Dual-Space Knowledge Distillation for Large Language Models

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen*, and Jinan Xu
Beijing Key Lab of Traffic Data Analysis and Mining,
Beijing Jiaotong University, Beijing, China
{smzhang22,zhang_xue,zengksun,chenyf,jaxu}@bjtu.edu.cn

Abstract

Knowledge distillation (KD) is known as a promising solution to compress large language models (LLMs) via transferring their knowledge to smaller models. During this process, white-box KD methods usually minimize the distance between the output distributions of the two models so that more knowledge can be transferred. However, in the current white-box KD framework, the output distributions are from the respective output spaces of the two models, using their own prediction heads. We argue that the space discrepancy will lead to low similarity between the teacher model and the student model on both representation and distribution levels. Furthermore, this discrepancy also hinders the KD process between models with different vocabularies, which is common for current LLMs. To address these issues, we propose a dual-space knowledge distillation (DSKD) framework that unifies the output spaces of the two models for KD. On the basis of DSKD, we further develop a cross-model attention mechanism, which can automatically align the representations of the two models with different vocabularies. Thus, our framework is not only compatible with various distance functions for KD (*e.g.*, KL divergence) like the current framework, but also supports KD between any two LLMs regardless of their vocabularies. Experiments on task-agnostic instruction-following benchmarks show that DSKD significantly outperforms the current white-box KD framework with various distance functions, and also surpasses existing KD methods for LLMs with different vocabularies[1].

1 Introduction

Existing large language models (LLMs) have exhibited strong generalization abilities on various tasks due to their huge model capacities (Chowdhery et al., 2023; Touvron et al., 2023; OpenAI, 2023). With faith in the scaling law (Kaplan et al., 2020), the amount of parameters in current LLMs is expanded steadily to achieve higher intelligence. However, the increasing parameters also bring high deployment costs in real scenarios. For this problem, knowledge distillation (KD; Hinton et al., 2015) is one of the promising solutions to compress large models with acceptable performance sacrifice. During the process of KD, the large model typically serves as the teacher and provides supervision signals for a small model (known as the student), and thus the knowledge and the abilities of the teacher can be transferred to the lightweight student.

Currently, KD algorithms for LLMs are usually under two frameworks, *i.e.*, black-box KD and white-box KD. Black-box KD uses the teacher's decoding sequences as the training data of the student and directly optimizes the cross-entropy loss on the one-hot target. (Kim and Rush, 2016; Fu et al., 2023; Li et al., 2023). By contrast, white-box KD methods usually minimize the distance (*e.g.*, KL divergence) between the output distributions of the teacher and the student, which theoretically transfer more information and usually perform better than black-box KD (Wen et al., 2023; Gu et al., 2023; Ko et al., 2024). Although the framework of white-box KD has shown its superiority, the distributions of the student and the teacher in this framework are from different output spaces since they are produced by different prediction heads. At the beginning of this work, we first reveal two inherent limitations in this framework due to the discrepancy of output spaces:

- **Low Teacher-Student Similarity:** The current framework usually yields low similarity between the teacher and the student on both representation and distribution levels (§2.2.1);
- **Requirements on the Same Vocabulary:** A key condition for current white-box KD is that the two models should share the same

[*] Yufeng Chen is the corresponding author.
[1] Our code is publicly available at https://github.com/songmzhang/DSKD.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 18164–18181
November 12-16, 2024 ©2024 Association for Computational Linguistics

vocabulary, which, however, is hardly satisfied for various LLMs in this era (§2.2.2).

Towards these limitations, we then propose a new framework for white-box KD, named dual-space knowledge distillation (DSKD), which is as simple as the current white-box KD framework but addresses the issues due to the space discrepancy. Specifically, DSKD unifies the output spaces of the two models by projecting the output hidden states[2] of the teacher/student to the representation spaces of the student/teacher, where we can use the shared prediction heads to produce the two distributions in the same output spaces. In particular, for models with different vocabularies, we further develop a cross-model attention (CMA) mechanism to automatically align the tokens in two differently tokenized sequences. Like the current framework, DSKD is also compatible with existing distance functions for distributions, including KL divergence, JS divergence, and so on. Meanwhile, with CMA, we can transform distributions of the two LLMs into the same shape, which makes our framework more general and can be applied to any two LLMs regardless of their vocabularies.

We evaluate our framework on instruction-following benchmarks under both settings that the two LLMs have the same/different vocabularies. Experimental results showcase that for LLMs with the same vocabulary, our DSKD framework significantly outperforms the current white-box KD framework on various distance functions. Moreover, DSKD with CMA surpasses all existing KD methods for LLMs with different vocabularies.

To sum up, the contributions are as follows:

- We empirically reveal that the current white-box KD framework limits the similarity between the student and the teacher due to their different output spaces.
- As a solution, we propose a new framework for white-box KD, named dual-space knowledge distillation (DSKD), which unifies the output spaces of the distributions from the teacher and the student for more effective KD.
- Based on DSKD, we further develop a cross-model attention mechanism to support KD between LLMs with different vocabularies.
- Experiments show that our DSKD framework significantly outperforms the current white-box KD framework on various distance functions and surpasses existing KD methods for LLMs with different vocabularies.

2 Background and Preliminary Study

2.1 Current Framework for White-Box KD

Given a sequence $\mathbf{x}$, current LLMs generally learn the casual language modeling objective at each token position i via the cross-entropy loss:

$$\mathcal{L}_{ce} = -\sum_{i}^{|\mathbf{x}|} \log q_\theta(x_i^*|\mathbf{x}_{<i}), \tag{1}$$

where $q_\theta(x_i^*|\mathbf{x}_{<i})$ denotes the probability of the student model on the target token x_i^* conditioning on the context $\mathbf{x}_{<i}$. On this basis, the current white-box KD framework first feeds this sequence into the teacher model to obtain its token-level probability distributions $p(x_i|\mathbf{x}_{<i})$. Then, the following loss is minimized to push the student distribution $q_\theta(x_i|\mathbf{x}_{<i})$ to the teacher distribution $p(x_i|\mathbf{x}_{<i})$:

$$\mathcal{L}_{kd} = \sum_{i} \mathcal{D}(p(x_i|\mathbf{x}_{<i};\tau)||q_\theta(x_i|\mathbf{x}_{<i};\tau)), \tag{2}$$

where $\mathcal{D}(\cdot||\cdot)$ is the distance function that measures the distance between the two distributions (*e.g.*, KL divergence) and τ is the temperature coefficient to control the sharpness of the distributions.

On the choice of the distance function $\mathcal{D}(\cdot||\cdot)$ in Eqn. (2), there have been several explorations (*e.g.*, reverse KL divergence) in recent literature that aim to improve the performance of KD for LLMs (Wen et al., 2023; Agarwal et al., 2024; Ko et al., 2024; Wu et al., 2024). However, in the following section, we will uncover that no matter which distance function is employed, the current white-box KD framework has two inherent limitations since the two distributions $p(x_i|\mathbf{x}_{<i};\tau)$ and $q_\theta(x_i|\mathbf{x}_{<i};\tau)$ are from different output spaces.

2.2 Limitations of the Current Framework

2.2.1 Low Teacher-Student Similarity

In the current white-box KD framework, the two output distributions in Eqn. (2) are calculated from different output spaces of two models using their respective prediction heads. Then, the student distribution will be optimized toward the teacher distribution by minimizing their distance. However,

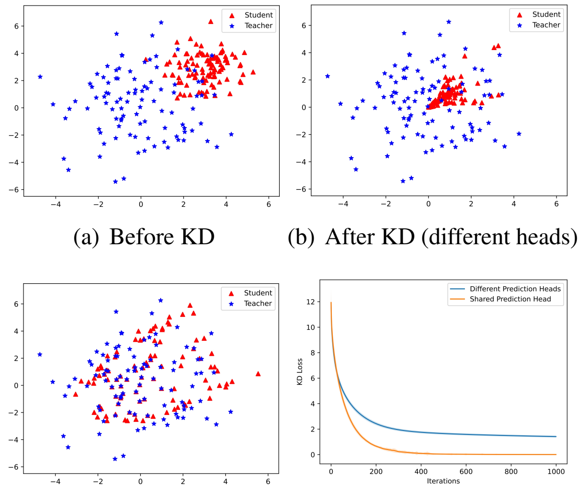
[2] In this paper, "output hidden states" means the hidden states output by the last layer of the model.

we suspect this practice will limit the final similarity between the student and the teacher from two aspects: **a) representation:** as the distributions are the results of the output hidden states through the prediction heads, if the prediction heads of the two models are different, even if the distributions are close, their hidden states will not be similar; **b) distribution:** If the output hidden states of the student and the teacher are not similar, the practical distance between their distributions is difficult to reach its theoretical minimum during optimization.

We verify the above conjectures by a simulation experiment. In this experiment, we randomly initialize two sets of 2-D vectors (one is trainable and the other is frozen) with different mean values and variances to represent the output hidden states of the student and the teacher, respectively (as plotted in Figure 1(a)). Besides, we set two prediction heads to produce probability distributions of the student and the teacher from these vectors. Then, we select KL divergence as the distance function $\mathcal{D}(\cdot||\cdot)$ and simulate the KD process with $\mathcal{L}_{kd}$ in Eqn. (2) for 1000 iterations. After the iterations, we plot the two sets of vectors again and record the loss curve during the whole process in Figure 1.

Firstly, we simulate the process of the current white-box KD framework, which uses distributions from different output spaces produced by different prediction heads. The result in Figure 1(b) shows that the student's hidden states optimized by the current KD framework exhibit distinct structure discrepancy from the teacher's hidden states, reflecting low similarity between them. As a comparison, we then unify the output spaces of the two distributions by sharing the same prediction head for the student and the teacher and conduct the same KD process as above. As shown in Figure 1(c), under this setting, the student's hidden states become more similar and closer to the teacher's hidden states. The significant difference between these two settings indicates that the current KD framework may lead to sub-optimal similarity between the student and the teacher **on the representation level**. By contrast, a better alternative is to unify the output spaces for the distributions of the student and the teacher.

