Knowledge Distillation with Differentiable Optimal Transport on Graph Neural Networks

Mengyao Li[1], Yanbin Liu[2], and Ling Chen[1](✉)

[1] AAII, University of Technology Sydney, Ultimo NSW 2007, AU
[2] Auckland University of Technology, Auckland 1010, NZ
ling.chen@uts.edu.au

Abstract. Knowledge distillation (KD) transfers knowledge from a large, well-trained teacher network to a smaller student network, improving student's performance without extra computational costs. Traditional KD methods usually focus on the logits or intermediate features. However, they might overlook the inherent correlation and suffer from capacity gaps due to the distinct architectures of the student and teacher. Relation-based distillation methods try to bridge these correlation but usually require a large memory bank for loss computation, being less efficient. To overcome those limitations, we propose a novel and efficient **O**ptimal **T**ransport-based **G**raph **D**istillation (OTGD) method. First, OTGD constructs attributed graphs for the teacher and student respectively, which are then utilized to capture both individual and relational knowledge through graph neural networks (GNNs). Then, we devise an innovative *differentiable optimal transport objective* to distill the teacher knowledge before and after GNNs learning, effectively incorporating both the feature-level and correlation-level knowledge. Specifically, our optimal transport objective is solved by the Sinkhorn algorithm without relying on an extra memory bank. This design makes our method efficient and numerically stable. Comprehensive experiments conducted on two benchmark datasets with diverse network architectures, and demonstrate that OTGD outperforms the state-of-the-art methods.

Keywords: Knowledge distillation · Graph neural network · Optimal transport.

1 Introduction

Deep Neural Networks (DNNs) have achieved remarkable success in a wide range of applications, including computer vision, natural language processing, and speech recognition. However, the increasing model size and computational cost associated with the high-capacity networks pose challenges for deployment in resource-constrained environments like mobile devices and embedded systems. To address this challenge, Knowledge Distillation (KD) [5] has emerged as a promising technique that transfers knowledge from a large, well-trained teacher network to a smaller, more efficient student network, thereby improving the student model's performance without incurring additional computational costs.

Distillation methods based on predicted logits and intermediate features have recently garnered widespread attention and achieved notable performance improvements. Logits-based distillation transfers information through the soft labels provided by the teacher model, with recent research focusing on temperature scaling on the student side [11,20] and exploring various KD variants [28,29]. In contrast, feature-based distillation aligns the feature activations between the teacher and student networks by extracting deep features from intermediate layers [2,13]. These intermediate features offer a rich source of information, facilitating more effective learning. Despite their achievements, these distillation methods typically extract knowledge independently, leading to two main issues. *First*, they overlook the inherent correlation between logits and features. This is particularly problematic when the teacher and student have different network architectures, as feature-based methods often suffer significant performance degradation. *Second*, the scale difference between teacher and student models results in a capacity gap, hindering the student network's ability to learn effectively.

To address these issues, relation-based distillation methods [15] have been proposed to achieve better knowledge transfer. For example, some studies [17,23] extract relationships from instances, and then keep the information derived from these instances to be consistent, regardless of the architectural differences between teacher and student networks. Although the relational information from instances provides additional structural knowledge, these methods still neglect the interaction between features and relationships, *i.e.*, the features and structure relations can affect each other during student learning.

Graphs, as natural carriers of information, capture both feature and structural information along with their intrinsic connections, which are crucial for student learning. Existing graph-based distillation methods utilize graph embeddings [1,10] or integrate additional graph neural networks (GNNs) to effectively capture and transfer relational knowledge. For example, IGR [12] constructs multi-level relational graphs from instances, aiding the student model in learning more comprehensive feature representations. HKD [30] further introduces GNNs into their method using features as nodes and instances as edges, capturing a broader scope of the teacher network's knowledge. Then HKD maximizes the mutual information (MI) between the graph representations of the teacher and student networks, which is implemented with an InfoNCE estimator [14] for contrastive learning. However, HKD adopts a large memory bank (*e.g.*, $16,384$) to estimate the InfoNCE, increasing the training cost and complexity.

To overcome these limitations, we propose a novel and efficient **O**ptimal **T**ransport-based **G**raph **D**istillation (OTGD) method. At first, OTGD constructs attribute graphs for the teacher and student models respectively, where each node represents an instance and node attributes correspond to the learned features and predicted logits. Specifically, graph edges are generated using the k-nearest neighbors (k-NN) algorithm to filter out irrelevant instances and reduce noise from the fully connected graph. Then, with attribute graphs, we adopt GNNs to create unified graph-based representations by aggregating node attributes from nearby instances on the graph. This way, both the individual and

relational knowledge from the graph are effectively extracted, thereby enhancing the learning process of the student network.

To facilitate effective knowledge distillation, we devise an innovative *differentiable optimal transport objective* to distill the teacher knowledge for student learning. While existing methods leverage a contrastive learning framework that requires large negative samples or memory bank, our objective formulates KD as an optimal feature matching problem. Specifically, the student features in each mini-batch are enforced to match their corresponding teacher features through a well-formulated optimal transport objective. The expected ideal matching, which is an identity matrix, serves as the ground-truth for model training. Our objective is applied to features before and after GNNs learning, enabling both feature-level and correlation-level knowlege transfer. Moreover, the optimal transport objective can be efficiently solved with the Sinkhorn algorithm without relying on any extra memory bank as negative samples, making our method efficient and numerically stable. Sinkhorn algorithm can be easily differentiated through tracing the Sinkhorn iteration for efficient backpropagation.

