SinKD: Sinkhorn Distance Minimization for Knowledge Distillation

Xiao Cui, Yulei Qin, Yuting Gao, Enwei Zhang, Zihan Xu, Tong Wu, Ke Li, Xing Sun, Wengang Zhou, Houqiang Li

SinKD: Sinkhorn Distance Minimization for Knowledge Distillation

Xiao Cui, Yulei Qin*, Yuting Gao, Enwei Zhang, Zihan Xu, Tong Wu, Ke Li, Xing Sun, Wengang Zhou, *Senior Member, IEEE,* and Houqiang Li†, *Fellow, IEEE*

***Abstract*—Knowledge distillation (KD) has been widely adopted to compress large language models (LLMs). Existing KD methods investigate various divergence measures including the Kullback-Leibler (KL), reverse Kullback-Leibler (RKL), and Jensen-Shannon (JS) divergences. However, due to limitations inherent in their assumptions and definitions, these measures fail to deliver effective supervision when few distribution overlap exists between the teacher and the student. In this paper, we show that the aforementioned KL, RKL, and JS divergences respectively suffer from issues of mode-averaging, mode-collapsing, and mode-underestimation, which deteriorates logits-based KD for diverse NLP tasks. We propose the Sinkhorn Knowledge Distillation (SinKD) that exploits the Sinkhorn distance to ensure a nuanced and precise assessment of the disparity between distributions of teacher and student models. Besides, thanks to the properties of the Sinkhorn metric, we get rid of sample-wise KD that restricts the perception of divergences inside each teacher-student sample pair. Instead, we propose a batch-wise reformulation to capture geometric intricacies of distributions across samples in the high-dimensional space. Comprehensive evaluation on GLUE and SuperGLUE, in terms of comparability, validity, and generalizability, highlights our superiority over state-of-the-art methods on all kinds of LLMs with encoder-only, encoder-decoder, and decoder-only architectures. Codes and models are available at https://github.com/2018cx/SinKD.**

***Index Terms*—Knowledge distillation, Wasserstein distance, Sinkhorn distance**

I. Introduction

LARGE language models (LLMs) such as BERT [1], RoBERTa [2], T0 [3], and GPT [4], [5] have set state-of-the-art (SOTA) records on various tasks in the field of natural language processing (NLP). On one hand, the scaling laws of LLMs undoubtedly stimulate the development of models with billions of parameters. On the other hand, the surge of model size makes it impractical to deploy LLMs under resource-constrained environments. Consequently, knowledge distillation (KD), emerging as a cost-efficient approach, has attracted attention from researchers to distill smaller models which maintain highly competitive performance.

One kind of the most representative KD methods is logits-based KD, where the divergence between the distributions of the predicted logits from teacher and student models is measured and minimized for knowledge transfer. The key to effective logits-based KD is exactly the proper measurement of such divergence. Existing studies have experimented with Kullback-Leibler (KL) divergence [6], reverse Kullback-Leibler (RKL) divergence [7], [8], and Jensen-Shannon (JS) divergence [9], [10], [11]. All these measures can be viewed as variants of the f-divergence measures, which are notoriously limited in quantification of any two distributions that lack substantial intersections [12]. Moreover, each measure has its own drawbacks (see Fig. 1). KL distillation results in **mode-averaging** [13], [14], causing the student to learn an excessively smooth distribution that encompasses the entire support of the teacher. RKL leads to **mode-collapsing** [15], [9], where the student focuses on one of the highly probable, salient regions of the teacher distribution and ignores the remaining ones. JS distillation gives rise to **mode-underestimation** [16], [17] where the student underestimates the probability of rare events due to insufficient penalty.

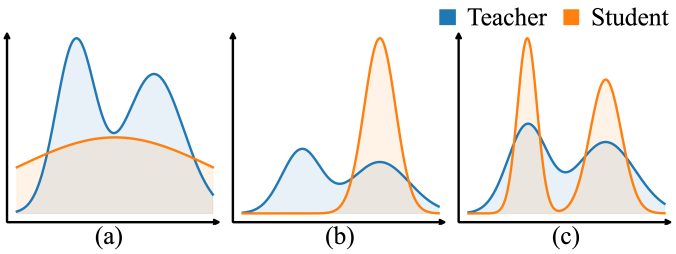

Fig. 1. Limitations of existing divergence measures in logits-based distillation. (a) Mode-averaging by Kullback-Leibler divergence. (b) Mode-collapsing by reverse Kullback-Leibler divergence. (c) Mode-underestimation by Jensen-Shannon divergence.

Another challenge of performing sample-wise KD on LLMs is that for discriminative tasks, the low-dimensional categorical outputs from the teacher provide limited insights into their underlying distributions in the high-dimensional hidden space. One intuitive solution is to bring in a batch of samples to collectively grasp the distribution differences. Nevertheless, existing divergence measures can only independently deal with each sample pair for logit-by-logit matching because they are not distance metrics. Given a batch of samples, they cannot locate the paired teacher and student logits from the same sample for overall distance minimization.

To address these challenges, we propose the **Sin**khorn **K**nowledge **D**istillation, termed as SinKD, for distillation of LLMs. In consideration of generalizability, we tackle **logits-based KD** in the present study, which would benefit a broad

Xiao Cui, Wengang Zhou and Houqiang Li are with the Department of Electrical Engineering and Information Science, University of Science and Technology of China, Hefei, 230027 China (e-mail: cuixiao2001@mail.ustc.edu.cn, zhwg@ustc.edu.cn, lihq@ustc.edu.cn), Yulei Qin, Yuting Gao, Enwei Zhang, Zihan Xu, Tong Wu, Ke Li and Xing Sun are with Tencent Youtu Lab, Shanghai, 200233 China (e-mail: yuleiqin@ tencent.com)

*Equal Contribution. †Corresponding author: Houqiang Li.

range of applications. Our SinKD employs the Sinkhorn distance [18], a variant of the Wasserstein distance [19], as divergence measure. The Wasserstein distance quantifies how different two distributions are by solving the optimal transport plan between them. Intuitively, it can be deemed as the minimum "effort" required to transform one distribution (i.e., student's logits distribution) into the other (i.e., teacher's logits distribution), where the "effort" can be defined as the product of the mass being moved and the moving distance. Compared with traditional divergence measures, it is more sensible as a cost function for distillation since it does not rely on implicit assumptions about distributions to be measured. Furthermore, it is differentiable almost everywhere, enabling easy optimization. Despite these advantages, the Wasserstein distance itself is difficult to be computed analytically. Its associated computational cost is prohibitively high for distilling LLMs. Under such circumstance, we propose to use Sinkhorn distance as an efficient approximation, which not only retains all the benefits of the Wasserstein distance but also greatly mitigates its cost issue for on-line distillation.

A straightforward application of Sinkhorn distance on sample-wise logits matching, though feasible, cannot take full advantage of its perception of structural differences in distributions. Fortunately, Sinkhorn distance is a symmetric metric and its derivation from the optimal transport (OT) imposes explicit constraints on **matching correctness**. It means that given a batch of logits outputs from the teacher and the student respectively as sets A and B, the minimization of the overall Sinkhorn distance between A and B enforces a precise element-wise matching between the two outputs coming from the same sample. Such properties allow it to work beyond sample-wise distillation and lay a solid foundation for batch-wise reformulation. As a result, we propose the "batchified" SinKD. In this way, we can capture geometric structures of the intricate and implicit distributions even through low-dimensional observations. We do not introduce additional modules or modify output formats specific to NLP tasks.

Extensive experiments are conducted in view of 1) **comparability**, 2) **validity**, and 3) **generalizability**. For comparability, we test SinKD with BERT on the GLUE benchmark [20] and it consistently outperforms the SOTA KD methods. For validity, we provide a comprehensive analysis on ablation studies and hyper-parameters tuning. Our findings advise practitioners on how to adopt SinKD in their own work. For generalizability, we test SinKD on the SuperGLUE benchmark [21] with LLMs of various architectures, ranging from the encoder-decoder T0 [3] to the decoder-only GPT-Neo [22] transformers. Our SinKD showcases robustness across model choices while previous studies merely investigate KD techniques on the encoder-only transformers (e.g., BERT).

In summary, our contributions are:

- We propose a knowledge distillation approach, SinKD, to employ the Sinkhorn distance for divergence measure. It not only addresses limitations of KL, RKL, and JS divergences under extreme scenarios, but also circumvents the computation burden of Wasserstein distance.
- We unearth the properties of Sinkhorn distance and further reformulate SinKD into batch-wise OT, extending its applicability in NLP tasks.
- Extensive experiments in terms of comparability, validity, and generalizability demonstrate the superiority of SinKD over SOTA methods. We offer practical guidelines of distillation for real-world applications.

Summary of Changes The present study is an extension of our previous work [23]. Major changes of the current extended version can be summarized below:

- The regression task is newly introduced for performance comparison and ablation studies.
- More experiments on the generalizability of SinKD are performed on larger models.
- The effect of prompt templates on distillation of generative LLMs is studied for further discussion.
- A more "glass-box" evaluation of our SinKD is conducted from the aspect of the representations of hidden states, the patterns of attention mechanism, and the layer-wise performance analysis.
- One new section in the related work for distillation with Sinkhorn distance is added.
- Extension of our SinKD to one-hot label fine-tuning is newly introduced to confirm its broad application.
- Experiments beyond NLP tasks (e.g., image classification) are added to verify the effectiveness of our SinKD in various domains (e.g., computer vision).

II. Related Work

A. Knowledge Distillation

Knowledge distillation (KD) is proposed to transfer the intrinsic and inherent knowledge of a teacher model to a student model via approximating the soft targets (e.g., output logits and intermediate representations) of the teacher model. The standard training approach of logits-based KD leverages both the cross-entropy loss and the distillation loss as a weighted combination for stable and efficient optimization of the student model. Existing KD methods can be simply classified into two categories: 1) logits-based KD and 2) representation-based KD. The logits-based KD is popularized by [6]. They force the student to match the predictions of the teacher as soft targets via cross-entropy loss, which is equivalent to minimize the KL divergence between teacher and student probabilities. [13] bring logits-based KD into generative language models and propose sequence KD. [24] and [25] apply KD on BERT for smaller models with minor degradation. [7] propose ENGINE to use the reverse KL for distillation of a non-autoregressive translation model. For representation-based KD, the hidden, intermediate representations of input tokens have been utilized as the matching targets of the student [26], [24], [27], [28], [29], [30]. There also exist methods that can adapt to either logits-based or representation-based KD [31], [32], [33], [34]. Notably, KD-Zero [33] and Auto-KD [34] leverage evolutionary and Monte Carlo tree search to autonomously identify effective distillation strategies for any teacher-student architectures. In contrast to KD-zero and Auto-KD, our approach focuses on addressing the limitations inherent in existing distillation divergence measures.

In this paper, we primarily focus on logits-based KD and investigate the fundamental problem: *how to transfer label-supplementary knowledge from the teacher to the student with an effective and reliable divergence measure*. Previous studies exploit KL divergence [6], RKL divergence [7], [8], JS divergence [9], [10], [11], and sophisticated distance measures [35], [36], [37], [38], [39]. However, these methods do not consistently capture subtle distribution differences and tend to take"shortcuts" in student imitating the teacher. To address these limitations, we combine the strengths of KL divergence in typical situations with the robustness of the Sinkhorn Distance in handling distribution mismatches and its ability to capture the geometry of probability distributions.

