Faculty Distillation with Optimal Transport

Su Lu  Han-Jia Ye  De-Chuan Zhan
State Key Laboratory for Novel Software Technology
Nanjing University, Nanjing, 210023, China
{lus,yehj}@lamda.nju.edu.cn, zhandc@nju.edu.cn

Abstract

The outpouring of various pre-trained models empowers knowledge distillation (KD) by providing abundant teacher resources. Meanwhile, exploring the massive model repository to select a suitable teacher and further extracting its knowledge become daunting challenges. Standard KD fails to surmount two obstacles when training a student with the availability of plentiful pre-trained teachers, i.e., the “faculty”. First, we need to seek out the most contributive teacher in the faculty efficiently rather than enumerating all of them for a student. Second, since the teacher may be pre-trained on different tasks w.r.t. the student, we must distill the knowledge from a more general label space. This paper studies this “faculty distillation” where a student performs teacher assessment and generalized knowledge reuse. We take advantage of optimal transport to construct a unifying objective for both problems, which bridges the semantic gap and measures the relatedness between a pair of models. This objective can select the most relevant teacher, and we minimize the same objective over student parameters to transfer the knowledge from the selected teacher subsequently. Experiments in various settings demonstrate the succinctness and versatility of our proposed method.

1 Introduction

Knowledge distillation (KD) [17] is a promising model reuse technique proposed by [20]. It has been proven to be effective in compressing models [48] and improving model performance [65]. The rapid spring of advanced deep learning algorithms [9, 18, 26] and models [19, 59, 13, 6, 57] brings the availability of plentiful pre-trained models, offering KD abundant teacher resources and the opportunity to be applied to more practical applications. Under such a background, new challenges arise when we want to distill knowledge from a repository of pre-trained models.

Consider product recognition models in supermarkets that target different class sets for different branches. When a new branch opens up, all existing models from other branches may be considered as a teacher to train the new model to improve learning efficiency. However, we do not know which one is the most valuable, and existing models have different label spaces from our target task. These dilemmas reveal the insufficiency of standard KD in two aspects. (1) Standard KD often assumes that a teacher is given in advance. However, with a repository of pre-trained models, we have to efficiently seek out the most contributive teacher. (2) Standard KD requires the label space of teacher and student to be identical, which is too strict to be satisfied for a teacher chosen from the repository.

Figure 1: Comparison between two settings. Left: KD extracts knowledge from a given teacher and assists the training of a student. It requires the teacher and the student to target a same task. Right: When we have access to a group of teachers (faculty), we need to select one and perform generalized knowledge reuse.

Preprint. Under review.
To tackle these two problems, we formulate a new setting called “faculty distillation” where the faculty contains a group of accessible teachers specializing in various tasks. Faculty distillation contains two main phases. In the first phase, a student needs to select the most relevant teacher from the faculty. Since the model repository can be huge, it is not affordable to try each model as a teacher and train the student repeatedly. The second phase is generalized knowledge reuse which means transferring the knowledge of the selected teacher to the target task even though it is pre-trained on a different label set. Figure 1 gives a comparison between standard KD and faculty distillation.

The first problem involves model transferability [58, 40, 67, 66, 51]. Taking no account of the computational budget, we can naively use every teacher belonging to the faculty to teach the student and pick the one that leads to the best result. In practice, we have to design a selection mechanism that evaluates the relatedness of a cross-task teacher precisely and efficiently. The second problem is caused by support mismatch between the teacher’s label distribution and the student’s prediction. Lying at the core of standard KD [20] is the simple idea of distribution matching via KL divergence. This approach does not work when the teacher and student target two different label spaces. An alternative is ignoring the teacher’s classifier and reusing its representation to assist student [44, 50, 64, 68, 8]. However, the potential information contained in the teacher’s predictions is wasted.

This paper proposes a framework named Faculty Distillation with Optimal Transport (FaDiOT) that jointly tackles two problems mentioned above. Although teacher and student target different classes, we can match their predictions by considering the relationship between two class sets, which inspires us to utilize Optimal Transport (OT) [46] to bridge the semantic gap and measure the relatedness between a teacher and the target student. Specifically, after constructing a cost matrix between two class sets by teacher’s embedding network, we use Sinkhorn distance [12] between teacher’s output and student’s output as an objective for both teacher assessment and generalized knowledge reuse. In the first phase, we traverse the faculty to seek the most relevant teacher with the nearest Sinkhorn distance to the target student. This process is free of model training and hence efficient. In the second phase, we optimize the student’s parameters to pull it closer to the teacher in Sinkhorn distance, which successfully transfers the teacher’s generalized knowledge to a distinct label space. Experiments in various settings demonstrate the succinctness and versatility of our proposed FaDiOT.

2 Related Work

Knowledge Distillation. KD is an essential model reuse approach that extracts knowledge from a teacher and assists a student’s training. It was first proposed in [20] and has been proven to be an effective technique for improving student performance. Recent advances in knowledge distillation involve exploring new types of knowledge [50, 45, 27, 23, 44, 64, 55, 2], developing new training styles [16, 62, 39, 30, 15, 43], studying the reasons for its success [47, 37, 56], and applying it to new applications [4, 33]. Existing works often assume that the teacher and student are trained on the same task, but we stress the problem of distilling knowledge from a generalized teacher whose label set can be same as, overlapped with, or totally different from the student’s label set. A similar setting was considered in [64] where the authors studied cross-task knowledge distillation and reused the comparison ability of teacher’s embedding. In FaDiOT, we simultaneously reuse the teacher’s representations and predictions and close the semantic gap between teacher and student through OT.

Optimal Transport for Knowledge Reuse. OT [46] is widely applied to many fields related to knowledge reuse. In [52], the authors softly aligned the neurons of two separate models to fuse them into a single one. Under the setting of domain adaptation, [11, 41, 42, 63] transported source features to target features and then trained a classifier for the target domain. These algorithms only consider the ground metric but ignore the histograms of transported objects and naively set them to empirical distributions (uniform distributions in each batch). A more related line of works is feature distillation through OT [32, 68, 8], which matched teacher’s and student’s representations to pull them closer. However, the information contained in the teacher’s classifier is overlooked. In FaDiOT, we directly use OT to match teacher’s and student’s predictions, and the ground metric is estimated based on class semantics. In this way, the teacher model is utilized to the maximum extent.

Assessment of Pre-Trained Models. More and more large-scale pre-trained models are available nowadays, and digging into the potential of these models is attracting increasing interest. In [66, 67], the authors studied how to rank a group of pre-trained models to find the most contributive ones without fine-tuning them on the target task. They proposed to estimate the maximum value of
label evidence given features extracted by pre-trained models. Similar works [40, 58] evaluated the transferability of existing models to current task. Our proposed FADiOT is different from existing works in two aspects: (1) Existing works mainly focus on the scenario of model fine-tuning, but we study knowledge distillation which is a more flexible setting. Fine-tuning can be seen as a special kind of self-distillation; (2) All these heuristic metrics are designed without considering the subsequent procedure of model reuse. In FADiOT, model assessment is relevant to the distillation process, i.e., they both rely on semantic gaps between label distributions and Sinkhorn distance.

3 Preliminary

Supervised Classification. In a $C$-class classification task, we denote the training set by $D = \{(x_i, y_i)\}_{i=1}^{N}$ where $x_i \in \mathbb{R}^D$ and $y_i \in \{0, 1\}^C$ are instances and the corresponding one-hot labels. A model $f(x) : \mathbb{R}^D \rightarrow \mathbb{R}^C$ receives an instance as input and outputs a $C$-dimensional logit vector. Taking deep neural network as an example, $f(x)$ can be written as $W^\top \phi(x)$ where $\phi(x) : \mathbb{R}^D \rightarrow \mathbb{R}^d$ is the feature extractor and $W \in \mathbb{R}^d \times C$ contains parameters of the classifier. We can empirically optimize some loss function $\ell$ on the training set, i.e., $f^* = \arg\min_{f} \sum_{i=1}^{N} \ell(y_i, f(x_i))$.

Standard Knowledge Distillation. If we already have a well-trained model $f_T$ on label set $C_T$, we can extract “dark knowledge” from it [47, 37] to assist the training of a new model $f_S$ on label set $C_S$. The standard distillation deals with the situation where $C_T = C_S$. The most well-known formulation [20] is $\min_{f_S} \sum_{i=1}^{N} \ell(y_i, f_S(x_i)) + \lambda \text{KL}(\rho_T(f_T(x_i)), \rho_T(f_S(x_i)))$. Here $\rho_T$ is the softmax function with temperature $\tau$, and $\lambda$ balances two terms. This method considers teacher’s output class distribution as a soft label, and uses KL divergence to pull student’s output closer to it. The teacher model usually has a larger capacity [39], and its output can encode more information than one-hot label such as class correlations and instance relationships [56]. Trained with both instance labels and teacher’s supervision, the student model can obtain a higher accuracy and converge faster.

Faculty Distillation. In faculty distillation we have $H$ teachers $\{f_T^h\}_{h=1}^{H}$ trained on diverse tasks. Let $C_T^h$ be the label set of $h$-th teacher, and $C_T^h \neq C_S$ usually holds. Our goal is to select one teacher from the faculty and reuse its general knowledge to assist the training of the student model $f_S$. Ideally, the selected teacher will result in a student that has the lowest generalization error. However, we do not have access to the test set during training phase, and so we consider the following problem:

$$\min_{h, f_S} \sum_{i=1}^{N} \ell(y_i, f_S(x_i)) + \lambda \text{D}(f_T^h(x_i), f_S(x_i)).$$

(1)

Two variables $h$ and $f_S$ are coupled together in Equation (1). Note that $h$ is the index of teacher, but it is unrealistic to enumerate $h \in [H]$ and solve Equation (1) repeatedly. Thus, we decompose faculty distillation into two sub-problems, i.e., Teacher Assessment (TA) and Generalized Knowledge Reuse (GKR). TA aims to use a practical metric to rank all teachers efficiently and select the best one for the subsequent GKR, and we expect the selected teacher to result in a good student. The core of GKR is the implementation of $\mathbb{D}$, which matches two outputs in different label spaces. Intuitively, the metric in TA should be binding with the distillation method, and so we first study GKR in Section 4 by stating the exact design of $\mathbb{D}$. After that, we propose a metric consistent with $\mathbb{D}$ in Section 5.

4 Generalized Knowledge Reuse

After introducing several basic concepts in Section 3, we are ready to present details of FADiOT. In this section, we first discuss how to reuse the knowledge of a teacher that may have a different label space from the student, and study optimization properties of the proposed method.

4.1 Sinkhorn Distance for Distribution Matching

The main obstacle to matching the outputs of teacher and student is support mismatch, which makes KL divergence not applicable in generalized knowledge reuse. However, if we consider the semantic relationship $C_T$ and $C_S$, the teacher’s output can still be instructive. For example, suppose $f_T$ classifies cat and dog while $f_S$ differs tiger from dog. In that case, we can approximately use the teacher’s output probability of cat as a reference because cat and tiger share similar appearance characteristics.
Sinkhorn distance [12] is a regularized variant of OT distance, and it is widely used for matching two distributions in various applications [38, 31] because it considers the metric space of probabilities. This property enables us to use Sinkhorn distance to measure the discrepancy between two predictions in different label spaces. To this end, a key component is the cost matrix that describes the similarities between semantic information encoded by probability dimensions. We first state the definition of Sinkhorn distance and then propose a method to generate the cost matrix.