In the following sections, we first review the related work on KD and Graph-based KD methods. We then present our OTGD method, including the construction of attribute graphs and the differentiable optimal transport objective. Finally, we provide experimental results and ablation studies to validate the effectiveness of our approach, followed by concluding remarks. We conduct experiments on two benchmark datasets (CIFAR-100 and TinyImageNet) across diverse teacher and student architectures, verifying the effectiveness and efficiency of our OTGD method. Our key contributions are summarized as follows:

- We propose a novel Optimal Transport-based Graph Distillation (OTGD) method that incorporates both feature and graph-based representations, ensuring a comprehensive and flexible knowledge transfer from the teacher to the student network.
- We devise an innovative differentiable optimal transport objective in OTGD to distill teacher knowledge at both the feature-level and correlation-level. Our objective is efficient, stable and free from any large extra memory bank.
- Comprehensive experiments are performed on two benchmark datasets, using diverse architecture configurations between the teacher and student networks. The results demonstrate that our OTGD method outperforms existing state-of-the-art methods while also gains better efficiency.

2 Related Work

Knowledge Distillation Knowledge distillation (KD), introduced by Hinton et al. [5], is a neural network compression technique that transfers knowledge from a large teacher model to a smaller student model by aligning their output distributions. This process minimizes the Kullback-Leibler divergence between the teacher and student outputs, capturing semantic similarities among categories. To improve distillation, various methods have extended KD by incorporating regularization on logits [28,11] , intermediate layers [2,27], and the distillation

process [6,26]. However, feature-based distillation methods often encounter challenges due to feature misalignment caused by the size difference between the teacher and student models. Moreover, these methods typically ignore the relation among instances, which are crucial for developing a robust student model.

Relation-based knowledge distillation addresses this issue by considering instance relations. RKD [15] pioneers the concept of relation-based distillation. SP [23] focuses on preserving pairwise similarity, while CC [17] incorporates instance-level information and correlations. Recent contrastive learning methods, such as CRD [22], maximize mutual information between teacher and student networks. CRCD [31] leverages both feature and gradient information. However, these methods depend on a memory bank to manage negative samples, resulting in additional memory and computational costs. Our method avoids these costs by eliminating the need for a memory bank, enhancing efficiency.

Graph-based Distillation Graphs effectively capture and represent complex relationships, making them invaluable across various domains. To leverage this, Graph Neural Networks (GNNs) [7] have been developed to learn node representations by aggregating information from neighboring instances within graph-structured data. By capturing both individual features and inter-instance relationships, these representations have enabled GNNs to achieve significant advancements in various learning tasks.

Several studies have explored the use of graphs to represent and transfer relational knowledge in the context of knowledge distillation. IRG [12] is the first to employ instance-wise graph structures to capture relational knowledge from the teacher model by introducing feature space transformation across layers. Building on this, MHGD [10] employs multi-head attention networks to create graph-based representations, capturing relational and intra-data information to enable effective knowledge transfer. GKD [9] constructs a KNN-based graph using cosine similarity of internal representations, but it requires teacher and student networks to have the same number of layers, which is not always feasible. More recently, HKD [30] employs GNNs to integrate both individual and relational knowledge into a unified graph-based representation. This method enhances knowledge transfer by maximizing mutual information between the representations of the teacher and student networks.

Optimal Transport Optimal transport distance [24], also known as Wasserstein distance or Earth Mover's distance, has a long-standing history in mathematics and has been widely applied to various tasks such as sequence-to-sequence learning [3], entity alignment [21], and graph matching [25]. Cuturi [4] initially proposed the Sinkhorn algorithm for computing an entropically regularized optimal transport distance. Peyré et al. [18] later utilized this algorithm to propose a fast Sinkhorn projection-based method for computing Gromov-Wasserstein discrepancy. Due to the Envelope Theorem, the Sinkhorn algorithm can be efficiently computed and easily integrated into neural networks.

Given the suitability of optimal transport for solving matching problems like shape and object matching, as well as its successful application in various domains like unsupervised machine translation and weighted directed network

matching, we extend its use to match teacher and student representation distributions in knowledge distillation.

3 Preliminaries

3.1 Knowledge Distillation

Vanilla knowledge distillation (KD) aims to transfer knowledge from a pre-trained teacher model to a smaller student model by leveraging the soft outputs produced by the teacher [5]. In particular, the soft outputs are generated using a temperature-scaled Softmax function, as following:

$$p_i(z;\tau) = \frac{\exp(z_i/\tau)}{\sum_{k=1}^{K}\exp(z_k/\tau)} \tag{1}$$

where z_i is the logit for the i-th class and τ is typically set to 1 by default. Higher values of τ lead to a softer probability distribution. A larger τ results in a softer probability distribution across the classes, while smaller leads to a sharper distribution. The student model is optimized by minimizing the Kullback-Leibler (KL) divergence between the soft targets $p^T = \{p^t(\boldsymbol{x}_i,\tau)\}_{i=1}^N$ from the teacher and the predicted probabilities $p^S = \{p^s(\boldsymbol{x}_i,\tau)\}_{i=1}^N$ from the student model:

$$\mathcal{L}_{KD}\left(p^S, p^T\right) = \frac{1}{N}\sum_{i=1}^{N}\mathrm{KL}\left(p^s(\boldsymbol{x}_i,\tau), p^t(\boldsymbol{x}_i,\tau)\right) . \tag{2}$$

Here, N is the number of examples, the superscripts S, s, T, t indicate the student and teacher models. Originally, the student model is trained with the Cross-Entropy (CE) loss $\mathcal{L}_{CE}$ on the labeled pairs $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$. Finally, the total loss for the student model is a weighted sum of the KL divergence and the CE loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{CE}\left(p^S, y\right) + \lambda\mathcal{L}_{KD}\left(p^S, p^T\right) , \tag{3}$$

where λ is a trade-off weight for the two losses.