B. Sinkhorn Distance

We first introduce the Wasserstein distance as a foundation for the Sinkhorn distance. It is a dissimilarity metric derived by the mass transportation theory of two probability measures. Since the Wasserstein distance takes into account the underlying geometry of the distribution space [19], [40], [41], [42], it enjoys high popularity in generative adversarial networks [12], [43], [44], unsupervised learning [45], [46], [47], causal discovery [48], [49], reinforcement learning [50], [51], [52], [53] and task similarity estimation [54]. However, the Wasserstein distance is too costly to be computed and its efficient approximation is a prerequisite for distillation. The Sinkhorn distance stems from it and incorporates an extra entropy regularization term to make the OT tractable. It is informally defined by the minimum transport cost of an entropy-regularized OT plan [18], and has been successful in classification [55], [56], machine translation [57], domain adaptation [58], [59], [60], teacher model selection [61], [62] and generative modeling [63], [64].

For distillation of LLMs, especially under discriminative tasks, the vanilla sample-wise SinKD cannot make the best use of its desirable properties in perceiving structural differences between distributions. On the contrary, we propose the batch-wise SinKD to make up the insufficient knowledge revealed from the low-dimensional outputs of the teacher, improving its generalization over tasks.

C. Distillation with Sinkhorn Distance

The divergence measures derived from the optimal transport theory have been recently proposed for distillation of NLP models. Specifically, Lu *et al.* [61] introduce the concept of 'faculty distillation', a novel approach where the student selects the most relevant teacher from a group. They use optimal transport to bridge the differences of task and label space between teacher and student models. Similarly, Bhardwaj *et al.* [62] develop multiple teachers for knowledge distillation. Inspired by the optimal transport, they introduce the semantic distance as a new metric to evaluate the quality of knowledge transfer under the federated learning settings.

The differences between our SinKD and previous distillation methods that use Sinkhorn distance [61], [62] are three-fold. First, we focus on addressing specific limitations in divergence measures and therefore exploit the Sinkhorn distance to improve the precision of assessing disparities between teacher and student distributions. In contrast, they propose to use Sinkhorn distance for selection of the most appropriate teacher from multiple ones. Second, unlike existing methods [61], [62], we directly calculate the Sinkhorn distance without gradient computation. Besides, we propose the batch-wise implementation for capturing geometric intricacies of distributions across samples in a batch. Third, in our setting, only one teacher is involved and we do not polish the target logits from the teacher. However, both [61] and [62] introduce teacher groups and their output logits are dynamically weighted for on-the-fly target adjustment.

III. Methodology

In this section, we first review classic divergence measures and analyze their drawbacks. Then, we present details of SinKD within an OT framework.

A. Problem Statement

Given a sample $\mathbf{x}_i$ and its ground-truth label $\mathbf{y}_i \in \mathbb{R}^d$ in the training set, the output logits with softmax activation σ_τ from the teacher f_T and the student f_S are respectively $\mathbf{t}_i \in \mathbb{R}^d$ and $\mathbf{s}_i \in \mathbb{R}^d$:

$$\mathbf{t}_i = \sigma_\tau(f_T(\mathbf{x}_i)), \quad \mathbf{s}_i = \sigma_\tau(f_S(\mathbf{x}_i)), \tag{1}$$

where τ is the temperature and d is the dimension of the output logits. The objective of KD is to minimize the measured divergence $J(\mathbf{t}_i, \mathbf{s}_i)$ for knowledge transfer.

B. Classic Divergence Measures

a) KL Divergence: It quantifies the amount of information lost when $\mathbf{s}_i$ approximates $\mathbf{t}_i$ as:

$$J_{\mathrm{KL}}(\mathbf{t}_i, \mathbf{s}_i) \approx \sum_{j=1}^{d} (-\mathbf{t}_{i(j)} \log \mathbf{s}_{i(j)} + \mathbf{t}_{i(j)} \log \mathbf{t}_{i(j)}). \tag{2}$$

Here, j denotes the index of an element in a vector. Despite its popularity, KL divergence suffers from three limitations. First, it is asymmetric with $J_{\mathrm{KL}}(\mathbf{t}_i, \mathbf{s}_i) \neq J_{\mathrm{KL}}(\mathbf{s}_i, \mathbf{t}_i)$, which introduces inconsistencies due to its violation of the property as a distance metric. Second, the student model optimized by the KL loss attempts to average the teacher's multimodal distribution, ending up with an underfitting of these modes. This is known as the mode-averaging problem. Consequently, the student fails to capture all crucial patterns of data and ultimately impacts performance. Third, the KL divergence corresponds to a non-smooth function, posing challenges to optimization.

b) RKL Divergence: It addresses the issue of mode-averaging associated with $J_{\mathrm{KL}}(\mathbf{t}_i, \mathbf{s}_i)$:

$$J_{\mathrm{RKL}}(\mathbf{t}_i, \mathbf{s}_i) \approx \sum_{j=1}^{d} (\mathbf{s}_{i(j)} \log \mathbf{s}_{i(j)} - \mathbf{s}_{i(j)} \log \mathbf{t}_{i(j)}). \tag{3}$$

However, it shares the inherent asymmetry with KL which leads to inconsistencies in capturing differences. Furthermore,

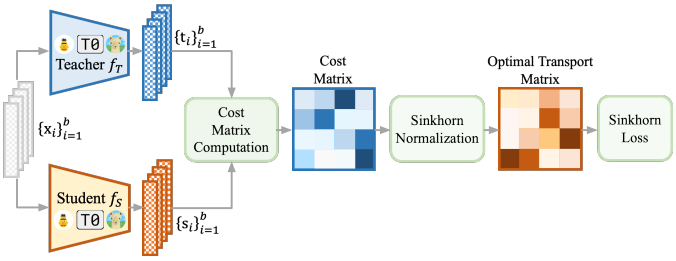

Fig. 2. Illustration of our SinKD pipeline: 1) cost matrix computation, 2) sinkhorn normalization, and 3) sinkhorn loss.

the student optimized by a RKL loss tends to pay attention only to highly likely events of the teacher's distribution, which is known as mode-collapsing. Accordingly, if the teacher assigns zero-probability to an event, the student is compelled to do the same. This "zero-forcing" effect could be problematic as the student lacks the capacity to track the complete distribution of the teacher, resulting in suboptimal performance.

c) JS Divergence: It combines both KL and RKL by:

$$J_{\mathrm{JS}}(\mathbf{t}_i,\mathbf{s}_i) \approx \frac{1}{2}\sum_{j=1}^{d}(-\mathbf{s}_{i(j)}\log\mathbf{m}_{i(j)}+\mathbf{s}_{i(j)}\log\mathbf{s}_{i(j)} - \mathbf{t}_{i(j)}\log\mathbf{m}_{i(j)}+\mathbf{t}_{i(j)}\log\mathbf{t}_{i(j)}), \tag{4}$$

where $\mathbf{m}_i=\frac{1}{2}(\mathbf{t}_i+\mathbf{s}_i)$. While the JS divergence overcomes the asymmetry shortcoming of the KL divergence, it is still subject to non-smoothness that makes it challenging to optimize. Moreoever, the student may excessively underestimate the probability of rare events as the JS loss does not penalize adequately for matching low probability regions. There also exists a risk of gradient vanishing when $J_{\mathrm{JS}}(\mathbf{t}_i,\mathbf{s}_i)$ degenerates as a constant on distributions with few or even no overlap.

C. Sinkhorn Distance

The Sinkhorn distance provides a fast approximation to the Wasserstein distance by incorporating entropy regularization. This method considers the minimum cost of mass transmission in converting one probability distribution into another.

a) Wasserstein Distance Definition: First, we define the Wasserstein distance, which involves the set of a transportation polytope $U(\mathbf{t}_i,\mathbf{s}_i)$. This set consists of all matrices of $\mathbf{P}\in\mathbb{R}_+^{d\times d}$ that satisfy the following constraints:

$$U(\mathbf{t}_i,\mathbf{s}_i)=\{\mathbf{P}\in\mathbb{R}_+^{d\times d}|\mathbf{P}\mathbf{1}_d=\mathbf{s}_i,\mathbf{P}^{\mathrm{T}}\mathbf{1}_d=\mathbf{t}_i\}, \tag{5}$$

where $\mathbf{1}_d\in\mathbb{R}^d$ is a vector of ones. Given a cost matrix $\mathbf{D}\in\mathbb{R}^{d\times d}$, the Wasserstein distance is:

$$J_{\mathrm{WD}}(\mathbf{t}_i,\mathbf{s}_i)=\min_{\mathbf{P}\in U(\mathbf{t}_i,\mathbf{s}_i)}\langle\mathbf{P},\mathbf{D}\rangle=\sum_{m,n}\mathbf{P}_{m,n}\mathbf{D}_{m,n}, \tag{6}$$

where $\mathbf{D}_{m,n}$ is usually the absolute difference between the m-th and n-th elements of $\mathbf{t}_i$ and $\mathbf{s}_i$:

$$\mathbf{D}_{m,n}=|\mathbf{t}_{i(m)}-\mathbf{s}_{i(n)}|. \tag{7}$$

b) Sinkhorn Distance Definition: To circumvent the substantial computation entailed by solving such an OT problem, Sinkhorn distance is proposed as a fast approximation to the Wasserstein distance for a constrained optimization [18]. It is defined as the inner product between the OT plan $\mathbf{P}^\lambda$ and the cost matrix $\mathbf{D}$:

$$J_{\mathrm{SD}}(\mathbf{t}_i,\mathbf{s}_i)=\left\langle\mathbf{P}^\lambda,\mathbf{D}\right\rangle, \tag{8}$$

where $\lambda>0$ is the weight for entropy regularization.

c) Obtaining the OT Plan $\mathbf{P}^\lambda$: The OT plan $\mathbf{P}^\lambda$ is obtained by minimizing:

$$\mathbf{P}^\lambda=\underset{\mathbf{P}\in U(\mathbf{t}_i,\mathbf{s}_i)}{\operatorname{argmin}}\ \langle\mathbf{P},\mathbf{D}\rangle-\lambda h(\mathbf{P}), \tag{9}$$

where $h(\mathbf{P})$ is the entropy of the matrix $\mathbf{P}$. The entropy term encourages the transport plan to be more spread out for easier optimization. The vanilla solution to $\mathbf{P}^\lambda$ by sample-wise Sinkhorn normalization [18] is performed between $\mathbf{t}_i$ and $\mathbf{s}_i$ in a manner of iterative updates:

$$\left(\mathbf{u}^t,\mathbf{v}^t\right)\leftarrow\left(\mathbf{t}_i\oslash\left(\mathbf{K}^{\mathrm{T}}\mathbf{v}^{t-1}\right),\mathbf{s}_i\oslash\left(\mathbf{K}\mathbf{u}^{t-1}\right)\right), \tag{10}$$

$$\mathbf{K}=\exp(-\frac{\mathbf{D}}{\lambda}), \tag{11}$$

where $\oslash$ indicates element-wise division and t denotes the iteration time. Two vectors $\mathbf{u}\in\mathbb{R}^d,\mathbf{v}\in\mathbb{R}^d$ are non-negative, representing scaling factors used to adjust the transport plan in the Sinkhorn algorithm. Both $\mathbf{u}$ and $\mathbf{v}$ are typically initialized as vectors of ones, i.e., $\mathbf{u}^{(0)}=\mathbf{1}$ and $\mathbf{v}^{(0)}=\mathbf{1}$. Such initialization with all-ones vectors has two benefits: 1) it makes the iterative updates of the involved variables easy to compute at the beginning stage, where some unnecessary complex operations can be saved. 2) it avoids large fluctuations of the values of $\mathbf{u}$ and $\mathbf{v}$ during continuous updates of the algorithm. The kernel matrix $\mathbf{K}\in\mathbb{R}^{d\times d}$ is constructed by applying the Gaussian kernel on $\mathbf{D}$ with the weight λ for entropy regularization. Finally, $\mathbf{P}^\lambda$ is defined as:

$$\mathbf{P}^\lambda=\mathrm{diag}\left(\mathbf{v}^t\right)\mathbf{K}\mathrm{diag}\left(\mathbf{u}^t\right). \tag{12}$$

