an expert on the target task, its experience will not negatively affect the training of the student.

5.2 Standard Knowledge Distillation

The techniques for GKD in REFILLED also facilitate standard KD where a student has the same classes with the teacher.

Datasets. Following [16], we test the KD ability of REFILLED on CIFAR-100 and CUB. All classes in these datasets are used during training based on the standard training-test split.

Implementation details. Three different families of the neural networks are used to test the ability of REFILLED, namely the ResNets [101] , Wide ResNets (WRN) [97], and MobileNets [96]. Towards getting different capacities of the model, we change the depth of the ResNet (through the number of layers), the (depth, width) pair of the Wide ResNet, and the width of the MobileNets (through the width multipliers). We use similar ways in GKD to train both the teacher and the student model for standard KD. We set the temperature of the teacher's model to 4 and $\lambda = 1$. For CIFAR-100, we pad 4 for each edge before the random crop.

Evaluations. Both teacher and student are trained on the same set with three different seeds of initialization, and we report the mean accuracy of the student on the test set. Two protocols are used in standard KD, where the teacher and the student come from the same or different model families.

(depth, width)	(40, 2)	(16, 2)	(40, 1)	(16, 1)
Teacher	74.44			
Teacher+BAN [7]	75.41			
Student	74.44	70.15	68.97	65.44
KD [3]	75.47	71.87	70.46	66.54
FitNet [5]	74.29	70.89	68.66	65.38
AT [46]	74.76	71.06	69.85	65.31
NST [103]	74.81	71.19	68.00	64.95
VID-I [16]	75.25	73.31	71.51	66.32
KD+VID-I [16]	76.11	73.69	72.16	67.19
SEED [86]	76.28	73.40	71.83	67.75
RKD [82]	76.62	72.56	72.18	65.22
LKD† [50]	-	75.44	-	67.72
SSKD‡ [85]	75.42	74.03	**72.71**	67.30
REFILLED	**77.90**	**75.71**	72.54	**68.13**

Table 9: The average classification results of knowledge distillation methods on CIFAR-100 based on the Wide ResNet. We fix the teacher with (depth, width) = $(40, 2)$, and set the student's capacity with different (depth, width) values. † The reported results of LKD in [50] is based on a stronger teacher with accuracy 75.61%. ‡ We apply SSKD for embedding distillation followed by our classifier distillation step.

- *Same-family knowledge distillation*. Both the teacher and the student come from the same model family. We use the same configuration as [16]. In CIFAR-100, both the teacher and the student are Wide ResNets. We set the (depth, width) pair of the teacher as $(40, 2)$, and change such configuration parameters of the student model among $(40, 2)$, $(16, 2)$, $(40, 1)$, and $(16, 1)$. On CUB, we consider the MobileNets, by setting the teacher's width multiplier to 1, we vary the width multipliers of the student among $\{1, 0.75, 0.5, 0.25\}$.
- *Different-family knowledge distillation*. The teacher and the student come from different architectural families, where the knowledge transfers from ResNets to MobileNets. Taking the computational burden into consideration, when in CIFAR-100, we choose the teacher as the ResNet-110, and we use ResNet-34 as the teacher in CUB. We only change the width multipliers of the student model in $\{0.75, 0.5, 0.25\}$ on CUB to keep the student model having a smaller capacity when compared with the teacher.

5.2.1 Distillation From Same Architecture Family Models

We first investigate the distillation ability when teacher and student come from the same model family. The results on CIFAR-100 and CUB are in Table 9 and Table 10, respectively. On CIFAR-100 we exactly follow the evaluation protocol in [16], which implements teacher and student with the Wide ResNet. We re-implement RKD [82] and cite the results of other comparison methods from [16], [50]. For CUB, we use MobileNets as the basic model. Since the teacher possesses more capacity, its learning experience assists the training of the student once utilizing the knowledge distillation methods. REFILLED achieves the best performance in almost all settings, which verifies transferring knowledge for both embedding and classifier is one of the key factors for KD.

Width Multiplier	1	0.75	0.5	0.25
Teacher	75.36			
Teacher+BAN [7]	76.87			
Student	75.36	74.87	72.41	69.72
KD [3]	77.61	76.02	74.24	72.03
FitNet [5]	75.10	75.03	72.17	69.09
AT [46]	76.22	76.10	73.70	70.74
NST [103]	76.91	77.05	74.03	71.54
KD+VID-I [16]	77.03	76.91	75.62	72.23
RKD [82]	77.72	76.80	74.99	72.55
REFILLED	**79.33**	**78.52**	**76.90**	**74.04**

Table 10: The average classification results of knowledge distillation methods on CUB based on MobileNets. We fix the teacher's width multiplier to 1.0, and change the student's multipliers.