Then, we repeat the simulations of the above two settings 100 times and plot their averaged curves of $\mathcal{L}_{kd}$ in Figure 1(d). As we suspected, when using different prediction heads, the value of KL divergence still leaves a large margin to its theoretical minimum (*i.e.*, 0) after convergence. On the contrary, when using a shared prediction head, the value of KL divergence will converge faster and finally be closer to this minimum. It sufficiently illustrates that the current KD framework also limits the similarity between the two models **on the distribution level**. Besides KL divergence, we also conduct these simulations with other distance functions (*e.g.*, reverse KL divergence, JS divergence, etc.). The results are shown in Appendix A.1, which also support the above conclusions. Additionally, we provide the pseudo code of the simulation experiment in Appendix A.2 to present more details.



(a) Before KD (b) After KD (different heads)

(c) After KD (shared head) (d) Loss curves of KD

Figure 1: Simulation results with KL divergence as the distance function $\mathcal{D}(\cdot||\cdot)$. (a), (b) and (c) plot the student's hidden states and the teacher's hidden states before and after the two KD processes. (d) shows the convergence curves of $\mathcal{L}_{kd}$ in the two KD processes.

2.2.2 Dependency on the Same Vocabulary

As stated in §2.1, the current KD framework minimizes the distance between the two distributions at each token position. However, when the teacher and the student have different vocabularies, the same text may be tokenized into different sequences like $\mathbf{x} = [x_1, x_2, ..., x_n]$ and $\mathbf{y} = [y_1, y_2, ..., y_m]$. Under this circumstance, the teacher distribution $p(y_i|\mathbf{y}_{<i})$ is probably incorrect for $q_\theta(x_i|\mathbf{x}_{<i})$. Additionally, as the output spaces are more different when the prediction heads contain different vocabularies, the produced distributions are even with different dimensions, which is obviously prohibited by Eqn. (2). Therefore, the current white-box KD framework fails to work between LLMs with different vocabularies.

3 Methodology

This section introduces our solutions to the above limitations of the current white-box KD framework. Firstly, we will introduce our new KD framework in §3.1. Then we present a cross-model attention mechanism in §3.2 to extend our framework to support LLMs with different vocabularies.

3.1 Dual-Space Knowledge Distillation Framework

Inspired by the observations in §2.2.1, we design our dual-space knowledge distillation (DSKD) framework. The core idea is to unify the output spaces of the two distributions in Eqn. (2). To achieve this, we project the output hidden states of the teacher/student model into the representation space of the student/teacher model, so that the distributions can be output by the same prediction head and thus lie in **the unified output space**. Next, we will detail how to conduct the projection and unify KD in student and teacher space.

KD in Student Space. In the student space, we first use a linear projector $\mathcal{P}^{t\to s}$ to transform the hidden states of the teacher model into the representation space of the student model. Here, we denote the output hidden states of the whole sequence from the teacher model as $\mathbf{h}^t_{1:n}$. Then the projection process can be formulated as follows:

$$\mathbf{h}^{t\to s}_{1:n} = \mathcal{P}^{t\to s}(\mathbf{h}^t_{1:n}; \theta^{t\to s}_{\mathcal{P}}) \in \mathbb{R}^{n\times d}, \tag{3}$$

where $\theta^{t\to s}_{\mathcal{P}}$ is the trainable parameter of the projector $\mathcal{P}^{t\to s}$ and d is the hidden size of the student model. With the projected hidden states $\mathbf{h}^{t\to s}$, we can obtain the transformed teacher distribution $\mathbf{p}^{t\to s}_{1:n}$ that shares the same output space with the student using the student's prediction head $\mathbf{W}^s \in \mathbb{R}^{d\times|V|}$:

$$\mathbf{p}^{t\to s}_{1:n} = \mathrm{softmax}(\mathbf{h}^{t\to s}_{1:n}\mathbf{W}^s) \in \mathbb{R}^{n\times|V|}_{+}, \tag{4}$$

where $|V|$ is the vocabulary size of the two models. As the projector is randomly initialized at the start of the training, we train the transformed distribution $\mathbf{p}^{t\to s}_{1:n}$ to predict the ground-truth target tokens in the student's sequence with cross-entropy loss[3]:

$$\mathcal{L}^{t\to s}_{ce} = -\sum_i \log(p^{t\to s}(x^*_i|\mathbf{x}_{<i})). \tag{5}$$

Meanwhile, we use this distribution $p^{t\to s}$ as the new teacher distribution and calculate the same loss for KD as Eqn. (2):

$$\mathcal{L}^{stu}_{kd} = \sum_i \mathcal{D}(p^{t\to s}(x_i|\mathbf{x}_{<i};\tau)||q_\theta(x_i|\mathbf{x}_{<i};\tau)), \tag{6}$$

where $\mathcal{D}(\cdot||\cdot)$ is as same as the one in Eqn. (2). Note that we stop the gradient of $p^{t\to s}(x_i|\mathbf{x}_{<i};\tau)$ in Eqn. (6) so that $\mathcal{L}^{stu}_{kd}$ will not collapse.

KD in Teacher Space. Similar to the process in the student space, we also project the hidden states of the student model into the teacher's dimension using another projector $\mathcal{P}^{s\to t}$:

$$\mathbf{h}^{s\to t}_{1:n} = \mathcal{P}^{s\to t}(\mathbf{h}^s_{1:n}; \theta^{s\to t}_{\mathcal{P}}) \in \mathbb{R}^{n\times D}, \tag{7}$$

where D is the hidden size of the teacher model. Then, we use the prediction head of the teacher model $\mathbf{W}^t \in \mathbb{R}^{D\times|V|}$ to obtain the distributions of the student model in the teacher's space:

$$\mathbf{q}^{s\to t}_{\theta 1:n} = \mathrm{softmax}(\mathbf{h}^{s\to t}_{1:n}\mathbf{W}^t) \in \mathbb{R}^{n\times|V|}_{+}, \tag{8}$$

As the teacher distributions in its own space are usually well-trained, we can directly calculate the KD loss in the teacher space:

$$\mathcal{L}^{tea}_{kd} = \sum_i \mathrm{KL}(p(x_i|\mathbf{x}_{<i};\tau)||q^{s\to t}_\theta(x_i|\mathbf{x}_{<i};\tau)), \tag{9}$$

where a difference from Eqn. (6) is that we directly fix KL divergence as $\mathcal{D}(\cdot||\cdot)$ since we found it more appropriate for KD in the teacher space.

The whole loss of DSKD sums the KD losses in both spaces and the cross-entropy loss in Eqn. (5):

$$\mathcal{L}_{dskd} = \mathcal{L}^{stu}_{kd} + \mathcal{L}^{tea}_{kd} + \mathcal{L}^{t\to s}_{ce}. \tag{10}$$

3.2 Cross-Model Attention Mechanism

In the above section, we have introduced our DSKD framework for LLMs with the same vocabulary. For LLMs with different vocabularies, since DSKD always produces distributions with the same dimensions for the student and the teacher via sharing the same prediction heads, the remaining requirement for KD is just to align the tokens in the two sequences tokenized by different tokenizers[4].

To this end, we develop a cross-model attention (CMA) mechanism to learn the alignment between tokens in the two sequences automatically. Specifically, we first concatenate the student's embeddings

[3] Note that we stop the gradient of $\mathbf{W}^s$ in Eqn. (4) to avoid negative effects to the student model

[4] Here we borrow the notations in §2.2.2 and assume that there are m tokens in the teacher's sequence.

of input tokens $\mathbf{e}^s_{1:n}$ and target tokens $\mathbf{e}^s_{2:n+1}$ in the sequence on the last dimension and project them as the query vectors with a query projector $\mathcal{P}^q$:

$$Q = \mathcal{P}^q([\mathbf{e}^s_{1:n}; \mathbf{e}^s_{2:n+1}]; \theta^q_{\mathcal{P}}) \in \mathbb{R}^{n \times 2D}.$$

Similarly, we use the teacher's embeddings and output hidden states to obtain the key and value vectors:

$$\begin{aligned} K &= \mathrm{N}([\mathbf{e}^t_{1:m}; \mathbf{e}^t_{2:m+1}]) \in \mathbb{R}^{m \times 2D}, \\ V &= \mathcal{P}^v(\mathrm{N}(\mathbf{e}^t_{2:m+1}) + \mathrm{N}(\mathbf{h}^t_{1:m}); \theta^v_{\mathcal{P}}) \in \mathbb{R}^{m \times d}, \end{aligned}$$

where we normalize the embeddings and the hidden states of the teacher with their standard deviations like $\mathrm{N}(x) = x/\mathrm{std}(x)$ for faster convergence.

Then, we calculate the attention matrix with the query and the key:

$$\mathbf{a}^{t \to s} = \mathrm{softmax}(\frac{QK^\top}{\sqrt{2D}}) \in \mathbb{R}^{n \times m}. \qquad (11)$$

The attention matrix reflects the alignment relationship from the teacher tokens to the student tokens. Based on this matrix, we can obtain the final projected and aligned hidden states of the teacher model from the weighted sum of the value vectors:

$$\tilde{\mathbf{h}}^{t \to s}_{1:n} = \mathbf{a}^{t \to s} V \in \mathbb{R}^{n \times d}. \qquad (12)$$

Then, we can substitute $\tilde{\mathbf{h}}^{t \to s}$ into Eqn. (4) and train $\tilde{\mathbf{h}}^{t \to s}$ to correctly predict the target tokens of the student model with Eqn. (5). Meanwhile, the teacher distributions produced from $\tilde{\mathbf{h}}^{t \to s}$ are also in the student space and can support the KD process in Eqn. (6)[5].

Besides, we also transpose the matrix to align the student tokens to the teacher tokens:

$$\mathbf{a}^{s \to t} = \mathrm{softmax}(\frac{KQ^\top}{\sqrt{2D}}) \in \mathbb{R}^{m \times n}. \qquad (13)$$

We can project and align the student's hidden states to the teacher's using this alignment matrix:

$$\tilde{\mathbf{h}}^{s \to t}_{1:m} = \mathbf{a}^{s \to t} \mathcal{P}^{s \to t}(\mathbf{h}^s_{1:n}; \theta^{s \to t}_{\mathcal{P}}) \in \mathbb{R}^{m \times D}. \qquad (14)$$

Then, we can substitute $\tilde{\mathbf{h}}^{s \to t}_{1:m}$ into Eqn. (8) and conduct KD in the teacher space with Eqn. (9).