D. Batch-wise Reformulation

In view of properties of the Sinkhorn distance metric, we can get rid of the sample-wise KD that only works on each teacher-student sample pair, and instead perform KD on groups of teacher and student samples. A batch of b samples all participate in divergence measures with their overall output logits $\mathbf{t}\in\mathbb{R}^{b\times d}$ and $\mathbf{s}\in\mathbb{R}^{b\times d}$ respectively from the teacher and the student. It thereby increases the dimension of the "observational" space via batch-wise reformulation especially when $d\ll b$ holds.

a) Cost Matrix Computation: To compute the cost matrix, we use the ℓ_p-norm to measure the pairwise differences between the i-th and j-th samples in a batch. This results in the entry $\mathbf{D}_{i,j}$ of the "batchified" cost matrix $\mathbf{D}\in\mathbb{R}^{b\times b}$:

$$\mathbf{D}_{i,j}=\|\mathbf{t}_i-\mathbf{s}_j\|_p. \tag{13}$$

b) Sinkhorn Normalization: Before we propose the batch-wise Sinkhorn normalization, we reformulate the sample-wise solution to $\mathbf{P}^{\lambda}$ (Eq. 10) into a equivalent vector-form with iterations only on $\mathbf{K}$:

$$\begin{aligned}\widehat{\mathbf{K}}^t &\leftarrow \operatorname{diag}\left(\mathbf{K}^{t-1}\mathbf{1}_d \oslash \mathbf{s}_i\right)^{-1}\mathbf{K}^{t-1},\\ \mathbf{K}^t &\leftarrow \widehat{\mathbf{K}}^t \operatorname{diag}\left(\left(\widehat{\mathbf{K}}^t\right)^{\mathrm{T}}\mathbf{1}_d \oslash \mathbf{t}_i\right)^{-1},\end{aligned} \tag{14}$$

where $\mathbf{K}^0 = \mathbf{K} \in \mathbb{R}^{d\times d}$ is defined in Eq. 11. For distillation beyond the d-dimensional space, we propose a more compact solution in the matrix-form for batch-wise normalization with $\mathbf{K} \in \mathbb{R}^{b\times b}$:

$$\begin{aligned}\widehat{\mathbf{K}}^t &\leftarrow \operatorname{diag}\left(\mathbf{K}^{t-1}\mathbf{1}_b \oslash \mathbf{w}_s\right)^{-1}\mathbf{K}^{t-1},\\ \mathbf{K}^t &\leftarrow \widehat{\mathbf{K}}^t \operatorname{diag}\left(\left(\widehat{\mathbf{K}}^t\right)^{\mathrm{T}}\mathbf{1}_b \oslash \mathbf{w}_t\right)^{-1},\end{aligned} \tag{15}$$

where $\mathbf{w}_s$ and $\mathbf{w}_t$ respectively represent the weights of each element involved in the batch-wise KD from the student and teacher. Without loss of generality, we assume uniform distributions with $\mathbf{w}_s = \mathbf{w}_t = \frac{1}{b}\mathbf{1}_b$. Given such conditions, updates on $\mathbf{K}^t$ (Eq. 15) can be further simiplified as:

$$\begin{aligned}\widehat{\mathbf{K}}^t &\leftarrow \mathbf{K}^{t-1} \oslash \left(\mathbf{K}^{t-1}\mathbf{1}_b\mathbf{1}_b^{\top}\right),\\ \mathbf{K}^t &\leftarrow \widehat{\mathbf{K}}^t \oslash \left(\mathbf{1}_b\mathbf{1}_b^{\top}\widehat{\mathbf{K}}^t\right).\end{aligned} \tag{16}$$

Out of simplicity, irrelevant constants are excluded from the equations above. With a pre-determined number of iterations T, the OT matrix is derived:

$$\mathbf{P}^{\lambda} = \mathbf{K}^T \tag{17}$$

c) Sinkhorn Loss: The batch-wise SinKD loss is:

$$\mathcal{L}_{\mathrm{SD}} = J_{\mathrm{SD}}(\mathbf{t},\mathbf{s}) = \left\langle \mathbf{P}^{\lambda}, \mathbf{D}\right\rangle = \sum_{i,j}\mathbf{K}_{i,j}^T\mathbf{D}_{i,j} \tag{18}$$

We illustrate the entire pipeline in Fig. 2 and Alg. 1.

d) Total Loss: For each batch of b samples, we follow the standard training approach to use a combination of the cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$, the KL loss $\mathcal{L}_{\mathrm{KL}}$, and our $\mathcal{L}_{\mathrm{SD}}$:

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{b}[&(1-\alpha)\mathcal{L}_{\mathrm{CE}}(\mathbf{y}_i,\mathbf{s}_i)\\ &+\alpha\mathcal{L}_{\mathrm{KL}}(\mathbf{t}_i,\mathbf{s}_i)] + \beta\mathcal{L}_{\mathrm{SD}},\end{aligned} \tag{19}$$

where α and β are weights, and $\mathcal{L}_{\mathrm{KL}}(\mathbf{t}_i,\mathbf{s}_i) \approx \mathcal{L}_{\mathrm{CE}}(\mathbf{t}_i,\mathbf{s}_i)$ given that the second term in $J_{\mathrm{KL}}(\mathbf{t}_i,\mathbf{s}_i)$ can be viewed as a constant during distillation. Such weighted combination (Eq. 19) takes advantages of the KL divergence in dealing with typical situations while benefiting from the robustness of Sinkhorn Distance in solving severe distribution mismatches.

E. Variants of our SinKD

a) Alternative $\mathbf{D}$*:* Apart from Eq. 21, we can further take into account all the d-dimensional logits of b samples by flattenning $\mathbf{t}$ and $\mathbf{s}$ for a $\mathbf{D} \in \mathbb{R}^{bd\times bd}$:

$$\mathbf{D}_{im,jn} = |\mathbf{t}_{i(m)} - \mathbf{s}_{j(n)}|. \tag{20}$$

The normalization is performed on $\mathbf{K} \in \mathbb{R}^{bd\times bd}$ with $\mathbf{w}_s = \mathbf{w}_t = \frac{1}{bd}\mathbf{1}_{bd}$. In this case, SinKD takes a broader perspective of batch distributions with a multiplied dimension bd, significantly exceeding the sample-wise KD.

b) Extension to Regression Tasks: For regression, the model does not generate probabilities for each option and only produces one scalar ($d = 1$) instead. For a batch of b samples, the outputs of the teacher and student models are denoted as $\mathbf{t} \in \mathbb{R}^{b\times 1}$ and $\mathbf{s} \in \mathbb{R}^{b\times 1}$, respectively. To calculate the batch-wise Sinkhorn distance between the teacher and student, entries of the cost matrix are determined by the absolute differences between the "batchified" regression outputs:

$$\mathbf{D}_{i,j} = |\mathbf{t}_i - \mathbf{s}_j|. \tag{21}$$

In the context of regression, the cross-entropy loss and KL loss are inapplicable. We substitute these terms in Eq. 19 with the mean squared error (MSE) loss $\mathcal{L}_{\mathrm{MSE}}$:

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{b}[&(1-\alpha)\mathcal{L}_{\mathrm{MSE}}(\mathbf{y}_i,\mathbf{s}_i)\\ &+\alpha\mathcal{L}_{\mathrm{MSE}}(\mathbf{t}_i,\mathbf{s}_i)] + \beta\mathcal{L}_{\mathrm{SD}}.\end{aligned} \tag{22}$$

c) Extension to One-hot Label Fine-Tuning: Our SinKD can also be applied for fine-tuning models only with one-hot labels where logits from the teacher model are unavailable. In such scenarios, the one-hot labels can be treated as the logits of a "hypothetical" one-hot teacher model. Given the predominant zeros in one-hot logits, traditional divergence measures (e.g., KL) become impotent as they fail to handle such extreme cases in divergence quantification. Therefore, we simply ignore the KL term in Eq. 19 and present:

$$\mathcal{L} = \sum_{i=1}^{b}\alpha\mathcal{L}_{\mathrm{CE}}(\mathbf{y}_i,\mathbf{s}_i) + \beta\mathcal{L}_{\mathrm{SD}}. \tag{23}$$

Algorithm 1 Sinkhorn Algorithm for Knowledge Distillation

Require:
Teacher output $\mathbf{t}$, Student's output $\mathbf{s}$,
Hyper-parameter λ, Maximum number of iterations T
Ensure:
Sinkhorn loss $\mathcal{L}_{SD}$
Initialize: Compute distance matrix $\mathbf{D}_{i,j} = \|\mathbf{t}_i - \mathbf{s}_j\|_p$
Initialize: Compute kernel matrix $\mathbf{K} \leftarrow \exp\left(-\frac{\mathbf{D}}{\lambda}\right)$
Set iteration counter $t \leftarrow 0$
while $t < T$ **do**
  Normalize rows: $\mathbf{K} \leftarrow \mathbf{K} \oslash \left(\mathbf{K}\mathbf{1}_b\mathbf{1}_b^{\mathrm{T}}\right)$
  Normalize columns: $\mathbf{K} \leftarrow \mathbf{K} \oslash \left(\mathbf{1}_b\mathbf{1}_b^{\mathrm{T}}\mathbf{K}\right)$
  Increment iteration counter $t \leftarrow t + 1$
end while
return Sinkhorn loss $\mathcal{L}_{SD} \leftarrow \langle \mathbf{K}, \mathbf{D}\rangle = \sum_{i,j}\mathbf{K}_{ij}\mathbf{D}_{ij}$

IV. Experimental Settings

A. Datasets

We evaluate our method on eight tasks of the GLUE benchmark [20], including seven discriminative tasks: CoLA [65], SST-2 [66], MNLI [67], MRPC [68], RTE [69], QNLI [70]

QQP [71], and one regression task STS-B [72]. For evaluation metrics, we follow previous works [27], [32], [31] to report accuracy (MNLI, SST-2, QNLI, QQP, and RTE), F1 score (MRPC), Matthews correlation coefficient (CoLA) and and Spearman's rank correlation coefficient (STS-B). Note that all discriminative tasks of GLUE are associated with extremely-low dimension of logits output ($d = 3$ for MNLI and $d = 2$ for the remainings tasks).

B. Implementation Details

Our SinKD is implemented in PyTorch Transformers [73]. For comparability, we follow AD-KD [27] to set $\text{BERT}_{\text{base}}$ as the teacher and a smaller BERT_6 [25] as the student for task-specific fine-tuning. For generalizability, we also validate SinKD on T0 [3] and GPT-Neo [22]. Note that for all GLUE tasks except MNLI, two definitions of $\mathbf{D}$ (Eqs. 21,20) are equivalent given the constraint of $\sum_{m=1}^{d} \mathbf{t}_{i(m)} = 1$ and $d = 2$. Consequently, we use the default $\mathbf{D}$ by Eq. 21. Out of simplicity, we set $p = 1$ (ℓ_1-norm) for $\mathbf{D}$. The hyper-parameters are optimized via grid search to determine the learning rate $lr \in \{2e-5, 3e-5, 4e-5, 5e-5\}$, $\alpha \in \{0.8, 0.9, 1.0\}$, $b \in \{16, 32, 64\}$, and $\tau_{\text{KL}} \in \{1, 2, 3, 4\}$. We empirically set $\tau_{\text{SD}} = 2$, $\lambda = 0.1$, $T = 20$, and $\beta = 0.8$. Discussions on the effect of T, λ, τ_{SD}, τ_{KL}, b, α, and β can be found in Sec. V-C.

