# KNOWLEDGE DISTILLATION FOR LARGE LANGUAGE MODELS THROUGH RESIDUAL LEARNING

**Anonymous authors**Paper under double-blind review

000

001

002 003 004

010 011

012

013

014

015

016

017

018

019

021

023

025

026

027

028

029

031

032

034

037

040

041

042

043

044

045

046

047

048

051

052

#### **ABSTRACT**

Knowledge distillation has become a crucial technique to transfer the capacities of large language models (LLMs) to smaller, more efficient models for practical deployment. While recent work exploits rich information from intermediate states of the teacher model for more effective knowledge transfer, imperfect knowledge from the teacher can also mislead student learning, restricting the student's generalization capacity. In this work, we propose a two-stage distillation framework that is effective for diverse knowledge distillation scenarios. In the first stage, we pretrain projectors to extract and compress teacher knowledge into a lowdimensional vector space via self-reconstruction. In the second stage, we perform distillation with a hybrid objective that combines learning from the compressed teacher representations with standard supervised fine-tuning on ground-truth data. Our key innovation is residual learning for LLM distillation, where the student learns to make predictions based on the differential between its representations and projected states from the teacher. This approach encourages the student to further improve its representations beyond potentially erroneous teacher knowledge. For Mixture-of-Experts (MoE) teacher models, we further fuse the experts' outputs using a self-attention mechanism for better utilizing the teacher knowledge. Moreover, to support the cross-tokenizer distillation setting, where the teacher and student models have different vocabularies, we adopt a cross-model attention mechanism that eliminates the need for explicit token alignment rules. Experimental results show the superior performance of our proposed framework under both same- and cross-tokenizer settings, demonstrating the effectiveness in preserving teacher knowledge and improving student generalization capability.

# 1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across diverse natural language tasks, from complex reasoning and mathematical problem solving to creative writing and code generation (Chowdhery et al., 2023; Grattafiori et al., 2024; Comanici et al., 2025; Yang et al., 2025). However, deploying large models with billions of parameters requires substantial computational power, which limits their adoption in resource-constrained environments. To this end, knowledge distillation (KD) (Buciluă et al., 2006; Hinton et al., 2015; Sanh et al., 2019) has emerged as a promising paradigm to address this challenge by transferring knowledge from large teacher models to smaller, more efficient student models that could preserve most of the teacher capability.

Knowledge distillation methods can be broadly categorized into black-box and white-box approaches. Under black-box KD, the student model learns from training data generated by the teacher model through standard supervised fine-tuning (Kim & Rush, 2016; Taori et al., 2023; Chiang et al., 2023; Xu et al., 2023). Despite its simplicity and broad applicability, black-box KD fails to utilize the rich knowledge embedded in the intermediate representations of the teacher model (Gu et al., 2023; Wen et al., 2023; Ko et al., 2024). Therefore, white-box KD methods aims to further improve student model training by leveraging more information from the teacher, including logit distributions (Sanh et al., 2019; Liang et al., 2023) and internal hidden states (Sun et al., 2019; Jiao et al., 2019; Hou et al., 2020; Zuo et al., 2022; Shen et al., 2025). However, applying white-box KD becomes particularly challenging when the teacher and student models use different tokenizers, as token misalignment makes direct knowledge transfer between sequence representations infeasible. To address these challenges, recent work has explored white-box KD for cross-tokenizer scenarios,

where not only the model structure but also the vocabulary are different between the teacher and student model (Boizard et al., 2024; Zhang et al., 2025b; Chen et al., 2025; Cui et al., 2025; Minixhofer et al., 2025).

Despite their success, recent work has mainly focused on divergence-based methods that optimize output distribution matching (Xu et al., 2024) without considering scenarios where teacher predictions are incorrect. In this case, the teacher can transfer detrimental bias to student, resulting in poor performance and generalization capacity of the student model (Zhang et al., 2025a). Moreover, teacher hacking (Tiapkin et al., 2025) could be another concerning issue, where the student model learns to mimic superficial patterns rather than acquiring meaningful knowledge. These limitations are further compounded by more challenging knowledge distillation settings when teacher and student models are significantly different in the model architectures, e.g. knowledge distillation from a Mixture-of-Experts (MoE) model to a dense model.