4 Experiments

4.1 Experimental Setup

Data. We evaluate our DSKD framework on several instruction-following datasets following Gu et al. (2023). Specifically, we choose `databricks-dolly-15k` dataset processed by Gu et al. (2023) to conduct the KD process, which contains about 11k samples for training, 1k for validation, and 500 for testing. Besides, we also select Self-Instruct (**SelfInst**), Vicuna-Evaluation (**VicunaEval**), Super Natural Instructions (**S-NI**), and Unnatural Instructions (**UnNI**) as the additional test sets for more comprehensive evaluation.

Models. For student LLMs, we select both GPT2-120M (Radford et al., 2019) and TinyLLaMA-1.1B (Zhang et al., 2024). For GPT2-120M, we employ GPT2-1.5B and Qwen1.5-1.8B (Bai et al., 2023) respectively as the teacher LLMs that have the same/different vocabularies with the student LLMs. For TinyLLaMA-1.1B, we choose LLaMA2-7B (Touvron et al., 2023) and Mistral-7B (Jiang et al., 2023) as the teacher LLMs that have the same/different vocabularies with the student LLMs.

Training and Evaluation. For KD on GPT2, we employ full-finetuning for the teachers and the students. For KD on TinyLLaMA, we finetune the students and the teachers with LoRA. In particular, we set the temperature τ to 2.0 according the performance on the validation set. Besides, all the projectors in our method are linear layers, which only increase few parameters in training (*e.g.*, ≈2M for DSKD on GPT2). For the evaluation, we sampling the responses from the models under 5 random seeds. The final performance is measured by Rouge-L (Lin, 2004) between the generated responses and the human-labeled ones. More details are provided in Appendix B.

4.2 Baselines

We compare our framework with existing methods under two settings:

KD with the same vocabulary. In this setting, we compare DSKD with the current white-box KD framework on the following distance functions:

- **KL.** The standard KL divergence used in KD proposed by Hinton et al. (2015).
- **RKL.** The reverse KL divergence that swaps the two distributions in KL divergence.

[5] For models with different vocabularies, the distribution in Eqn. (4) usually has lower accuracy, so we mask the KD loss in Eqn. (6) when the teacher distribution is incorrect.

Methods	Dolly	SelfInst	VicunaEval	S-NI	UnNI	Avg.
SFT	$22.94_{\pm 0.28}$	$10.11_{\pm 0.36}$	$15.17_{\pm 0.63}$	$16.21_{\pm 0.19}$	$18.68_{\pm 0.09}$	16.62
GPT2-1.5B → GPT2-120M (Same Vocabulary)						
Teacher	$27.19_{\pm 0.23}$	$14.64_{\pm 0.64}$	$16.30_{\pm 0.37}$	$27.55_{\pm 0.30}$	$31.42_{\pm 0.11}$	23.42
SeqKD	$23.68_{\pm 0.25}$	$10.03_{\pm 0.23}$	$14.41_{\pm 0.46}$	$16.36_{\pm 0.18}$	$18.48_{\pm 0.11}$	16.59
KL	$24.54_{\pm 0.48}$	$10.43_{\pm 0.24}$	$15.66_{\pm 0.42}$	$17.24_{\pm 0.27}$	$20.28_{\pm 0.18}$	17.63
w/ DSKD (ours)	$24.70_{\pm 0.24}$	$10.65_{\pm 0.30}$	$15.67_{\pm 0.30}$	$19.51_{\pm 0.21}$	$22.94_{\pm 0.07}$	18.69 (+1.06↑)
RKL	$24.38_{\pm 0.55}$	$10.73_{\pm 0.61}$	$15.71_{\pm 0.39}$	$17.31_{\pm 0.11}$	$20.96_{\pm 0.12}$	17.82
w/ DSKD (ours)	$24.61_{\pm 0.59}$	$11.01_{\pm 0.45}$	$14.98_{\pm 0.48}$	$19.32_{\pm 0.28}$	$22.27_{\pm 0.13}$	18.44 (+0.62↑)
JS	$23.86_{\pm 0.14}$	$10.20_{\pm 0.40}$	$15.50_{\pm 0.23}$	$16.20_{\pm 0.23}$	$19.17_{\pm 0.06}$	16.98
w/ DSKD (ours)	$24.61_{\pm 0.27}$	$11.41_{\pm 0.35}$	$15.40_{\pm 0.28}$	$18.94_{\pm 0.20}$	$21.48_{\pm 0.17}$	18.37 (+1.39↑)
SKL (Ko et al., 2024)	$24.03_{\pm 0.23}$	$10.66_{\pm 0.51}$	$14.70_{\pm 0.37}$	$17.99_{\pm 0.15}$	$21.18_{\pm 0.16}$	17.71
w/ DSKD (ours)	$25.24_{\pm 0.28}$	$10.50_{\pm 0.13}$	$15.76_{\pm 0.43}$	$18.34_{\pm 0.44}$	$20.87_{\pm 0.11}$	18.14 (+0.43↑)
SRKL (Ko et al., 2024)	$24.48_{\pm 0.19}$	$10.35_{\pm 0.38}$	$14.88_{\pm 0.24}$	$16.53_{\pm 0.23}$	$19.68_{\pm 0.05}$	17.19
w/ DSKD (ours)	$25.23_{\pm 0.25}$	$11.19_{\pm 0.22}$	$15.91_{\pm 0.45}$	$17.92_{\pm 0.16}$	$21.20_{\pm 0.12}$	18.29 (+1.10↑)
AKL (Wu et al., 2024)	$24.75_{\pm 0.60}$	$10.46_{\pm 0.24}$	$15.37_{\pm 0.41}$	$17.48_{\pm 0.17}$	$20.11_{\pm 0.05}$	17.63
w/ DSKD (ours)	$25.13_{\pm 0.14}$	$10.63_{\pm 0.43}$	$16.18_{\pm 0.35}$	$18.58_{\pm 0.48}$	$21.45_{\pm 0.16}$	18.39 (+0.76↑)
Qwen1.5-1.8B → GPT2-120M (Different Vocabularies)						
Teacher	$27.42_{\pm 0.33}$	$19.42_{\pm 0.11}$	$19.31_{\pm 0.21}$	$34.87_{\pm 0.30}$	$36.00_{\pm 0.10}$	27.40
SeqKD	$23.40_{\pm 0.21}$	$9.36_{\pm 0.38}$	$15.37_{\pm 0.35}$	$15.16_{\pm 0.17}$	$17.34_{\pm 0.11}$	16.13
MinED (Wan et al., 2024)	$24.41_{\pm 0.61}$	$10.60_{\pm 0.39}$	$15.86_{\pm 0.42}$	$16.76_{\pm 0.28}$	$19.68_{\pm 0.12}$	17.46
ULD (Boizard et al., 2024)	$23.77_{\pm 0.41}$	$9.67_{\pm 0.50}$	$14.99_{\pm 0.55}$	$17.60_{\pm 0.21}$	$19.49_{\pm 0.12}$	17.11
DSKD-CMA-SRKL (ours)	$25.23_{\pm 0.17}$	$10.99_{\pm 0.26}$	$15.56_{\pm 0.41}$	$17.76_{\pm 0.23}$	$20.54_{\pm 0.07}$	18.02

Table 1: Rouge-L scores (%) on several benchmarks with GPT2-120M as the student. We list the mean values and the standard deviations among 5 random seeds. The average scores (**Avg.**) on all benchmarks are also listed. "*w/* DSKD" denotes our DSKD using the corresponding distance function as $\mathcal{D}(\cdot||\cdot)$ in Eqn. (6). And "DSKD-CMA-SRKL" denotes our DSKD framework equipped with cross-model attention with SRKL as $\mathcal{D}(\cdot||\cdot)$ in Eqn. (6).

- **JS.** Jenson-Shannon (JS) divergence, a symmetric variant of KL divergence.
- **SKL.** The skewed KL proposed by Ko et al. (2024), which skews the student distribution q_θ in KL as $\lambda p + (1-\lambda)q_\theta$.
- **SRKL.** The skewed RKL proposed by Ko et al. (2024), which skews the teacher distribution p in RKL as $\lambda q_\theta + (1-\lambda)p$.
- **AKL.** The adaptive fusion of KL and RKL proposed by Wu et al. (2024).

KD with different vocabularies. We also compare DSKD with cross-model attention to the KD methods for different vocabularies:

- **MinCE.** The method proposed by Wan et al. (2024), aligns the logits between different models via dynamic programming that minimizes the edit distances of token strings.
- **ULD.** The method proposed by Boizard et al. (2024), replaces the usual KL divergence with a closed-form solution of Wasserstein distance to overcome the limitation on the same tokenizers between the teacher and the student.

Besides, we also compare our framework with the black-box KD method, *i.e.*, sequence-level KD (**SeqKD**; Kim and Rush, 2016), under both settings. Nevertheless, we did not compare our framework with on-policy KD methods such as ImitKD (Lin et al., 2020), GKD (Agarwal et al., 2024), MiniLLM (Gu et al., 2023) and DistiLLM (Ko et al., 2024) since we only focus on the more general off-policy scenarios.