C. Baselines

We compare SinKD with SOTA KD methods. For logits-based KD, we include the vanilla KD [6], RCO [36], DML [74], and PD [25]. For representation-based KD, we compare PKD [35], TinyBERT [26], RKD [37], CKD [75], SFTN [38], TAKD [76], ProKT [77], MGSKD [77], MetaDistill [31], ReAugKD [32], and AD-KD [27]. For fair comparison, we follow [27] to exclude MiniLM [78] and MobileBERT [79] as their two-stage settings involve both task-agnostic and task-specific distillation. In contrast, we emphasize a generalized one-stage setting without "customized" pre-training. Baseline results are quoted [27], [32].

V. Results and Discussions

A. Comparison with SOTA

Table I shows that SinKD outperforms all baselines on most datasets. Specifically, SinKD achieves an average increase of 0.47% and 1.17% over AD-KD [27] and ReAugKD [32], respectively, for classification tasks. Compared with AD-KD, SinKD reduces the performance gap between the student and the teacher over 57%, highlighting that SinKD effectively narrows such gap by injecting structural knowledge from teacher to student. In the regression task of STS-B, our approach also attains state-of-the-art performance relative to other baseline methods compared. Our improvements can be attributed to the unique properties of Sinkhorn distillation, where the integrated characteristics of distributions are respected during distillation and thereafter facilitate impartial, efficient knowledge transfer for robust convergence. We also notice that SinKD does not rank the top on QNLI, possibly due to suboptimal hyper-parameters for this specific task. Meticulous tuning of hyper-parameters might yield better results, but will impair comparability and therefore is beyond the scope of the present study.

B. Ablation Study

a) Sinkhorn loss benefits the student the most among all losses: In order to study the impact of each loss component, we carry out ablation studies on three variations of SinKD: 1) SinKD without Sinkhorn loss, 2) SinKD without KL divergence loss, and 3) SinKD without cross-entropy loss. As revealed in Table II, significant decreases over all tasks can be observed when Sinkhorn loss is removed. In addition, the drop of performance without KL divergence loss suggests that the proposed SinKD is supplementary to the vanilla KL divergence in distribution measurements. With respect to the cross-entropy loss, its supervision from ground-truth labels directly improves the student model and consequently should be kept intact during distillation. Each component contributes to diminishing the gap between the student and the teacher. Our proposed Sinkhorn loss brings the most pronounced gains over other losses, confirming the validity of Sinkhorn distance as a stable metric for convergence to global optimum.

b) Batch-wise SinKD excels sample-wise SinKD: Table III demonstrates the superiority of the batch-wise SinKD over the sample-wise SinKD on all tasks, implying that the Sinkhorn distance is indeed adept in handling the deviation of the student from the teacher in a high-dimensional space. The sample-wise distillation treats each instance independently while neglecting the overall tendency in a batch of samples.

c) SinKD surpasses distillation methods based on variants of f-divergence: To investigate if the existing distillation methods with f-divergence measures can achieve competitive results, we replace our Sinkhorn loss with losses based on: 1) RKL divergence, 2) JD divergence, and 3) total variation distance (TVD). To fairly compare with SinKD, each loss mentioned above is combined with cross-entropy loss and KL divergence loss during distillation. Table IV shows that our SinKD outperforms three other distillation methods on all datasets, verifying the superiority of Sinkhorn distance over variants of f-divergence measures in matching distributions. It is notable that traditional divergence measures like KL, RKL and JS are inapplicable for regression tasks like STS-B as they rely on specific probability values, which showcases the superiority and broad applicability of batch- wise knowledge distillation Additionally, it is worth noting that among the other three methods, TVD exhibits slight advantages over RKL and JS divergence on average. Such finding is consistent with [9].

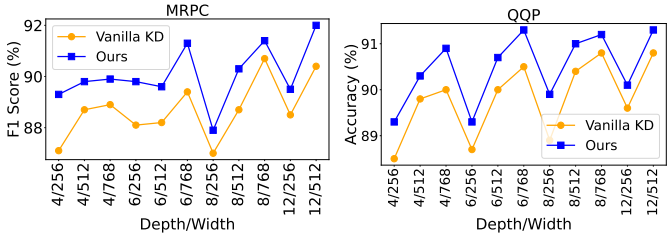

Fig. 3. Performance at different student scales on MRPC (left) & QQP (right). Best viewed magnified.

d) SinKD generalizes well on student LLMs across scales: To thoroughly assess the influence of size of student

TABLE I
COMPARISON WITH SOTA METHODS ON GLUE WITH $\text{BERT}_{\text{BASE}}$ AS THE TEACHER (T) AND BERT_6 AS THE STUDENT (S). THE FINAL COLUMN PRESENTS TWO AVERAGES: THE FIRST EXCLUDES BOTH THE MNLI-(M/MM) AND STS-B SCORES, AND THE SECOND EXCLUDES ONLY THE MNLI-(M/MM) SCORES.

Method	**#Params.**	**COLA** (MCC)	**SST-2** (ACC)	**MNLI-(m/mm)** (ACC)	**MRPC** (F1)	**RTE** (ACC)	**QNLI** (ACC)	**QQP** (ACC)	**STS-B** (Spear)	**Avg**
$\text{BERT}_{\text{base}}$ (T) [1]	110M	60.3	93.7	84.9/84.8	91.4	71.1	91.7	91.5	89.4	83.28/84.16
BERT_6 (S) [25]	66M	51.2	91.0	81.7/82.6	89.2	66.1	89.3	90.4	88.3	79.53/80.79
Task-specific Representation-based Distillation										
PKD [35]	66M	45.5	91.3	81.3/-	85.7	66.5	88.4	88.4	86.2	77.63/78.86
TinyBERT [26]	66M	53.8	92.3	83.1/83.4	88.8	66.5	89.9	90.5	88.3	80.30/81.44
RKD [37]	66M	53.4	91.7	-	86.1	68.6	89.5	90.9	-	80.03/-
CKD [75]	66M	55.1	93.0	83.6/84.1	89.6	67.3	90.5	91.2	89.0	81.11/82.24
SFTN [38]	66M	53.6	91.5	-	85.3	68.5	89.5	90.4	88.5	79.80/81.04
TAKD [76]	66M	53.8	91.4	-	85.0	68.5	89.6	90.7	88.0	79.83/81.00
ProKT [77]	66M	54.3	91.3	-	86.3	68.4	89.7	90.9	88.6	80.15/81.36
MGSKD [80]	66M	49.1	91.7	83.3/83.9	89.8	67.9	90.3	91.2	88.5	80.00/81.21
MetaDistill [31]	66M	58.6	92.3	-	86.8	69.4	90.4	91.0	89.1	81.42/82.51
ReAugKD [32]	66M	59.4	92.5	-	86.3	70.4	90.7	91.2	-	81.75/-
AD-KD [27]	66M	58.3	91.9	83.4/**84.2**	91.2	70.9	**91.2**	91.2	**89.2**	82.45/83.41
Task-specific Logits-based Distillation										
Vanilla KD [6]	66M	53.6	91.1	82.7/83.1	89.4	66.8	90.1	90.5	88.7	80.25/81.46
RCO [36]	66M	53.6	91.4	-	85.1	67.6	89.7	90.6	88.3	79.67/80.90
DML [74]	66M	53.7	91.5	-	85.1	68.4	89.6	90.3	88.1	79.77/80.96
PD [25]	66M	-	91.1	82.5/83.4	89.4	66.7	89.4	90.7	-	-/-
SinKD (ours)	66M	**60.2**	**93.1**	**83.8/84.2**	**91.3**	**71.1**	90.5	**91.3**	**89.2**	**82.92/83.81**

TABLE II
EFFECT OF DIFFERENT LOSS TERMS ON GLUE. FOR STS-B, $\mathcal{L}_{CE}$ AND $\mathcal{L}_{KL}$ ARE REPLACED BY $\mathcal{L}_{\text{MSE}}(\mathbf{y}_i, \mathbf{s}_i)$ AND $\mathcal{L}_{\text{MSE}}(\mathbf{t}_i, \mathbf{s}_i)$.

Method	**COLA** (MCC)	**SST-2** (ACC)	**MNLI-(m/mm)** (ACC)	**MRPC** (F1)	**RTE** (ACC)	**QNLI** (ACC)	**QQP** (ACC)	**STS-B** (Spear)
SinKD (ours)	60.2	93.1	83.8/84.2	91.3	71.1	90.5	91.3	89.2
w/o $\mathcal{L}_{SD}$	53.6	91.1	82.7/83.1	89.4	66.8	90.1	90.5	88.7
w/o $\mathcal{L}_{KL}$	56.2	91.7	82.3/83.0	90.1	69.3	90.2	90.7	88.4
w/o $\mathcal{L}_{CE}$	58.0	92.3	83.5/84.1	91.1	70.4	90.4	91.3	88.4
w/o $\mathcal{L}_{KL}$&$\mathcal{L}_{SD}$	51.2	91.0	81.7/82.6	89.2	66.1	89.3	90.4	88.3

TABLE III
COMPARISON BETWEEN THE SAMPLE-WISE AND BATCH-WISE SINKD ON GLUE. N/A INDICATES THAT THE METHOD IS NOT APPLICABLE FOR THIS TASK.

Level	**COLA** (MCC)	**SST-2** (ACC)	**MNLI-(m/mm)** (ACC)	**MRPC** (F1)	**RTE** (ACC)	**QNLI** (ACC)	**QQP** (ACC)	**STS-B** (Spear)
Sample-wise	54.2	91.5	83.2/83.7	90.4	69.0	90.3	91.2	N/A
Batch-wise	60.2	93.1	83.8/84.2	91.3	70.0	90.5	91.3	89.2

LLMs on the performance of SinKD, we conduct an extensive analysis with comparison between the vanilla KD and SinKD. Without loss of generality, we take two tasks (MRPC and QQP) for demonstration. A broad range of model scales [25] are employed to explore the adaptability and robustness of SinKD when applied on student models with various configurations. Note that both the vanilla KD and our SinKD are logits-based KD methods, which are independent of model structure by nature and thus enjoy high versatility. As illustrated in Fig. 3, SinKD consistently outperforms the vanilla KD on both two tasks across all scales. Such generalizability on model size confirms the potential of SinKD as an efficient and reliable KD method.

C. Discussion on Hyper-parameters

a) T as the number of Sinkhorn iterations: We vary the number of iterations T and results (see Table V) reflect the importance of selecting an appropriate T. An increase of T to 20 respectively improves F1 scores for MRPC (91.3) and accuracy for QQP (91.3), suggesting that sufficient iterations is crucial to approximation and convergence. Nevertheless, raising the iterations to 50 yields no further improvement. It indicates the existence of a saturation point, beyond which additional iterations are not beneficial but redundant. Hence, we set $T = 20$ throughout experiments.

b) λ as the weight of entropy-regularization: The Sinkhorn distance is derived from the entropy-regularized OT problem, where the regularization term promotes a more dispersed, less concentrated OT plan. In other words, entropy-regularization would enhance the numerical stability and

TABLE IV
COMPARISON WITH DISTILLATION METHODS BASED ON VARIANTS OF f-DIVERGENCE ON GLUE. N/A INDICATES THAT THE METHOD IS NOT APPLICABLE FOR THIS TASK.

Method	**Complexity**	**COLA** (MCC)	**SST-2** (ACC)	**MNLI-(m/mm)** (ACC)	**MRPC** (F1)	**RTE** (ACC)	**QNLI** (ACC)	**QQP** (ACC)	**STS-B** (Spear)
RKL [8]	$O(bd)$	53.9	91.6	82.9/83.4	90.5	67.1	90.1	91.1	N/A
JS [9]	$O(bd)$	54.2	92.2	83.1/83.7	90.7	68.9	90.3	91.2	N/A
TVD [9]	$O(bd)$	54.1	92.1	83.3/83.8	90.9	70.0	90.2	91.2	87.9
SinKD	$O\left(b^2(d+T)\right)$	60.2	93.1	83.8/84.2	91.3	71.1	90.5	91.3	89.2

TABLE V
EFFECT OF T ON MRPC & QQP.