To this end, we propose a novel white-box KD framework to address the above challenges. Our approach consists of two stages: pretraining and distillation. In the pretraining stage, we compress teacher hidden states into a low-dimensional space and train the teacher projectors via a self-reconstruction mechanism, which extracts task-relevant information from the teacher model. In the distillation stage, we introduce a novel mechanism called  $residual\ learning$ , in which the student learns to predict next tokens using the residual hidden states computed by subtracting projected teacher hidden states from those of student whenever the teacher's predictions are inaccurate. This mechanism encourages the student to learn complementary knowledge that captures what it understands differently from the teacher and avoid replicating teacher errors. To support the cross-tokenizer distillation, we adopt a cross-model attention mechanism, inspired by Zhang et al. (2025b), that automatically establishes the alignments between teacher and student token representations through hidden state similarity. Finally, to better exploit the rich knowledge distributed across experts in a MoE teacher model, we introduce a lightweight scaled dot-production self-attention mechanism that enriches expert outputs before top-k aggregation. These components form a unified framework which is effective for distillation across diverse architectures and tokenization schemes.

We evaluate our framework on standard instruction-following benchmarks following recent works (Gu et al., 2023; Zhang et al., 2025b; Chen et al., 2025) under different distillation settings including cross-tokenizer distillation and distillation from MoE to dense models. Experimental results demonstrate that our method consistently outperforms the existing white-box KD approaches on average Rouge-L scores for different scenarios.

In summary, our contributions are as follows:

- We introduce a two-stage knowledge distillation framework that effectively extracts teacher knowledge for fine-tuning a student model.
- We introduce a residual learning approach that effectively leverages teacher's hidden states and errors as learning signals, enabling student model to generalize better.
- We adopt a cross-model attention mechanism that aligns teacher and student tokens based on their hidden state similarity to handle token mismatches in cross-tokenizer distillation.
- We enhance knowledge extraction for MoE teachers by using a scaled dot-product attention mechanism, fully leveraging all experts' knowledge.
- Experiments show that our method significantly outperforms the existing white-box methods under diverse distillation settings.

## 2 RELATED WORK

**Knowledge Distillation for LLMs**. A straightforward method for knowledge distillation is blackbox KD (Kim & Rush, 2016). Let  $\mathcal{T}$  and  $\mathcal{S}$  represent the teacher and student models, respectively. Black-box KD fine-tunes the student model using the standard causal language modeling objective on teacher-generated texts:

$$\mathcal{L}_{KD} = \mathbb{E}_{\mathbf{x} \sim \mathcal{X}, \mathbf{y} \sim \mathbb{P}_{\mathcal{T}}(\cdot | \mathbf{x})} \left[ -\log \mathbb{P}_{\mathcal{S}}(\mathbf{y} | \mathbf{x}) \right], \tag{1}$$

where y is the output sequence generated by teacher model,  $\mathbb{P}_{\mathcal{T}}$  and  $\mathbb{P}_{\mathcal{S}}$  are the output probability distributions from the teacher and student, respectively. Although simple, black-box KD has proven to be an effective method in diverse tasks (Wang et al., 2023; Taori et al., 2023; Chiang et al., 2023).

Recent research has shifted to white-box KD to leverage the intermediate information from LLMs beyond their final outputs. Specifically, these methods attempt to align output distributions or maximize the similarity between the hidden states of the teacher and student models (Sanh et al., 2019; Liang et al., 2023; Sun et al., 2019; Zuo et al., 2022; Shen et al., 2025). One common approach is minimizing the divergence between the output probability distributions of the teacher and student:

$$\mathcal{L}_{\text{divergence}} = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{X} \times \mathcal{Y}} \left[ D_{\text{KL}} \left( \mathbb{P}_{\mathcal{T}}(\mathbf{y} | \mathbf{x}) \parallel \mathbb{P}_{\mathcal{S}}(\mathbf{y} | \mathbf{x}) \right) \right]$$
(2)

where the data point  $(\mathbf{x}, \mathbf{y})$  are sampled from the data space  $\mathcal{X} \times \mathcal{Y}$  and the closed-form expression of  $D_{\mathrm{KL}}$  is determined based on the specific divergence metric adopted. Popular choices for  $D_{\mathrm{KL}}$  include Kullback-Leibler (KL) divergence, reverse KL (Gu et al., 2023), Jensen-Shannon divergence (Agarwal et al., 2024), and their skew variants (Lee, 2001; Ko et al., 2024; 2025).

While proven to be effective, divergence-based methods can potentially limit the student's learning capacity. The information from the teacher model is not perfect as it can still make wrong predictions and exhibit certain bias (Zhang et al., 2025a). As a result, forcing the student to mimic the teacher's distributions may propagate such imperfection to the student model and prevent the student from learning beyond the teacher's knowledge boundaries.