3.2 Optimal Transport

Optimal Transport (OT) provides a framework to determine a minimal cost transportation plan between a source distribution $\boldsymbol{\mu}^s$ and a target distribution $\boldsymbol{\mu}^t$. Specifically, we focus on the discrete problem with $\boldsymbol{\mu}^s$ and $\boldsymbol{\mu}^t$ being discrete empirical distributions, *i.e.*, $\boldsymbol{\mu}^s = \sum_{i=1}^{n_s} p_i^s \delta_{x_i}, \boldsymbol{\mu}^t = \sum_{j=1}^{n_t} p_j^t \delta_{y_j}$. Here, $\delta(\cdot)$ denotes the Dirac function, n_s and n_t are the number of samples, and p_i^s and p_j^t are the probability masses of the examples x_i and y_j, belonging to the probability simplex, *i.e.*, $\sum_{i=1}^{n_s} p_i^s = \sum_{j=1}^{n_t} p_j^t = 1$.

To formulate the optimal transport problem, we define a cost matrix $\mathbf{C}$ with C_{ij} representing the distance between x_i and y_j. Then the transport problem is:

$$\mathbf{T}^* = \underset{\mathbf{T}\in\mathbb{R}_+^{n_s\times n_t}}{\operatorname{argmin}} \sum_{i=1}^{n_s}\sum_{j=1}^{n_t} T_{ij}C_{ij} \quad \text{s.t.} \quad \mathbf{T}\mathbf{1}_{n_t} = \boldsymbol{\mu}^s,\ \mathbf{T}^\top\mathbf{1}_{n_s} = \boldsymbol{\mu}^t, \tag{4}$$

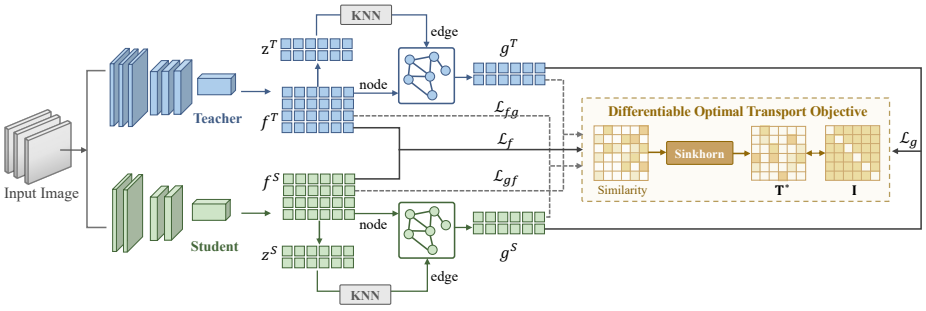


Fig. 1: An overview of the proposed OTGD. The features f^S, f^T and logits z^S, z^T are extracted from the teacher and student networks. Then, attribute graphs are constructed by using features as the nodes, whose edges are calculated from the logits with the KNN strategy. Graph Neural Networks are applied to generate the relational features g^S, g^T. Finally, the Optimal Transport objective is enforced on different levels of features: (f^S, f^T), (f^S, g^T), (g^S, f^T) and (g^S, g^T) to distill the comprehensive knowledge.

$\mathbf{T}^*$ is the optimal transport plan or transport matrix, where T_{ij} denotes the optimal amount of mass to move from x_i to y_j to achieve the minimum cost.

It is time-consuming to directly solve the original optimal transport problem 4. Therefore, [4] introduced an entropy regularization term to enhance the computational efficiency, which converted the original problem 4 to:

$$\mathbf{T}^* = \underset{\mathbf{T} \in \Pi(\boldsymbol{\mu}^s, \boldsymbol{\mu}^t)}{\operatorname{argmin}} \sum_{i,j} T_{ij} C_{ij} - \epsilon H(\mathbf{T}), \tag{5}$$

where $\Pi(\boldsymbol{\mu}^s, \boldsymbol{\mu}^t)$ denotes the probability constraints same as the constraints in Eq. 4, $H(\mathbf{T}) = -\sum_{i,j} T_{ij}(\log T_{ij} - 1)$ is the entropy regularization of $\mathbf{T}$, and ϵ is the weight of regularization. After introducing the entropy term, this problem can be efficiently solved with a Sinkhorn iteration algorithm. We will describe the detailed algorithm and the differentiation in Sec. 4.2.

4 Methodology

Fig. 1 shows the architecture of OTGD. Specifically, both the features f^S, f^T and logits z^S, z^T from the teacher and student networks are used in our method. Leveraging the logits similarities with the KNN strategy, we can calculate the graph edges for the feature nodes. With this graphs structure, GNNs are adopted to aggregate and extract the relational features from both the student and teacher models. To facilitate knowledge transfer for both features and relations, we propose a differentiable optimal transport objective for optimal feature matching on different levels of features: (f^S, f^T), (f^S, g^T), (g^S, f^T) and (g^S, g^T). This way, different levels of knowledge can be effectively incorporated for comprehensive knowledge distillation.

4.1 Attribute Graph and Graph Neural Networks

While most existing works only focus on either the features or logits for the knowledge distillation tasks and ignore their interactions, we leverage both features and logits for effective distillation. In particular, we build two graphs $G^S = (V^S, E^S)$ and $G^T = (V^T, E^T)$, where V^S, V^T represent the nodes (*i.e.*, the features f^S or f^T) and E^S, E^T are the set of edges constructed from the logits z^S or z^T. Concretely, A^S denotes the student adjacent matrix and $A^S_{ij} = \texttt{sim}(z^S_i, z^S_j)$, where $\texttt{sim}$ is the similarity function. After that, we apply the KNN processing on A^S to only keep the top-k similarities. This way, the irrelevant and noisy correlations can be filtered out. A^T can be obtained in a similar way.