**Definition 4.1 (Sinkhorn Distance).** Let $\mathcal{P}_R$ be the set of R-dimensional probability simplexes, i.e., $\mathcal{P}_R = \{p \in \mathbb{R}^R | p_r \geq 0, \sum_{r=1}^R p_r = 1\}$. Let $\mu \in \mathcal{P}_{R_1}$ and $\nu \in \mathcal{P}_{R_2}$ be two discrete probability distributions. Define the set of transport polytopes as $\mathbb{H}(\mu, \nu) = \{T \in \mathbb{R}^{R_1 \times R_2} | T_{1R_2} = \mu, T_{R_11} = \nu\}$ which contains all legal transportation plans from $\mu$ to $\nu$. Let $\mathbb{H}(T) = -\sum_{m=1}^{R_1} \sum_{n=1}^{R_2} T_{mn} (\log T_{mn} - 1)$ be discrete entropy of transportation plan $T$. Given a cost matrix $M \in \mathbb{R}^{R_1 \times R_2}$, Sinkhorn distance between $\mu$ and $\nu$ is defined as

$$S_\epsilon(\mu, \nu) = \min_{T \in \mathbb{H}(\mu, \nu)} \langle T, M \rangle - \epsilon \mathbb{H}(T) .$$

Let $p_T$ and $p_S$ be the teacher’s and student’s predicted label distributions of an instance, computing $S_\epsilon(p_T, p_S)$ amounts to finding an optimal plan to transport teacher’s confidences in its own classes to student’s confidences. Each dimension will have priority if transported to a target dimension with a similar semantic since this leads to a lower transportation cost. If a teacher is related to current task, values in $M$ should be small in general, resulting in a small $S_\epsilon(p_T, p_S)$. On the other hand, the transportation cost will be very high if the task gap between teacher and student cannot be ignored. In Equation (2), $\epsilon$ is a parameter controlling the strength of regularization term $\mathbb{H}(T)$. This term smooths the objective and forces the transportation plan to spread over the dimensions [46].

**Cost Matrix.** In this paper, we reuse teacher’s representation network to obtain class centers of both teacher’s classes and student’s classes. After that, we can compute Euclidean distances between class centers in the embedding space to determine $M$. Specifically, $M_{mn} = \|e_{T,m} - e_{S,n}\|_2$ where $e_{T,m}$ is the $m$-th class center in $C_T$ and $e_{S,n}$ is the $n$-th class center in $C_S$.

**Discussion on Cost Matrix.** (1) We do not require the teacher to cover the student’s classes. A common case in KD is that teacher’s task is related to the student’s task. Although there often exist representation drifts between different class sets, $M$ only needs to describe relative relationships between classes, and we can transfer the teacher’s comparison ability across classes [64, 60, 54]. (2) When the task gap between teacher and student is enormous, we empirically show that the student performance will not degenerate compared to vanilla training. Refer to Section 6.1 for details.

With cost matrix $M$, we can minimize the Sinkhorn distance $S_\epsilon(p_T, p_S)$ to train the student:

$$\min_{f_S} \sum_{i=1}^N \ell(y_i, f_S(x_i)) + \lambda S_\epsilon(p_T(f_T(x_i)), p_S(f_S(x_i))) .$$

Equation (3) replaces KL divergence in standard KD with Sinkhorn distance, bridging the support gap between $p_T$ and $p_S$. Moreover, the teacher’s representations and predictions are simultaneously reused, which is a more adequate utilization compared to feature-based methods [44, 68, 8, 32].

### 4.2 Optimization

Now we study the optimization properties of Equation (3). It is a bi-level optimization problem [10] because it involves a nested optimization problem (Equation (2)). Our target is learning the student, and we need to compute the gradient of $S_\epsilon$ w.r.t. model parameters through inner optimization. Moreover, solving the inner problem once can only provide a single update for the student, so accelerating the inner problem is necessary. Thus, there are two problems we need to tackle: (1) How to compute the gradient w.r.t. model parameters? (2) How to accelerate inner optimization?

**Gradient Computation.** Directly computing the gradient of $S_\epsilon$ is intractable, and we solve this problem from the dual of Equation (2), which is written as the following equation:

$$\max_{\alpha, \beta} \alpha^\top p_T + \beta^\top p_S - \epsilon \sum_{m,n} \exp \left(\frac{\alpha_m + \beta_n - M_{mn}}{\epsilon}\right) .$$

(4)
Proposition 4.2 (Gradient of Sinkhorn Distance). Let $p_T = \rho_T(f_T(x))$ and $p_S = \rho_T(f_S(x))$ be the output probability distributions of teacher and student respectively. Let $\beta^*$ be the optimal solution to Equation (4). The gradient of $S_t(p_T, p_S)$ w.r.t. $p_S$ and $f(x)$ can be written as $\frac{\partial S_t(p_T, p_S)}{\partial p_S} = \beta^*$ and $\frac{\partial S_t(p_T, p_S)}{\partial f(x)} = (\beta^* - \langle \beta^*, p_S \rangle) \odot p_S$ respectively. $\odot$ means element-wise multiplication.

Proposition 4.2 solves our first question, and its detailed proof is in Appendix A. To update the student’s parameters, we need to obtain the optimal solution to Equation (4). In [12], the authors reformulated the problem as a matrix scaling problem and proposed a practical iterative algorithm (Sinkhorn algorithm) to solve it. Although the Sinkhorn algorithm has time complexity $O(|C_T||C_S|)$ [22, 3], which is much faster than raw OT problem, in our method we need to update student for many times, and this requires us to further accelerate Sinkhorn algorithm.

Acceleration of Sinkhorn Algorithm. We first review Sinkhorn algorithm proposed by [12], and then show that early stopping is a convenient and efficient strategy to accelerate Sinkhorn algorithm. Let $T_{\alpha, \beta}$ be the optimal solution to problem $\min_{T \in \Pi_{|p_T|, p_S}} \langle T, M \rangle - \epsilon \|\Pi(T) - \alpha^T T 1\|_F - \beta^T T 1$, and first order condition yields that $\frac{\partial L(T, \alpha, \beta)}{\partial T} = M_{mn} + \epsilon \log T_{mn} = \exp(\alpha_m / \epsilon) \exp(-M_{mn} / \epsilon) \exp(\beta_n / \epsilon)$. This can be written in matrix form $T_{\alpha, \beta} = \text{diag}(\alpha) K \text{diag}(\beta)$ where matrix $K = \exp(-M / \epsilon)$, $\alpha = \exp(\alpha / \epsilon)$, $\beta = \exp(\beta / \epsilon)$. Therefore, solving the Sinkhorn problem amounts to obtaining vectors $u, v \geq 0$. This can be solved by Sinkhorn’s fixed point iteration [53] $(u, v) \leftarrow (p_T / (K u), p_S / (K^T v))$. A natural idea is performing fixed point iteration only for a few times. This raises another question: what is the influence of early stopping on the gradient used to optimize student model? Although the convergence of Sinkhorn algorithm is well analyzed, the convergence rate of gradient used in our method is not clear. Thus, we study this problem to ensure the rationality of early stopping.

Proposition 4.3 (Convergence of Gradient). Let $u^{(t)}$ and $v^{(t)}$ be vectors after $t$ Sinkhorn iterations. Let $\beta^{(t)}$ be the approximate gradient w.r.t. student’s output probability $p_S$ based on $u^{(t)}$ and $v^{(t)}$. Let $\beta^*$ be the accurate gradient. $\beta^{(t)}$ has a linear convergence rate $\kappa(K)^2$ in variation seminorm $\|z\|_{\text{var}} = \max(z) - \min(z)$, as shown in Equation (5). $\kappa(K) \in (0, 1)$ is a constant about $K$.

$$\frac{\|\beta^{(t+1)} - \beta^*\|_{\text{var}}}{\|\beta^{(t)} - \beta^*\|_{\text{var}}} \leq \kappa(K)^2$$

Figure 2: An empirical evaluation of Proposition 4.3. We randomly sample 256 instances from CIFAR-100 [29] and form 256 OT problems. We illustrate the convergence curves of $\|\beta^{(t)} - \beta^*\|_{\text{var}}$, theoretical bound in Proposition 4.3, and $\|\beta^{(t)} - \beta^*\|_2$. The values are averaged over 256 problems. Refer to Appendix C for more details.

5 Teacher Assessment

Until now we have discussed how to reuse the knowledge of a teacher model that may have a different label space from the student and have studied the optimization properties of the proposed method. Another important topic is how to pick out the most contributive teacher from the faculty.

Recent works [58, 40, 67, 66] have proposed several empirical metrics to evaluate the relevance of a pre-trained model for fine-tuning. However, these metrics are heuristic and are not binding with the subsequent model reuse process. In FADLO, we equip the knowledge distillation algorithm with a complementary teacher selection method. Since we minimize the Sinkhorn distance to reuse a cross-task teacher, we also use Sinkhorn distance to measure the relatedness of teacher models.
Appendix

D. Terminal State

E. We check the gap between the fictitious and true student. The selected class subset changes with the state of sliding window. We can use the class overlap ratio to measure the task gap between teacher and student. For CUB, the class overlap ratio is given by 

\[ \frac{\sum_{i=1}^{N} S_i \left( \rho_f \left( f_T \left( x_i \right) \right), \rho_f \left( f_F \left( x_i \right) \right) \right)}{N} \]

where \( S_i \) is the similarity between the teacher's and fictitious student's predictions, \( \rho_f \) is the embedding network, \( f_T \) is the teacher, and \( f_F \) is the fictitious student. This forward process exists in all related works [38, 40, 67, 66] and does not cost too much time. After that, in virtue of the simplicity of linear classifiers and our acceleration of Sinkhorn algorithm, the computation of metrics can be completed very quickly. Refer to Section 6.2 for an empirical evaluation. (2) The gap between the fictitious and true student does not affect teacher assessment too much. Although the complexity of a linear classifier is limited, we do not require the fictitious student to mimic the true student perfectly but only use it to evaluate the relatedness between a teacher and the target student. In Appendix E, we check the gap between the fictitious and true student, and show that they lead to similar evaluation metrics.

Summary of FADiOT. We have discussed Generalized Knowledge Reuse (GKR) in Section 4 and Teacher Assessment (TA) in Section 5, which are two basic components of faculty distillation. In summary, given the faculty \( \{ f_T \}_{h=1}^{H} \) containing \( H \) models, we first perform TA and pick the model with the highest score as our teacher. After that, we solve Equation (3) to learn a student with the selected teacher. Algorithm schemes of our method can be found in Appendix D.

6 Experiment

Since faculty distillation includes two main stages, our experiments contain two main parts. Our experiments are conducted mainly on CIFAR-100 [29] and Caltech-UCSD Birds-200-2011 (CUB) [61]. Descriptions of these datasets can be found in Appendix F. Implementation details of experiments are listed in Appendix H.

6.1 Generalized Knowledge Reuse

In the first part, we assume that a fixed teacher is given and check the generalized knowledge reuse ability of FADiOT. In detail, we want to study following questions: (1) Can FADiOT reuse the knowledge of a generalized teacher? (2) Will FADiOT also perform well in standard KD? (3) Can cost matrix \( M \) capture class relationships well? (4) What will happen if task gap between teacher and student is too large? (5) Ablation study on the cost matrix (Appendix K). (6) Influence of hyper-parameters \( \tau, \lambda \) and \( \epsilon \) in Equation (3) over model performance (Appendix L).