Number of Iterations T	**MRPC** (F1)	**QQP** (ACC)
2	90.6	91.0
5	90.9	91.0
10	90.9	91.1
20	91.3	91.3
50	91.3	91.3

TABLE VI
EFFECT OF b ON MRPC & SST-2.

Batchsize b	**MRPC** (F1)	**SST-2** (ACC)
2	90.5	91.3
8	90.8	92.4
16	91.3	92.8
32	91.1	93.1
64	91.3	93.1

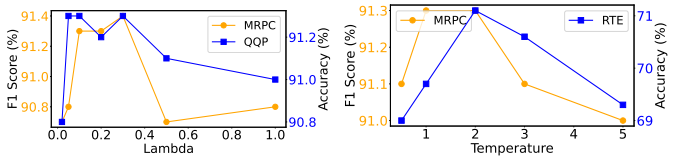

Fig. 4. Effect of λ on MRPC & QQP (left) and τ_{SD} on MRPC & RTE (right). Best viewed magnified.

computational tractability of the solution to OT problem. Theoretically, λ dictates the balance between the accuracy of the OT approximation and the stability of the solution. A larger λ results in a smoother and more stable solution, albeit potentially less accurate. A smaller λ yields a more accurate solution at the risk of numerical instability. As demonstrated in Fig. 4(a), a λ within the range of 0.1 to 0.3 appears to achieve an optimal trade-off among various aspects. Out of consistency, we choose $\lambda = 0.1$ throughout experiments.

c) τ_{SD} as the temperature in Sinkhorn loss: Fig. 4(b) systematically investigates the influence of τ_{SD} on distillation on the tasks of MRPC and QQP. Our findings indicate that the default empirical setting $\tau_{\text{SD}} = 2$ is appropriate for both two tasks. A smaller τ_{SD} may cause the student model to concentrate solely on learning the most salient features, neglecting the nuanced but valuable information present in less probable categories for classification. On the other hand, a larger τ_{SD} results in smoother and more uniform probability distributions, which confuses the student model to discern between essential and irrelevant information.

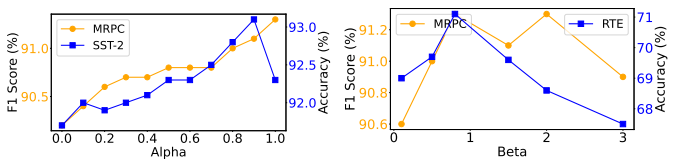

Fig. 5. Effect of α on MRPC & SST-2 (left) and β on MRPC & RTE (right). Best viewed magnified.

d) b as the number of batch size: The batch size is closely associated with the efficiency of geometric structural learning since the distribution divergences are measured within each batch of samples for the proposed Sinkhorn distance minimization. An increased batch size contributes to enhancing the student's understanding of complex geometric inter-sample relations present within the dataset. Empirical evidence, as presented in Table VI, elucidates a positive correlation between the batch size and evaluation results (F1 scores for the MRPC and accuracy for SST-2). Such performance gains are theoretically grounded in the premise that larger batches provide a more expansive dimensional space, allowing for a more comprehensive representation of the geometric configuration during each optimization step. A larger batch size b effectively provides the model with exposure to the intrinsic geometric variance of the dataset, potentially accelerating the transfer and assimilation of the teacher model's knowledge. However, such benefit becomes negligible when the batch size increases beyond 32, where both metrics for MRPC and SST-2 remain almost unchanged. This observation suggests the existence of a saturation point, which delineates the boundary where the advantages of augmenting the geometric sampling space are outweighed by the computational overhead.

TABLE VII
EFFECT OF τ_{KL} ON MRPC & SST-2.

Temperature τ_{KL}	**MRPC** (F1)	**SST-2** (ACC)
1	90.5	92.6
2	90.8	93.1
3	91.1	92.7
4	91.3	92.5

e) τ_{KL} as the temperature in KL loss: Table VII provides the results of how the temperature τ_{KL} affects the knowledge distillation. For the MRPC dataset, a monotonically increasing trend in the F1-score is observed as τ_{KL} ranges from 1 to 4. The best results of F1-score are achieved at $\tau_{\text{KL}} = 4$. Conversely, the accuracy for SST-2 is maximized at a lower temperature ($\tau_{\text{KL}} = 2$), beyond which a diminution occurs. It exemplifies the dualistic role of τ_{KL}: 1) refining the granularity of probability distributions at lower temperatures and 2) fostering generalization at higher settings. The optimal value of τ_{KL} is to be task-dependent, underscoring the necessity of task-specific hyperparameter tuning for SinKD applications.

f) α and β as the loss weights: In the total loss (Eq. 19) of SinKD, we introduce α and β to balance the contributions from the cross-entropy loss, KL divergence loss, and Sinkhorn distance loss. A comprehensive evaluation of various

TABLE VIII
RESULTS OF T0 ON SUPERGLUE.

Method	RTE (ACC)	CB (ACC)
$T0_{11B}$ (T)	89.1	100
$T0_{3B}$ (S)	87.1	94.6
KL	87.4	94.6
KL+RKL	87.8	96.4
KL+JS	88.1	96.4
KL+SinKD	**89.9**	**98.2**

combinations of α and β can be found in Fig. 5. Each time, we only adjust one parameter and keep the other one fixed. Our findings indicate that a larger α generally produces better performance, corroborating that knowledge transfer from the teacher model does play an indispensable role. In line with the results of SinKD without the cross-entropy loss (see Table II), $\alpha = 1$ causes a drastic decline on SST-2, suggesting that "soft" guidance from the teacher model is not equivalent to "hard" supervision from ground-truth labels. Additionally, we observe that $\beta = 0.8$ yields promising results for both two tasks. Consequently, we keep $\beta = 0.8$ fixed and find the optimal α in $\{0.8, 0.9, 1.0\}$ respectively for each task.

D. Training Cost and Complexity

In our experiments, the training cost for SinKD depends on the architecture of the language model and the complexity of the tasks. We report the runtime on GLUE in Table IX. We have endeavored to optimize the training process of SinKD to maintain a balance between efficiency and performance. Also, we compare the computational complexity of the training process with SOTA methods in Table X. Under the same experimental settings where $b \leq 32$, $d \leq 3$, $T = 20$, and $e = 768$, our complexity is smaller when $b(d + T) < e$. This indicates that the computational complexity of our method is lower than any representation-based KD method under these conditions. When comparing with the SOTA logits-based KD methods, our proposed SinKD ($O(b^2(d + T))$) has a higher complexity than their $O(bd)$ since $b(d + T) > d$. However, given that our results are achieved with moderate values of b and T, the proposed SinKD still maintains a competitive computational complexity.

E. Generalizability on Generative LLMs

To demonstrate the potential of our SinKD on generative LLMs, we perform distillation on various transformer architectures including the encoder-decoder T0 [3] and the decoder-only GPT-Neo [22]. Specifically, $T0_{11B}$, GPT-$Neo_{1.3B}$ and GPT-$Neo_{2.7B}$ serve as the teacher while $T0_{3B}$, GPT-Neo_{125M} and GPT-$Neo_{2.7B}$ as the student. We validate SinKD on the SuperGLUE [21] benchmark against SOTA KD methods based on 1) KL divergence, 2) RKL divergence, and 3) JS divergence. We choose two datasets of RTE [69], CB [81] for demonstrative experiments of typical real-word NLP tasks.

a) Comparison with SOTA methods: Table VIII, XI and XII show that the proposed SinKD surpasses all other KD methods. Specifically, when compared to the teacher model

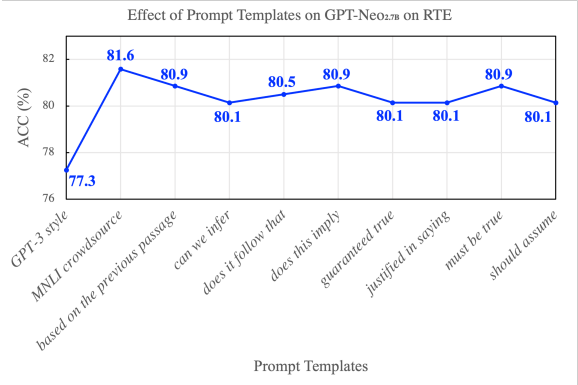

Fig. 6. Effect of prompt templates on GPT-$Neo_{2.7B}$ on the RTE task of SuperGLUE. All templates are chosen from the templates of the *PromptSource* [82].

GPT-Neo1.3B, a student model with an order of magnitude fewer parameters (10 times fewer) achieves commendable performance utilizing our SinKD method. On the RTE task, all three existing methods of KL, RKL, and JS perform similarly with respect to the vanilla fine-tuning performance of GPT-Neo_{125M}. In contrast, the proposed SinKD improved the vanilla performance by 0.6%. For the CB task, the performance gains by SinKD are much more obvious, where an increase of 4.5% in accuracy is achieved. Furthermore, in scenarios where the parameter sizes of the student and teacher models are more closely matched, our method facilitates the student model in reaching performance levels comparable to those of the teacher model. Such findings showcase that SinKD can generalize to generative LLMs whose output logits are of high dimension equivalent to the size of the tokenizer vocabulary. Moreover, the performance gap between T0 and GPT-Neo can be ascribed to two reasons: 1) Architecture. The encoder-decoder architectures are generally more suitable for discriminative tasks compared with the decoder-only architectures since the former better comprehend the input-output relationships with bi-directional modeling. 2) Model scale. According to the scaling laws [5], the performance of GPT-Neo is expected to grow exponentially with billions of parameters increased. Under the limited GPU budget, experiments on larger decoder-only models are currently unavailable.

b) Effect of Prompt Templates: In consideration of the effect of prompt design on performance of generative models, we also compare the results of GPT-$Neo_{2.7B}$ under different prompt templates. Out of simplicity and comparability, we obtain the prompt templates of the SuperGLUE benchmark from the *PromptSource* [82]. From Figs. 6 and 7, it is observed that the GPT-Neo is **not** robust to prompt templates and the variance of performance is non-negligible for both two tasks of the RTE and CB. It is noteworthing that the GPT-3 style prompt template is not the best template even though the GPT-Neo shares a similar architecture with GPT-3 [5]. We believe that due to the nature of these discrimination tasks, prompts which are originally designed for encoder-only and encoder-decoder transformers, are not appropriate for direct adaptation for the GPT-style models. It requires both a reformulation of

TABLE IX
TRAINING COSTS OF SINKD ON GLUE. TEACHER: $\text{BERT}_{\text{BASE}}$, STUDENT: BERT_6

Task	**CoLA**	**MNLI**	**SST-2**	**QNLI**	**MRPC**	**QQP**	**RTE**	**STS-B**
Training Time	20 minutes	6 hours	30 minutes	1 hour	15 minutes	6 hours	5 minutes	12 minutes

TABLE X
COMPUTATIONAL COMPLEXITIES PER BATCH FOR EXISTING SOTA METHODS IN KNOWLEDGE DISTILLATION. VARIABLES: b = BATCH SIZE, d = NUMBER OF CATEGORICAL LOGITS (CLASSES), k = NUMBER OF INTERMEDIATE LAYERS, e = EMBEDDING DIMENSION, N, F, C = PARAMETERS FROM SFTN [38], m = NUMBER OF RELATION HEADS IN MGSKD [80], T = NUMBER OF SINKHORN ITERATIONS. ATTENTION-BASED DISTILLATION COSTS ARE EXCLUDED.