4.3 Results

KD with the same vocabulary. The results of KD for models with the same vocabulary are presented at the top parts of Table 1 and Table 2. Firstly, it is shown that all white-box KD methods exhibit better performance than the black-box KD method SeqKD, which demonstrates that token-level distributions can transfer more knowledge than single target tokens. Furthermore, our DSKD framework significantly outperforms the current white-box KD framework for both GPT2 and TinyLLaMA on various distance functions. On the one hand, it showcases the effectiveness of our DSKD framework that conducts KD in unified output spaces. On the other hand, the improvements on all distance functions also demonstrate that our

Methods	Dolly	SelfInst	VicunaEval	S-NI	UnNI	Avg.
SFT	23.20$_{\pm 0.13}$	14.88$_{\pm 0.54}$	16.42$_{\pm 0.35}$	27.79$_{\pm 0.27}$	26.12$_{\pm 0.11}$	21.68
LLaMA2-7B → TinyLLaMA-1.1B (Same Vocabulary)						
Teacher	28.32$_{\pm 0.46}$	20.95$_{\pm 0.69}$	18.76$_{\pm 0.35}$	32.05$_{\pm 0.28}$	32.41$_{\pm 0.12}$	26.50
SeqKD	23.21$_{\pm 0.22}$	16.46$_{\pm 0.72}$	16.58$_{\pm 0.38}$	26.33$_{\pm 0.26}$	27.69$_{\pm 0.10}$	22.05
KL	25.46$_{\pm 0.63}$	17.21$_{\pm 0.25}$	16.43$_{\pm 0.53}$	29.27$_{\pm 0.29}$	29.28$_{\pm 0.09}$	23.53
w/ DSKD (ours)	26.31$_{\pm 0.26}$	18.27$_{\pm 0.56}$	18.04$_{\pm 0.37}$	31.43$_{\pm 0.26}$	31.20$_{\pm 0.09}$	25.05 (+1.52↑)
RKL	24.49$_{\pm 0.41}$	17.14$_{\pm 0.61}$	16.87$_{\pm 0.26}$	29.50$_{\pm 0.28}$	29.36$_{\pm 0.08}$	23.47
w/ DSKD (ours)	26.93$_{\pm 0.34}$	18.14$_{\pm 0.54}$	18.81$_{\pm 0.39}$	31.79$_{\pm 0.31}$	32.49$_{\pm 0.11}$	25.63 (+2.17↑)
JS	24.03$_{\pm 0.31}$	15.75$_{\pm 0.51}$	16.64$_{\pm 0.30}$	28.08$_{\pm 0.10}$	28.68$_{\pm 0.08}$	22.62
w/ DSKD (ours)	24.79$_{\pm 0.42}$	17.10$_{\pm 0.47}$	16.78$_{\pm 0.20}$	29.06$_{\pm 0.18}$	29.47$_{\pm 0.22}$	23.44 (+0.82↑)
SKL (Ko et al., 2024)	24.14$_{\pm 0.53}$	15.98$_{\pm 0.72}$	16.89$_{\pm 0.22}$	29.30$_{\pm 0.18}$	28.71$_{\pm 0.12}$	23.01
w/ DSKD (ours)	25.88$_{\pm 0.22}$	17.59$_{\pm 0.56}$	17.17$_{\pm 0.34}$	29.52$_{\pm 0.33}$	30.69$_{\pm 0.16}$	24.17 (+1.16↑)
SRKL (Ko et al., 2024)	24.28$_{\pm 0.58}$	16.91$_{\pm 0.67}$	16.88$_{\pm 0.20}$	29.55$_{\pm 0.19}$	28.64$_{\pm 0.21}$	23.25
w/ DSKD (ours)	25.44$_{\pm 0.22}$	17.34$_{\pm 0.69}$	17.19$_{\pm 0.34}$	30.29$_{\pm 0.29}$	31.23$_{\pm 0.13}$	24.30 (+1.05↑)
AKL (Wu et al., 2024)	24.80$_{\pm 0.70}$	16.79$_{\pm 1.09}$	16.80$_{\pm 0.44}$	29.29$_{\pm 0.35}$	28.81$_{\pm 0.09}$	23.30
w/ DSKD (ours)	26.33$_{\pm 0.45}$	20.17$_{\pm 0.46}$	17.43$_{\pm 0.48}$	34.93$_{\pm 0.39}$	34.40$_{\pm 0.20}$	26.65 (+3.35↑)
Mistral-7B → TinyLLaMA-1.1B (Different Vocabularies)						
Teacher	31.56$_{\pm 0.19}$	25.10$_{\pm 0.36}$	20.50$_{\pm 0.32}$	36.07$_{\pm 0.24}$	36.27$_{\pm 0.15}$	29.90
SeqKD	23.56$_{\pm 0.39}$	15.87$_{\pm 0.54}$	15.99$_{\pm 0.55}$	25.50$_{\pm 0.37}$	26.64$_{\pm 0.09}$	21.51
MinED (Wan et al., 2024)	20.96$_{\pm 0.51}$	14.49$_{\pm 0.35}$	15.98$_{\pm 0.45}$	27.21$_{\pm 0.13}$	26.47$_{\pm 0.11}$	21.77
ULD (Boizard et al., 2024)	22.80$_{\pm 0.28}$	15.93$_{\pm 0.74}$	16.43$_{\pm 0.60}$	26.94$_{\pm 0.28}$	24.83$_{\pm 0.13}$	20.64
DSKD-CMA-AKL (ours)	26.45$_{\pm 0.56}$	19.57$_{\pm 0.69}$	17.95$_{\pm 0.55}$	35.99$_{\pm 0.19}$	35.00$_{\pm 0.16}$	26.99

Table 2: Rouge-L scores (%) on several benchmarks with TinyLLaMA-1.1B as the student. We list the mean values and the standard deviations among 5 random seeds. "*w/* DSKD" denotes our DSKD using the corresponding distance function as $\mathcal{D}(\cdot||\cdot)$ in Eqn. (6). And "DSKD-CMA-AKL" denotes our DSKD framework equipped with cross-model attention with AKL as $\mathcal{D}(\cdot||\cdot)$ in Eqn. (6).

framework is highly compatible with current distance functions in KD.

Objective	Diff. Space	Student Space	DSKD
GPT2-1.5B → GPT2-120M			
KL	17.63	18.00	18.69
RKL	17.82	18.03	18.44
JS	16.98	17.17	18.37
SKL	17.71	17.99	18.14
SRKL	17.19	17.47	18.29
AKL	17.63	17.77	18.39
LLaMA2-7B → TinyLLaMA-1.1B			
KL	23.53	24.99	25.05
RKL	23.47	25.50	25.63
JS	22.62	22.64	23.44
SKL	23.01	23.55	24.17
SRKL	23.25	23.64	24.30
AKL	23.30	26.23	26.65

Table 3: The averaged Rouge-L (%) among all test sets. The detailed scores on each test set are in Appendix C.

KD with different vocabularies. At the bottom parts of Table 1 and Table 2, we also show the results of KD methods for models with different vocabularies[6]. As mentioned in §2.2.2, the key challenge in this setting is to deal with the mismatch distributions due to different vocabulary sizes and tokenization. Facing this challenge, existing KD methods only pre-define coarse alignment and thus yield limited performance, lagging behind KD methods for models with the same vocabulary. In contrast, our CMA mechanism learns the alignment automatically, with which our DSKD performs better than existing methods. Particularly, as the teacher models under this setting are stronger, DSKD-CMA can sometimes achieve better performance than DSKD with the same vocabulary (*e.g.*, DSKD-CMA-AKL in Table 2). It suggests the potential of our method to train better students with stronger teachers, even if they have different vocabularies.

5 Analysis

5.1 KD in Different Spaces *vs.* Unified Space

In this section, we further evaluate whether unifying the space for KD leads to better performance. Specifically, we only keep the KD process in the

[6] In this setting, we only list the results of our method with the best performing distance functions due to space limitation. The full results are listed in Table 5 and Table 6.

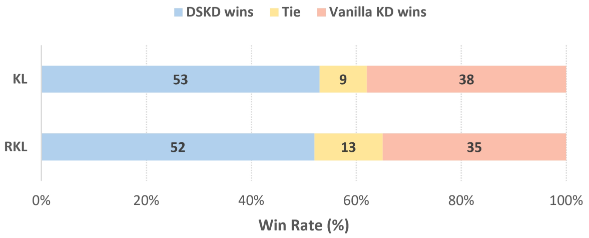
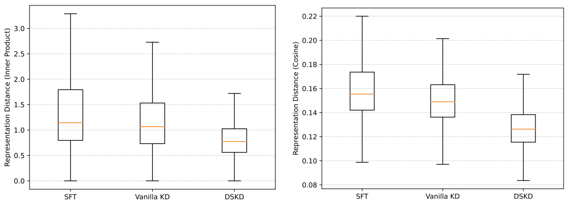


Figure 2: Win rates (%) on the response quality between TinyLLaMA trained by DSKD and the current white-box KD framework.



(a) Cosine as Structure (b) Inner Product as Structure

Figure 3: Distance between the representation structures of the teacher and the student.

student space in our DSKD, *i.e.*, only calculate the losses in Eqn. (5) and Eqn. (6), since it optimizes the same student distribution q_θ as the current KD framework does in Eqn. (2). The only difference is that the teacher distribution $p^{t\rightarrow s}$ in Eqn. (6) shares the same output space with the student distribution. The results are shown in Table 3. For all distance functions, KD in the student space (**Student Space**) consistently surpasses KD in different spaces (**Diff. Space**). These results sufficiently reflect the superiority of unifying the output spaces of the distributions for KD. Furthermore, when combined with KD in the teacher space, KD in dual spaces, *i.e.*, DSKD, achieves further improvement, indicating that KD in the student space and the teacher space can complement each other.

5.2 Evaluation via GPT-4

We also use GPT-4 to evaluate and compare our DSKD and the current white-box KD framework. Specifically, we randomly pick 100 instructions in the test set of Dolly and generate responses with TinyLLaMA trained by DSKD and the current framework. Then we use GPT-4 to judge which responses are better and plot the win rates in Figure 2. It is shown that our DSKD can beat the current KD framework in most cases for both KL divergence and reverse KL divergence. More details and the complete results for other distance functions can be referred to in Appendix D.

5.3 Representation Similarity between the Teacher and the Student

In the simulation experiment, we find that the current KD framework will lead to limited representation similarities between the student and the teacher (as shown in Figure 1(b)). Thus, we evaluate whether this phenomenon also holds in the real KD scenario. Since the dimensions are usually different for the teacher and student models, we measure the similarity of representation structures of the two models instead of their hidden states. Specifically, we use cosine similarity and normalized inner product between output hidden states to represent the representation structure of a model (see Eqn. (16) and (17) in Appendix E for the definitions). Then we calculate the L1 distance between the representation structures to reflect their similarity, where lower distance denotes higher similarity between representation structures (see Eqn. (18) and (19) in Appendix E for the detailed calculations). The average distances between the structure of the teacher and the student on 1000 training samples are plotted in Figure 3. It shows that on both types of representation structures, the current KD framework (**Vanilla KD**) only reduces minor distances between the teacher and the student compared to fine-tuning without KD (**SFT**). However, our DSKD achieves significantly lower distances between the teacher and the student, which indicates that DSKD can enhance the similarity between the student and the teacher.