Sliding Window Protocol. First of all, we introduce a special dataset split method called “sliding window” protocol, as shown in Figure 3a. Taking CUB as an example, we first sort its 200 classes randomly and then use a sliding window that covers 100 classes to select a class subset. Initially, the window covers the first 100 classes, moving 25 classes forward every step until covering the last 100 classes. This procedure generates 5 different label sets. In this subsection, we fix the class subset of the teacher as the initial one (first 100 classes) and range the class subset of the student, which means the task gap between teacher and student increases when the sliding window moves forward. We can use the class overlap ratio \( g \) to measure the task gap between teacher and student. For CUB, \( g \in \{ 100\%, 75\%, 50\%, 25\%, 0\% \} \). Refer to Appendix G for more details about this protocol.

In this subsection, teacher is trained on a fixed class subset (determined by the initial sliding window) with cross-entropy loss. Students are trained on different class subsets. When we train a model, we extract training instances belonging to its class subset to train it, and test in on test instances belonging...
Figure 4: [best viewed in color] Results of generalized knowledge distillation on CIFAR-100 (upper) and CUB (bottom). Class overlap ratio changes from 100% to 0%. All the values can be found in Appendix I.

Table 1: Average test accuracies on two datasets. Teacher architecture is WideResNet-(40,2) for CIFAR-100 and MobileNet-1.0 for CUB. When the student shares a same architecture as the teacher, we also use self-distillation (BAN) to learn the student, and the test accuracies are 75.41% and 76.87% on CIFAR-100 and CUB. † indicates self-supervised methods while ‡ indicates methods based on OT. Best results are in bold.

(a) Results on CIFAR-100.

| (Depth, Width) | (40, 2) | (16, 2) | (40, 1) | (16, 1) |
|----------------|---------|---------|---------|---------|
| Teacher        | 74.44   | 70.15   | 68.97   | 65.44   |
| Student        | 74.44   | 70.15   | 68.97   | 65.44   |
| KD [20]        | 75.47   | 71.87   | 70.46   | 66.54   |
| FitNet [50]    | 74.29   | 70.89   | 68.66   | 65.38   |
| VID [2]        | 75.25   | 73.31   | 71.51   | 66.32   |
| RKD [44]       | 76.62   | 72.56   | 72.18   | 65.22   |
| SFTN [45]      | 76.93   | 75.23   | 72.04   | 67.41   |
| AFD [23]       | 77.42   | 75.58   | 72.50   | 67.37   |
| SEED† [14]     | 76.28   | 73.40   | 71.83   | 67.75   |
| SSKD† [62]     | 75.42   | 74.03   | 72.71   | 67.30   |
| WCoRD† [8]     | 77.36   | 74.29   | 72.78   | 67.35   |
| MGD† [68]      | 76.40   | 74.25   | 72.17   | 66.80   |
| ReFilled [64]  | 77.49   | 74.01   | 72.72   | 67.56   |
| FADiOT         | 77.95   | 75.33   | 73.28   | 67.84   |

(b) Results on CUB.

| Width Multiplier | 1.0  | 0.75 | 0.5  | 0.25 |
|------------------|------|------|------|------|
| Teacher          | 75.36 | 74.87 | 72.41 | 69.72 |
| Student          | 75.36 | 74.87 | 72.41 | 69.72 |
| KD [20]          | 77.61 | 76.02 | 74.24 | 72.03 |
| FitNet [50]      | 75.10 | 75.03 | 72.17 | 69.09 |
| VID [2]          | 77.03 | 76.91 | 75.62 | 72.23 |
| RKD [44]         | 77.72 | 76.80 | 74.99 | 72.55 |
| SFTN [45]        | 77.64 | 77.90 | 77.34 | 73.55 |
| AFD [23]         | 78.67 | 78.11 | 77.42 | 73.60 |
| SEED† [14]       | 77.93 | 78.14 | 77.50 | 73.23 |
| SSKD† [62]       | 78.34 | 78.22 | 77.10 | 72.18 |
| WCoRD† [8]       | 79.02 | 78.20 | 77.83 | 74.22 |
| MGD† [68]        | 78.55 | 77.69 | 76.68 | 73.40 |
| ReFilled [64]    | 79.33 | 78.52 | 76.90 | 74.04 |
| FADiOT           | 79.69 | 78.37 | 78.05 | 74.49 |

to its class subset. For CIFAR-100, we instantiate teacher and student as WideResNets [69]. Teacher is WideResNet-(40,2) while students are WideResNet-{(40,2), (16,2), (40,1), (16,1)}. For CUB, teacher is MobileNet-1.0 [21] while students are MobileNet-{1.0, 0.75, 0.5, 0.25}.

**Generalized Knowledge Distillation.** Since standard KD [20] does not work when teacher and student target at different label spaces, we compare our FADiOT to several feature-based distillation methods including RKD [44], AML [7], and ReFilled [64]. Two OT-based methods, i.e., MGD [68] and WCoRD [8] are also considered. Results are shown in Figure 4. We can see that FADiOT outperforms other methods in most cases. Among all compared methods, ReFilled [64] is the only one designed for cross-task knowledge distillation, and it also achieves competitive performance. All these compared methods ignore teacher’s classifier and only utilizes its representation ability, which is the main reason for their unsatisfying performances. Both training set and test set change with class overlap ratio $g$, and this makes accuracies across different overlap ratios incomparable.

**Standard Knowledge Distillation.** Standard knowledge distillation is a special case of generalized knowledge distillation. In this part, we show that our proposed FADiOT can achieve competitive performance on standard knowledge distillation. We compare our method to several distillation methods on CIFAR-100 and CUB. Architectures of teachers and students are same as those in previous part. Results are listed in Table 1a and Table 1b. Our method achieves best results in most cases. Specifically, when teacher and student share a same architecture, we also try to use self-distillation to learn a student [16]. SEED [14] and SSKD [62] are recent proposed self-supervised
distillation methods, and we can see that our method is better because instance labels are used. Another interesting topic is different-family distillation which means the architectures of teacher and student come from different families. We study this problem in Appendix J.

**Effect of Cost Matrix.** An important module in our proposed FAD1OT is the cost matrix $\mathcal{M}$, which is constructed by the teacher’s embedding network $\phi_T$. In this part, we check whether this embedding network can characterize class semantics well. Specifically, we conduct a clustering experiment on CIFAR-100 [29]. CIFAR-100 contains 20 superfamilies and 5 classes in each superclass. Classes belonging to the same superclass share similar semantic information. For a given class overlap ratio $g$, we compute the embedded centers of student’s classes by $\phi_T$ and a randomly initialized network, and then perform K-means clustering on them. The superclass of each class center is considered as the ground truth, and we report normalized mutual information (NMI) in Table 2. We can see that the teacher’s embedding network successfully capture semantic information of unseen classes. We also try other kinds of cost matrices in the ablation study in Appendix K.

**Extremely Large Task Gap.** Since we assume that a fixed teacher is given, an interesting question is: what will happen if this teacher is irrelevant to the current task? Ideally, the student’s performance will not drop compared to training without the teacher. To verify this, we additionally train student models on Stanford Dog [25] with assistance of the teacher trained on the whole CUB dataset. Results are listed in Table 3. Some methods that fit the cross-task setting are compared. We can see that our method does not suffer from performance drop while most of the compared methods do harm to student’s performance. This means our method is robust to the task gap between teacher and student.

6.2 Teacher Assessment

In the second part, we construct a group of teachers to check the ability of FAD1OT to assess teachers. Several questions are under consideration: (1) Can our proposed metric successfully rank all teachers according to their contributions? (2) Is the metric efficient enough to be applied in practical applications? (3) Can the metric work well in both coarser-grained and fine-grained teacher selection?

**Double Sliding Window Protocol.** In order to generate multiple teachers, we expand the aforementioned sliding window protocol to “double sliding window protocol”, which contains two sliding windows for the teacher and the student, respectively, as shown in Figure 3b. Now both the teacher’s and the student’s subsets can change with their own sliding window.

In this subsection, the architecture of student is fixed as MobileNet-0.25 for CUB. We consider another architecture (ResNet [19]) to enrich repository of teacher models. In CUB we have 5 class subsets and 2 teacher architectures, which means we can construct 10 different teachers in total.

**Teacher Assessment.** Now we study whether our proposed metric can precisely select the most contributive teacher compared to other metrics including NCE [58], LEEP [40], and LogME [67]. Assuming that we have $H$ teachers, given a target task (a class subset for student), we denote by $Q^h$ the test accuracy of the student trained with assistance of $h$-th teacher. We use $Q^h$ to represent some evaluation metric of the $h$-th teacher. Ideally, $Q$ and $G$ should be highly correlated. In our assessment method, we set $Q^h = -\mathcal{M}(f^h_T)$ where $\mathcal{M}$ is introduced in Section 5. On CUB dataset, we have $H = 10$, and we show the values of $G$ and $Q$ in Figure 5. We can see that LogME and our proposed metric are positively associated with the student accuracy while other two metrics fail. In this figure, we can see that LEEP is architecture-sensitive since all triangles (ResNets) are assigned low confidence. In addition, Pearson correlation coefficients between each metric and the ground truth accuracy on 5 tasks are listed in Table 4. Our metric achieves the best results on 5 tasks.

| Table 2: NMI based on randomly initialized embedding network and teacher’s embedding $\phi_T$. |
|---|---|---|---|---|---|---|
| $g$ (%) | 100 | 80 | 60 | 40 | 20 | 0 |
| random | 0.52 | 0.52 | 0.59 | 0.56 | 0.55 | 0.55 |
| $\phi_T$ | 0.80 | 0.72 | 0.77 | 0.76 | 0.79 | 0.81 |

| Table 3: Average test accuracies on DOG. The teacher is trained on CUB. Best results are in bold. |
| Channel Width | 1.0 | 0.75 | 0.5 | 0.25 |
| Student | 72.35 | 70.69 | 70.11 | 68.57 |
| FitNet [50] | 71.14 | 68.37 | 69.90 | 68.41 |
| RKD [44] | 72.38 | 70.15 | 69.90 | 68.42 |
| ReFilled [64] | 73.07 | 70.23 | 69.35 | 68.18 |
| WCoRD [8] | 72.84 | 69.98 | 69.42 | 68.20 |
| FAD1OT | 73.55 | 70.74 | 70.35 | 68.60 |

| Table 4: Pearson correlation coefficients between each metric and the ground truth accuracy on 5 tasks. |
|---|---|---|---|---|---|
| Metrics | Task1 | 2 | 3 | 4 | 5 |
| NCE [58] | -0.03 | 0.16 | 0.35 | 0.50 | 0.19 | 0.23 |
| LEEP [40] | 0.14 | 0.60 | 0.17 | 0.14 | 0.50 | 0.31 |
| LogME [67] | 0.72 | 0.96 | 0.66 | 0.40 | 0.73 | 0.69 |
| FAD1OT | 0.74 | 0.95 | 0.69 | 0.40 | 0.73 | 0.70 |
we further evaluate our method on a coarse-grained Time Consumption. We list the time consumption for computing metrics in Table 5. NCE and LEEP are two efficient metrics, but they fail to evaluate teacher qualities in our experiment. LogME and our proposed metric are slower but effective. The extra time cost of our metric mainly comes from training the linear classifier. In general, our metric can efficiently select the most relevant teacher.