Method	**Complexity**
PKD [35]	$O(b(ke+d))$
TinyBERT [26]	$O(b(ke+d))$
RKD [37]	$O(b^3e)$
CKD [75]	$O(b^2e+bd)$
SFTN [38]	$O(Nb(F^2Ce+d))$
TAKD [76]†	-
ProKT [77]†	-
MGSKD [80]	$O(mb^2e+mb^3)$
MetaDistill† [31]	-
ReAugKD [32]	$O(b^2e)$
AD-KD [27]	$O(b(e+d))$
Vanilla KD [6]	$O(bd)$
RCO [36]	$O(bd)$
DML [74]	$O(bd)$
PD [25]	$O(bd)$
SinKD (ours)	$O(b^2(d+T))$

† The complexity of these methods is not comparable since the teacher model and the additionally introduced teacher assistant model are optimized simultaneously during distillation.

TABLE XI
THE AVERAGED RESULTS OF GPT-NEO ON SUPERGLUE UNDER THREE DIFFERENT PROMPTS (MEAN±STD.). FOR EACH TASK, THREE PROMPT TEMPLATES ARE RANDOMLY CHOSEN FROM THE *PromptSource* [82] RESPECTIVELY. FOR RTE, WE USE THE PROMPT TEMPLATES OF *GPT-3 style*, *MNLI crowdsource*, AND *must be true*. FOR CB, WE USE THE PROMPT TEMPLATES OF *GPT-3 style*, *must be true*, AND *should assume*. THE $\text{GPT-NEO}_{1.3\text{B}}$ ACTS AS THE TEACHER WHILE THE $\text{GPT-NEO}_{125\text{M}}$ ACTS AS THE STUDENT.

Method	**RTE** (ACC)	**CB** (ACC)
$\text{GPT-Neo}_{1.3\text{B}}$ (T)	75.4±1.8	86.9±1.0
$\text{GPT-Neo}_{125\text{M}}$ (S)	64.4±3.2	80.4±1.8
KL	64.7±2.7	83.3±2.1
KL+RKL	64.3±3.2	83.3±2.7
KL+JS	64.6±2.9	82.1±3.1
KL+SinKD	**65.0±3.1**	**84.5±2.7**

the task and an explicitly defined and re-organized prompt for generative models to comprehend the task. Furthermore, the best design of prompt templates is specific to downstream tasks and beyond the scope of the present study.

F. Glass-box Evaluation

To enhance our intrinsic evaluation, we investigate and propose the following additional analyses:

1) Representation of Hidden States: We delve deeper into how the SinKD reshapes the feature space by focusing on the representations at various layers of the student model. Given the same input examples, we compared the differences

TABLE XII
THE AVERAGED RESULTS OF GPT-NEO ON SUPERGLUE UNDER THREE DIFFERENT PROMPTS (MEAN±STD.). FOR EACH TASK, THREE PROMPT TEMPLATES ARE RANDOMLY CHOSEN FROM THE *PromptSource* [82] RESPECTIVELY. FOR RTE, WE USE THE PROMPT TEMPLATES OF *GPT-3 style*, *MNLI crowdsource*, AND *must be true*. FOR CB, WE USE THE PROMPT TEMPLATES OF *GPT-3 style*, *must be true*, AND *should assume*. THE $\text{GPT-NEO}_{2.7\text{B}}$ ACTS AS THE TEACHER WHILE THE $\text{GPT-NEO}_{1.3\text{B}}$ ACTS AS THE STUDENT.

Method	**RTE** (ACC)	**CB** (ACC)
$\text{GPT-Neo}_{2.7\text{B}}$ (T)	79.9±2.3	91.7±6.7
$\text{GPT-Neo}_{1.3\text{B}}$ (T)	75.4±1.8	86.9±1.0
KL	75.8±0.9	90.5±4.5
KL+RKL	77.1±1.5	88.7±2.7
KL+JS	77.6±0.9	91.0±1.8
KL+SinKD	**78.1±0.6**	**91.7±2.1**

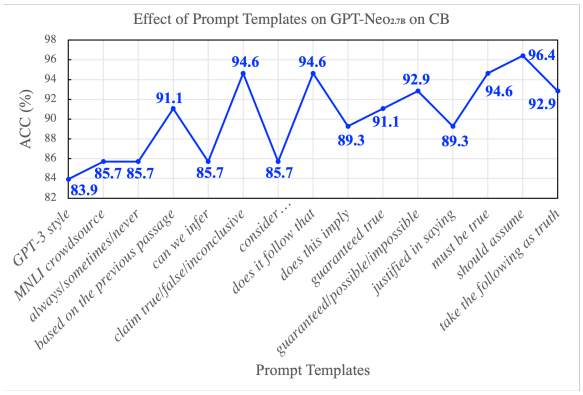

Fig. 7. Effect of prompt templates on $\text{GPT-Neo}_{2.7\text{B}}$ on the CB task of SuperGLUE. All templates are chosen from the templates of the *PromptSource* [82].

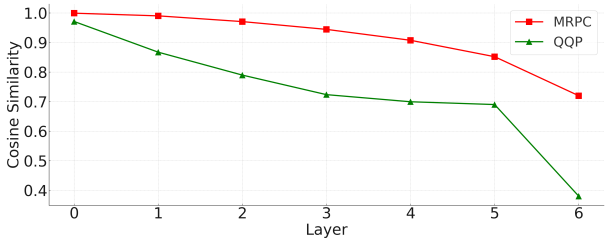

Fig. 8. Cosine similarity of hidden space representation on MRPC and QQP.

in representations of intermediate layers of the student models before and after distillation. As shown in Fig. 8, it is observed

TABLE XIII
MAE BETWEEN THE TEACHER AND STUDENT IN MRPC.

	Base	KL	RKL	JS	SinKD
MAE	0.0485	0.0464	0.0463	0.0461	**0.0458**
Reduction in MAE	0	0.0021	0.0022	0.0024	**0.0027**

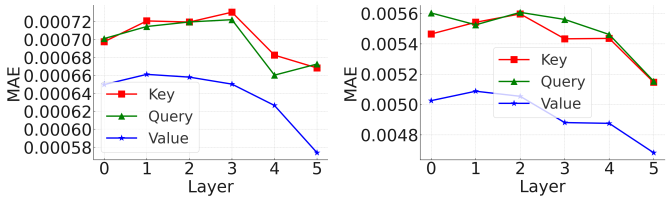

Fig. 9. MAE of weights across differnet layers on MRPC (left) and QQP (right). Best viewed magnified.

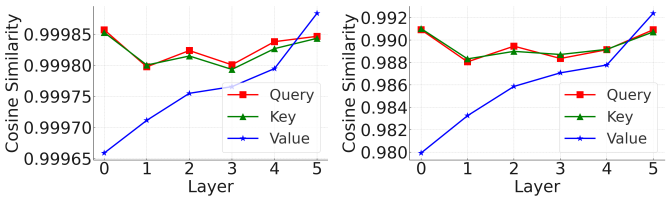

Fig. 10. Cosine similarity of weights across differnet layers on MRPC (left) and QQP (right). Best viewed magnified.

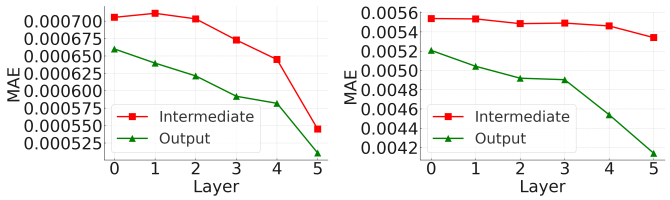

Fig. 11. MAE of attention matrix across differnet layers on MRPC (left) and QQP (right). Best viewed magnified.

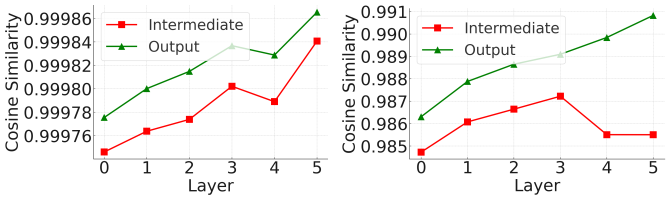

Fig. 12. Cosine similarity of attention matrix across differnet layers on MRPC (left) and QQP (right). Best viewed magnified.

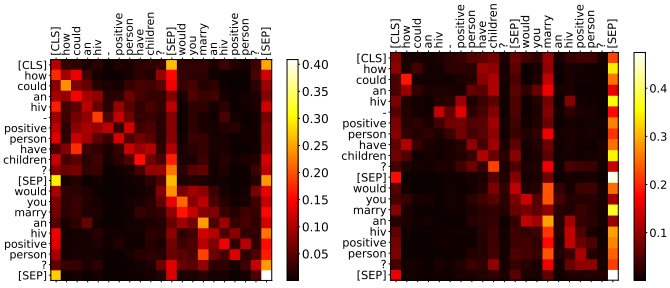

Fig. 13. Visualization of the attention map at the 4-th layer of the student model before (left) and after (right) distillation. The example is cherry-picked from the QQP task. Best viewed magnified.

that the cosine similarity of representations decreases with an increasing layer depth (e.g., 0.97→0.38 for QQP and 1.00→0.72 for MRPC), indicating that the extent of change in representations is expected to be higher in latter layers. Notably, representations from the last hidden state undergo the most significant transform, suggesting that logits-based distillatiosn predominantly impacts representations of the final layer. Moreover, we also find that the variation of representations might depend on: a) the number of training samples of the downstream tasks, b) the knowledge gap between the pre-training and the downstream tasks.

For tasks such as MRPC and RTE, given limited amount of training data, representations of shallow layers exhibit minimal variation while those of deeper layers exhibit significant changes. This is evidenced by an upward trajectory in the slope of the curve of the cosine similarity versus layer depth. For instance, in the MRPC task, the absolute value of cosine similarity decreases along the layer depth (1 at layer 0→0.7 at layer 6). However, the volume of such changes, namely the slope of the curve, continuously increased by a factor up to 13 (0.01 at layer 1 → 0.02 at layer 2 → 0.03 at layer 3 → 0.04 at layer 4 → 0.06 at layer 5 → 0.13 at layer 6). This confirms that changes are expected to be stronger in deeper layers.

On the other hand, for tasks with abundant training data, the amount of variations is related to the knowledge gap where the task-specific knowledge is missing during pre-training. Substantial variations of representations in shallow layers are observed while changes in upper layers are less pronounced, demonstrated by the downward slope of the curve of the cosine similarity versus layer depth. For instance, in QQP, the slope has been reduced to one-tenth of its original value (0.10 at layer 1 → 0.08 at layer 2 → 0.07 at layer 3 → 0.02 at layer 4 → 0.01 at layer 5). Since the QQP dataset contains questions across various domains from the Quora website, such task of identifying multiple duplicated questions poses challenges to language models on the extent of knowledge. Therefore, a large amount of data from specific domains and tasks would provide models with supplemental knowledge by fundamentally affecting parameters throughout layers, bridging the gap between pre-training and fine-tuning.

2) Patterns of Attention Mechanism: In consideration of the importance of the attention mechanism, we measure the similarity of self-attention matrices (query, key, and value) before and after distillation across layers: a) between the base and fine-tuned student models, b) between the student and teacher models. In the present study, we take the tasks of QQP and Microsoft Research Paraphrase Corpus (MRPC) for demonstration (see Figs. 9, 10, 11, 12).

a) Differences between the Base and Fine-tuned Student Models: The consistent patterns across tasks are observed when comparing student models before and after distillation. The layer-wise cosine similarity of the 'value' matrices in the multi-head attention increases as layer deepens. In contrast, the cosine similarity for 'query' and 'key' matrices does not show strong correlation with the layer depth. Furthermore, the mean absolute errors (MAEs) for the query, key, and value components all decrease as the layer deepens, suggesting a trend towards lower variance in deeper layers. For instance, for the student model under the MRPC task, the MAE of 'value' matrices decreases by 13%. The cosine similarity of

TABLE XIV
COMPARISON WITH SOTA METHODS ON ONE-HOT EVALUATION. N/A INDICATES THAT THE METHOD IS NOT APPLICABLE FOR THIS TASK.