6 Related Work

White-Box KD for Language Models. The white-box KD framework for language models stems from the standard KD method proposed by Hinton et al. (2015). As pre-trained language models (PLMs) become prevalent for various NLP tasks, numerous KD methods within this framework were proposed to compress the excessive model sizes of PLMs (Sun et al., 2019; Sanh et al., 2019; Sun et al., 2020; Jiao et al., 2020). Besides minimizing the distance between distributions, there are also feature-based KD methods that distill the knowledge in intermediate hidden states and attention maps of the teacher model (Jiao et al., 2020; Wang et al., 2020, 2021b). Additionally, white-box KD is also widely used in text generation tasks, such as neural machine translation

(Tan et al., 2019; Wang et al., 2021a; Zhang et al., 2023) and text summarization (Chen et al., 2020; Liu et al., 2021). Since LLMs are predominate for various tasks, several KD techniques have also been proposed for LLMs (Gu et al., 2023; Ko et al., 2024; Wu et al., 2024; Xu et al., 2024). Unlike the previous work that follows the current white-box KD framework, we challenge this framework by revealing its inherent limitations and proposing a simple yet more effective and general KD framework as the solution.

KD with the Shared Prediction Head. In the previous literature on KD, SimKD (Chen et al., 2022) also proposed to share the teacher's prediction head for KD, which was similar to the process of KD in the teacher space in our DSKD. However, the aim of SimKD is to equip the prediction head of the teacher model to the student model, and thus the student model will be larger after KD and suffer from higher inference costs. In contrast, our DSKD only leverages this process to transfer the representation information from the teacher and has no influence on the original model size of the student.

7 Conclusion

In this work, we first reveal two limitations in the current white-box KD framework for LLMs, *i.e.*, leading to low similarity between the student and the teacher and the requirements of the same vocabulary between two LLMs. To address them, we propose a novel white-box KD framework, named dual-space knowledge distillation (DSKD), which unifies the output spaces of the student and the teacher for KD. On this basis, we further develop a cross-model attention mechanism to solve the vocabulary mismatch between different LLMs, so that our DSKD framework supports KD between any two LLMs, regardless of their vocabularies. Experimental results on several instruction-following benchmarks showcase that our framework significantly outperforms the current white-box KD framework on various distance functions. Meanwhile, for LLMs with different vocabularies, DSKD also surpasses all existing KD methods.

Limitations

Although our DSKD supports KD between LLMs with different vocabularies via the cross-model attention mechanism, the final performance of DSKD-CMA in most cases still lags slightly behind the performance of DSKD when LLMs have the same vocabularies (see Table 5 and Table 6). We attribute this gap to the alignment error between the tokens in two differently tokenized sequences. Nevertheless, we still believe that our cross-model attention is a simple yet relatively effective method to solve the KD for LLMs with different vocabularies and may inspire more effective methods in future work.

Acknowledgements

The research work described in this paper has been supported by the National Nature Science Foundation of China (No. 62476023, 61976016, 62376019, 61976015), and the authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Nicolas Boizard, Kevin El-Haddad, Céline Hudelot, and Pierre Colombo. 2024. Towards cross-tokenizer distillation: the universal logit distillation loss for llms. *arXiv preprint arXiv:2402.12030*.

Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen. 2022. Knowledge distillation with the reused teacher classifier. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11933–11942.

Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. 2020. Distilling knowledge learned in BERT for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7893–7905, Online. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, march 2023. *URL https://lmsys. org/blog/2023-03-30-vicuna*, 3(5).

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul

Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning*, pages 10421–10430. PMLR.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic chain-of-thought distillation: Small models can also “think” step-by-step. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2665–2679, Toronto, Canada. Association for Computational Linguistics.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. 2020. Autoregressive knowledge distillation through imitation learning. *arXiv preprint arXiv:2009.07253*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu, Sheng Shen, and Mirella Lapata. 2021. Noisy self-knowledge distillation for text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 692–703, Online. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. *arXiv preprint arXiv:1902.10461*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge fusion of large language models. *arXiv preprint arXiv:2401.10491*.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021a. Selective knowledge distillation for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6456–6466, Online. Association for Computational Linguistics.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021b. Minilmv2: Multi-head self-attention relation distillation for compressing pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *arXiv preprint arXiv:2204.07705*, 2.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10817–10834.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Zhe Zhao, and Ngai Wong. 2024. Rethinking kullback-leibler divergence in knowledge distillation for large language models. *arXiv preprint arXiv:2404.02657*.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.

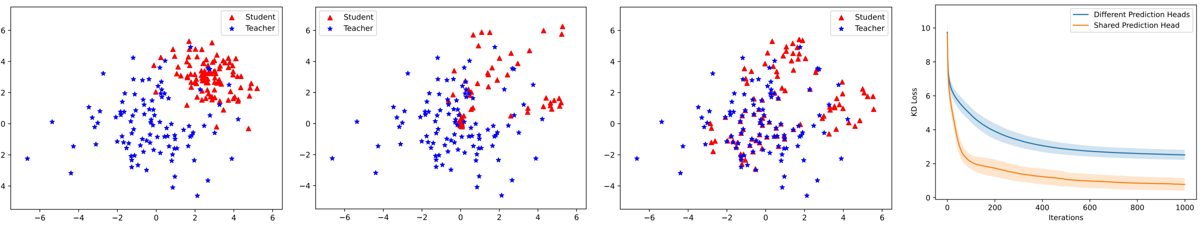
Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Songming Zhang, Yunlong Liang, Shuaibo Wang, Yufeng Chen, Wenjuan Han, Jian Liu, and Jinan Xu. 2023. Towards understanding and improving knowledge distillation for neural machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8062–8079, Toronto, Canada. Association for Computational Linguistics.

A Appendix

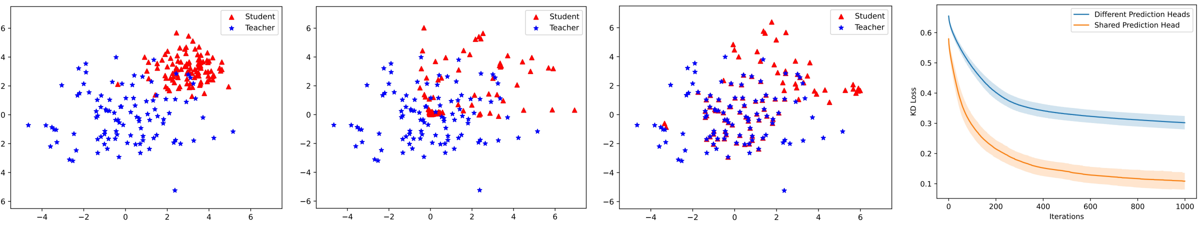
A.1 Simulation Results for Other Distance Functions

We complement the remaining results of simulation experiments for the following objectives: reverse KL divergence, JS divergence, skewed KL divergence, skewed RKL divergence, and adaptive KL divergence. The results are plotted in Figure 4, Figure 5, Figure 6, Figure 7 and Figure 8. It is shown that no matter which distance function is used, the student after KD will have low representation similarity with the teacher and leave large margin to the minimum distance between the two distributions when using different prediction heads. Thus, all these results lead to the consistent conclusion in §2.2.1, and also suggest that current KD framework may have inherent flaws on enhancing the similarity between the student model and the teacher model. As a solution, unifying the output spaces by sharing the prediction head for teacher and student may achieve more effective KD process.



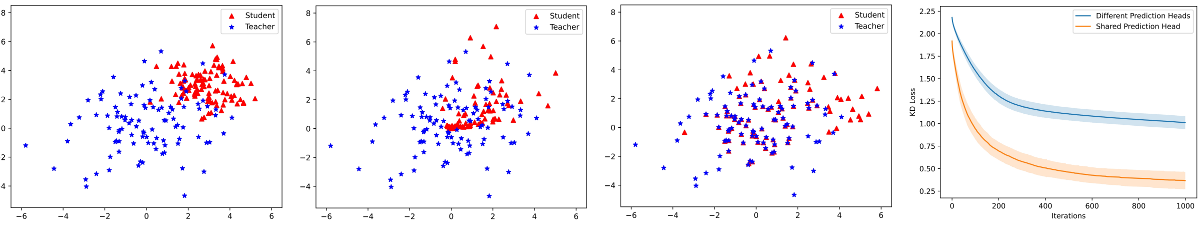
(a) Before KD (b) After KD (different heads) (c) After KD (shared head) (d) Loss curves of KD

Figure 4: Simulation results with reverse KL divergence as the KD objective. (a), (b) and (c) plot the student's hidden states and the teacher's hidden states before and after the two KD processes. (d) shows the convergence curves of the KD objective in the two KD processes.



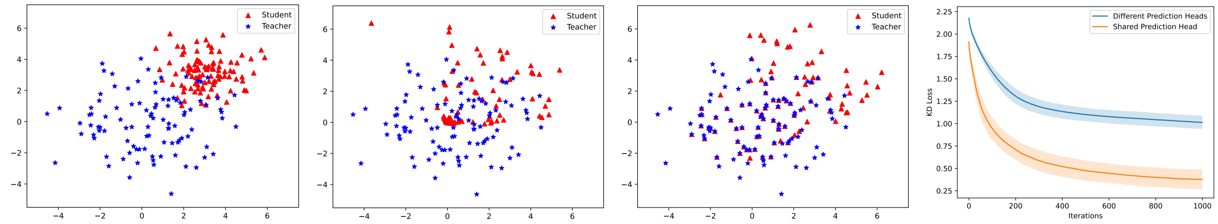
(a) Before KD (b) After KD (different heads) (c) After KD (shared head) (d) Loss curves of KD

Figure 5: Simulation results with JS divergence as the KD objective. (a), (b) and (c) plot the student's hidden states and the teacher's hidden states before and after the two KD processes. (d) shows the convergence curves of the KD objective in the two KD processes.