**Coarse-Grained Teacher Selection.** We have tried to select a suitable teacher from a fine-grained repository, i.e., the class subsets for all the teachers are sampled from a same dataset (CUB). In this part, we further evaluate our method on a coarse-grained repository containing 5 teachers trained on MIT Indoor Scenes (Indoor) [49], Stanford Dog (Dog) [25], Caltech-UCSD Birds-200-2011 (CUB) [61], Stanford Car-196 (Car) [28], and FGVC-AirCraft (AirCraft) [36], respectively. These 5 datasets constitute 5 separate domains and the semantic gap between them is very large. We use FaDiOT to select the teacher for each target domain, and train 5 student models. In Figure 6, we can see that our proposed metric (values in the figure) easily finds the corresponding domain, and the student performance is improved a lot.

### 7 Further Discussion

(1) In our experiment, we only reuse the knowledge of one selected teacher model, and reusing multiple models simultaneously is another interesting topic [51, 66, 15]. Actually, we can distill multiple teachers by adding Sinkhorn distance terms in our method, but the weights of these terms are not easy to determined. A possible solution is weighting multiple teachers by their assessment scores, which is a meaningful future work. Another significant future work is finding vectorized representations of models and tasks [70, 1, 24]. (2) Although our proposed method does not require to train the student repeatedly, all the existing works [67, 58, 40] including ours need to perform forward process for each teacher, which is infeasible for a huge repository. If each teacher and task is represented by a vector, we can directly match them in the embedding space to realize efficient model recommendation. (3) In our paper, we assume that all the teacher models in the repository are well trained. If an offensive client upload a bad model to the teacher repository, it may be evaluated as a “related” model to the target task while does harm to student’s performance. Defending adversarial attacks on the repository is also an essential future work.

| Method | Time per Model (s) | NCE | LEEP | LogME | FaDiOT |
|--------|--------------------|-----|------|-------|--------|
| A      | 0.12               | 0.11| 0.42 | 0.91  |        |
| B      | 0.34               | 0.39| 0.39 | 0.39  |        |
| C      | 0.25               | 0.19| 0.19 | 0.19  |        |
| D      | 0.14               | 0.14| 0.14 | 0.14  |        |
| E      | 0.34               | 0.39| 0.39 | 0.39  |        |

Figure 6: **Left:** Our metric can easily find the most relevant teacher from the coarse-grained model repository. **Right:** Distillation from the selected teacher improves student performance.
References

[1] Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. In Proceedings of the 17th IEEE/CVF International Conference on Computer Vision, pages 6429–6438, 2019.

[2] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9163–9171, 2019.

[3] Jason Altschuler, Jonathan Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In Advances in Neural Information Processing Systems 30, pages 1965–1975, 2017.

[4] Glen Berseth, Cheng Xie, Paul Cernek, and Michiel Van de Panne. Progressive reinforcement learning with distillation for multi-skilled motion control. In Proceedings of the 6th International Conference on Learning Representations, 2018.

[5] Garrett Birkhoff. Extensions of jentzsch’s theorem. Transactions of the American Mathematical Society, 85(1):219–227, 1957.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Advances in Neural Information Processing Systems 33, pages 1877–1901, 2020.

[7] Mateusz Budnik and Yannis Avrithis. Asymmetric metric learning for knowledge transfer. In Proceedings of the 34th IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8228–8238, 2021.

[8] Liqun Chen, Dong Wang, Zhe Gan, Jingjing Liu, Ricardo Henao, and Lawrence Carin. Wasserstein contrastive representation distillation. In Proceedings of the 34th IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16296–16305, 2021.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Proceedings of the 37th International Conference on Machine Learning, pages 1597–1607, 2020.

[10] Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. Annals of Operations Research, 153(1):235–256, 2007.

[11] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. CoRR, abs/1507.00504, 2015.

[12] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In Advances in Neural Information Processing Systems 26, pages 2292–2300, 2013.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In Proceedings of the 9th International Conference on Learning Representations, 2021.

[14] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation. CoRR, abs/2101.04731, 2021.

[15] Takashi Fukuda, Masayuki Suzuki, Gakuto Kurata, Samuel Thomas, Jia Cui, and Bhuvana Ramabhadran. Efficient knowledge distillation from an ensemble of teachers. In Proceedings of the 18th Annual Conference of the International Speech Communication Association, pages 3697–3701, 2017.

[16] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In Proceedings of the 35th International Conference on Machine Learning, pages 1607–1616, 2018.

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

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9726–9735, 2020.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the 29th IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.
[20] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. CoRR, abs/1503.02531, 2015.
[21] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017.
[22] Gao Huang, Chuan Quo, Matt J Kusner, Yu Sun, Kilian Q Weinberger, and Fei Sha. Supervised word mover’s distance. In Advances in Neural Information Processing Systems 29, pages 4869–4877, 2016.
[23] Mingi Ji, Byeongho Heo, and Sungrae Park. Show, attend and distill: Knowledge distillation via attention-based feature matching. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, pages 7945–7952, 2021.
[24] Hadi S. Jomaa, Lars Schmidt-Thieme, and Josif Grabocka. Dataset2vec: learning dataset meta-features. Data Mining and Knowledge Discovery, 35(3):964–985, 2021.
[25] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In Proceedings of CVPR Workshop on Fine-Grained Visual Categorization, 2011.
[26] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In Advances in Neural Information Processing Systems 33, pages 18661–18673, 2020.
[27] Nikos Komodakis and Sergey Zagoruyko. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In Proceedings of the 5th International Conference on Learning Representations, 2017.
[28] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In Proceedings of the 4th International IEEE Workshop on 3D Representation and Recognition, 2013.
[29] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
[30] Seung Hyun Lee, Dae Ha Kim, and Byung Cheol Song. Self-supervised knowledge distillation using singular value decomposition. In Proceedings of the 15th European Conference on Computer Vision, pages 335–350, 2018.
[31] Chi-Heng Lin, Mehdi Azabou, and Eva Dyer. Making transport more robust and interpretable by moving data through a small number of anchor points. In Proceedings of the 38th International Conference on Machine Learning, pages 6631–6641, 2021.
[32] Suhas Lohit and Michael Jones. Model compression using optimal transport. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 3645–3654, 2022.
[33] Su Lu, Han-Jia Ye, Le Gan, and De-Chuan Zhan. Towards enabling meta-learning from target models. In Advances in Neural Information Processing Systems 34, 2021.
[34] Giulia Luise. Entropic Optimal Transport in Machine Learning: Applications to Distributional Regression, Barycentric Estimation and Probability Matching. PhD thesis, University College London, 2021.
[35] Giulia Luise, Alessandro Rudi, Massimiliano Pontil, and Carlo Ciliberto. Differential properties of sinkhorn approximation for learning with wasserstein distance. CoRR, abs/s1805.11897, 2018.
[36] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. CoRR, abs/1306.5151, 2013.
[37] Aditya K Menon, Ankit Singh Rawat, Sashank Reddi, Seungyeon Kim, and Sanjiv Kumar. A statistical perspective on distillation. In Proceedings of the 38th International Conference on Machine Learning, pages 7632–7642, 2021.
[38] Grégoire Mialon, Dexiong Chen, Alexandre d’Aspremont, and Julien Mairal. A trainable optimal transport embedding for feature aggregation and its relationship to attention. In Proceedings of the 9th International Conference on Learning Representations, 2021.
[39] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassanzadak. Improved knowledge distillation via teacher assistant. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, pages 5191–5198, 2020.
[40] Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. Leep: A new measure to evaluate transferability of learned representations. In Proceedings of the 37th International Conference on Machine Learning, pages 7294–7305, 2020.
[41] Tuan Nguyen, Trung Le, Nhan Đam, Quan Hung Tran, Truyen Nguyen, and Dinh Q. Phung. TIDOT: A teacher imitation learning approach for domain adaptation with optimal transport. In Proceedings of the 30th International Joint Conference on Artificial Intelligence, pages 2862–2868, 2021.
[42] Tuan Nguyen, Trung Le, He Zhao, Quan Hung Tran, Truyen Nguyen, and Dinh Q. Phung. Most: multi-source domain adaptation via optimal transport for student-teacher learning. In Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence, volume 161, pages 225–235, 2021.

[43] Dae Young Park, Moon-Hyun Cha, Changwook Jeong, Daesin Kim, and Bohyung Han. Learning student-friendly teacher networks for knowledge distillation. In Advances in Neural Information Processing Systems 34, pages 13292–13303. 2021.

[44] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3967–3976, 2019.

[45] Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. Alp-kd: Attention-based layer projection for knowledge distillation. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, pages 13657–13665, 2021.

[46] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. Foundations and Trends® in Machine Learning, 11(5-6):355–607, 2019.

[47] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In Proceedings of the 36th International Conference on Machine Learning, pages 5142–5151, 2019.

[48] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In Proceedings of the 6th International Conference on Learning Representations, 2018.

[49] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In Proceedings of the 22nd IEEE Conference on Computer Vision and Pattern Recognition, pages 413–420, 2009.

[50] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In Proceedings of the 3rd International Conference on Learning Representations, 2015.

[51] Yang Shu, Zhi Kou, Zhangjie Cao, Jianmin Wang, and Mingsheng Long. Zoo-tuning: Adaptive transfer from a zoo of models. In Proceedings of the 38th International Conference on Machine Learning, pages 9626–9637, 2021.

[52] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In Advances in Neural Information Processing Systems 33, pages 22045–22055, 2020.

[53] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. The Annals of Mathematical Statistics, 35(2):876–879, 1964.

[54] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In Advances in Neural Information Processing Systems 30, pages 4077–4087, 2017.

[55] Suraj Srinivas and François Fleuret. Knowledge transfer with jacobian matching. In Proceedings of the 35th International Conference on Machine Learning, pages 4723–4731, 2018.

[56] Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H Chi, and Sagar Jain. Understanding and improving knowledge distillation. CoRR, abs/2002.03532, 2020.

[57] Ilya O. Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In Advances in Neural Information Processing Systems 34, pages 24261–24272, 2021.

[58] Anh T Tran, Cuong V Nguyen, and Tal Hassner. Transferability and hardness of supervised classification tasks. In Proceedings of the 17th IEEE/CVF International Conference on Computer Vision, pages 1395–1405, 2019.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems 30, pages 5998–6008. 2017.

[60] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In Advances in Neural Information Processing Systems 29, pages 3630–3638. 2016.

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

[62] Guodong Xu, Ziwei Liu, Xiaoxiao Li, and Chen Change Loy. Knowledge distillation meets self-supervision. In Proceedings of the 16th European Conference on Computer Vision, pages 588–604, 2020.

[63] Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen, and Jindong Wang. Reliable weighted optimal transport for unsupervised domain adaptation. In Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4394–4403, 2020.
[64] Han-Jia Ye, Su Lu, and De-Chuan Zhan. Distilling cross-task knowledge via relationship matching. In Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12396–12405, 2020.

[65] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4133–4141, 2017.

[66] Kaichao You, Yong Liu, Jianmin Wang, Michael I Jordan, and Mingsheng Long. Ranking and tuning pre-trained models: A new paradigm of exploiting model hubs. CoRR, abs/2110.10545, 2021.

[67] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In Proceedings of the 38th International Conference on Machine Learning, pages 12133–12143, 2021.

[68] Kaiyu Yue, Jiangfan Deng, and Feng Zhou. Matching guided distillation. In Proceedings of the 16th European Conference on Computer Vision, volume 12360, pages 312–328, 2020.

[69] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. CoRR, abs/1605.07146, 2016.