(depth, width)	(40, 2)	(16, 2)	(40, 1)	(16, 1)
w/o CM	55.47	50.14	45.04	38.06
SEED† [86]	61.32	52.57	51.24	43.50
SSKD [85]	61.25	**54.77**	52.22	43.90
w/ CM	**62.12**	53.86	**52.71**	**44.33**

Table 11: The nearest class mean accuracy on CIFAR-100 to evaluate the embedding quality before and after the Comparison Matching (CM) step, the first stage, in REFILLED. † We transform the distillation strategy of SEED [86] to a supervised version using class semantics, and implement the embedding distillation over all instances in a mini-batches.

Will embedding distillation help? There are two stages in REFILLED. In the first stage we distill the discriminative embedding from the teacher and improve the comparison ability of the student. We evaluate the quality of the embedding via its classification accuracy based on Nearest Class Mean classifier (NCM). In detail, we extract features on instances from the student's training set, and then compute the center for each class. Finally, we predict a test instance based on the label of its nearest class in the embedding space. In Table 11, we compute NCM accuracy for student model's embedding trained with and without aligning the teacher's tuples (denoted as comparison matching, CM) in CIFAR-100. We compare our comparison matching with the relationship distillation strategy SSKD in [85], where self-distillation strategy is incorporated. We also transform the relationship distillation in SEED [86] to a supervised version, which determines the similarity between instances through their class labels. Table 9 contains the classification results when we distill the teacher with our LKD based on the embedding learned by SSKD, SEED, and REFILLED. Figure 8 visualizes the embedding quality over 10 sampled classes using tSNE [70]. Both quantitative and qualitative results validate that the quality of the student's embedding is improved after distilling the knowledge from the teacher. Thus the comparison matching step in REFILLED is effective for knowledge distillation. The results also verify using the NCM predictions to direct the classifier training in the 2nd stage of REFILLED is reasonable.

Will local knowledge distillation help? Results in Table 12

(depth, width)	(40, 2)	(16, 2)	(40, 1)	(16, 1)
w/ KD	77.08	73.57	72.24	67.14
w/ LKD	**77.90**	**74.82**	**72.54**	**68.13**

Table 12: The mean accuracy on CIFAR-100 to evaluate the effectiveness of Local Knowledge Distillation (LKD) in REFILLED.
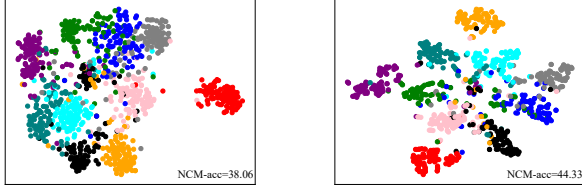

Figure 8: The tSNE [70] of the vanilla student training (left) and the improved embedding after the 1st stage of REFILLED (right) over 10 classes sampled from CIFAR-100.

verify the further improvement of Local Knowledge Distillation (LKD) in Eq. 8 compared with the vanilla Knowledge Distillation (KD) when training based on the distilled embedding after the first stage of REFILLED. A local consideration of probability matching helps.

5.2.2 Distillation From Different Model Families

To further evaluate the performance of REFILLED, we use REFILLED in standard KD but with a teacher with cross-family architecture. For CIFAR-100, we set the teacher as ResNet-110, and use the MobileNets with different channels as the student. For CUB, we set the teacher as ResNet-34, and use the MobileNets with different width multipliers (from $\{0.75, 0.5, 0.25\}$) as the student. Results on CIFAR-100 and CUB are reported in Table 13 and Table 14, respectively. REFILLED keeps its superiority in all cases, which indicates its practical utility with different teacher's configurations.

5.3 KD for One-Step Incremental Learning

We claim that the distillation ability in REFILLED acts as a better way to prevent catastrophic forgetting [52], and facilitates to augment the discerning ability in the one-step incremental learning environment. In other words, the student not only distills the dark knowledge in f_T to improve its classification ability in the target C classes but also augments the classifier to discern those $C' = |\mathcal{C}'|$ classes from the teacher's task. Finally, f becomes a *joint classifier* over $\mathcal{C} \cup \mathcal{C}'$ with $C + C'$ classes. The student's classifier W is augmented with $\hat{W} \in \mathbb{R}^{d \times C'}$ for classes in $\mathcal{C}'$ simultaneously. Note that different from the vanilla incremental learning, REFILLED calibrates both old and target classes well and can be applied *across different architectures*.

Datasets. Following [16], we test REFILLED on CUB, based on the scenario of the cross-task KD in section 5.1 (there is no class shared between teacher and student).