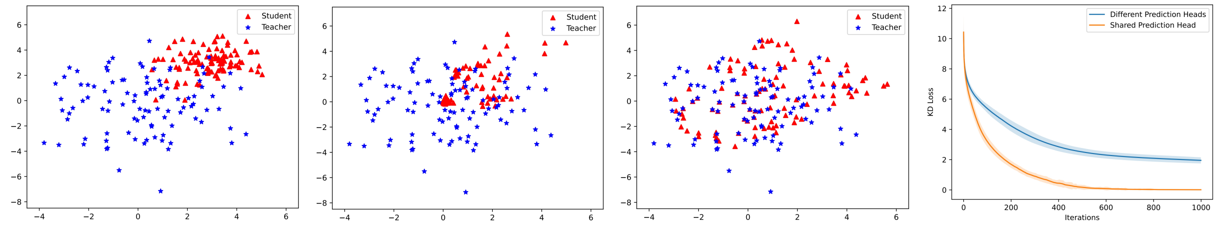
(a) Before KD (b) After KD (different heads) (c) After KD (shared head) (d) Loss curves of KD

Figure 6: Simulation results with skewed KL divergence as the KD objective. (a), (b) and (c) plot the student's hidden states and the teacher's hidden states before and after the two KD processes. (d) shows the convergence curves of the KD objective in the two KD processes.



(a) Before KD (b) After KD (different heads) (c) After KD (shared head) (d) Loss curves of KD

Figure 7: Simulation results with skewed reverse KL divergence as the KD objective. (a), (b) and (c) plot the student's hidden states and the teacher's hidden states before and after the two KD processes. (d) shows the convergence curves of the KD objective in the two KD processes.



(a) Before KD (b) After KD (different heads) (c) After KD (shared head) (d) Loss curves of KD

Figure 8: Simulation results with adaptive KL divergence as the KD objective. (a), (b) and (c) plot the student's hidden states and the teacher's hidden states before and after the two KD processes. (d) shows the convergence curves of the KD objective in the two KD processes.

A.2 Pseudo Code for Simulation Experiments

We also provide the pseudo code for re-implementing the key parts of our simulation experiments:

```
class Teacher(nn.Module):
    def __init__(self):
        super(Teacher, self).__init__()
        # the initial teacher hiddens are sampled from Gaussian Distribution N(0, 2)
        self.hidden = torch.randn(100, 2) * 2
        # the head contains 10000 classes
        self.head = torch.randn(10000, 2)

class Student(nn.Module):
    def __init__(self):
        super(Student, self).__init__()
        # the initial student hiddens are sampled from Gaussian Distribution N(3, 1)
        self.hidden = nn.Parameter(torch.randn(100, 2) + 3)
        # the head contains 10000 classes
        self.head = nn.Parameter(torch.randn(10000, 2))

def kd_with_different_head(student, teacher):
    student_logits = student.hidden.matmul(student.head.transpose(-1, -2))
    # calculating logits with the respective heads
    teacher_logits = teacher.hidden.matmul(teacher.head.transpose(-1, -2))
    kd_loss = distance_func(student_logits, teacher_logits)
    return kd_loss

def kd_with_shared_head(student, teacher):
    student_logits = student.hidden.matmul(student.head.transpose(-1, -2))
    # calculating logits with the same head (student's head)
    teacher_logits = teacher.hidden.matmul(student.head.transpose(-1, -2))
    kd_loss = distance_func(student_logits, teacher_logits)
    return kd_loss
```

As shown in the code, we manually separate the hidden states of the student and teacher in initialization, so that the difference before and after KD will be more clear. Besides, to unify the output spaces of the two models, we share the prediction head of the student with the teacher in "`kd_with_shared_head`". In this way, the output distributions of the student being optimized are as same as the ones in "`kd_with_different_head`" and thus the results will be more comparable with the ones in "`kd_with_different_head`". The student models are optimized by the SGD optimizer with appropriate learning rates in $[1.0, 40.0]$ for different distance functions.

B Experimental Details

B.1 Data

All the test sets in our experiments are processed by (Gu et al., 2023). For all these test sets, Dolly contains 500 samples, Self-Instruction (Wang et al., 2023) contains 242 samples, Vicuna-Evaluation (Chiang et al., 2023) contains 80 samples, Super-Natural Instructions (Wang et al., 2022) contains 1694 samples with response lengths in $[11, +\infty]$, and Unnatural Instructions (Honovich et al., 2023) contains 10000 samples with response lengths in $[11, +\infty]$.

B.2 Training

For GPT2-1.5B, we directly use the checkpoint released by Gu et al. (2023). For other models, the detailed training configurations are listed in Table 4. Note that we do not use the pre-training corpus while distillation as (Gu et al., 2023) did for simplicity. Each training requires several hours on 4×RTX 3090 or 8×RTX A4000.

Settings	**KD for GPT2**		**KD for TinyLLaMA**		
	GPT2	Qwen1.5	TinyLLaMA	LLaMA2	Mistral
Epoch	20	10	10	10	10
Learning Rate	5e-4	2e-5	1e-3	1e-3	1e-3
Projector Learning Rate	1e-3	1e-3	1e-3	1e-3	1e-3
Batch Size	32	32	32	32	32
LR Scheduler	Cosine	Cosine	Cosine	Cosine	Cosine
Fine-Tuning Method	Full	Full	LoRA	LoRA	LoRA
Lora Rank	N/A	N/A	256	256	256
Lora Alpha	N/A	N/A	8	8	8
Lora Dropout	N/A	N/A	0.1	0.1	0.1

Table 4: Detailed training configurations of KD for GPT2 and TinyLLaMA.

Besides, we combine the original cross-entropy loss on the target tokens in Eqn. (1) and the KD loss in Eqn. (2) and Eqn. (10) as the overall training loss for all the white-box KD methods in our main experiments:

$$\mathcal{L} = 0.5 * \mathcal{L}_{ce} + 0.5 * \mathcal{L}_{(ds)kd}. \tag{15}$$

B.3 Evaluation

For the evaluation, we use random sampling to decode the responses from all models. For decoding, we set both the decoding temperature and top_p to 1.0. Then, we generate the responses with random seeds in [10, 20, 30, 40, 50] and report the averaged Rouge-L scores of each seed following Gu et al. (2023).

B.4 Effect of Temperature for KD

As an important hyper-parameter in KD, the temperature coefficient τ significantly affects the final performance of KD. As stated by the previous literature, a larger temperature (>1.0) will smooth the teacher's distribution and transfer more class relationship information to the student model. Thus, we search for the best temperatures among [1.0, 1.5, 2.0, 3.0, 4.0] for two representative objectives (*i.e.*, KL divergence and reverse KL divergence) on the validation set and report the results in Figure 9. The results show that both objectives perform best when the temperature is 2.0. Thus, we keep the temperature to 2.0 for all objectives in our experiments.

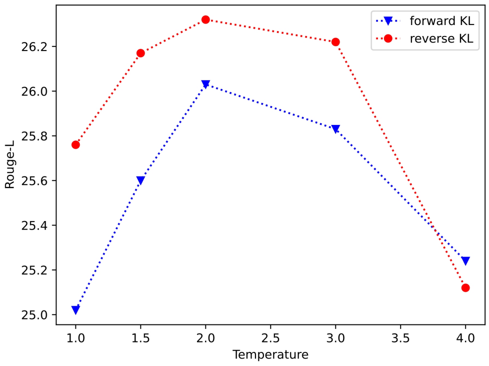


Figure 9: Rouge-L scores (%) on the validation set for different temperature coefficients in KL divergence and reverse KL divergence.

C Full Results

We provide the full results of our main experiments in Table 5 and Table 6. For KD between LLMs with the same vocabulary, we complement the detailed results of all distance functions in both the student and the teacher space. For KD between LLMs with different vocabularies, we also present the full results of our DSKD with CMA for all the distance functions.

As shown in Table 5 and Table 6, KD in the student space yields better performance than vanilla KD (in the different spaces) on all distance functions. However, KD in the teacher space only leads to limited improvement for some distance functions. The reason is that the student distribution $q_\theta^{s\to t}$ optimized by KD in the teacher space is different from the original student distribution q_θ, and thus the KD process has no direct influence on q_θ. Nevertheless, we found that KL divergence has relatively good performance for KD in the teacher space. Therefore, we directly choose KL divergence as the distance function for KD in the teacher space in our DSKD.

D Details and Full Results for GPT-4 Evaluation

We use the API of `gpt4-turbo-0409` to evaluate the quality of the responses. As we conduct pairwise comparison between the responses from two models, to alleviate the order bias in the evaluation process of GPT-4, we randomly shuffle the two responses as the Response A/B in the system prompts.

Please act as an impartial judge and compare the quality of response A and response B provided by two AI assistants to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response.
Just tell me which response do you think is better:
- If A is significantly better than B, just answer me "A";
- If B is significantly better than A, just answer me "B";
- If A and B have similar quality (both good or both wrong), just answer me "Tied".

[Question]
{question or instruction}

[Response A]
{response A}

[Response B]
{response B}

Figure 10: Prompt for GPT-4 Evaluation.

The full results for GPT-4 Evaluation on all distance functions are shown in Figure 11. For all distance functions, the students trained by our DSKD always win more than the student trained by the current white-box KD framework, indicating the consistent superiority of our DSKD framework on existing distance functions.

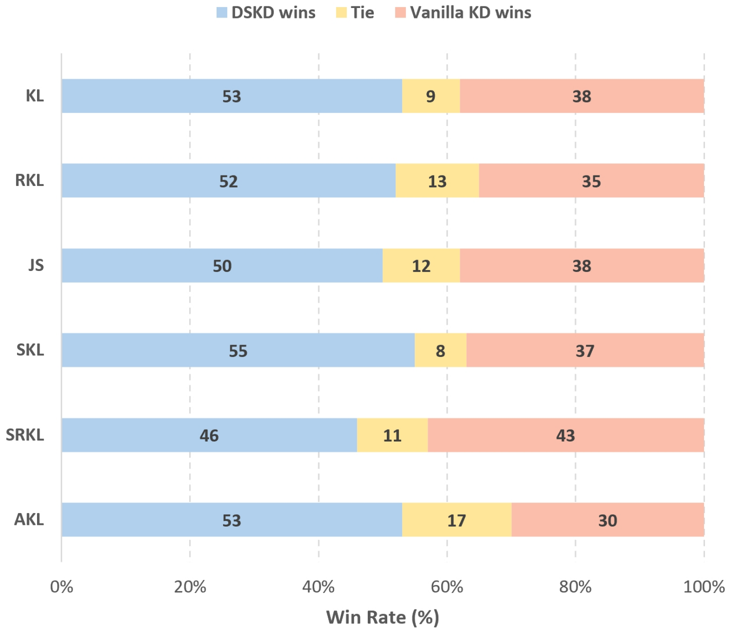


Figure 11: GPT-4 Evaluation Results for all the distance functions.