Method	**COLA** (MCC)	**SST-2** (ACC)	**MNLI** (ACC)	**MRPC** (F1)	**RTE** (ACC)	**QNLI** (ACC)	**QQP** (ACC)	**STS-B** (Spear)
CE/MSE	51.2	91.0	81.7/82.6	89.2	66.1	89.3	90.4	88.3
TaiLr [83]	49.3	91.1	81.9/82.7	89.1	66.1	89.3	90.5	N/A
MixCE [84]	51.9	91.3	82.0/82.8	89.7	66.4	**89.9**	90.4	N/A
CE/MSE + sample-wise SinKD	53.7	91.4	81.9/82.8	89.7	67.1	89.4	90.4	N/A
CE/MSE + batch-wise SinKD	**55.4**	**91.9**	**82.3/83.1**	**90.0**	**67.9**	**89.9**	**90.6**	**88.6**

TABLE XV
EXPERIMENTAL RESULTS ON THE TESTING SET OF CIFAR-100 [85]. RESULTS OF BASELINES REPORTED IN CRD [86] AND METADISTILL [31] ARE DIRECTLY CITED HERE.

Teacher	ResNet-56	ResNet-110	ResNet-110	VGG-13	ResNet32x4	ResNet32x4
Student	ResNet-20	ResNet-20	ResNet-32	VGG-8	ShuffleNetV1	ShuffleNetV2
Teacher	72.34	74.31	74.31	74.64	79.42	79.42
Student	69.06	69.06	71.14	70.36	70.50	71.82
KD [6]	70.66	70.67	73.08	72.98	74.07	74.45
FitNet [87]	69.21	68.99	71.06	71.02	73.59	73.54
AT [88]	70.55	70.22	72.31	71.43	71.73	72.73
SP [89]	69.67	70.04	72.69	72.68	73.48	74.56
CC [90]	69.63	69.48	71.48	70.71	71.14	71.29
VID [91]	70.38	70.16	72.61	71.23	73.38	73.40
RKD [37]	69.61	69.25	71.82	71.48	72.28	73.21
PKT [92]	70.34	70.25	72.61	72.88	74.10	74.69
AB [93]	69.47	69.53	70.98	70.94	73.55	74.31
FT [94]	69.84	70.22	72.37	70.58	71.75	72.50
ProKT [77]	70.98	70.74	72.95	74.12	71.74	74.68
CRD [86]	71.16	71.46	73.48	**73.94**	75.11	75.65
MetaDistil [31]	71.25	71.40	73.35	73.65	-	-
SinKD	**71.65**	**71.54**	**73.74**	**73.94**	**75.34**	**75.94**

TABLE XVI
TRAINING COST OF SINKD ON CIFAR-100 [85].

Teacher	ResNet-56	ResNet-110	ResNet-110	VGG-13	ResNet32x4	ResNet32x4
Student	ResNet-20	ResNet-20	ResNet-32	VGG-8	ShuffleNetV1	ShuffleNetV2
Training Time	60 minutes	80 minutes	100 minutes	30 minutes	120 minutes	130 minutes

the output dense weight increases with layer depth, while the intermediate dense weight does not exhibit a clear trend with respect to the layer. A decrease of MAE can be observed in both the output and intermediate dense matrices with an increasing layer depth. Overall, these observations suggest that larger variations are more prevalent in earlier layers.

In Fig. 13, we provide the visualization of the attention map from the 4-th layer of the student model before and after distillation. The example is cherry-picked from the the QQP task, which requires the model to determine whether two input questions share the same semantics. We opt to visualize the 4-th attention layer because the initial layers primarily "digest" information at the token-level with exchange flow across adjacent tokens, whereas the subsequent layers gradually aggregate information from [SEP] and [CLS] tokens. The intermediate layer (e.g., 4-th) provides insight into the behavior of the [CLS] token and the network's overall dynamics. The distilled [CLS] token can better focus on information critical for making judgments (such as 'have children', 'marry'), whereas the attention distribution of the model before distillation is relatively uniform across all normal tokens.

b) Differences between the Student and Teacher Models: With respect to the similarity between the student and the teacher, an increase in the cosine similarity of the classifier's weight, together with a reduction in MAE, can be observed in the distilled student model. Such improvement appears more pronounced than those observed with other divergence measures. For instance, as shown in Table XIII, compared to the baseline without distillation, the MAE reduction between student and teacher models using SinKD exceeded the MAE reduction achieved by KL divergence distillation by 30%. These findings imply that the SinKD exhibits superior efficacy in capturing the characteristics of the teacher model, which is indicative of a successful knowledge transfer.

3) Layer-wise Performance Analysis: We meticulously investigate the impact of integrating the classifier heads at various layers as side outputs. Our findings indicate that for architectures with relatively sparse neurons (e.g., lower dimension of the hidden states), there exists a negligible correlation between the layer-depth and the overall model performance, where the side outputs from various layers share similar performance. Conversely, for denser networks, performance of side outputs consistently increases along the layer depth. We believe for large language models, the capacity of intermediate representations get continuously augmented as layers pile up. Therefore, logits-based KD, where supervision signals intervene from the very last layer towards front layers, should take advantage of representations at each layer to

pinpoint semantic nuances between the student and the teacher.

G. One-hot Label Fine-tuning

One-hot labels can be conceptualized as logit outputs from a "theoretical" one-hot teacher model. Specifically for both discriminative and generative language models, the one-hot labels, either in the dimension of the number of categories or the size of vocabulary, are widely adopted in maximizing likelihood estimation (MLE). Unlike existing divergence measures (e.g., KL), our SinKD method can also be extended to fine-tuning language models with one-hot labels.

For performance comparison, we contrast our SinKD against MLE and two of its variants: TaiLr [83] and MixCE [84]. TaiLr decomposes the TVD and seeks to minimize its upper bound alongside the MLE loss. MixCE introduces an approximation technique and employs an interpolation coefficient to merge it with the forward cross-entropy. While both TaiLr and MixCE utilize novel distance metrics that theoretically surpass forward cross-entropy in certain aspects, they inherit the fundamental limitation of existing divergence measures in representing complex data distributions. For a fair comparison, we follow TaiLr [83] and MixCE [84] to adopt their default hyper-parameter settings. For our SinKD method, we keep hyper-parameters of CE/MSE configurations unchanged and additionally set $\alpha = 1$, $\beta = 0.8$, $\tau_{\text{SD}} = 2$, $\lambda = 0.1$ and $T = 30$. Table XIV reveals that our batch-wise SinKD consistently surpasses both MLE and other training paradigms across all tasks in the GLUE benchmark.

Due to the limited information of one-hot labels, the sample-wise SinKD does not adequately capture the characteristics of the underlying distribution. Moreover, it is the dependence on probability distributions that impedes MLE, TaiLr, MixCE, and the sample-wise SinKD from being applicable for regression tasks (e.g., STS-B). Such limitation further highlights the necessity and superiority of our batch-wise reformulation.

H. Sinkhorn Distillation on Image Classification

Beyond the large language models, we investigate whether the proposed SinKD can be extended to distillation of vision models. Besides, we explore the versatility and efficacy of our SinKD across scenarios where the teacher and student models do and do NOT share the same architecture.

a) Experimental Settings: We align our experimental setup with the protocols established in CRD [86] and MetaDistill [31], focusing on the CIFAR-100 [85] classification challenge as a typical task in computer vision. Our experiments span a variety of student-teacher pairs which differ in capacity and architectural design. We mainly investigate ResNet [85] and VGG [95] configurations. In addition, we study the distillation between heterogeneous architectures. Specifically, we set ResNet as the teacher and ShuffleNet [96], [97] as the student. Results of the top-1 accuracy are reported for the distilled student models. For hyper-parameters, we set $p = 1$ (ℓ_1-norm) for $\mathbf{D}$. Their settings are optimized via grid search to determine $\alpha \in \{0.85, 0.9, 0.95\}$ and $\beta \in \{0.08, 0.8, 8\}$. We empirically set the learning rate $lr = 0.05$, $b = 8$, $\tau_{\text{KL}} = 4$, $\tau_{\text{SD}} = 2$, $\lambda = 0.1$ and $T = 30$. Although a more exhaustive grid search might yield improved outcomes, we are aimed at demonstrating the robustness and applicability of SinKD instead of pursuing the optimal hyper-parameter combinations.

b) Baselines: Our method is compared with 13 distillation methods, including one SOTA distillation method in image classification [86], one cutting-edge language model distillation framework [31], and other prevalent knowledge distillation methodologies such as ProKT [77].

c) Results: We show the experimental results of SinKD distilling ResNet [85], VGG [95] and ShuffleNet [96], [97] with six different teacher-student pairs. As shown in Table XV, SinKD consistently surpasses all baseline methods in every configuration tested, including the current benchmark in image classification distillation, CRD [86], as well as alternative approaches with sophisticated feature and loss function designs. Remarkably, SinKD achieves superior results without resorting to CRD's additional mechanisms like negative sampling and contrastive learning. Such performance underscores the adaptability of our method, confirming its effectiveness across both homogeneous and heterogeneous architectures. The training cost is reported in Table XVI.

VI. Conclusion

We propose the Sinkhorn distance for divergence measure and present the SinKD to address limitations of existing distillation methods. Besides, we propose the batch-wise reformulation to capture geometric intricacies of distributions across samples in the high-dimensional space. Extensive experiments on GLUE and SuperGLUE benchmarks confirm the superiority of our SinKD over SOTA methods from the aspect of comparability, validity, and generalizability. Furthermore, additional experiments on one-hot label finetuning and vision tasks further demonstrate the universality of our method, showcasing its effectiveness across a broader range of applications.

Future work includes exploring application to representation-based KD, attention-based KD, and their extension to other tasks (e.g., document summarization and machine translation). Moreover, it would be an interesting topic to delve deep into conducting rigorous mathematical analysis and proofs for the effect of the batch-wise Sinkhorn distillation on intermediate representations.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[3] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja *et al.*, "Multitask prompted training enables zero-shot task generalization," *arXiv preprint arXiv:2110.08207*, 2021.

[4] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI*, 2019.

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," in *Proceedings of the Advances in Neural Information Processing Systems*, 2020, pp. 1877–1901.

[6] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[7] L. Tu, R. Y. Pang, S. Wiseman, and K. Gimpel, "Engine: Energy-based inference networks for non-autoregressive machine translation," *arXiv preprint arXiv:2005.00850*, 2020.

[8] Y. Gu, L. Dong, F. Wei, and M. Huang, "Knowledge distillation of large language models," *arXiv preprint arXiv:2306.08543*, 2023.

[9] Y. Wen, Z. Li, W. Du, and L. Mou, "f-divergence minimization for sequence-level knowledge distillation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 10 817–10 834.

[10] H. Yin, P. Molchanov, J. M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz, "Dreaming to distill: Data-free knowledge transfer via deepinversion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8715–8724.

[11] G. Fang, Y. Bao, J. Song, X. Wang, D. Xie, C. Shen, and M. Song, "Mosaicking to distill: Knowledge distillation from out-of-domain data," in *Proceedings of the Advances in Neural Information Processing Systems*, 2021, pp. 11 920–11 932.

[12] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2017, pp. 214–223.