With the graphs, we adopt the graph neural networks for relational feature extraction. Specifically, we adopt the graph convolutional networks (GCNs) structure. The degree matrix $D_{ii} = \sum_j A_{ij}$, then A is normalized as $\hat{A} = D^{-1/2}AD^{-1/2}$. Initially, the node features of the first layer H^1 is set to f^S or f^T. Then this can be computed layerwisely:

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)}), l = 1, 2, \ldots, L\,. \tag{6}$$

Here, σ is the activation function such as ReLU and $W^{(l)}$ is the learnable weights. Finally, we take the last layer features as the graph representation: $g = H^{(L)}$. The student and teacher encounter two different GCNs with the above computation to obtain the graph representations as g^S and g^T.

We take the logits z^S and z^T to construct the graph edges instead of the features f^S and f^T for two reasons: (1) the logits contain useful semantic information and (2) this can avoid potential collapse since the features are already used as node representations.

4.2 Differentiable Optimal Transport Objective

After obtaining the features and graph representations, a straightforward way is to directly apply distance function to transfer knowledge from teacher to student, *e.g.*, mutual information, InfoNCE or KL divergence. We adopt optimal transport as the loss objective due to its good mathematical property.

Pre-processing. Taking $g^S \in \mathbb{R}^{N\times d}$ and $g^T \in \mathbb{R}^{N\times d}$ as an example, we first compute the cost matrix $\mathbf{C} = \texttt{cos\_similarity}(g^S, g^T) \in \mathbb{R}^{N\times N}$, which measures the difficulty of matching student representation and teacher representation. Since C might vary significantly for different mini-batch examples, we further apply the z-score normalization to reduce the variance and improve the training stability, *i.e.*, $\mathbf{C}_{ij} = \frac{\mathbf{C}_{ij} - \texttt{mean}(\mathbf{C})}{\texttt{std\_dev}(\mathbf{C})}$, where $\texttt{mean}$ and $\texttt{std\_dev}$ denote the mean value and standard deviation.

Optimal Transport Objective. The entropy optimal transport problem (Eq. 4) can be efficiently solved with the Sinkhorn Algorithm (Alg. 1). Then, the optimal matching between the teacher and student features can be calculated

$$\mathbf{T}^* = \texttt{Sinkorn}(\mathbf{C}, \epsilon, t_{max}), \tag{7}$$

where ϵ is the same weight as Eq. 4 and t_{max} is the maximum iteration number. Ideally, the N student features should match the teacher features one-by-one, *i.e.*, $\mathbf{T}^*$ is expected to be a diagonal matrix. As $\mathbf{T}^*$ is a 2D probability, we normalize the rows and columns to make it comparable with an identity matrix:

Algorithm 1: Sinkhorn algorithm

1. **Input:** $\boldsymbol{\mu^s}, \boldsymbol{\mu^t}, \mathbf{C}, \epsilon, t_{max}$
2. Initialize $\mathbf{K} = e^{-\mathbf{C}/\epsilon}, \mathbf{b} \leftarrow \mathbf{1}, i \leftarrow 0$

while $i \leq t_{max}$ and not converge **do**

3. $\mathbf{a} = \boldsymbol{\mu^t}/(\mathbf{K}\mathbf{b})$
4. $\mathbf{b} = \boldsymbol{\mu^t}/(\mathbf{K}^\top \mathbf{a})$

end while

5. **Output:** $\mathbf{T}^* = \mathrm{diag}(\mathbf{a})\mathbf{K}\mathrm{diag}(\mathbf{b})$

$\mathbf{T}^* \leftarrow \frac{\mathbf{T}^*}{\mathbf{T}^*.\mathrm{sum}(\mathrm{dim}=1)}$, and $\mathbf{T}^* \leftarrow \frac{\mathbf{T}^*}{\mathbf{T}^*.\mathrm{sum}(\mathrm{dim}=0)}$. Then, we calculate the loss as

$$\mathcal{L} = \|\mathbf{T}^* - \mathbf{I}\|_2 \,. \tag{8}$$

To make the best use of the teacher network for knowledge distillation, we apply the loss to different levels of the architecture. Formally, we have $(g^S, g^T) \rightarrow \mathbf{C}_g \rightarrow \mathbf{T}_g^* \rightarrow \mathcal{L}_g$. Similarly, we can obtain $\mathcal{L}_f$. To increase diversity and avoid trivial solutions, we further apply $(f^S, g^T) \rightarrow \mathbf{C}_{fg} \rightarrow \mathbf{T}_{fg}^* \rightarrow \mathcal{L}_{fg}$. The OT loss is $\mathcal{L}_{ot} = \mathcal{L}_f + \mathcal{L}_g + \mathcal{L}_{fg} + \mathcal{L}_{gf}$. And, the final loss for KD is

$$\mathcal{L} = \mathcal{L}_{CE} + \beta \mathcal{L}_{ot} \,, \tag{9}$$

where β is the weight to balance two losses.

Differentiation. For end-to-end training, we formulate the chain rule for the Sinkhorn algorithm (an example of g^S, g^T):

$$\frac{\partial \mathcal{L}}{\partial \mathbf{T}^*} \rightarrow \underset{\text{iterate from } t_{max} \text{ to } 0}{\frac{\partial \mathbf{T}^*}{\partial \mathbf{K}}, \frac{\partial \mathbf{T}^*}{\partial \mathbf{a}}, \frac{\partial \mathbf{T}^*}{\partial \mathbf{b}}} \rightarrow \frac{\partial \mathbf{K}}{\partial \mathbf{C}} \rightarrow \frac{\partial \mathbf{C}}{\partial g^S}, \frac{\partial \mathbf{C}}{\partial g^T} \tag{10}$$

Each step only involves matrix-vector multiplications, so the gradient calculation can be efficiently implemented.

5 Experiment

5.1 Experimental Settings

Datasets We conduct experiments on two well-established benchmarks for KD: CIFAR-100 [8] and Tiny-ImageNet [19]. CIFAR-100 consists of 60,000 images, each sized 32×32, divided into 100 classes. Tiny-ImageNet, a subset of ImageNet, has 200 classes with 500 training and 50 validation images per class, totaling 100K training images and 10K validation images.