E Details of the Distance between Representation Structure

Since the student models and the teacher models generally have different dimensions on representations, it is difficult to directly measure the representation similarity between the student and the teacher. Thus, we calculate the similarity on the structure of sentences in their own representation spaces of the student and the teacher. Specifically, given a sentence with n tokens, we calculate structure matrices with both the cosine similarity and normalized inner-product values between the output hidden states of this sentence:

$$\mathcal{M}_{cosine}(i,j) = \frac{{h_i}^\top h_j}{|h_i||h_j|} \in \mathbb{R}^{n \times n}, \tag{16}$$

$$\mathcal{M}_{prod}(i,j) = \frac{{h_i}^\top h_j}{\sum_k {h_i}^\top h_k} \in \mathbb{R}^{n \times n}, \tag{17}$$

where $\mathcal{M}_{cosine}$ and $\mathcal{M}_{prod}$ are structure matrices calculated by cosine and normalized inner-product between output hidden states, respectively. Then we calculate the L1 distance between the matrices of the student and the teacher:

$$\mathcal{D}_{cosine} = \sum_i^n \sum_j^n |\mathcal{M}^t_{cosine}(i,j) - \mathcal{M}^s_{cosine}(i,j)|, \tag{18}$$

$$\mathcal{D}_{prod} = \sum_i^n \sum_j^n |\mathcal{M}^t_{prod}(i,j) - \mathcal{M}^s_{prod}(i,j)|. \tag{19}$$

The smaller distance values means the representations of the student and the teacher are more similar. In Figure 3, we calculate and average the two distances $\mathcal{D}_{cosine}$ and $\mathcal{D}_{prod}$ on 1000 samples in the training set for GPT2 models that trained without KD (SFT), trained by the current white-box KD framework (Vanilla KD) and trained by our DSKD framework (DSKD).

Methods	**Dolly**	**SelfInst**	**VicunaEval**	**S-NI**	**UnNI**	**Avg.**
SFT	$22.94_{\pm0.28}$	$10.11_{\pm0.36}$	$15.17_{\pm0.63}$	$16.21_{\pm0.19}$	$18.68_{\pm0.09}$	16.62
GPT2-1.5B → GPT2-120M (Same Vocabulary)						
Teacher	$27.19_{\pm0.23}$	$14.64_{\pm0.64}$	$16.30_{\pm0.37}$	$27.55_{\pm0.30}$	$31.46_{\pm0.12}$	23.43
SeqKD	$23.68_{\pm0.25}$	$10.03_{\pm0.23}$	$14.41_{\pm0.46}$	$16.36_{\pm0.18}$	$18.48_{\pm0.11}$	16.59
KL	$24.54_{\pm0.48}$	$10.43_{\pm0.24}$	$15.66_{\pm0.42}$	$17.24_{\pm0.27}$	$20.28_{\pm0.18}$	17.63
KL in Student Space	$23.83_{\pm0.30}$	$10.46_{\pm0.36}$	$15.79_{\pm0.51}$	$18.82_{\pm0.31}$	$21.08_{\pm0.07}$	18.00
KL in Teacher Space	$24.07_{\pm0.67}$	$10.34_{\pm0.38}$	$14.94_{\pm0.24}$	$18.83_{\pm0.25}$	$21.02_{\pm0.11}$	17.84
KL in Student Space + KL in Teacher Space	$24.70_{\pm0.24}$	$10.65_{\pm0.30}$	$15.67_{\pm0.30}$	$19.51_{\pm0.21}$	$22.94_{\pm0.07}$	18.69
RKL	$24.38_{\pm0.55}$	$10.73_{\pm0.61}$	$15.71_{\pm0.39}$	$17.31_{\pm0.11}$	$20.96_{\pm0.12}$	17.82
RKL in Student Space	$25.12_{\pm0.25}$	$10.60_{\pm0.27}$	$15.25_{\pm0.26}$	$17.96_{\pm0.24}$	$21.19_{\pm0.09}$	18.03
RKL in Teacher Space	$23.54_{\pm0.33}$	$10.48_{\pm0.55}$	$15.21_{\pm0.52}$	$16.59_{\pm0.18}$	$19.49_{\pm0.16}$	17.06
RKL in Student Space + KL in Teacher Space	$24.61_{\pm0.59}$	$11.01_{\pm0.45}$	$14.98_{\pm0.48}$	$19.32_{\pm0.28}$	$22.27_{\pm0.13}$	18.44
JS	$23.86_{\pm0.14}$	$10.20_{\pm0.40}$	$15.50_{\pm0.23}$	$16.20_{\pm0.23}$	$19.17_{\pm0.06}$	16.98
JS in Student Space	$24.46_{\pm0.34}$	$10.02_{\pm0.24}$	$15.59_{\pm0.46}$	$16.53_{\pm0.19}$	$19.25_{\pm0.14}$	17.17
JS in Teacher Space	$23.28_{\pm0.52}$	$9.76_{\pm0.37}$	$15.08_{\pm0.26}$	$15.89_{\pm0.20}$	$18.34_{\pm0.12}$	16.47
JS in Student Space + KL in Teacher Space	$24.61_{\pm0.27}$	$11.41_{\pm0.35}$	$15.40_{\pm0.28}$	$18.94_{\pm0.20}$	$21.48_{\pm0.17}$	18.37
SKL (Ko et al., 2024)	$24.03_{\pm0.23}$	$10.66_{\pm0.51}$	$14.70_{\pm0.37}$	$17.99_{\pm0.15}$	$21.18_{\pm0.16}$	17.71
SKL in Student Space	$24.06_{\pm0.38}$	$11.03_{\pm0.18}$	$15.11_{\pm0.44}$	$18.67_{\pm0.27}$	$21.13_{\pm0.05}$	18.00
SKL in Teacher Space	$23.44_{\pm0.25}$	$10.06_{\pm0.43}$	$14.86_{\pm0.51}$	$16.52_{\pm0.21}$	$19.60_{\pm0.15}$	16.90
SKL in Student Space + KL in Teacher Space	$25.24_{\pm0.28}$	$10.50_{\pm0.13}$	$15.76_{\pm0.43}$	$18.34_{\pm0.44}$	$20.87_{\pm0.11}$	18.14
SRKL (Ko et al., 2024)	$24.48_{\pm0.19}$	$10.35_{\pm0.38}$	$14.88_{\pm0.24}$	$16.53_{\pm0.23}$	$19.68_{\pm0.05}$	17.19
SRKL in Student Space	$24.84_{\pm0.08}$	$10.50_{\pm0.59}$	$15.16_{\pm0.30}$	$16.80_{\pm0.26}$	$20.04_{\pm0.05}$	17.47
SRKL in Teacher Space	$23.10_{\pm0.39}$	$10.00_{\pm0.42}$	$14.83_{\pm0.39}$	$16.07_{\pm0.34}$	$18.45_{\pm0.17}$	16.49
SRKL in Student Space + KL in Teacher Space	$25.23_{\pm0.25}$	$11.19_{\pm0.22}$	$15.91_{\pm0.45}$	$17.92_{\pm0.16}$	$21.20_{\pm0.12}$	18.29
AKL (Wu et al., 2024)	$24.75_{\pm0.60}$	$10.46_{\pm0.24}$	$15.37_{\pm0.41}$	$17.48_{\pm0.17}$	$20.11_{\pm0.05}$	17.63
AKL in Student Space	$25.08_{\pm0.36}$	$10.70_{\pm0.15}$	$14.56_{\pm0.74}$	$17.80_{\pm0.20}$	$20.72_{\pm0.11}$	17.77
AKL in Teacher Space	$23.82_{\pm0.60}$	$10.10_{\pm0.59}$	$15.40_{\pm0.16}$	$17.04_{\pm0.16}$	$20.13_{\pm0.09}$	17.30
AKL in Student Space + KL in Teacher Space	$25.13_{\pm0.14}$	$10.63_{\pm0.43}$	$16.18_{\pm0.35}$	$18.58_{\pm0.48}$	$21.45_{\pm0.16}$	18.39
Qwen1.5-1.8B → GPT2-120M (Different Vocabulary)						
Teacher	$27.19_{\pm0.23}$	$14.64_{\pm0.64}$	$16.30_{\pm0.37}$	$27.55_{\pm0.30}$	$31.42_{\pm0.11}$	23.42
SeqKD	$23.40_{\pm0.21}$	$9.36_{\pm0.38}$	$15.37_{\pm0.35}$	$15.16_{\pm0.17}$	$17.34_{\pm0.11}$	16.13
MinED (Wan et al., 2024)	$24.41_{\pm0.61}$	$10.60_{\pm0.39}$	$15.86_{\pm0.42}$	$16.76_{\pm0.28}$	$19.68_{\pm0.12}$	17.46
ULD (Boizard et al., 2024)	$23.77_{\pm0.41}$	$9.67_{\pm0.50}$	$14.99_{\pm0.55}$	$17.60_{\pm0.21}$	$19.49_{\pm0.12}$	17.11
DSKD-CMA-KL (ours)	$24.73_{\pm0.47}$	$11.15_{\pm0.34}$	$15.31_{\pm0.38}$	$17.20_{\pm0.24}$	$20.57_{\pm0.08}$	17.79
DSKD-CMA-RKL (ours)	$23.99_{\pm0.29}$	$10.89_{\pm0.46}$	$15.15_{\pm0.28}$	$17.82_{\pm0.11}$	$21.05_{\pm0.13}$	17.78
DSKD-CMA-JS (ours)	$23.95_{\pm0.29}$	$10.44_{\pm0.60}$	$15.38_{\pm0.23}$	$16.69_{\pm0.14}$	$20.27_{\pm0.10}$	17.35
DSKD-CMA-SKL (ours)	$24.67_{\pm0.13}$	$10.82_{\pm0.46}$	$15.30_{\pm0.51}$	$17.95_{\pm0.28}$	$20.65_{\pm0.13}$	17.88
DSKD-CMA-SRKL (ours)	$25.23_{\pm0.17}$	$10.99_{\pm0.26}$	$15.56_{\pm0.41}$	$17.76_{\pm0.23}$	$20.54_{\pm0.07}$	18.02
DSKD-CMA-AKL (ours)	$24.72_{\pm0.33}$	$10.67_{\pm0.29}$	$15.84_{\pm0.67}$	$16.59_{\pm0.25}$	$19.78_{\pm0.10}$	17.52

Table 5: Detailed Rouge-L scores (%) of all our models on several benchmarks with GPT2-120M as the student. We present the mean values and the standard deviations among 5 random seeds. The average scores (**Avg.**) on all benchmarks are also listed. “XX in Student Space + KL in Teacher Space” represents our DSKD with XX as the distance function in Eqn. (6).