Evaluations. We first consider the classification accuracy over all 200 classes, which encodes the 200-way classification for instances from both old and current 100 classes. The same number of instances from old and new classes are

Width Multiplier	1	0.75	0.5	0.25
Student	68.57	67.92	65.66	60.87
KD [3]	70.34	68.21	66.06	61.38
FitNet [5]	67.99	67.85	65.12	61.01
AT [46]	68.97	67.88	66.44	62.15
NST [103]	70.62	70.49	69.15	61.32
KD+VID-I [16]	71.94	70.13	68.51	62.50
RKD [82]	70.41	68.93	66.24	61.44
REFILLED	**73.75**	**72.65**	**70.32**	**62.90**

Table 13: The average classification accuracy of standard KD on CIFAR-100. The teacher is trained with ResNet-110, which gets 74.09% test accuracy. The student is learned with MobileNets, whose width multiplier is changed.

width multiplier	0.75	0.5	0.25
Student	74.87	72.41	69.72
KD [3]	76.02	74.17	71.97
FitNet [5]	75.03	72.17	70.03
AT [46]	76.11	72.94	70.99
NST [103]	75.89	73.82	71.92
KD+VID-I [16]	76.41	74.04	72.20
RKD [82]	76.11	75.24	72.84
REFILLED	**78.40**	**76.52**	**73.44**

Table 14: The mean classification accuracy of standard KD on CUB. Teacher is trained with ResNet-34, which gets 75.31% test accuracy. Student is learned with MobileNets, whose width multiplier is changed.

used to evaluate the model. To avoid a biased accuracy towards a certain split, we follow [105], [106] and compute the *harmonic mean* value of the two types of mean accuracy, *i.e.*, the mean accuracy for instances from classes in teacher's and student's tasks (denoted as "$\mathcal{C}' \rightarrow \mathcal{C} \cup \mathcal{C}'$" and "$\mathcal{C} \rightarrow \mathcal{C} \cup \mathcal{C}'$", respectively). A model has a high harmonic mean if it performs well on both first and second splits of all classes. Detailed formulations are in the supplementary.

Implementation details. We follow [107], [108] to discard the learned bias and ℓ_2-normalize the learned weights of all 200 classes, which leads to a better calibrated joint classifier for compared and our methods.

Comparison methods.

- *Combined classifier with teacher*. We concatenate the target class classifier with the teacher's classifier and get a joint one for all classes. We normalize classifiers for LR and FT.
- *Variants of standard KD*. We match the student's predictions on the old classes with the teacher's classifier [52] (denoted as LwoF) to avoid catastrophic forgetting. We finally concatenate the learned student's classifier with the teacher's one and normalize them for better calibration.
- *Incremental learning methods*. Since in our one-step incremental learning task, only a fixed model is provided, so we compare with incremental learning method EWC [104] using the fixed model only *without using a small number of examples from the previous task for experience replay* [9], [43].

Results. The results are in Table 15, and the student is

Criterion	Accuracy $\mathcal{C}' \to \mathcal{C} \cup \mathcal{C}'$				Accuracy $\mathcal{C} \to \mathcal{C} \cup \mathcal{C}'$				Accuracy				Harmonic Mean			
Width	1	0.75	0.5	0.25	1	0.75	0.5	0.25	1	0.75	0.5	0.25	1	0.75	0.5	0.25
Teacher	LR: 63.73, FT: 49.85				LR: 31.97, FT: 45.70				LR: 47.85, FT: 47.77				LR: 42.58, FT: 47.68			
EWC [104]	**35.66**				55.65				45.66				43.47			
Student + LwoF [52]	33.79	33.55	29.39	21.47	56.84	56.62	55.76	53.46	45.32	45.09	42.58	37.47	42.39	42.14	38.50	30.64
RKD [82] + LwoF [52]	34.30	34.02	30.21	21.98	57.27	57.63	55.90	53.81	45.79	45.83	43.06	37.90	42.91	42.79	39.23	31.22
REFILLED	35.50	**35.13**	**31.05**	**24.51**	**58.57**	**58.70**	**58.00**	**53.98**	**47.04**	**46.92**	**44.53**	**39.25**	**44.21**	**43.96**	**40.45**	**33.72**

Table 15: One-step incremental learning on CUB, where we follow the same setting as the GKD (overlap ratio equals 0%). There are four criteria: 1. the mean accuracy over instances from teacher's old classes $\mathcal{C}' \to \mathcal{C} \cup \mathcal{C}'$, 2. the mean accuracy over instances from target classes $\mathcal{C} \to \mathcal{C} \cup \mathcal{C}'$, 3. the mean accuracy over all classes, 4. the harmonic mean value of the mean accuracy over instances from teacher's and student's classes. The two baselines of the teacher denote we concatenate the teacher's classifier with the LR and fine-tuned (FT) model trained over the student's instances, given teacher and student *have the same architecture*. EWC only works for a student network that shares the same architecture as the teacher (width=1).