Baseline We compare with strong baselines and state-of-the-art KD methods, which can be divided into three categories:

- **Individual Distillation** includes KD [5], AT [27], SemCKD [2], DKD [28] and TTM [29], focusing on knowledge from individual instances.
- **Relation-based Distillation** includes SP [23], PKT [16], RKD [15], and CRD [22], capturing pairwise relational knowledge.

– **Graph-based Distillation** includes IRG [12] and HKD [30], utilizing graph structures for knowledge distillation.

Training Details For training, we adopt the SGD optimizer with 0.9 momentum. The mini-batch is set to 64 and the weight decay to 5×10^{-4}. We set the initial learning rate to 0.01 for MobileNetV2 and ShuffleNet, and at 0.05 for other networks. The learning rate is scheduled to decay by a factor of 10 at epochs 150, 180, and 210, with a max ecpoch of 240. The regularization ϵ is set to 1×10^{-5}, max iteration t_{max} is set to 2 in our experiments for Algorithm 1. For consistency with conventional KD methods, the temperature parameter τ is set to 4. All results are reported as means $\pm$ standard deviations over 3 trials.

5.2 Comparison with State-of-the-Art (SoTA) Methods

We experiment with two scenarios to evaluate the effectiveness of our OTGD method: **same architecture** and **heterogeneous architecture**, *i.e.*, if the student and teacher network come from the same network family such as ResNet). Results of CIFAR-100 are shown in Table 1 and 2, and TinyImageNet in Table 3.

The plain version of our method ("OTGD") achieves better or comparable results without the KD loss. When KD loss is added, our method ("OTGD+KD") can be further improved and beats most of the baseline results, demonstrating the effectiveness of our optimal transport objective. Note that we do not rely on any extra memory bank as negative samples, which is more efficient.

To study the comparability of our method and existing SoTA (*i.e.*, HKD), we combine our objective with the InfoNCE loss of HKD. "OTGD+HKD" refers to combining the OTGD loss with the HKD. The performance of "OTGD+HKD" can outperform the original HKD results, verifying the complementary potential of our method with existing KD methods. Comparing Table 1 and Table 2, we find that our method works extremely well for the heterogeneous teacher-student architectures. For example, "OTGD+KD" generally outperforms "IRG+KD" by approximately 2% and outperforms "HKD+KD" by around 1%.

5.3 Ablation Studies

Batch Size Fig. 2 (a) presents the results of varying batch sizes, ranging from 32 to 512. For two network settings, peak accuracy is observed at 64. However, as the batch size increases beyond 128, there is a clear decline. This suggests that large batch sizes hinder the distillation process, likely due to the difficulty in capturing relationships among a larger number of nodes, emphasizing the importance of selecting an appropriate batch size for optimal performance. We also ablated ϵ and t_{max}, but the variation is small due to the numerical stability of Sinkhorn algorithm. Here, we omit those results due to space limitation.

Hyperparameters We further ablate the loss weight β on CIFAR-100 dataset to study the effect of the OTGD loss. The highest accuracy for ResNet8×4 is 75.95 with $\beta = 1.0$, while for ShuffleV1, it peaks at 76.76 with $\beta = 3.5$ (but $\beta = 1.0$ is also reasonably good). This suggest that although it is overall stable,

Table 1: **Same architecture comparison** with the state-of-the-art methods on CIFAR-100. Top-1 test accuracy (%) is reported with standard deviation over 3 repeated experiments. Best results are in **bold**, and second best with underline.

-	Teacher	ResNet32×4	ResNet110	ResNet56	WRN-40-2	WRN-40-2
	Acc	79.42	74.31	73.44	76.31	76.31
Type	Student	ResNet8×4	ResNet20	ResNet20	WRN-40-1	WRN-16-2
	Acc	72.63±.21	69.06±.27	69.06±.27	71.98±.17	73.26±.22
Individual	KD	73.55±.20	70.67±.27	70.66±.22	73.42±.22	74.92±.20
	AT	74.57±.17	70.65±.17	71.16±.14	74.43±.11	74.28±.13
	SemCKD	75.58±.22	N/A	**71.79**±.21	74.75±.21	75.42±.15
	DKD	76.23±.12	71.28±.20	71.43±.13	74.54±.12	75.70±.06
	TTM	76.17±.28	71.46±.16	71.65±.23	74.32±.31	76.23±.15
Relation	SP	74.69±.17	70.74±.23	70.84±.25	73.90±.17	74.88±.28
	PKT	73.64±.09	70.88±.16	70.85±.22	74.01±.23	74.82±.19
	RKD	75.11±.13	69.25±.05	69.61±.06	72.22±.20	73.35±.09
	CRD	75.59±.07	71.38±.04	71.69±.15	74.36±.10	75.53±.10
Graph	IRG	72.69±.13	69.38±.19	70.13±.42	71.96±.16	73.74±.31
	HKD	75.73±.07	70.56±.19	70.49±.10	73.43±.24	75.42±.20
	OTGD	75.95±.13	71.31±.15	70.92±.11	73.84±.13	75.27±.07
	IRG+KD	74.10±.28	71.32±.07	71.19±.14	74.16±.24	75.53±.12
	HKD+KD	76.19±.14	71.62±.17	71.64±.23	74.66±.05	75.89±.14
	OTGD+KD	**76.24**±.14	71.63±.07	71.22±.06	74.48±.08	75.45±.12
	OTGD+HKD	76.09±.05	**71.95**±.02	**71.79**±.03	**75.00**±.14	**75.93**±.09

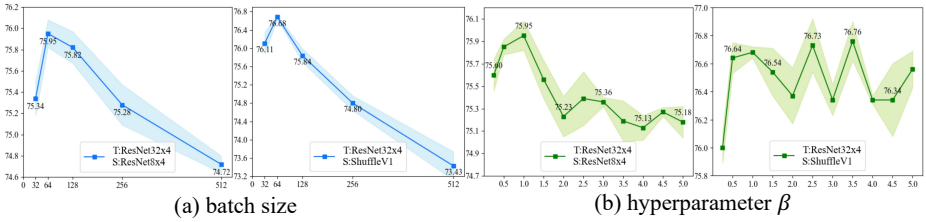
Table 2: **Heterogeneous architecture comparison** with state-of-the-art methods on CIFAR-100. Top-1 test accuracy (%) is reported with standard deviation over 3 repeated experiments. Best results are in **bold**, and second best with underline.