[70] Zhi-Hua Zhou. Learnware: on the future of machine learning. Frontiers Computer Science, 10(4):589–590, 2016.
A Proof of Proposition 4.2

Proposition A.1 (Gradient of Sinkhorn Distance, Restate). Let $p_T = \rho_T(f_T(x))$ and $p_S = \rho_S(f_S(x))$ be the output probability distributions of teacher and student respectively. Let $\beta^*$ be one optimal solution to Equation (9). The gradient of $S_\epsilon(p_T, p_S)$ w.r.t. $p_S$ and $f(x)$ can be written in the following forms where $\odot$ means element-wise multiplication:

\[
\frac{\partial S_\epsilon(p_T, p_S)}{\partial p_S} = \beta^*,
\]

\[
\frac{\partial S_\epsilon(p_T, p_S)}{\partial f(x)} = (\beta^* - \langle \beta^*, p_S \rangle) \odot p_S.
\]

In order to give a detailed proof of Proposition A.1, we first review the optimization problem of Sinkhorn distance and its dual form. Sinkhorn distance between teacher’s output label distribution and student’s output label distribution can be written as

\[
\min_{T \in \Pi(p_T, p_S)} \langle T, M \rangle - \epsilon H(T).
\]

The dual problem of Equation (8) is Equation (9):

\[
\max_{\alpha, \beta} \alpha^T p_T + \beta^T p_S - \epsilon \sum_{m,n} \exp \left( \frac{\alpha_m + \beta_n - M_{mn}}{\epsilon} \right).
\]

With existence of entropic regularization term $-\epsilon H(T)$, the target function of Equation (8) is $\epsilon$-strictly convex, resulting in strong duality between Equation (8) and Equation (9). Therefore, the gradient of Sinkhorn distance w.r.t. student’s output probability $p_S$ can be directly obtained by dual form given the differentiability of Sinkhorn distance [33], say

\[
\frac{\partial S_\epsilon(p_T, p_S)}{\partial p_S} = \frac{\partial \beta^T p_S}{\partial p_S} = \beta.
\]

This means when we have solved an optimal transport problem and obtain optimal dual variable $\alpha^*$ and $\beta^*$, we can use $\beta^*$ to update the class probability distribution of student. The remaining part of Proposition A.1 can be directly obtained by chain rule.

B Proof of Proposition 4.3

Proposition B.1 (Convergence of Gradient, Restate). Let $u^{(t)}$ and $v^{(t)}$ be the potential vectors after Sinkhorn iterations. Let $\beta^{(t)}$ be the approximate gradient w.r.t. student’s output probability $p_S$ based on $u^{(t)}$ and $v^{(t)}$. Let $\beta^*$ be the accurate gradient. $\beta^{(t)}$ has a linear convergence rate $\kappa(K)^2$ in variation seminorm $\|z\|_{\text{var}} = \max(z) - \min(z)$, as shown in Equation (11). $\kappa(K)$ is a constant about $K$ and $0 < \kappa(K) < 1$.

\[
\frac{\|\beta^{(t+1)} - \beta^*\|_{\text{var}}}{\|\beta^{(t)} - \beta^*\|_{\text{var}}} \leq \kappa(K)^2.
\]

Our proof is build on previous convergence analysis of potential vectors in Sinkhorn algorithm [34] and a fundamental theorem proofed in [5]. In order to give the detailed proof of Proposition B.1, we first introduce some basic concepts.

Definition B.2 (Hilbert Projective Metric). Let $x$ and $y$ be two vectors with positive values, i.e., $x, y \in \mathbb{R}_{++}^d$. The Hilbert projective metric over $x$ and $y$ is defined as Equation (12):

\[
d_{\text{HP}}(x, y) = \log \max_{i,j} \frac{x_i y_j}{x_j y_i}.
\]
We can see that \( d_{\text{HP}}(x, y) = 0 \) if and only if there exists some \( t > 0 \) such that \( x = ty \). An important property of Hilbert projective metric is invariance with respect to element-wise division, i.e., \( d_{\text{HP}}(x, y) = d_{\text{HP}}(x/y, 1) = d_{\text{HP}}(1, x/y) \). This can be easily proven from the definition of Hilbert projective metric.

Also, we have \( d_{\text{HP}}(x, y) = \| \log(x) - \log(y) \|_{\text{var}} \) which is the variation seminorm of the difference between logarithm of two vectors. This relationship can be shown based on the definitions of variation seminorm and Hilbert projective metric.

**Theorem B.3** ([5]). Let \( K \in \mathbb{R}^{R_1 \times R_2} \) be a matrix with positive values. Define \( \psi(K) = \max_{s,t,j,k,l} K_{sk} K_{tl} K_{jk} K_{il} \) and \( \kappa(K) = \frac{\sqrt{\psi(K)}}{\sqrt{\psi(K) + 1}} \). Then for any pair of vectors \( x, y \in \mathbb{R}^{R_2}_{++} \), we have

\[
\| \beta^{(t+1)} - \beta^* \|_{\text{var}} = \| \log(\beta^{(t+1)}) - \log(\beta^*) \|_{\text{var}} \text{ Sinkhorn algorithm}
\]

\[
= \epsilon \| \log(\beta^{(t+1)}) - \log(\beta^*) \|_{\text{var}} \quad \text{(linearity of variation seminorm)}
\]

\[
= \epsilon d_{\text{HP}}(\beta^{(t+1)}, \beta^*) \quad \text{(relation between \( \| \cdot \|_{\text{var}} \) and \( d_{\text{HP}}(\cdot, \cdot) \))}
\]

\[
= \epsilon d_{\text{HP}}(K^\top u^{(t+1)}, K^\top u^*) \quad \text{(Sinkhorn iteration)}
\]

\[
= \epsilon d_{\text{HP}}(K^\top u^{(t+1)}, K^\top u^*) \quad \text{(property of \( d_{\text{HP}}(\cdot, \cdot) \))}
\]

\[
\leq \epsilon \kappa(K) d_{\text{HP}}(u^{(t+1)}, u^*) \quad \text{(Theorem B.3)}
\]

On the other hand, we have

\[
\epsilon \ d_{\text{HP}}(u^{(t+1)}, u^*) = \epsilon \ d_{\text{HP}}(K^\top v^{(t)}, K^\top v^*) \quad \text{(Sinkhorn iteration)}
\]

\[
= \epsilon \ d_{\text{HP}}(K^\top v^{(t)}, K^\top v^*) \quad \text{(property of \( d_{\text{HP}} \))}
\]

\[
\leq \kappa(K) \epsilon \ d_{\text{HP}}(v^{(t)}, v^*) \quad \text{(Theorem B.3)}
\]

\[
= \kappa(K) \epsilon \| \log(v^{(t)}) - \log(v^*) \|_{\text{var}} \quad \text{(relation between \( \| \cdot \|_{\text{var}} \) and \( d_{\text{HP}}(\cdot, \cdot) \))}
\]

\[
= \kappa(K) \epsilon \| \log(v^{(t)}) - \log(v^*) \| \quad \text{(linearity of variation seminorm)}
\]

\[
= \kappa(K) \| \beta^{(t)} - \beta^* \|_{\text{var}} \quad \text{(Sinkhorn algorithm)}
\]

(15)

Substituting Equation (15) into Equation (14) yields \( \| \beta^{(t+1)} - \beta^* \|_{\text{var}} \leq \kappa(K)^2 \| \beta^{(t)} - \beta^* \| \), which ends the proof of Proposition B.1. The proposition can also be written as \( \| \beta^{(t)} - \beta^* \|_{\text{var}} = O(\kappa(K)^{2t}) \).

### C Experiment Details of Figure 2

In this part we give experimental details of Figure 2 in the main body, and show more results. In this experiment, we use a well-trained WideResNet-(40,2) [69] as teacher model and a WideResNet-(16,1) as student model. We randomly sample a mini-batch of instances (256 instances) from the training split of CIFAR-100, and forms 256 optimal transport problems. The cost matrix \( M \) is computed by teacher’s embedding network (refer to Section 4.1 for details). For \( i \)-th instance, the teacher’s output
probability is \( \rho_\tau(f_T(x_i)) \), and the student’s output probability is \( \rho_\tau(f_S(x_i)) \). Hyper-parameter \( \tau \) is set to 3 and hyper-parameter \( \epsilon \) is set to 0.1. We need to solve the following dual form of optimal transport problem:

\[
\max_{\alpha, \beta} \alpha^T \rho_\tau(f_T(x_i)) + \beta^T \rho_\tau(f_S(x_i)) - \epsilon \sum_{m, n} \exp\left(\frac{\alpha_m + \beta_n - M_{mn}}{\epsilon}\right).
\]

When using Sinkhorn algorithm to solve Equation (16), we run 100 iterations and record \( \beta^{(0)}, \beta^{(1)}, \ldots, \beta^{(100)} \). The last vector is considered as optimal solution to the problem, i.e., \( \beta^* = \beta^{(100)} \).

There are two curves in Figure 2, one denotes the variation seminorm of the difference between \( \beta^{(t)} \) and \( \beta^* \) which is computer according to the definition, and another denotes the theoretical upper bound derived from Proposition B.1. In detail, we have \( \|\beta^{(t)} - \beta^*\|_{\text{var}} \leq \|\beta^{(0)} - \beta^*\|_{\text{var}} \kappa(K)^{2t} \) where \( K = \exp\left(\frac{M}{\tau}\right) \) and \( \kappa \) is defined in Theorem B.3. We record the value at right hand side during iterations. Finally, two groups of values are average across all 256 tasks, and we plot them in Figure 2. We also sample more mini-batches to draw more figures like Figure 2, as shown in Figure 7.

D Algorithm Schemes

Our proposed FADiOT contains two main phases, i.e., Teacher Assessment (TA) and Generalized Knowledge Reuse (GKR), which are discussed in Section 5 and Section 4, respectively. In this part, we give detailed algorithm schemes of TA, GKR, and the whole FADiOT. Algorithm 1 and Algorithm 2 are algorithms schemes of TA and GKR. Algorithm 3 is the algorithm scheme of FADiOT. We also state the accelerated Sinkhorn algorithm in Algorithm 4.

E Experiments about Fictitious Student

In Section 5 of the main body, we have discussed how to measure the relatedness of pre-trained teachers efficiently. In our proposed method, we construct a fictitious student to mimic the behaviour of the true student and then measure the relatedness between the predictions between the teacher and the fictitious student. Although our proposed method to assess teachers is efficient, it is not clear how the fictitious student affects the results of teacher assessment. Thus, in this part, we conduct more experiments on the fictitious student to check its rationality.

Firstly, we empirically check the consistency between the fictitious student and the true student on Caltech-UCSD Birds-200-2011 (CUB) [61]. Experimental details are same as those described in Section H. In Table 6, we list the KL divergence between the outputs of the fictitious student and the true student on 5 different tasks of CUB. We can see that the gap between the fictitious student and the true student is acceptable. In this experiment, the teacher model is pre-trained on Task 1, and the task gap between Task \( i \) and Task 1 increases with \( i \) (Refer to Section G for details). We can further see that the KL divergence between the fictitious student and the true student increases with
Algorithm 1 Teacher assessment process in FADiOT.

**Input:** Number of teachers $H$, Faculty $\{f^h_T\}_{h=1}^H$, Training set of the student $D = \{x_i, y_i\}_{i=1}^N$.
Teacher’s class centers $\{e_{T,m}\}$, Hyper-parameters $\epsilon$ and $\tau$.