Method	Vanilla	REFILLED$^{\text{EMB}}$	REFILLED$^-$	REFILLED
Acc.	45.32	45.87	46.39	**47.04**
HM Acc.	42.39	42.90	43.45	**44.21**

Table 16: The mean accuracy of GKD on CUB dataset. Two criteria, *i.e.*, accuracy and Harmonic Mean (HM) accuracy are shown. We fix the width multiplier of the student to 1.

required to make a holistic classification on all classes, *i.e.*, the union of teacher's and student's class sets. Four criteria are utilized to evaluate a GKD model. Since the accuracy is computed over all classes, a model may predict target class instances well and forget the knowledge the teacher introduced at the initial KD stage. The harmonic mean reveals the joint ability of the classifier over both previous and target classes, and a model achieves a high value only if it predicts both sets of classes well. GKD improves the classification ability of the model than using the teacher's embeddings directly. Similar to LwoF [52], we apply a KD term on all old classes to prevent forgetting during the training progress of the vanilla student model and RKD. With LwoF, the student and RKD can handle all classes. We find comparison methods could not balance the predictions of old and target classes. Our REFILLED calibrates the two sets of classes better and gets the best performance w.r.t. both the accuracy and harmonic mean criteria. The margin with other methods becomes larger when we distill the knowledge to a much smaller student. The results validate that the proposed REFILLED expands the ability of the student model for more classes effectively.

Does REFILLED calibrate better on GKD? Similar to Table 1, we evaluate the GKD performance of REFILLED variants in Table 16. We keep both teacher and student have the same architecture. We normalize the classifier for all comparison methods. For comparison baselines, we concatenate the student's model with the teacher's classifier. For "vanilla", the student model is trained from scratch; for "REFILLED$^{\text{EMB}}$", the student model is trained over the embedding learned by REFILLED; while for "REFILLED$^-$", we tune the student model with the local knowledge distillation term without instance-specific weighting. We find REFILLED gets the best results in both accuracy and harmonic mean accuracy.

5.4 Few-Shot and Middle-Shot Learning

Training a deep neural network with limited data is a challenging task, where models are prone to over-fit. We apply our REFILLED approach for few-shot and middle-shot learning, where the classification ability from a teacher trained on SEEN class can be used to help the student model training for UNSEEN few-shot and middle-shot tasks.

Datasets. We use *Mini*ImageNet dataset [55] with 100 classes in total and 600 images per class. All images are resized to 84×84 before inputting into the models. Following [55], [109], there are 64 classes (SEEN class) to train the teacher (a.k.a. the meta-train set), 16 classes for validation (a.k.a. the meta-val set), and we sample tasks from the remaining 20 classes (a.k.a. the meta-test set) to train the student.

Implementation details. We set the student as a 4-layer ConvNet [55], [56], [57], and consider two types of the teacher model, *i.e.*, the same 4-layer ConvNet (but trained on different classes in the meta-train set) and the ResNet [95], [59]. The ConvNet contains 4 identical blocks, and each block is a sequential of convolution operator, batch normalization [110], ReLU, and Max pooling. We add another global max-pooling layer to reduce the computational burden after the 4 blocks, which gives rise to a 64-dimensional embedding before the top-layer classifier. ResNet removes the two down-sampling layers in the vanilla version [95], [59], and outputs 640-dimension embeddings. We train a teacher on SEEN classes set with ResNet/ConvNet. Supervised by the cross-entropy loss, we use random crop and horizontal flip as the data augmentation, SGD w/ momentum 0.9 as the optimizer, and 128 as the batch size. The student is trained with the help of the teacher and limited UNSEEN class examples.

Evaluations. Define a M-shot C-way task as a C-class classification problem with M instances in each class. Different from the few-shot learning where $C = 5$ and $K \in \{1, 5\}$, here we consider there are a bit more instances in each class, *i.e.*, $K = \{10, 30\}$. Although the value of M increases in middle-shot learning, it is still small to train a complicated neural network from scratch. We sample tasks from the 20-class split (meta-test set) to train the student and evaluate the classification accuracy over another 15 instances from each of the C classes. Mean accuracy over 10,000 trials are reported. We omit the confidence interval since they all have similar values around 0.2%.

Comparison methods. There are two branches of baselines:

M-Shot	1	5	10	30
1NN	49.73	63.11	66.56	69.80
SVM	51.61	69.17	74.24	77.87
Fine-Tune	45.89	68.61	74.95	78.62
SimpleShot [111]	49.95	69.62	74.48	78.43
MAML [57]	48.70	63.11	-	-
ProtoNet [56]	51.79	70.38	74.42	78.10
Neg-Cosine [56]	52.84	70.41	-	-
FEAT [59]	55.15	71.61	74.86	78.84
REFILLED$^{\text{Res}}$	**55.13**	**72.05**	**76.93**	**80.75**
REFILLED$^{\text{Conv}}$	53.81	71.86	75.56	78.98

Table 17: The mean accuracy over 10,000 trials of 5-Way M-shot tasks from few-shot ($K \in \{1, 5\}$) to middle-shot ($K \in \{10, 30\}$). We set student as the ConvNet, and investigate both ResNet and ConvNet as the teacher. REFILLED$^{\text{Res}}$ and REFILLED$^{\text{Conv}}$ denote the results with a ResNet and a ConvNet teacher, respectively.