[13] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," *arXiv preprint arXiv:1606.07947*, 2016.

[14] T. Kim, J. Oh, N. Y. Kim, S. Cho, and S.-Y. Yun, "Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 8 2021, pp. 2628–2635.

[15] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *Proceedings of the International Conference on Learning Representations*, 2017.

[16] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," in *Proceedings of the Advances in Neural Information Processing Systems*, 2016.

[17] L. Yu, Y. Song, J. Song, and S. Ermon, "Training deep energy-based models with f-divergence minimization," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2020, pp. 10 957–10 967.

[18] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 26, 2013.

[19] S. Vallender, "Calculation of the wasserstein distance between probability distributions on the line," *Theory of Probability & Its Applications*, vol. 18, no. 4, pp. 784–786, 1974.

[20] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018.

[21] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "Superglue: A stickier benchmark for general-purpose language understanding systems," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 32, 2019.

[22] S. Black, L. Gao, P. Wang, C. Leahy, and S. Biderman, "GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow," Mar. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5297715

[23] X. Cui, Y. Qin, Y. Gao, E. Zhang, Z. Xu, T. Wu, K. Li, X. Sun, W. Zhou, and H. Li, "Sinkhorn distance minimization for knowledge distillation," *arXiv preprint arXiv:2402.17110*, 2024.

[24] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[25] I. Turc, M.-W. Chang, K. Lee, and K. Toutanova, "Well-read students learn better: On the importance of pre-training compact models," *arXiv preprint arXiv:1908.08962*, 2019.

[26] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," in *Findings of the Association for Computational Linguistics: EMNLP*, 2020, pp. 4163–4174.

[27] S. Wu, H. Chen, X. Quan, Q. Wang, and R. Wang, "Ad-kd: Attribution-driven knowledge distillation for language model compression," *arXiv preprint arXiv:2305.10010*, 2023.

[28] S. Li, M. Lin, Y. Wang, Y. Wu, Y. Tian, L. Shao, and R. Ji, "Distilling a powerful student model via online knowledge distillation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8743–8752, 2022.

[29] C. Yang, Z. An, L. Cai, and Y. Xu, "Knowledge distillation using hierarchical self-supervision augmented distribution," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 2, pp. 2094–2108, 2022.

[30] J. Gou, Y. Chen, B. Yu, J. Liu, L. Du, S. Wan, and Z. Yi, "Reciprocal teacher-student learning via forward and feedback knowledge distillation," *IEEE Transactions on Multimedia*, 2024.

[31] W. Zhou, C. Xu, and J. McAuley, "Bert learns to teach: Knowledge distillation with meta learning," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 7037–7049.

[32] J. Zhang, A. Muhamed, A. Anantharaman, G. Wang, C. Chen, K. Zhong, Q. Cui, Y. Xu, B. Zeng, T. Chilimbi *et al.*, "Reaugkd: Retrieval-augmented knowledge distillation for pre-trained language models," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2023, pp. 1128–1136.

[33] L. Li, P. Dong, A. Li, Z. Wei, and Y. Yang, "Kd-zero: Evolving knowledge distiller for any teacher-student pairs," *Advances in Neural Information Processing Systems*, vol. 36, pp. 69 490–69 504, 2023.

[34] L. Li, P. Dong, Z. Wei, and Y. Yang, "Automated knowledge distillation via monte carlo tree search," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 413–17 424.

[35] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient knowledge distillation for bert model compression," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, 2019, pp. 4323–4332.

[36] X. Jin, B. Peng, Y. Wu, Y. Liu, J. Liu, D. Liang, J. Yan, and X. Hu, "Knowledge distillation via route constrained optimization," in *Proceedings of the International Conference on Computer Vision*, 2019, pp. 1345–1354.

[37] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3967–3976.

[38] D. Y. Park, M.-H. Cha, D. Kim, B. Han *et al.*, "Learning student-friendly teacher networks for knowledge distillation," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 34, pp. 13 292–13 303, 2021.

[39] C. Yang, Z. An, H. Zhou, F. Zhuang, Y. Xu, and Q. Zhang, "Online knowledge distillation via mutual contrastive learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 8, pp. 10 212 – 10 227, 2023.

[40] S. Vallender, "Calculation of the wasserstein distance between probability distributions on the line," *Theory of Probability & Its Applications*, vol. 18, no. 4, pp. 784–786, 1974.

[41] C. Villani and C. Villani, "The wasserstein distances," *Optimal Transport: Old and New*, pp. 93–111, 2009.

[42] J. Zhang, T. Liu, and D. Tao, "An optimal transport analysis on generalization in deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 6, pp. 2842–2853, 2021.

[43] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 30, 2017.

[44] G. Peyré, M. Cuturi *et al.*, "Computational optimal transport: With applications to data science," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[45] J. Gu, X. Qian, Q. Zhang, H. Zhang, and F. Wu, "Unsupervised domain adaptation for covid-19 classification based on balanced slice wasserstein distance," *Computers in Biology and Medicine*, p. 107207, 2023.

[46] P. Chen, R. Zhao, T. He, K. Wei, and Q. Yang, "Unsupervised domain adaptation of bearing fault diagnosis based on join sliced wasserstein distance," *ISA transactions*, vol. 129, pp. 504–519, 2022.

[47] S. He, Y. Jiang, H. Zhang, J. Shao, and X. Ji, "Wasserstein unsupervised reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 6, 2022, pp. 6884–6892.

[48] Y. Wei, X. Li, L. Lin, D. Zhu, and Q. Li, "Causal discovery on discrete data via weighted normalized wasserstein distance," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[49] C. Weilin, Q. Jie, C. Ruichu, and H. Zhifeng, "On the role of entropy-based loss for learning causal structure with continuous optimization," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[50] B. Du, W. Xie, Y. Li, Q. Yang, W. Zhang, R. R. Negenborn, Y. Pang, and H. Chen, "Safe adaptive policy transfer reinforcement learning for distributed multiagent control," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[51] C. Bai, T. Xiao, Z. Zhu, L. Wang, F. Zhou, A. Garg, B. He, P. Liu, and Z. Wang, "Monotonic quantile network for worst-case offline reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[52] Y. Lan, X. Xu, Q. Fang, and J. Hao, "Sample efficient deep reinforcement learning with online state abstraction and causal transformer

model prediction," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
[53] Q. Zhang, S. Leng, X. Ma, Q. Liu, X. Wang, B. Liang, Y. Liu, and J. Yang, "Cvar-constrained policy optimization for safe reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
[54] F. Zhou, C. Shui, M. Abbasi, L.-É. Robitaille, B. Wang, and C. Gagné, "Task similarity estimation through adversarial multitask neural network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 466–480, 2020.
[55] C. Frogner, C. Zhang, H. Mobahi, M. Araya, and T. A. Poggio, "Learning with a wasserstein loss," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 28, 2015.
[56] Y. Liu, L. Zhu, X. Wang, M. Yamada, and Y. Yang, "Bilaterally normalized scale-consistent sinkhorn distance for few-shot image classification," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
[57] S. Li, I. J. Unanue, and M. Piccardi, "Improving machine translation and summarization with the sinkhorn divergence," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2023, pp. 149–161.
[58] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," *Proceedings of the Advances in Neural Information Processing Systems*, 2017.
[59] T. T. Nguyen and A. T. Luu, "Improving neural cross-lingual abstractive summarization via employing optimal transport distance for knowledge distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 11 103–11 111.
[60] J. Xu, C. Li, F. Huang, Z. Li, X. Xie, and S. Y. Philip, "Sinkhorn distance minimization for adaptive semi-supervised social network alignment," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
[61] S. Lu, H.-J. Ye, and D.-C. Zhan, "Faculty distillation with optimal transport," *arXiv preprint arXiv:2204.11526*, 2022.
[62] R. Bhardwaj, T. Vaidya, and S. Poria, "Knot: Knowledge distillation using optimal transport for solving nlp tasks," *arXiv preprint arXiv:2110.02432*, 2021.
[63] A. Genevay, G. Peyré, and M. Cuturi, "Learning generative models with sinkhorn divergences," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2018, pp. 1608–1617.
[64] S. Kammammettu and Z. Li, "Scenario reduction and scenario tree generation for stochastic programming using sinkhorn distance," *Computers & Chemical Engineering*, vol. 170, p. 108122, 2023.
[65] A. Warstadt, A. Singh, and S. R. Bowman, "Neural network acceptability judgments," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 625–641, 2019.
[66] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1631–1642.
[67] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1112–1122.
[68] B. Dolan and C. Brockett, "Automatically constructing a corpus of sentential paraphrases," in *International Workshop on Paraphrasing*, 2005.
[69] L. Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo, "The fifth pascal recognizing textual entailment challenge." *IEEE Transactions on Automatic Control*, vol. 7, p. 8, 2009.
[70] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2383–2392.
[71] Z. Chen, H. Zhang, X. Zhang, and L. Zhao, "Quora question pairs," 2018.
[72] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation," *arXiv preprint arXiv:1708.00055*, 2017.
[73] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.
[74] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4320–4328.
[75] G. Park, G. Kim, and E. Yang, "Distilling linguistic context for language model compression," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 364–378.
[76] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5191–5198.
[77] W. Shi, Y. Song, H. Zhou, B. Li, and L. Li, "Learning from deep model via exploring local targets," 2020.
[78] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 33, pp. 5776–5788, 2020.
[79] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "Mobilebert: a compact task-agnostic bert for resource-limited devices," *arXiv preprint arXiv:2004.02984*, 2020.
[80] C. Liu, C. Tao, J. Feng, and D. Zhao, "Multi-granularity structural knowledge distillation for language model compression," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 1001–1011.
[81] M.-C. De Marneffe, M. Simons, and J. Tonhauser, "The commitmentbank: Investigating projection in naturally occurring discourse," in *Proceedings of Sinn und Bedeutung*, vol. 23, no. 2, 2019, pp. 107–124.
[82] S. H. Bach, V. Sanh, Z.-X. Yong, A. Webson, C. Raffel, N. V. Nayak, A. Sharma, T. Kim, M. S. Bari, T. Fevry, Z. Alyafeai, M. Dey, A. Santilli, Z. Sun, S. Ben-David, C. Xu, G. Chhablani, H. Wang, J. A. Fries, M. S. Al-shaibani, S. Sharma, U. Thakker, K. Almubarak, X. Tang, X. Tang, M. T.-J. Jiang, and A. M. Rush, "Promptsource: An integrated development environment and repository for natural language prompts," *arXiv preprint arXiv:2202.01279*, 2022.
[83] H. Ji, P. Ke, Z. Hu, R. Zhang, and M. Huang, "Tailoring language generation models under total variation distance," in *Proceedings of the International Conference on Learning Representations*, 2022.
[84] S. Zhang, S. Wu, O. İrsoy, S. Lu, M. Bansal, M. Dredze, and D. Rosenberg, "Mixce: Training autoregressive language models by mixing forward and reverse cross-entropies," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 9027–9050.
[85] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[86] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *Proceedings of the International Conference on Learning Representations*, 2019.
[87] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets. arxiv 2014," *arXiv preprint arXiv:1412.6550*, 2014.
[88] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proceedings of the International Conference on Learning Representations*, 2016.
[89] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1365–1374.
[90] B. Peng, X. Jin, J. Liu, D. Li, Y. Wu, Y. Liu, S. Zhou, and Z. Zhang, "Correlation congruence for knowledge distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.
[91] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019, pp. 9163–9171.
[92] N. Passalis and A. Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 268–284.
[93] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge transfer via distillation of activation boundaries formed by hidden neurons," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3779–3787.
[94] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 31, 2018.
[95] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[96] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6848–6856.

[97] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 116–131.