Methods	**Dolly**	**SelfInst**	**VicunaEval**	**S-NI**	**UnNI**	**Avg.**
SFT	23.20$_{\pm0.13}$	14.88$_{\pm0.54}$	16.42$_{\pm0.35}$	27.79$_{\pm0.27}$	26.12$_{\pm0.11}$	21.68
LLaMA2-7B → TinyLLaMA-1.1B (Same Vocabulary)						
Teacher	28.32$_{\pm0.46}$	20.95$_{\pm0.69}$	18.76$_{\pm0.35}$	32.05$_{\pm0.28}$	32.41$_{\pm0.12}$	26.50
SeqKD	23.21$_{\pm0.22}$	16.46$_{\pm0.72}$	16.58$_{\pm0.38}$	26.33$_{\pm0.26}$	27.69$_{\pm0.10}$	22.05
KL	25.46$_{\pm0.63}$	17.21$_{\pm0.25}$	16.43$_{\pm0.53}$	29.27$_{\pm0.29}$	29.28$_{\pm0.09}$	23.53
KL in Student Space	26.20$_{\pm0.30}$	18.69$_{\pm0.72}$	17.71$_{\pm0.43}$	32.40$_{\pm0.21}$	29.94$_{\pm0.09}$	24.99
KL in Teacher Space	22.86$_{\pm0.77}$	15.80$_{\pm0.53}$	15.90$_{\pm0.22}$	27.58$_{\pm0.29}$	28.03$_{\pm0.20}$	22.04
KL in Student Space + KL in Teacher Space	26.31$_{\pm0.26}$	18.27$_{\pm0.56}$	18.04$_{\pm0.37}$	31.43$_{\pm0.26}$	31.20$_{\pm0.09}$	25.05
RKL	24.49$_{\pm0.41}$	17.14$_{\pm0.61}$	16.87$_{\pm0.26}$	29.50$_{\pm0.28}$	29.36$_{\pm0.08}$	23.47
RKL in Student Space	26.74$_{\pm0.36}$	19.16$_{\pm0.29}$	18.85$_{\pm0.41}$	31.76$_{\pm0.42}$	31.01$_{\pm0.06}$	25.50
RKL in Teacher Space	22.60$_{\pm0.43}$	16.04$_{\pm1.15}$	15.81$_{\pm0.40}$	28.88$_{\pm0.23}$	28.86$_{\pm0.10}$	22.44
RKL in Student Space + KL in Teacher Space	26.93$_{\pm0.34}$	18.14$_{\pm0.54}$	18.81$_{\pm0.39}$	31.79$_{\pm0.31}$	32.49$_{\pm0.11}$	25.63
JS	24.03$_{\pm0.31}$	15.75$_{\pm0.51}$	16.64$_{\pm0.30}$	28.08$_{\pm0.10}$	28.68$_{\pm0.08}$	22.62
JS in Student Space	23.86$_{\pm0.26}$	17.16$_{\pm0.85}$	16.98$_{\pm0.39}$	27.61$_{\pm0.27}$	27.65$_{\pm0.08}$	22.64
JS in Teacher Space	22.74$_{\pm0.34}$	15.28$_{\pm0.74}$	16.33$_{\pm0.26}$	26.54$_{\pm0.28}$	26.07$_{\pm0.14}$	21.39
JS in Student Space + KL in Teacher Space	24.79$_{\pm0.42}$	17.10$_{\pm0.47}$	16.78$_{\pm0.20}$	29.06$_{\pm0.18}$	29.47$_{\pm0.22}$	23.44
SKL (Ko et al., 2024)	24.14$_{\pm0.53}$	15.98$_{\pm0.72}$	16.89$_{\pm0.22}$	29.30$_{\pm0.18}$	28.71$_{\pm0.12}$	23.01
SKL in Student Space	25.15$_{\pm0.24}$	17.16$_{\pm0.84}$	17.27$_{\pm0.18}$	29.19$_{\pm0.19}$	28.98$_{\pm0.20}$	23.55
SKL in Teacher Space	22.72$_{\pm0.75}$	15.88$_{\pm0.64}$	15.89$_{\pm0.41}$	28.37$_{\pm0.23}$	26.84$_{\pm0.15}$	21.94
SKL in Student Space + KL in Teacher Space	25.88$_{\pm0.22}$	17.59$_{\pm0.56}$	17.17$_{\pm0.34}$	29.52$_{\pm0.33}$	30.69$_{\pm0.16}$	24.17
SRKL (Ko et al., 2024)	24.28$_{\pm0.58}$	16.91$_{\pm0.67}$	16.88$_{\pm0.20}$	29.55$_{\pm0.19}$	28.64$_{\pm0.21}$	23.25
SRKL in Student Space	25.92$_{\pm0.39}$	16.76$_{\pm0.71}$	17.13$_{\pm0.46}$	29.69$_{\pm0.17}$	28.67$_{\pm0.04}$	23.64
SRKL in Teacher Space	22.88$_{\pm0.57}$	16.40$_{\pm0.46}$	16.24$_{\pm0.40}$	27.23$_{\pm0.37}$	27.16$_{\pm0.04}$	21.98
SRKL in Student Space + KL in Teacher Space	25.44$_{\pm0.22}$	17.34$_{\pm0.69}$	17.19$_{\pm0.34}$	30.29$_{\pm0.29}$	31.23$_{\pm0.13}$	24.30
AKL (Wu et al., 2024)	24.80$_{\pm0.70}$	16.79$_{\pm1.09}$	16.80$_{\pm0.44}$	29.29$_{\pm0.35}$	28.81$_{\pm0.09}$	23.30
AKL in Student Space	26.07$_{\pm0.51}$	19.57$_{\pm0.83}$	17.57$_{\pm0.46}$	34.50$_{\pm0.33}$	33.45$_{\pm0.15}$	26.23
AKL in Teacher Space	22.81$_{\pm0.56}$	16.33$_{\pm0.73}$	16.00$_{\pm0.14}$	27.05$_{\pm0.15}$	28.09$_{\pm0.19}$	22.05
AKL in Student Space + KL in Teacher Space	26.33$_{\pm0.45}$	20.17$_{\pm0.46}$	17.43$_{\pm0.48}$	34.93$_{\pm0.39}$	34.40$_{\pm0.20}$	26.65
Mistral-7B → TinyLLaMA-1.1B (Different Vocabularies)						
Teacher	31.56$_{\pm0.19}$	25.10$_{\pm0.36}$	20.50$_{\pm0.32}$	36.07$_{\pm0.24}$	36.27$_{\pm0.15}$	29.90
SeqKD	23.56$_{\pm0.39}$	15.87$_{\pm0.54}$	15.99$_{\pm0.55}$	25.50$_{\pm0.37}$	26.64$_{\pm0.09}$	21.51
MinED (Wan et al., 2024)	20.96$_{\pm0.51}$	14.49$_{\pm0.35}$	15.98$_{\pm0.45}$	27.21$_{\pm0.13}$	26.47$_{\pm0.11}$	21.77
ULD (Boizard et al., 2024)	22.80$_{\pm0.28}$	15.93$_{\pm0.74}$	16.43$_{\pm0.60}$	26.94$_{\pm0.28}$	24.83$_{\pm0.13}$	20.64
DSKD-CMA-KL (ours)	26.52$_{\pm0.45}$	17.90$_{\pm0.69}$	18.20$_{\pm0.59}$	30.66$_{\pm0.39}$	31.03$_{\pm0.11}$	24.86
DSKD-CMA-RKL (ours)	25.41$_{\pm0.18}$	18.31$_{\pm0.45}$	16.83$_{\pm0.46}$	34.79$_{\pm0.16}$	34.05$_{\pm0.12}$	25.88
DSKD-CMA-JS (ours)	24.09$_{\pm0.71}$	16.77$_{\pm0.75}$	16.96$_{\pm0.27}$	30.01$_{\pm0.15}$	30.00$_{\pm0.10}$	23.56
DSKD-CMA-SKL (ours)	25.28$_{\pm0.24}$	17.33$_{\pm0.62}$	17.57$_{\pm0.43}$	30.27$_{\pm0.30}$	31.14$_{\pm0.35}$	24.32
DSKD-CMA-SRKL (ours)	24.87$_{\pm0.50}$	17.63$_{\pm0.53}$	17.16$_{\pm0.24}$	29.77$_{\pm0.19}$	30.78$_{\pm0.14}$	24.04
DSKD-CMA-AKL (ours)	26.45$_{\pm0.56}$	19.57$_{\pm0.69}$	17.95$_{\pm0.55}$	35.99$_{\pm0.19}$	35.00$_{\pm0.16}$	26.99

Table 6: Rouge-L scores (%) of all models on several benchmarks with TinyLLaMA-1.1B as the student. We present the mean values and the standard deviations among 5 random seeds. The average scores (**Avg.**) on all benchmarks are also listed. “XX in Student Space + KL in Teacher Space” represents our DSKD with XX as the distance function in Eqn. (6).