Cross-Tokenizer Knowledge Distillation. One restriction in white-box KD is that it requires the same vocabulary between the student and the teacher for distribution alignment. Recent work has further explored white-box KD for cross-tokenizer distillation settings, where the student and teacher have different vocabularies. Cross-tokenizer KD setting poses more challenges than conventional same-tokenizer settings, including mismatches in sequence length and output dimensionality. To tackle these challenges, ULD (Boizard et al., 2024) applies simple truncation strategies to align sequence lengths and logit dimensions, enabling direct comparison of output distributions. Based on ULD, MultiLevelOT (Cui et al., 2025) integrates an approximate optimal transport objective to reduce the divergence at both token and sequence levels. Dual-space KD (DSKD) (Zhang et al., 2025b) projects the student's hidden states into the teacher space and vice versa for calculating the divergence between output distributions. It further introduces a cross-model attention mechanism to align tokens across disparate tokenizers using token embeddings similarity. ALM (Minixhofer et al., 2025) further refines token alignment by exhaustively enumerating decoded chunks and identifying equivalent segments between teacher and student outputs.

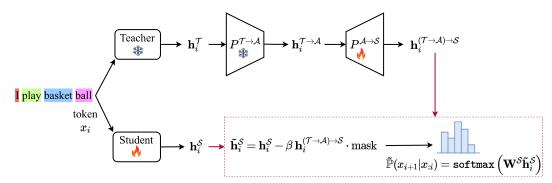
Despite these advances, existing cross-tokenizer KD methods exhibit several inherent limitations. Truncation-based strategies such as ULD may discard valuable information and suffer from alignment errors. MultiLevelOT improves performance but incurs computational overhead due to optimal transport, limiting its scalability. Projection-based approaches like DSKD rely on KL divergence on the output distributions, which could suffer from the limitations discussed above. Brute-force methods such as ALM achieve precise alignment but scale poorly with large vocabularies or extended sequences. Moreover, none of these state-of-the-art methods address teacher hacking (Tiapkin et al., 2025) or mitigate teacher bias (Zhang et al., 2025a), which could restrict the generalization capacity of the student when the teacher itself is imperfect.

Distillation from Mixture-of-Experts. Despite the superior capacity of Mixture-of-Experts (MoE) models on various natural language tasks, few works have specifically explored how to fully distill their inherent power (Salinas et al., 2022; Kim et al., 2025). While Kim et al. (2025) highlighted the importance of incorporating knowledge from all experts during distillation and proposed two mechanisms for expert knowledge extraction, their approaches face practical limitations. The first approach relies on stochastic expert sampling, which requires multiple forward passes to access different expert combinations and increases computational overhead during training. The second approach attempts to adjust router probabilities based on student preferences, but this may result in suboptimal expert selection as it prioritizes the student's current state over the teacher's expertise. These limitations motivate the need for a more efficient and effective method to distill knowledge from MoE teachers while preserving the specialized capabilities of individual experts.

#### 3 METHODOLOGY

In this section, we introduce a two-stage white-box KD framework that is applicable to various KD settings to address the existing limitations. Figure 1 illustrates the overall workflow of our framework. In the first stage, we employ learnable projectors to compress teacher hidden states into

(a) Stage 1: Pretraining teacher projectors. The projectors  $P^{\mathcal{T} \to \mathcal{A}}$  and  $P^{\mathcal{A} \to \mathcal{T}}$  are learned by optimizing next-token prediction objective using the reconstructed hidden states.



(b) Stage 2: Residual learning objective. We first compute the residual hidden states  $\tilde{\mathbf{h}}_{i}^{\mathcal{S}}$ , then use this quantity to obtain the adjusted output probability distribution  $\tilde{\mathbb{P}}(x_{i+1}|x_{:i})$  for next-token prediction.

Figure 1: Overview of our proposed two-stage framework. Modules marked with fire icons have learnable parameters that are updated during training, while those marked with snow icons have frozen weights.

a low-dimensional space that is agnostic to any student architecture and the projectors are optimized using a **self-reconstruction** mechanism as shown in Figure 1(a). Section 3.1 describes the details of the first stage. Figure 1(b) illustrates the workflow of the second stage which performs knowledge distillation from the teacher to the model. We introduce a novel mechanism called **residual learning** in Section 3.2 that leverages the residual hidden states calculated from the student hidden states and projected teacher hidden states to effectively guide student learning. Furthermore, we present the **expert knowledge fusion** mechanism in Section 3.3, which utilizes the knowledge from all experts to enhance distillation from MoE teachers. Finally, we introduce a **cross-model attention** mechanism in Section 3.4, aiming at handling sequence mismatch in cross-tokenizer KD settings. The total objective function for distillation is given in Section 3.5.