**Output:** The selected teacher index $h^*$.

Initialize $h^* = 1$;
Initialize $M^* = +\infty$;

for $h \in [H]$ do

Extract feature representations $\{\phi^h_T(x_i)\}_{i=1}^N$ with the embedding network $\phi_T$ of $f^h_T$;
Obtain student’s class centers $\{e_{S,n}\}$;
Train a linear classifier $f^h_F$ on dataset $\{\phi^h_T(x_i), y_i\}_{i=1}^N$;

Compute cost matrix $M$ where $M_{mn} = \|e_{T,m} - e_{S,n}\|_2$;
Compute $M(f^h_T) = \frac{1}{N} \sum_{i=1}^N S_\epsilon(\rho_\tau(f^h_T(x_i)), \rho_\tau(f^h_F(x_i)))$ through Algorithm 4;
if $M(f^h_T) < M^*$ then
   $M^* \leftarrow M(f^h_T)$;
   $h^* \leftarrow h$;
end if
end for
return $h^*$.

Algorithm 2 Generalized knowledge reuse process in FADiOT.

**Input:** A selected teacher $f_T$. Training set of the student $D = \{x_i, y_i\}_{i=1}^N$.
Teacher’s class centers $\{e_{T,m}\}$, Hyper-parameters $\epsilon$, $\tau$, and $\lambda$.

**Output:** Trained student $f_S$.

Randomly initialize $f_S$;
Obtain student’s class centers $\{e_{S,n}\}$ with teacher’s embedding network $\phi_T$;
Compute cost matrix $M$ where $M_{mn} = \|e_{T,m} - e_{S,n}\|_2$;

while not converge do
   Compute $L_1 = \sum_{i=1}^N \ell(y_i, f_S(x_i))$;
   Compute $L_2 = \sum_{i=1}^N S_\epsilon(\rho_\tau(f^h_T(x_i)), \rho_\tau(f^h_F(x_i)))$ through Algorithm 4;
   $L \leftarrow L_1 + \lambda L_2$;
   Use total loss $L$ to update $f_S$;
end while
return $f_S$.

Algorithm 3 The whole process of FADiOT.

**Input:** Number of teachers $H$, Faculty $\{f^h_T\}_{h=1}^H$, Training set of the student $D = \{x_i, y_i\}_{i=1}^N$.
Teacher’s class centers $\{e_{T,m}\}$, Hyper-parameters $\epsilon$, $\tau$, and $\lambda$.

**Output:** Trained student $f_S$.

Obtain $h^*$ through Algorithm 1;
Fetch the selected teacher $f^h_T$;
Train the student $f_S$ through Algorithm 2;
return $f_S$. 
Algorithm 4  The sinkhorn algorithm for gradient computation.

**Input:** Cost matrix $M$, Teacher’s output probability distribution $p_T$, Student’s output probability distribution $p_S$, Hyper-parameter $\epsilon$, Maximum number of fixed point iterations $I$;

**Output:** Sinkhorn loss $S_\epsilon(p_T, p_S)$.

Randomly initialize $u$ and $v$;

$T \leftarrow \text{diag}(u)K\text{diag}(v)$;

Compute matrix $K = \exp\left(\frac{-M}{\epsilon}\right)$;

$i \leftarrow 0$;

while not converge do

$u \leftarrow P_T/(Kv)$;

$v \leftarrow P_S/(K^TU)$;

$T \leftarrow \text{diag}(u)K\text{diag}(v)$;

$i \leftarrow i + 1$;

if $i \geq I$ then

Break;

end if

end while

return $\langle T, M \rangle - \epsilon \bar{H}(T)$

---

Table 6: KL divergence between outputs of $f_F$ and $f_S$ on 5 CUB tasks. The gap is acceptable. Note that the teacher is trained on Task 1, and the KL gap increases when task semantic gap increases.

| CUB Task | 1   | 2   | 3   | 4   | 5   |
|----------|-----|-----|-----|-----|-----|
| KL($f_F || f_S$) | 0.32 | 0.41 | 0.39 | 0.47 | 0.53 |

---

F  Descriptions of Datasets

In this section, we give detailed descriptions of two datasets, i.e., CIFAR-100 [29] and Caltech-UCSD Birds-200-2011 (CUB) [61]. CIFAR-100 is a small dataset which contains 100 class.
The selected class subset changes with the state of sliding window.

Figure 8: Illustration of the “sliding window” protocol. In a given dataset, we first order its classes randomly, and then use a sliding window to choose the class subset. While the sliding window moves forward, the selected class subset changes.

32 × 32 images per class. In each class, there are 500 images for training and 100 images for test. In CIFAR-100, there are 20 superclasses and 5 classes in each superclass. CUB is a fine-grained dataset with 200 different species of birds. It contains 200 categories, 11788 images with label annotations and bounding boxes. In this work, we mainly focus on classification, and the bounding boxes are not used at all.

G Sliding Window Protocol for Dataset Split

In our experiments, three kinds of dataset split are considered. They are used for standard knowledge distillation, generalized knowledge distillation, and model assessment.

(1) Standard Knowledge Distillation. In standard knowledge distillation, the teacher and student are trained on a same label set. Thus, we consider all classes in three datasets, and follow the standard split of these datasets to obtain training dataset and test dataset.

(2) Generalized Knowledge Distillation. When we need to consider generalized knowledge distillation, we pick a class subset from the whole class set of a dataset. Taking CIFAR-100 as an example, it contains 100 classes, and we pick a subset that contains 50 classes to form a new class set. After ordering these 100 classes, the subset construction follows a “sliding window” protocol as shown in Figure 8. For CIFAR-100, the window covers 50 classes, and its step size is 10 classes. Therefore, we can generate 6 different class subsets. For CIFAR-100, the window covers 50 classes, and its step size is 10 classes, which offers us 6 different class subsets. For CUB, the window covers 100 classes, and its step size is 25 classes, which offers us 5 different class subsets. In generalized knowledge distillation, we fix the class subset of teacher as the one selected by the initial sliding window, and change the class subsets of student within all possible class subsets. This means we need to learn a number of students that have different class overlap ratios with the teacher. Taking CIFAR-100 as an example again, when the class subset of student is determined by the initial sliding window, student model targets at a totally same label set as teacher (overlap ratio 100%), and the setting degenerates to standard knowledge distillation. However, when the class subset of student is determined by the terminal sliding window, student model targets at a totally different label set from teacher (overlap ratio 0%), and the task gap between teacher and student is largest. For CIFAR-100, class overlap ratio $g \in \{100\%, 80\%, 60\%, 40\%, 20\%, 0\%\}$ for different students. For CUB, $g \in \{100\%, 75\%, 50\%, 25\%, 0\%\}$.

(3) Teacher Assessment. When we consider teacher assessment, we expand the ‘sliding window’ protocol to ‘double sliding window’ protocol as shown in Figure 9. Obviously, now both teacher and student may have different label spaces, and this offers us different teacher models to form the faculty. The window sizes for CIFAR-100, and CUB are 50 classes and 100 classes respectively. The step sizes of both sliding windows are 10 classes and 25 classes for CIFAR-100, and CUB respectively.

H Implementation Details

In this section, we describe the implementation details in our experiments.
Teacher Pre-Training. Given a class subset and a specific architecture, we first randomly initialize the parameters of the teacher model, and then train the teacher using standard cross-entropy loss. The teacher architecture is WideResNet-(40,2) [69] for CIFAR-100 and MobileNet-1.0 [21] for CUB in most experiments. In teacher assessment part, we also use ResNet-50 [19] as our teacher model. We train the teacher model for 200 epochs. The teacher is optimized using SGD optimizer with initial learning rate 0.1. For CIFAR-100, the learning rate is multiplied by 0.2 after 50, 100, and 150 epochs. For CUB, the learning rate is multiplied by 0.2 after 150, 170, 180 epochs. The batch size is 256 for CIFAR-100 and 128 for CUB. As for optimizer hyper-parameters, weight decay is set to 0.0005 and momentum is set to 0.9.

Generalized Knowledge Distillation. In generalized knowledge distillation, teacher and student may target at different label spaces, and we follow “sliding window protocol” (Figure 8) to sample several class subsets for different students. The class subset for teacher is fixed. Teacher architecture is WideResNet-(40,2) [69] for CIFAR-100 and MobileNet-1.0 [21] for CUB. Student architectures are WideResNet-\{(40,2), (16,2), (40,1), (16,1)\} for CIFAR-100 and MobileNet-\{1.0, 0.75, 0.5, 0.25\} for CUB. We train each student with SGD optimizer for 200 epochs. Weight decay is set to 0.0005 and momentum is set to 0.9. The initial learning rate is set to 0.1. For CIFAR-100, the learning rate is multiplied by 0.2 after 50, 100, and 150 epochs. For CUB, the learning rate is multiplied by 0.2 after 150, 170, 180 epochs. The batch size is 256 for CIFAR-100 and 128 for CUB. Hyper-parameter $\lambda$ in Equation (3) of the main body is set to 10 for CIFAR-100 and 100 for CUB. The temperature $\tau$ of softmax function is set to 3 for both datasets. Smoothing factor $\epsilon$ in Sinkhorn distance is set to 0.1. For accelerating the computation of Sinkhorn distance, we only run 10 iterations to approximate the gradient.

Standard Knowledge Distillation. In standard knowledge distillation, teacher and student share a same label space. Teacher architecture is WideResNet-(40,2) [69] for CIFAR-100 and MobileNet-1.0 [21] for CUB. Student architectures are WideResNet-\{(40,2), (16,2), (40,1), (16,1)\} for CIFAR-100 and MobileNet-\{1.0, 0.75, 0.5, 0.25\} for CUB. We train each student with SGD optimizer for 200 epochs. Weight decay is set to 0.0005 and momentum is set to 0.9. The initial learning rate is set to 0.1. For CIFAR-100, the learning rate is multiplied by 0.2 after 50, 100, and 150 epochs. For CUB, the learning rate is multiplied by 0.2 after 150, 170, 180 epochs. The batch size is 256 for CIFAR-100 and 128 for CUB. Hyper-parameter $\lambda$ in Equation (3) in the main body is set to 10 for CIFAR-100 and 100 for CUB. The temperature $\tau$ of softmax function is set to 3 for both datasets. Smoothing factor $\epsilon$ in Sinkhorn distance is set to 0.1. For accelerating the computation of Sinkhorn distance, we only run 10 iterations to approximate the gradient.

Teacher Assessment. In teacher assessment part, we adopt a “double sliding window protocol” as shown in Figure 9. Teacher architectures are WideResNet-(40,2) [69] and ResNet-50 [19] for CIFAR-100 and MobileNet-1.0 [21] and ResNet-50 [64] for CUB. Student architectures are WideResNet-\{(40,2), (16,2), (40,1), (16,1)\} for CIFAR-100 and MobileNet-\{1.0, 0.75, 0.5, 0.25\} for CUB. We train each student with SGD optimizer for 200 epochs. Weight decay is set to 0.0005 and momentum is set to 0.9. The initial learning rate is set to 0.1. For CIFAR-100, the learning rate is multiplied by 0.2 after 50, 100, and 150 epochs. For CUB, the learning rate is multiplied by 0.2 after 150, 170, 180 epochs. The batch size is 256 for CIFAR-100 and 128 for CUB. Hyper-parameter $\lambda$ in Equation (3) in the main body is set to 10 for CIFAR-100 and 100 for CUB. The temperature $\tau$ of softmax function is set to 3 for both datasets. Smoothing factor $\epsilon$ in Sinkhorn distance is set to 0.1. For accelerating the computation of Sinkhorn distance, we only run 10 iterations to approximate the gradient.