- *Meta-learning methods*. Meta-learning mimics the test case by sampling C-Way M-Shot tasks from the SEEN class set to learn task-level inductive bias like embedding [55], [56]. However, the computational burden (*e.g.*, the batch size) sampling episodes of tasks becomes large when the number of shots increases. Besides, meta-learning needs to specify the way to obtain a meta-model from the SEEN classes. We compare REFILLED with the embedding-based meta-learning approaches ProtoNet [56] and FEAT [59].

- *Embedding-based baselines*. We can make predictions directly with the teacher's embedding, the penultimate layer of the teacher model, by leveraging the nearest neighbor classifier. Based on which, we also train linear classifiers like SVM on the current task's data or fine-tune the whole model. It is notable that we tune the hyper-parameters with sampled few/middle-shot tasks on the validation split. We compare REFILLED with SimpleShot [111] and Neg-Cosine [112].

Results. The results of 5-way M-shot classification are reported in Table 17. Both ProtoNet and FEAT are meta-learned over the pre-trained embeddings (the ConvNet teacher) from the SEEN class set (meta-train set). When $K = \{1, 5\}$ as in the standard few-shot learning setting, the meta-learning approaches perform well than the embedding-based baselines, but fine-tuning becomes a very strong baseline when the number of shots becomes large, which gets better results than ProtoNet or FEAT. REFILLED gets better results when distilling the knowledge from a stronger teacher (*i.e.*, the ResNet), which demonstrates the influence of teacher's capacity. REFILLED achieves competitive results when M is small, and gets better results than other methods with large M, which validates the importance of distilling the knowledge of a cross-task teacher for training a classifier.

6 CONCLUSION

Although knowledge distillation makes it easier to transfer learning experiences between related heterogeneous models, reusing models learned from a *general* label space is still difficult. We propose generalized knowledge distillation where the student is not restricted to having identical classes with the teacher. Our REFILLED improves the learning efficiency of the target model with the help of two stages, *i.e.*, comparison matching and adaptive local knowledge distillation. REFILLED aligns the comparison ability w.r.t. embeddings, removing the label space constraint while simultaneously capturing high order relationships among instances. Then, emphasizing the teacher's confident supervision makes REFILLED automatically match the predictions between two models locally. Experiments validate that REFILLED improves classification performance in a variety of tasks, including general and standard knowledge distillation.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[2] C. Bucila, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *KDD*, 2006, pp. 535–541.

[3] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015.

[4] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *CVPR*, 2017, pp. 7130–7138.

[5] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *ICLR*, 2015.

[6] L. Yu, V. O. Yazici, X. Liu, J. van de Weijer, Y. Cheng, and A. Ramisa, "Learning metrics from teachers: Compact networks for image embedding," in *CVPR*, 2019, pp. 2907–2916.

[7] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born-again neural networks," in *ICML*, 2018, pp. 1602–1611.

[8] C. Yang, L. Xie, C. Su, and A. L. Yuille, "Snapshot distillation: Teacher-student optimization in one generation," in *CVPR*, 2019, pp. 2859–2868.

[9] Y. Liu, S. Parisot, G. G. Slabaugh, X. Jia, A. Leonardis, and T. Tuytelaars, "More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning," in *ECCV*, 2020, pp. 699–716.

[10] N. Passalis, M. Tzelepi, and A. Tefas, "Heterogeneous knowledge distillation using information flow modeling," in *CVPR*, 2020, pp. 2336–2345.

[11] L. Hou, Z. Huang, L. Shang, X. Jiang, X. Chen, and Q. Liu, "Dynabert: Dynamic BERT with adaptive width and depth," in *NeurIPS*, 2020.

[12] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," in *NeurIPS*, 2020.

[13] Q. Guo, X. Wang, Y. Wu, Z. Yu, D. Liang, X. Hu, and P. Luo, "Online knowledge distillation via collaborative learning," in *CVPR*, 2020, pp. 11 017–11 026.

[14] T. Li, J. Li, Z. Liu, and C. Zhang, "Few sample knowledge distillation for efficient network compression," in *CVPR*, 2020, pp. 14 627–14 635.

[15] T. Wang, J.-Y. Zhu, A. Torralba, and A. A. Efros, "Dataset distillation," *CoRR*, vol. abs/1811.10959, 2018.

[16] S. Ahn, S. X. Hu, A. C. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *CVPR*, 2019, pp. 9163–9171.

[17] S.-I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *AAAI*, 2020, pp. 5191–5198.

[18] G. K. Nayak, K. R. Mopuri, V. Shaj, V. B. Radhakrishnan, and A. Chakraborty, "Zero-shot knowledge distillation in deep networks," in *ICML*, 2019, pp. 4743–4751.

[19] J. H. Cho and B. Hariharan, "On the efficacy of knowledge distillation," in *ICCV*, 2019, pp. 4794–4802.

[20] B. Zhao, K. R. Mopuri, and H. Bilen, "Dataset condensation with gradient matching," in *ICLR*, 2021.

[21] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *CVPR*, 2018, pp. 4320–4328.