-	Teacher	ResNet32×4	ResNet32×4	VGG-13	ResNet50	WRN-40-2
	Acc	79.42	79.42	74.64	73.44	76.31
Type	Student	ShuffleV2	ShuffleV1	MobileV2	MobileV2	ShuffleV1
	Acc	72.60±.12	70.49±.25	64.60±.32	64.60±.32	70.49±.25
Individual	KD	75.60±.21	74.30±.16	67.37±.22	67.35±.32	74.83±.13
	AT	74.41±.10	71.93±.31	60.01±.20	58.58±.54	73.32±.35
	SemCKD	77.37±.31	75.41±.11	68.78±.22	68.73±.11	74.81±.21
	DKD	76.48±.08	75.44±.20	69.71±.22	69.11±.19	76.70±.13
	TTM	76.57±.26	74.18±.26	68.98±.85	69.24±.28	75.39±.33
Relation	SP	75.77±.08	73.48±.21	66.30±.13	68.08±.35	74.52±.37
	PKT	75.10±.11	74.39±.16	67.13±.30	66.52±.33	73.89±.16
	RKD	73.21±.16	72.28±.39	64.52±.45	64.43±.42	72.21±.16
	CRD	75.90±.15	75.13±.33	69.73±.42	69.11±.28	76.05±.14
Graph	IRG	73.35±.09	71.50±.23	65.37±.07	65.27±.25	71.86±.34
	HKD	76.19±.16	75.58±.13	69.88±.17	69.83±.15	76.22±.25
	OTGD	76.87±.02	76.68±.04	70.37±.19	70.00±.39	76.31±.12
	IRG+KD	75.61±.24	74.84±.24	66.62±.23	68.61±.32	75.03±.14
	HKD+KD	76.78±.07	75.93±.05	70.21±.09	70.72±.32	76.45±.20
	OTGD+KD	**77.63**±.19	**76.78**±.15	**70.58** ±.34	**70.82**±.56	76.76±.10
	OTGD+HKD	77.14±.11	76.20±.05	70.56±.25	70.75±.52	**76.93**±.14

the choice of β can influence the distillation process, highlighting the importance of choosing a proper β to balance CE loss and OTGD loss. In our experiments, we adopt $\beta = 1$ to keep consistency across student-teacher settings.

Table 3: Comparison with state-of-the-art methods on TinyImageNet.Top-1 test accuracy (%) is reported with standard deviation over 3 repeated experiments. Best results are in **bold**, and second best with underline.

Teacher	ResNet32×4	ResNet32×4	VGG-13	VGG-13	ResNet50	ResNet50
Acc	57.92	57.92	52.02	52.02	55.44	55.44
Student	ResNet8×4	ShuffleV1	MobileV2	VGG-8	MobileV2	VGG-8
Acc	49.91±.16	50.45±.33	44.20±.22	47.00±.17	44.20±.22	47.00±.17
KD	52.28±.07	56.98±.24	45.39±.59	51.34±.08	46.13±.26	51.50±.36
IRG	50.28±.05	50.43±.14	44.11±.31	46.31±.15	44.00±.06	46.37±.10
HKD	55.53±.07	58.65±.20	49.10±.21	51.97±.33	47.85±.06	52.20±.20
OTGD	55.51±.09	58.67±.18	49.30±.05	51.55±.22	49.00±.10	52.42±.05
IRG+KD	53.93±.11	57.16±.15	47.36±.36	50.77±.09	47.50±.22	50.83±.03
HKD+KD	**56.18**±.12	58.86±.08	49.56±.18	**52.62**±.03	49.33±.14	53.30±.33
OTGD+KD	56.01±.11	59.08±.14	49.61±.26	52.27±.12	49.58±.06	53.31±.28
OTGD+HKD	56.08±.06	**59.09**±.11	**50.06**±.19	52.46±.09	**49.93**±.18	**53.54**±.14



(a) batch size (b) hyperparameter β

Fig. 2: (a) The effect of batch size on CIFAR-100. (b) The effect of the hyperparameter β (Eq. 9) on CIFAR-100.

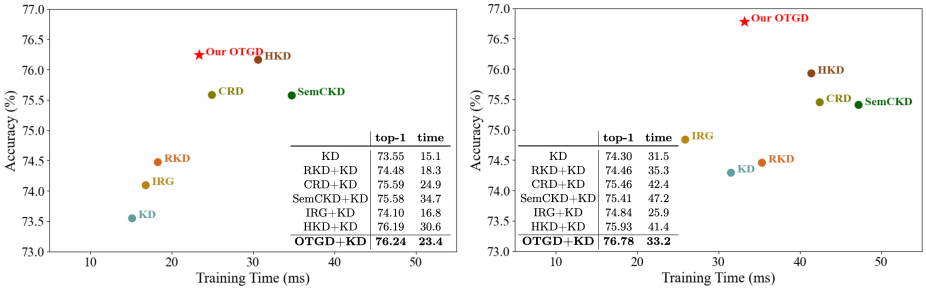
Different Losses We performed an additional ablation study on the proposed loss terms using the CIFAR-100 dataset, and the results are in Table 4. The baseline model, which only uses the primary cross-entropy loss, achieves an accuracy of 72.51 on ResNet8×4 and 70.50 on ShuffleV1. The introduction of all loss components ($\mathcal{L}_{kd}$, $\mathcal{L}_f$, $\mathcal{L}_g$, and $\mathcal{L}_{dual} = \mathcal{L}_{fg} + \mathcal{L}_{gf}$) in the OTGD+KD configuration achieves the highest accuracy of 76.24 for ResNet8×4 and 76.78 for ShuffleV1. Specifically, the inclusion of $\mathcal{L}_{dual}$ improves accuracy by 0.08 and 0.29, respectively. This configuration provides the best performance across both network architectures, with an improvement of 3.73 for ResNet8×4 and 6.28 for ShuffleV1 compared to the baseline. These findings demonstrate that integrating the proposed loss components significantly enhances the knowledge transfer process, with the OTGD+KD yielding the best performance on both networks.