Figure 9: Illustration of the “double sliding window” protocol. In a given dataset, we first order its classes randomly, and then use two sliding windows to choose the class subsets of teacher and student. While two sliding windows move forward, the selected class subsets of teacher and student change.
the computation of Sinkhorn distance, we only run 10 iterations to approximate the gradient. After obtaining well-trained students with assistance of every single teacher in the faculty, we consider their test accuracies as ground truth, and use several metrics to select the most contributive teacher. Assuming that we have $H$ teachers, given a target task (a class subset for student), we denote by $G^h$ the test accuracy of the student trained with assistance of $h$-th teacher. We use $Q^h$ to represent some evaluation metric of $h$-th teacher. Ideally, $Q$ and $G$ should be highly correlated. The Pearson correlation coefficient between $Q$ and $G$ can be computed as
\[
\eta(Q,G) = \frac{\text{cov}(Q,G)}{\sigma_Q \sigma_G}
\]
where $\text{cov}(Q,G)$ is the covariance between $G$ and $G$, and $\sigma_Q$ and $\sigma_G$ are standard deviations of $Q$ and $G$ respectively.

I Generalized Knowledge Distillation: Values in Figure 4

In this section, we list all values for drawing Figure 4 in the main body. Table 16 and Table 17 are results of generalized knowledge distillation on CIFAR-100 and CUB respectively. Since standard KD [20] does not work when teacher and student target at different label spaces, we compare our FA DiOT to several feature-based distillation methods including RKD [44], AML [7], and ReFilled [64]. Two OT-based methods, i.e., MGD [68] and WCoRD [8] are also considered. We can see that FA DiOT outperforms other methods in most cases. Among all compared methods, ReFilled [64] is the only one designed for cross-task knowledge distillation, and it also achieves competitive performance. All these compared methods ignore teacher’s classifier and only utilizes its representation ability, which is the main reason for their unsatisfying performances. In Table 16 and Table 17, we can see that deeper students often achieve higher test accuracies. Both training set and test set change with class overlap ratio $g$, and this makes accuracies across different overlap ratios incomparable.

J Standard Knowledge Distillation: Different Model Families

In Section 6.1 of the main body, we not only check the student’s performances when the teacher has a different label space from student, but also simplify the setting to standard knowledge distillation. As we know, standard knowledge distillation is a special case of generalized knowledge distillation, and our proposed FA DiOT can also handle this case. In Table 1, we list student’s performances on CIFAR-100 and CUB when the teacher shares a same label space with student. However, we conduct these experiments under ‘same-family’ setting which means the architectures of teacher and student come from a same family, i.e., both teacher and student are WideResNets or MobileNets.

In this section, we conduct a complementary experiment where teacher and student do not share a same architecture family. Specifically, we set teacher as ResNet-50 [19] for both CIFAR-100 and CUB. Results are shown in Table 8 and Table 9. In Table 8, we can see that the accuracy of teacher (72.17) is lower than that of a WideResNet-(40,2) student (74.44). Thus, the improvements of all distillation methods in the first column are limited. Our proposed FA DiOT achieves best accuracy in most cases, showing that FA DiOT can distill the knowledge of a cross-family teacher. Similar phenomenon can be observed in Table 9.

Table 8: Average test accuracies on CIFAR-100. Teacher architecture is ResNet-50. Its accuracy on test set is 72.17. Teacher and student share a same label space. † means self-supervised methods. Best results are in bold.

| (Depth, Width) | (40, 2) | (16, 2) | (40, 1) | (16, 1) |
|---------------|---------|---------|---------|---------|
| Teacher       | 72.17   | 74.44   | 70.15   | 68.97   | 65.44   |
| Student       |         |         |         |         |         |
| KD [20]       | 74.15   | 71.23   | 69.30   | 66.12   |
| FitNet [50]   | 74.00   | 70.22   | 68.06   | 64.72   |
| VID [2]       | 74.25   | 72.83   | 71.00   | 65.76   |
| RKD [44]      | 74.82   | 71.99   | 71.35   | 65.20   |
| SEED† [14]    | **75.03** | 72.25   | 70.93   | 66.44   |
| SSKD† [62]    | 74.86   | 72.75   | 71.96   | 66.87   |
| ReFilled [64] | 74.92   | 72.57   | 71.85   | 66.90   |
| FA DiOT       | 74.85   | **73.60** | **72.14** | **67.32** |
Table 9: Average test accuracies on CUB. Teacher architecture is ResNet-50. Its accuracy on test set is 74.29. Teacher and student share a same label space. † means self-supervised methods. Best results are in bold.

| Width Multiplier | 1.0  | 0.75  | 0.5  | 0.25  |
|------------------|------|-------|------|-------|
| Teacher          | 74.29| 74.87 | 72.41| 69.72 |
| Student          | 75.36| 74.87 | 72.41| 69.72 |
| KD [20]          | 75.66| 74.50 | 72.50| 70.92 |
| FitNet [50]      | 75.20| 74.28 | 72.66| 70.52 |
| VID [2]          | 75.82| 74.14 | 72.50| 70.92 |
| RKD [44]         | 75.29| 74.34 | 72.87| 71.02 |
| SEED† [14]       | 76.23| 75.12 | 73.49| 71.88 |
| SSKD† [62]       | 76.34| 75.22 | 73.80| 72.35 |
| ReFilled [64]    | 76.24| 75.50 | 74.97| 73.87 |
| FADiOT           | 75.92| 75.67 | 75.14| 73.99 |

K Generalized Knowledge Distillation: Ablation Study

In this section, we give some important ablation studies about our proposed FADiOT under the setting of generalized knowledge distillation. When distilling the knowledge from a cross-task teacher, we utilize Sinkhorn distance to bridge the support gap between two predictions. Here an important component is the cost matrix $M$. In FADiOT, we propose to construct the cost matrix $M$ by the teacher’s embedding network $\phi_T$. Now we try some other cost matrices and report model performance on both CIFAR-100 and CUB. Specifically, two more kinds of cost matrices are considered:

- Constant cost matrix: $M_{mn} = 1$ if $m \neq n$ and $M_{mn} = 0$ if $m = n$. In this case, the semantic relationships between source classes and target classes are ignored;
- Cost matrix generated by randomly initialized embedding network: we first compute class centers by a randomly initialized embedding network which has the same structure as $\phi_T$, and then compute the Euclidean distances between class centers to determine $M$.

Results are shown in Table 10. This experiment is performed using class overlap ratio $g = 60\%$ for CIFAR-100 and $g = 50\%$ for CUB. We can see that our proposed cost matrix brings us best performances. This is because $M$ constructed by teacher’s embedding network can capture semantic information of both source classes and target classes well.

Table 10: Ablation study on cost matrix. We report student’s test accuracies on both CIFAR-100 and CUB. Three kinds of cost matrices are considered. For CIFAR-100, class overlap ratio $g$ is set to 60%. For CUB, class overlap ratio $g$ is set to 50%. Best results are in bold.

| (Depth, Width) | (40, 2) | (16, 2) | (40, 1) | (16, 1) | CIFAR-100 ($g = 60\%$) | 0.25 |
|----------------|---------|---------|---------|---------|-------------------------|------|
| Constant       | 76.42   | 71.50   | 73.33   | 66.69   | Constant                | 61.42|
| Random         | 75.17   | 72.38   | 73.69   | 67.95   | Random                  | 62.54|
| FADiOT         | 80.51   | 76.92   | 77.87   | 70.34   | FADiOT                  | 66.17|

L Generalized Knowledge Distillation: Influence of Hyper-Parameters

In this section, we study the influence of several important hyper-parameters in FADiOT. To be specific, there are three hyper-parameters under our consideration, i.e., temperature $\tau$, weight of distillation term $\lambda$, and regularization strength $\epsilon$ in Sinkhorn distance. All these hyper-parameters appear in our objective Equation (3). Now, we range $\tau \in \{0.1, 1, 2, 3\}$, $\lambda \in \{1, 10, 100, 1000\}$, $\epsilon \in \{0.01, 0.1, 1\}$, and show their influences on model performance. This experiment is conducted on CIFAR-100 (class overlap ratio $g = 60\%$) and CUB (class overlap ratio $g = 50\%$). The student architecture is fixed as WideResNet-(16,1) for CIFAR-100 and MobileNet-0.25 for CUB. Results are shown in Figure 10a, Figure 10b, and Figure 10c. We can see that weight of distillation term has a remarkable influence on model performance, and we set it to different values for different datasets. $\epsilon$ controls the strength of entropy regularization term in Sinkhorn distance, and a large $\epsilon$ tends to
Figure 10: Influence of three hyper-parameters in FADtOT. Without specification, $\tau$ is usually set to 3 for both datasets. $\lambda$ is usually set to 10 for CIFAR-100 and 100 for CUB. $\epsilon$ is usually set to 0.1 for both datasets.
Table 11: Values for drawing the first column of Figure 5.

| Teacher Id | Student Accuracy | NCE [58] | LEEP [40] | LogME [67] | FAdIO T |
|------------|------------------|----------|-----------|------------|---------|
| 1          | 65.72            | 1.00     | 0.91      | 0.99       | 1.00    |
| 2          | 64.47            | 0.28     | 0.68      | 0.73       | 0.73    |
| 3          | 60.19            | 0.80     | 0.49      | 0.50       | 0.49    |
| 4          | 63.59            | 0.88     | 0.28      | 0.26       | 0.24    |
| 5          | 62.60            | 0.92     | 0.09      | 0.04       | 0.03    |
| 6          | 68.75            | 0.99     | 0.09      | 0.82       | 0.88    |
| 7          | 65.13            | 0.24     | 0.08      | 0.63       | 0.66    |
| 8          | 63.81            | 0.74     | 0.06      | 0.43       | 0.45    |
| 9          | 62.28            | 0.80     | 0.04      | 0.22       | 0.22    |
| 10         | 60.30            | 0.87     | 0.01      | 0.01       | 0.03    |

Table 12: Values for drawing the second column of Figure 5.

| Teacher Id | Student Accuracy | NCE [58] | LEEP [40] | LogME [67] | FAdIO T |
|------------|------------------|----------|-----------|------------|---------|
| 1          | 64.17            | 0.22     | 0.67      | 0.71       | 0.70    |
| 2          | 69.59            | 1.00     | 0.92      | 0.99       | 0.99    |
| 3          | 64.21            | 0.34     | 0.72      | 0.74       | 0.73    |
| 4          | 63.55            | 0.73     | 0.46      | 0.48       | 0.47    |
| 5          | 59.71            | 0.86     | 0.28      | 0.26       | 0.25    |
| 6          | 62.78            | 0.22     | 0.07      | 0.59       | 0.63    |
| 7          | 66.63            | 0.99     | 0.10      | 0.85       | 0.90    |
| 8          | 64.10            | 0.33     | 0.08      | 0.63       | 0.66    |
| 9          | 60.59            | 0.59     | 0.07      | 0.43       | 0.44    |
| 10         | 58.61            | 0.78     | 0.04      | 0.21       | 0.23    |

decrease the effect of semantic transport. \( \tau \) smooths the output probability distributions of teacher and student, and a proper \( \tau \) will improve model performance to some extent.