[22] J. N. Kundu, N. Lakkakula, and R. V. Babu, "Um-adapt: Unsupervised multi-task adaptation using adversarial cross-task distillation," in *ICCV*, 2019, pp. 1436–1445.

[23] P. Zhou, L. Mai, J. Zhang, N. Xu, Z. Wu, and L. S. Davis, "M2KD: multi-model and multi-level knowledge distillation for incremental learning," *CoRR*, vol. abs/1904.01769, 2019.

[24] K. Javed and F. Shafait, "Revisiting distillation and incremental classifier learning," in *ACCV*, 2018, pp. 3–17.

[25] Y.-C. Hsu, Z. Lv, and Z. Kira, "Learning to cluster in order to transfer across domains and tasks," in *ICLR*, 2018.

[26] A. Achille and S. Soatto, "Emergence of invariance and disentanglement in deep representations," *JMLR*, vol. 19, no. 50, pp. 1–34, 2018.

[27] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *CVPR*, 2016, pp. 4004–4012.

[28] R. Manmatha, C.-Y. Wu, A. J. Smola, and P. Krähenbühl, "Sampling matters in deep embedding learning," in *CVPR*, 2017, pp. 2859–2867.

[29] Z.-H. Zhou, "Learnware: on the future of machine learning," *FCS*, vol. 10, no. 4, pp. 589–590, 2016.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015, pp. 1026–1034.

[31] I. Kuzborskij and F. Orabona, "Fast rates by transferring from auxiliary hypotheses," *Machine Learning*, vol. 106, no. 2, pp. 171–195, 2017.

[32] S. S. Du, J. Koushik, A. Singh, and B. Póczos, "Hypothesis transfer learning via transformation functions," in *NeurIPS*, 2017, pp. 574–584.

[33] X. Li, Y. Grandvalet, and F. Davoine, "Explicit inductive bias for transfer learning with convolutional networks," in *ICML*, 2018, pp. 2830–2839.

[34] S. Srinivas and F. Fleuret, "Knowledge transfer with jacobian matching," in *ICML*, 2018, pp. 4730–4738.

[35] H.-J. Ye, D.-C. Zhan, Y. Jiang, and Z.-H. Zhou, "Rectify heterogeneous models with semantic mapping," in *ICML*, 2018, pp. 1904–1913.

[36] ——, "Heterogeneous few-shot model rectification with semantic mapping," *TPAMI*, vol. 43, no. 11, pp. 3878–3891, 2021.

[37] V. Vapnik and A. Vashist, "A new learning paradigm: Learning using privileged information," *Neural Networks*, vol. 22, no. 5-6, pp. 544–557, 2009.

[38] V. Vapnik and R. Izmailov, "Learning using privileged information: similarity control and knowledge transfer," *JMLR*, vol. 16, pp. 2023–2049, 2015.

[39] ——, "Learning with intelligent teacher," in *COPA*, 2016, pp. 3–19.

[40] Z.-H. Zhou and Y. Jiang, "Nec4.5: Neural ensemble based C4.5," *TKDE*, vol. 16, no. 6, pp. 770–773, 2004.

[41] M. Phuong and C. Lampert, "Towards understanding knowledge distillation," in *ICML*, 2019, pp. 5142–5151.

[42] A. Gotmare, N. S. Keskar, C. Xiong, and R. Socher, "A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation," in *ICLR*, 2019.

[43] X. Tao, X. Chang, X. Hong, X. Wei, and Y. Gong, "Topology-preserving class-incremental learning," in *ECCV*, 2020, pp. 254–270.

[44] H. Bagherinezhad, M. Horton, M. Rastegari, and A. Farhadi, "Label refinery: Improving imagenet classification through label progression," *CoRR*, vol. abs/1805.02641, 2018.

[45] B. B. Sau and V. N. Balasubramanian, "Deep model compression: Distilling knowledge from noisy teachers," *CoRR*, vol. abs/1610.09650, 2016.

[46] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *ICLR*, 2017.

[47] W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, and R. Pascanu, "Sobolev training for neural networks," in *NeurIPS*, 2017, pp. 4281–4290.

[48] A. Koratana, D. Kang, P. Bailis, and M. Zaharia, "LIT: learned intermediate representation training for model compression," in *ICML*, 2019, pp. 3509–3518.

[49] S. H. Lee, D. H. Kim, and B. C. Song, "Self-supervised knowledge distillation using singular value decomposition," in *ECCV*, 2018, pp. 339–354.

[50] X. Li, J. Wu, H. Fang, Y. Liao, F. Wang, and C. Qian, "Local correlation consistency for knowledge distillation," in *ECCV*, 2020, pp. 18–33.

[51] G. Li, J. Zhang, Y. Wang, C. Liu, M. Tan, Y. Lin, W. Zhang, J. Feng, and T. Zhang, "Residual distillation: Towards portable deep neural networks without shortcuts," in *NeurIPS*, 2020.