5.4 Visualization

Training efficiency Fig. 3 presents the trade-off between accuracy and training time for various knowledge distillation methods, evaluated on two different teacher-student pairs: (a) ResNet32×4 as the teacher and ResNet8×4 as the student, and (b) ResNet32×4 as the teacher and ShuffleV1 as the student. In both

Table 4: Ablation study on different loss components using the CIFAR-100 dataset ($\mathcal{L}_{dual} = \mathcal{L}_{fg} + \mathcal{L}_{gf}$).

Module	Losses				ResNet8×4	ShuffleV1
	$\mathcal{L}_{kd}$	$\mathcal{L}_f$	$\mathcal{L}_g$	$\mathcal{L}_{dual}$		
Baseline	-	-	-	-	72.51	70.50
KD	✓	-	-	-	73.55	74.30
OTGD	-	✓	-	-	75.41	76.67
OTGD	-	-	✓	-	75.36	76.08
OTGD	-	✓	✓	-	75.87	76.39
OTGD	-	✓	✓	✓	75.95	76.68
OTGD+KD	✓	✓	✓	✓	**76.24**	**76.78**



(a) T: ResNet32×4 S: ResNet8×4 (b) T: ResNet32×4 S: ShuffleV1

Fig. 3: Top-1 Accuracy versus Training time (per batch) on CIFAR-100 with both *same architecture* and *heterogeneous architectures* of the student-teacher.

settings, our OTGD achieves the highest accuracy of 76.24 and 76.78 with training time of 23.4ms 33.2ms, respectively, demonstrating a superior balance between performance and efficiency. Other methods like HKD, CRD and SemCKD show competitive accuracy but require longer training times, while methods like KD and IRG offer lower accuracy. To ensure a fair comparison, all methods were evaluated under consistent conditions (using official implementations where available), and trained with an NVIDIA RTX6000 GPU.

t-SNE Visualization We provide t-SNE visualizations for three graph-based methods (IRG, HKD and our OTGD), evaluated on two different teacher-student pairs. As shown in Fig. 4, our method achieves superior class separability compared to the others especially on ShuffleV1. The t-SNE plot for OTGD demonstrates distinct boundaries between different classes, indicating enhanced robustness and discernibility of feature representations. This improved separation is likely to significantly benefit classification performance.

Transport plan change To understand the progression of the optimal transport plan $\mathbf{T}^*$, we adopt ResNet-32×4 as the teacher as the teacher and experiment two different student architectures: ResNet8x4 and ShuffleV1. Fig. 5 compares the transport plan $\mathbf{T}^*$ from our OT objective at the beginning (epoch 0) and end (epoch 240) of training. The visualizations highlight the progression

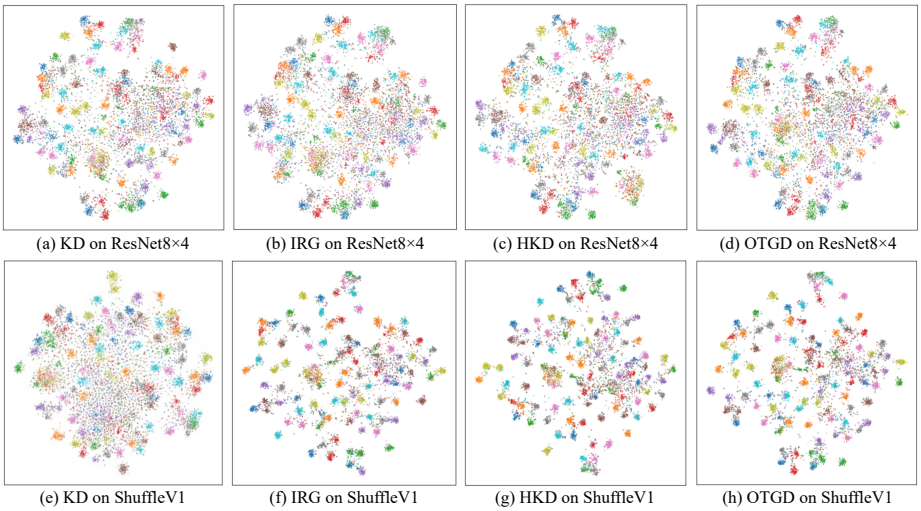


Fig. 4: t-SNE visualization results of test images from CIFAR-100 with ResNet32×4 as the teacher.

of an optimal transport plan, which ideally converges to the identity matrix $\mathbf{I}$. The evolution of $\mathbf{T}^*$ demonstrates the effectiveness of our optimal transport objective in achieving more accurate distribution matching over training epochs.