M **Teacher Assessment: Values in Figure 5**

In this section, we list all values for drawing Figure 5 in the main body. For each target task (each column of Figure 5), we give the accuracies of 10 students trained with assistance of 10 different teachers (x-axis values in each column of Figure 5) and four metrics to evaluate 10 teachers (y-axis values in each column). All the metrics are normalized into \([0, 1]\) for convenience. These values are listed in Table 11, Table 12, Table 13, Table 14, and Table 15.
Table 13: Values for drawing the third column of Figure 5.

| Teacher Id | Student Accuracy | NCE [58] | LEEP [40] | LogME [67] | FaDiOT |
|------------|------------------|----------|-----------|------------|--------|
| 1          | 66.17            | 0.80     | 0.47      | 0.46       | 0.45   |
| 2          | 63.36            | 0.11     | 0.70      | 0.73       | 0.72   |
| 3          | 71.28            | 1.00     | 0.95      | 1.00       | 0.99   |
| 4          | 64.24            | 0.36     | 0.62      | 0.69       | 0.69   |
| 5          | 59.51            | 0.82     | 0.45      | 0.45       | 0.44   |
| 6          | 63.14            | 0.74     | 0.05      | 0.39       | 0.41   |
| 7          | 66.66            | 0.20     | 0.08      | 0.63       | 0.66   |
| 8          | 68.97            | 0.99     | 0.10      | 0.85       | 0.90   |
| 9          | 64.24            | 0.30     | 0.08      | 0.61       | 0.64   |
| 10         | 65.78            | 0.77     | 0.06      | 0.37       | 0.39   |

Table 14: Values for drawing the fourth column of Figure 5.

| Teacher Id | Student Accuracy | NCE [58] | LEEP [40] | LogME [67] | FaDiOT |
|------------|------------------|----------|-----------|------------|--------|
| 1          | 71.12            | 0.88     | 0.25      | 0.23       | 0.21   |
| 2          | 66.81            | 0.71     | 0.47      | 0.50       | 0.48   |
| 3          | 65.27            | 0.42     | 0.71      | 0.71       | 0.70   |
| 4          | 72.72            | 1.00     | 0.88      | 0.97       | 0.98   |
| 5          | 65.71            | 0.16     | 0.69      | 0.71       | 0.71   |
| 6          | 64.18            | 0.75     | 0.03      | 0.20       | 0.21   |
| 7          | 66.37            | 0.50     | 0.06      | 0.44       | 0.45   |
| 8          | 68.89            | 0.33     | 0.08      | 0.60       | 0.62   |
| 9          | 70.09            | 0.99     | 0.11      | 0.87       | 0.91   |
| 10         | 68.23            | 0.08     | 0.10      | 0.61       | 0.64   |

Table 15: Values for drawing the fifth column of Figure 5.

| Teacher Id | Student Accuracy | NCE [58] | LEEP [40] | LogME [67] | FaDiOT |
|------------|------------------|----------|-----------|------------|--------|
| 1          | 69.66            | 0.90     | 0.05      | 0.01       | 0.00   |
| 2          | 66.44            | 0.86     | 0.25      | 0.27       | 0.26   |
| 3          | 66.12            | 0.78     | 0.48      | 0.46       | 0.46   |
| 4          | 69.18            | 0.21     | 0.65      | 0.70       | 0.70   |
| 5          | 73.66            | 1.00     | 1.00      | 0.99       | 0.99   |
| 6          | 65.57            | 0.82     | 0.00      | 0.00       | 0.00   |
| 7          | 66.77            | 0.71     | 0.03      | 0.23       | 0.24   |
| 8          | 68.96            | 0.70     | 0.06      | 0.38       | 0.40   |
| 9          | 68.52            | 0.00     | 0.09      | 0.62       | 0.65   |
| 10         | 71.80            | 0.99     | 0.13      | 0.87       | 0.91   |
Table 16: Average test accuracies of students on CIFAR-100. Teacher architecture is WideResNet-(40,2). The class subset of teacher is fixed while the class subset of student changes. Best results are in **bold**.

| Class Overlap Ratio $g = 0\%$ | (Depth, Width) | (40, 2) | (16, 2) | (40, 1) | (16, 1) |
|------------------------------|----------------|---------|---------|---------|---------|
| **Student**                  |                |         |         |         |         |
| RKD                          | 81.46          | 79.23   | 78.80   | 73.45   |
| AML                          | 79.99          | 79.11   | 78.99   | 73.68   |
| ReFilled                     | **82.60**      | 80.70   | 80.18   | **74.42** |
| MGD                          | 79.24          | 79.31   | 77.74   | 73.79   |
| WCoRD                        | 80.29          | 80.38   | 77.83   | 73.90   |
| FA2OT                        | 81.75          | **80.91** | **81.04** | 74.33   |

| Class Overlap Ratio $g = 20\%$ | (Depth, Width) | (40, 2) | (16, 2) | (40, 1) | (16, 1) |
|------------------------------|----------------|---------|---------|---------|---------|
| **Student**                  |                |         |         |         |         |
| RKD                          | 80.34          | 76.12   | 75.43   | 70.84   |
| AML                          | 80.05          | 76.95   | 75.37   | 70.79   |
| ReFilled                     | **81.40**      | 77.82   | 77.24   | 72.28   |
| MGD                          | 80.25          | 76.25   | 75.57   | 71.22   |
| WCoRD                        | 80.50          | 77.19   | 75.77   | 71.52   |
| FA2OT                        | 81.28          | **78.14** | **77.84** | **72.45** |

| Class Overlap Ratio $g = 40\%$ | (Depth, Width) | (40, 2) | (16, 2) | (40, 1) | (16, 1) |
|------------------------------|----------------|---------|---------|---------|---------|
| **Student**                  |                |         |         |         |         |
| RKD                          | 78.66          | 75.58   | 74.69   | 69.07   |
| AML                          | 79.98          | 75.84   | 74.28   | 68.87   |
| ReFilled                     | **80.40**      | **76.26** | 75.62   | 70.08   |
| MGD                          | 79.44          | 75.24   | 74.08   | 69.13   |
| WCoRD                        | 81.18          | 76.22   | 74.50   | 69.37   |
| FA2OT                        | **81.33**      | **76.14** | **75.99** | **70.90** |

| Class Overlap Ratio $g = 60\%$ | (Depth, Width) | (40, 2) | (16, 2) | (40, 1) | (16, 1) |
|------------------------------|----------------|---------|---------|---------|---------|
| **Student**                  |                |         |         |         |         |
| RKD                          | 78.90          | 76.37   | 75.14   | 68.48   |
| AML                          | 78.69          | 76.20   | 75.50   | 68.23   |
| ReFilled                     | **80.66**      | 76.66   | 76.52   | 69.50   |
| MGD                          | 78.95          | 75.55   | 75.03   | 68.30   |
| WCoRD                        | 78.92          | 75.25   | 76.48   | 69.72   |
| FA2OT                        | 80.51          | **76.92** | **77.87** | **70.34** |

| Class Overlap Ratio $g = 80\%$ | (Depth, Width) | (40, 2) | (16, 2) | (40, 1) | (16, 1) |
|------------------------------|----------------|---------|---------|---------|---------|
| **Student**                  |                |         |         |         |         |
| RKD                          | 80.50          | 77.43   | 76.96   | 72.16   |
| AML                          | 81.21          | 77.65   | 77.34   | 72.05   |
| ReFilled                     | **82.56**      | **78.76** | 79.29   | 73.92   |
| MGD                          | 81.46          | 76.60   | 77.86   | 71.78   |
| WCoRD                        | 81.54          | 77.16   | 77.80   | 73.09   |
| FA2OT                        | 82.33          | 78.29   | **80.08** | **74.17** |

| Class Overlap Ratio $g = 100\%$ | (Depth, Width) | (40, 2) | (16, 2) | (40, 1) | (16, 1) |
|------------------------------|----------------|---------|---------|---------|---------|
| **Student**                  |                |         |         |         |         |
| RKD                          | 80.66          | 77.94   | 76.35   | 71.56   |
| AML                          | 80.52          | 78.03   | 76.82   | 72.04   |
| ReFilled                     | 81.58          | **78.60** | **78.04** | 73.52   |
| MGD                          | 80.13          | 77.64   | 76.35   | 71.09   |
| WCoRD                        | 81.50          | 77.78   | 76.85   | 71.94   |
| FA2OT                        | **82.32**      | 78.33   | 77.85   | **74.17** |
Table 17: Average test accuracies of students on CUB. Teacher architecture is MobileNet-1.0. The class subset of teacher is fixed while the class subset of student changes. Best results are in **bold**.

| Overlap Ratio | Width Multiplier | 1.0   | 0.75  | 0.5   | 0.25  |
|---------------|------------------|-------|-------|-------|-------|
| 0%            | Student          | 71.25 | 67.56 | 66.85 | 64.48 |
|               | RKD              | 72.24 | 68.42 | 66.85 | 65.74 |
|               | AML              | 72.86 | 68.79 | 68.59 | 66.83 |
|               | ReFilled         | **75.13** | **71.67** | 71.06 | 68.22 |
|               | MGD              | 72.24 | 69.43 | 69.78 | 67.13 |
|               | WCoRD            | 72.78 | 69.81 | 69.20 | 67.00 |
|               | FAD1OT           | 75.03 | 71.44 | **71.35** | **69.66** |
| 25%           | Width Multiplier | 1.0   | 0.75  | 0.5   | 0.25  |
|               | Student          | 71.30 | 71.08 | 68.56 | 65.71 |
|               | RKD              | 72.07 | 71.70 | 68.56 | 66.43 |
|               | AML              | 72.35 | 72.05 | 70.37 | 67.20 |
|               | ReFilled         | 75.09 | 73.92 | **72.99** | 70.04 |
|               | MGD              | 73.41 | 72.52 | 69.99 | 67.99 |
|               | WCoRD            | 73.97 | 72.99 | 70.89 | 67.28 |
|               | FAD1OT           | **75.52** | **74.33** | 72.85 | **71.12** |
| 50%           | Width Multiplier | 1.0   | 0.75  | 0.5   | 0.25  |
|               | Student          | 68.20 | 66.11 | 65.23 | 62.26 |
|               | RKD              | 68.72 | 66.82 | 65.58 | 62.79 |
|               | AML              | 67.94 | 67.34 | 66.29 | 63.64 |
|               | ReFilled         | 70.25 | **68.39** | **68.50** | 65.33 |
|               | MGD              | 68.10 | 67.90 | 66.81 | 64.24 |
|               | WCoRD            | 68.45 | 67.42 | 66.89 | 63.95 |
|               | FAD1OT           | **70.94** | 68.13 | 67.78 | **66.17** |
| 75%           | Width Multiplier | 1.0   | 0.75  | 0.5   | 0.25  |
|               | Student          | 65.53 | 66.73 | 64.10 | 60.81 |
|               | RKD              | 65.89 | 67.28 | 64.66 | 61.35 |
|               | AML              | 66.32 | 66.92 | 65.03 | 62.09 |
|               | ReFilled         | 67.28 | 68.35 | **66.72** | 63.03 |
|               | MGD              | 67.01 | 67.56 | 65.43 | 63.29 |
|               | WCoRD            | 66.89 | 67.60 | 66.00 | 63.11 |
|               | FAD1OT           | **67.79** | **69.20** | 66.46 | **64.17** |
| 100%          | Width Multiplier | 1.0   | 0.75  | 0.5   | 0.25  |
|               | Student          | 67.76 | 67.98 | 64.91 | 62.17 |
|               | RKD              | 67.23 | 68.25 | 65.73 | 62.04 |
|               | AML              | 67.06 | 68.35 | 66.27 | 62.69 |
|               | ReFilled         | 68.77 | 69.10 | 68.44 | 63.33 |
|               | MGD              | 67.46 | 68.35 | 67.31 | 62.50 |
|               | WCoRD            | 67.72 | 68.75 | 68.07 | 63.24 |
|               | FAD1OT           | **70.03** | **70.87** | **69.55** | **65.72** |