[52] Z. Li and D. Hoiem, "Learning without forgetting," *TPAMI*, vol. 40, no. 12, pp. 2935–2947, 2018.

[53] F. Mao, B. Ma, H. Chang, S. Shan, and X. Chen, "Learning efficient text-to-image synthesis via interstage cross-sample similarity distillation," *Science China Information Sciences*, vol. 64, no. 2, 2021.

[54] X.-Z. Wu, S. Liu, and Z.-H. Zhou, "Heterogeneous model reuse via optimizing multiparty multiclass margin," in *ICML*, 2019, pp. 6840–6849.

[55] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *NeurIPS*, 2016, pp. 3630–3638.

[56] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *NeurIPS*, 2017, pp. 4080–4090.

[57] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017, pp. 1126–1135.

[58] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *CVPR*, 2018, pp. 5822–5830.

[59] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *CVPR*, 2020, pp. 8805–8814.

[60] N. Pang, X. Zhao, W. Wang, W. Xiao, and D. Guo, "Few-shot text classification by leveraging bi-directional attention and cross-class knowledge," *Science China Information Sciences*, vol. 64, no. 3, 2021.

[61] D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Co-transport for class-incremental learning," in *ACM MM*, 2021, pp. 1645–1654.

[62] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *CVPR*, 2017, pp. 5533–5542.

[63] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *JMLR*, vol. 10, pp. 207–244, 2009.

[64] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015, pp. 815–823.

[65] H.-J. Ye, D.-C. Zhan, Y. Jiang, and Z.-H. Zhou, "What makes objects similar: A unified multi-metric learning approach," *TPAMI*, vol. 41, no. 5, pp. 1257–1270, 2019.

[66] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, vol. 119, 2020, pp. 1597–1607.

[67] H.-J. Ye, D.-C. Zhan, N. Li, and Y. Jiang, "Learning multiple local metrics: Global consideration helps," *TPAMI*, vol. 42, no. 7, pp. 1698–1712, 2020.

[68] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *ICML*, 2007, pp. 209–216.

[69] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *JMLR*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[70] L. van der Maaten and K. Q. Weinberger, "Stochastic triplet embedding," in *MLSP*, 2012, pp. 1–6.

[71] E. Amid and A. Ukkonen, "Multiview triplet embedding: Learning attributes in multiple maps," in *ICML*, 2015, pp. 1472–1480.

[72] H. Chen, Y. Wang, C. Xu, C. Xu, and D. Tao, "Learning student networks via feature embedding," *TNNLS*, vol. 32, no. 1, pp. 25–35, 2021.

[73] M. Budnik and Y. Avrithis, "Asymmetric metric learning for knowledge transfer," in *CVPR*, 2021, pp. 8228–8238.

[74] L. Chen, D. Wang, Z. Gan, J. Liu, R. Henao, and L. Carin, "Wasserstein contrastive representation distillation," in *CVPR*, 2021, pp. 16 296–16 305.

[75] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *ICCV*, 2019, pp. 1921–1930.

[76] B. Peng, X. Jin, J. Liu, D. Li, Y. Wu, Y. Liu, S. Zhou, and Z. Zhang, "Correlation congruence for knowledge distillation," in *ICCV*, 2019, pp. 5007–5016.

[77] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *ICCV*, 2019, pp. 1365–1374.

[78] Y. Liu, J. Cao, B. Li, C. Yuan, W. Hu, Y. Li, and Y. Duan, "Knowledge distillation via instance relationship graph," in *CVPR*, 2019, pp. 7096–7104.

[79] H. Hu, L. Xie, R. Hong, and Q. Tian, "Creating something from nothing: Unsupervised knowledge distillation for cross-modal hashing," in *CVPR*, 2020, pp. 3120–3129.

[80] H. Oki, M. Abe, J. Miyao, and T. Kurita, "Triplet loss for knowledge distillation," in *IJCNN*, 2020, pp. 1–7.

[81] J. Zhu, S. Tang, D. Chen, S. Yu, Y. Liu, M. Rong, A. Yang, and X. Wang, "Complementary relation contrastive distillation," in *CVPR*, 2021, pp. 9260–9269.

[82] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *CVPR*, 2019, pp. 3967–3976.

[83] S. A. Koohpayegani, A. Tejankar, and H. Pirsiavash, "Compress: Self-supervised learning by compressing representations," in *Advances in Neural Information Processing Systems 33*, 2020.

[84] A. Tejankar, S. A. Koohpayegani, V. Pillai, P. Favaro, and H. Pirsiavash, "ISD: self-supervised learning by iterative similarity distillation," in *ICCV*, 2021, pp. 9609–9618.

[85] G. Xu, Z. Liu, X. Li, and C. C. Loy, "Knowledge distillation meets self-supervision," in *ECCV*, 2020, pp. 588–604.