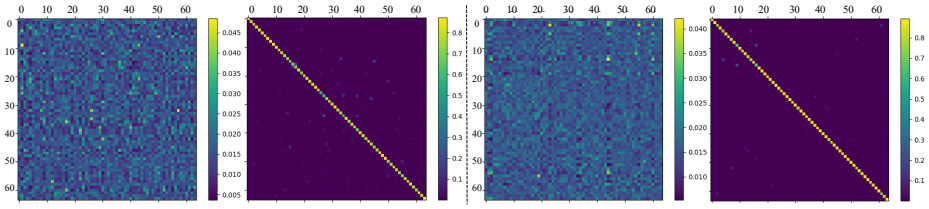


Fig. 5: The variation of the Transport Plan $\mathbf{T}^*$ of the Optimal Transport, from the initial stage (epoch 0; (a) left and (b) left) to the final stage (epoch 240; (a) right and (b) right). After training, the Transport Plan $\mathbf{T}^*$ approaches the expected ideal plan, *i.e.*, an Identity matrix $\mathbf{I}$, showing the effectiveness of the optimal transport objective.

6 Conclusion

In this paper, we present Optimal Transport-based Graph Distillation (OTGD), a new method for efficient knowledge distillation. By constructing attribute graphs for both the teacher and student model, we adopt GNNs to generate

graph-based representations for effective knowledge transfer. This manner, both the individual and relational knowledge can be extracted and transferred from the teacher graph to the student one. Different from existing methods which used InfoNCE estimator as the objective, we propose an innovative differentiable optimal transport objective to train the model. This objective is implemented with the Sinkhorn algorithm due to the good numerical stability and differentiable ability. Comprehensive experiments conducted on benchmark datasets and diverse model architectures demonstrate that OTGD consistently outperforms existing state-of-the-art methods. The results validate the efficacy of our strategy in achieving superior performance while reducing computational complexity. Our findings suggest that OTGD is a promising direction for improving the efficiency and effectiveness of knowledge distillation in resource-constrained environments.

Acknowledgments. This project is partially supported by ARC. (grant number DP240101349). Yanbin Liu is supported by the Google Cloud Research Credits program with the award GCP19980904.

References

1. Cai, H., Zheng, V.W., Chang, K.C.C.: A comprehensive survey of graph embedding: Problems, techniques, and applications. IEEE transactions on knowledge and data engineering **30**(9), 1616–1637 (2018)
2. Chen, D., Mei, J.P., Zhang, Y., Wang, C., Wang, Z., Feng, Y., Chen, C.: Cross-layer distillation with semantic calibration. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 7028–7036 (2021)
3. Chen, L., Zhang, Y., Zhang, R., Tao, C., Gan, Z., Zhang, H., Li, B., Shen, D., Chen, C., Carin, L.: Improving sequence-to-sequence learning via optimal transport. arXiv preprint arXiv:1901.06283 (2019)
4. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems **26** (2013)
5. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
6. Kim, K., Ji, B., Yoon, D., Hwang, S.: Self-knowledge distillation with progressive refinement of targets. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6567–6576 (2021)
7. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
8. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
9. Lassance, C., Bontonou, M., Hacene, G.B., Gripon, V., Tang, J., Ortega, A.: Deep geometric knowledge distillation with graphs. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 8484–8488. IEEE (2020)
10. Lee, S., Song, B.C.: Graph-based knowledge distillation by multi-head attention network. arXiv preprint arXiv:1907.02226 (2019)
11. Li, Z., Li, X., Yang, L., Zhao, B., Song, R., Luo, L., Li, J., Yang, J.: Curriculum temperature for knowledge distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1504–1512 (2023)

12. Liu, Y., Cao, J., Li, B., Yuan, C., Hu, W., Li, Y., Duan, Y.: Knowledge distillation via instance relationship graph. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7096–7104 (2019)
13. Miles, R., Rodriguez, A.L., Mikolajczyk, K.: Information theoretic representation distillation. arXiv preprint arXiv:2112.00459 (2021)
14. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
15. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3967–3976 (2019)
16. Passalis, N., Tefas, A.: Learning deep representations with probabilistic knowledge transfer. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 268–284 (2018)
17. Peng, B., Jin, X., Liu, J., Li, D., Wu, Y., Liu, Y., Zhou, S., Zhang, Z.: Correlation congruence for knowledge distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5007–5016 (2019)
18. Peyré, G., Cuturi, M., et al.: Computational optimal transport: With applications to data science. Foundations and Trends® in Machine Learning **11**(5-6), 355–607 (2019)
19. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**, 211–252 (2015)
20. Sun, S., Ren, W., Li, J., Wang, R., Cao, X.: Logit standardization in knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15731–15740 (2024)
21. Tang, J., Zhao, K., Li, J.: A fused gromov-wasserstein framework for unsupervised knowledge graph entity alignment. arXiv preprint arXiv:2305.06574 (2023)
22. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. arXiv preprint arXiv:1910.10699 (2019)
23. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1365–1374 (2019)
24. Villani, C., et al.: Optimal transport: old and new, vol. 338. Springer (2009)
25. Xu, H., Luo, D., Zha, H., Duke, L.C.: Gromov-wasserstein learning for graph matching and node embedding. In: International conference on machine learning. pp. 6932–6941. PMLR (2019)
26. Yang, C., Xie, L., Su, C., Yuille, A.L.: Snapshot distillation: Teacher-student optimization in one generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2859–2868 (2019)
27. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016)
28. Zhao, B., Cui, Q., Song, R., Qiu, Y., Liang, J.: Decoupled knowledge distillation. arXiv preprint arXiv:2203.08679 (2022)
29. Zheng, K., Yang, E.H.: Knowledge distillation based on transformed teacher matching. arXiv preprint arXiv:2402.11148 (2024)
30. Zhou, S., Wang, Y., Chen, D., Chen, J., Wang, X., Wang, C., Bu, J.: Distilling holistic knowledge with graph neural networks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10387–10396 (2021)
31. Zhu, J., Tang, S., Chen, D., Yu, S., Liu, Y., Rong, M., Yang, A., Wang, X.: Complementary relation contrastive distillation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9260–9269 (2021)