[86] Z. Fang, J. Wang, L. Wang, L. Zhang, Y. Yang, and Z. Liu, "SEED: self-supervised distillation for visual representation," in *ICLR*, 2021.

[87] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *ECCV*, 2016, pp. 499–515.

[88] X. He, Y. Zhou, Z. Zhou, S. Bai, and X. Bai, "Triplet-center loss for multi-view 3d object retrieval," in *CVPR*, 2018, pp. 1945–1954.

[89] E. Amid and A. Ukkonen, "Multiview triplet embedding: Learning attributes in multiple maps," in *ICML*, 2015, pp. 1472–1480.

[90] T. Mensink, J. J. Verbeek, F. Perronnin, and G. Csurka, "Distance-based image classification: Generalizing to new classes at near-zero cost," *TPAMI*, vol. 35, no. 11, pp. 2624–2637, 2013.

[91] H. Chen, T. Guo, C. Xu, W. Li, C. Xu, C. Xu, and Y. Wang, "Learning student networks in the wild," in *CVPR*, 2021, pp. 6428–6437.

[92] M. Gao, Y. Shen, Q. Li, C. C. Loy, and X. Tang, "Feature matters: A stage-by-stage approach for knowledge transfer," *CoRR*, vol. abs/1812.01819, 2018.

[93] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[94] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[95] B. N. Oreshkin, P. R. López, and A. Lacoste, "TADAM: task dependent adaptive metric for improved few-shot learning," in *NeurIPS*, 2018, pp. 719–729.

[96] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.

[97] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *BMVC*, 2016.

[98] X. Li, D.-P. Fan, F. Yang, A. Luo, H. Cheng, and Z. Liu, "Probabilistic model distillation for semantic correspondence," in *CVPR*, 2021, pp. 7505–7514.

[99] Y. Zhang, Z. Lan, Y. Dai, F. Zeng, Y. Bai, J. Chang, and Y. Wei, "Prime-aware adaptive distillation," in *ECCV*, 2020, pp. 658–674.

[100] L. Yuan, F. E. H. Tay, G. Li, T. Wang, and J. Feng, "Revisiting knowledge distillation via label smoothing regularization," in *CVPR*, 2020, pp. 3902–3910.

[101] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[102] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *IEEE Conference on Computer Vision and Pattern Recognition First Workshop on Fine-Grained Visual Categorization*, 2011.

[103] Z. Huang and N. Wang, "Like what you like: Knowledge distill via neuron selectivity transfer," *CoRR*, vol. abs/1707.01219, 2017.

[104] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *PNAS*, vol. 114, no. 13, pp. 3521–3526, 2017.

[105] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning - the good, the bad and the ugly," in *CVPR*, 2017, pp. 3077–3086.

[106] H.-J. Ye, H. Hu, and D.-C. Zhan, "Learning adaptive classifiers synthesis for generalized few-shot learning," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1930–1953, 2021.

[107] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia, "Maintaining discrimination and fairness in class incremental learning," in *CVPR*, 2020, pp. 13 205–13 214.

[108] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, "Decoupling representation and classifier for long-tailed recognition," in *ICLR*, 2020.

[109] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *ICLR*, 2017.

[110] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[111] Y. Wang, W.-L. Chao, K. Q. Weinberger, and L. van der Maaten, "Simpleshot: Revisiting nearest-neighbor classification for few-shot learning," *CoRR*, vol. abs/1911.04623, 2019.

[112] B. Liu, Y. Cao, Y. Lin, Q. Li, Z. Zhang, M. Long, and H. Hu, "Negative margin matters: Understanding margin in few-shot classification," in *ECCV*, 2020, pp. 438–455.

Han-Jia Ye received the Ph.D. degree in computer science, Nanjing University, China in 2019. At the same year, he became a faculty member in the School of Artificial Intelligence at Nanjing University His research interests lie primarily in machine learning, including distance metric learning, multi-modal/multi-task learning, meta-learning, and semantic mining. He has published over 20 papers in leading international journal/conferences. He serves as SPC/PC in leading conferences such as ICLR, CVPR, ICCV, ICML, NeurIPS, IJCAI, ECML-PKDD, etc.

Su Lu received his B.Sc. degree in 2018. He is currently a Ph.D. candidate in the State Key Laboratory for Novel Software Technology at Nanjing University, China. His main research interests include machine learning and its applications in computer vision and data mining, especially meta-learning and knowledge distillation. He served as PC member in leading conferences like CVPR, AAAI, etc.

De-Chuan Zhan received the Ph.D. degree in computer science, Nanjing University, China in 2010. At the same year, he became a faculty member in the Department of Computer Science and Technology at Nanjing University, China. He is currently a professor the School of Artificial Intelligence at Nanjing University. His research interests are mainly in machine learning, data mining and mobile intelligence. He has published over 40 papers in leading international journal/conferences. He serves as an editorial board member of IDA and IJAPR, and serves as SPC/PC in leading conferences such as IJCAI, AAAI, ICML, NeurIPS, etc.