#### 3.1 Pretraining Projectors with Self-Reconstruction

The pretraining stage learns to compress teacher representations while preserving their semantics and task-relevant information. We denote  $\mathcal A$  as a low-dimensional space whose dimension is  $d_{\mathcal A}$ , and the hidden state dimensions of the student and teacher as  $d_{\mathcal S}$  and  $d_{\mathcal T}$ , respectively. Let  $P^{\mathcal T \to \mathcal A}$  and  $P^{\mathcal A \to \mathcal T}$  denote the projection functions that map hidden states between  $\mathcal T$  and  $\mathcal A$ . As shown in Figure 1(a), these projectors form an autoencoder in the self-reconstruction process, where teacher hidden states are compressed into  $\mathcal A$ , and then reconstructed. The reconstructed hidden states  $\mathbf h_i^{\mathcal T'}$  are subsequently used for next-token prediction, and the projectors are optimized via the crossentropy objective:

$$\mathcal{L}_{\text{CE}} = -\sum_{i} \log \mathbb{P}(x_{i+1}|x_{:i}), \quad \text{with} \quad \mathbb{P}(x_{i+1}|x_{:i}) = \operatorname{softmax}\left(\mathbf{W}^{\mathcal{T}}\mathbf{h}_{i}^{\mathcal{T}'}\right)$$
(3)

where  $\mathbf{W}^{\mathcal{T}}$  is the teacher's prediction head. We denote  $\mathbf{h}_i^{\mathcal{T} \to \mathcal{A}}$  as the compressed teacher hidden states, *i.e.*,  $\mathbf{h}_i^{\mathcal{T} \to \mathcal{A}} = P^{\mathcal{T} \to \mathcal{A}} \mathbf{h}_i^{\mathcal{T}}$ . In the subsequent stage,  $P^{\mathcal{T} \to \mathcal{A}}$  remains frozen to stabilize teacher information during distillation.

#### 3.2 RESIDUAL LEARNING

Recent work highlights two key challenges in distillation: *teacher hacking*, where the student exploits superficial patterns in the teacher's outputs rather than learning meaningful knowledge (Tiapkin et al., 2025), and *teacher bias*, where prediction errors are directly transferred to the student. To address these challenges, we propose *residual learning* for LLM distillation, which leverages not only the projected hidden states from the teacher but also the teacher errors as informative signals to guide student learning.

Let  $\mathbf{h}_i^{(\mathcal{T} \to \mathcal{A}) \to \mathcal{S}}$  denote the projected teacher hidden states of token i that are mapped to the student space  $\mathcal{S}$ , *i.e.*,  $\mathbf{h}_i^{(\mathcal{T} \to \mathcal{A}) \to \mathcal{S}} = P^{\mathcal{A} \to \mathcal{S}} \mathbf{h}_i^{\mathcal{T} \to \mathcal{A}}$ . We compute the **residual hidden states** by taking the difference between the student hidden states and the projected teacher representations in the student space at the positions where the teacher makes incorrect predictions:

$$\tilde{\mathbf{h}}_{i}^{\mathcal{S}} = \mathbf{h}_{i}^{\mathcal{S}} - \beta \mathbf{h}_{i}^{(\mathcal{T} \to \mathcal{A}) \to \mathcal{S}} \cdot \mathbb{1} \left[ \arg \max \mathbb{P}_{\mathcal{T}}(x_{i}|x_{:i-1}) \neq x_{i} \right], \tag{4}$$

where  $\tilde{\mathbf{h}}_i^S$  represents the residual hidden states,  $\beta$  is a scaling factor, and the indicator function equals 1 only when the teacher's top-1 prediction is different from the ground-truth token. The scaling factor  $\beta$  is crucial for balancing the contribution of teacher and student in the residual term. If  $\beta$  is too large, student representations are overwhelmed by those of teachers, making the residuals ineffective. In contrast, if  $\beta$  is too small, the residual terms approximate the student hidden states, and the benefits of residual learning would diminish. Therefore, we adaptively compute proper  $\beta$  values according to the dimensions and magnitudes of the hidden states as below:

$$\beta = \underbrace{\sqrt{\frac{d_{\mathcal{S}}}{d_{\mathcal{A}}}}}_{\text{rescale dimension}} \times \underbrace{\frac{1}{n} \sum_{i=1}^{n_{\mathcal{S}}} \frac{\left\| \mathbf{h}_{i}^{\mathcal{S}} \right\|}{\left\| \mathbf{h}_{i}^{(\mathcal{T} \to \mathcal{A}) \to \mathcal{S}} \right\|}}_{\text{rescale magnitude}}$$
(5)

where  $n_{\mathcal{S}}$  stands for the student's sequence length. The first component in  $\beta$  normalizes for dimensional differences between spaces, while the second term aligns the magnitudes of teacher and student representations to ensure balanced residual computation. These ratios prevent either the student or teacher representations from dominating the residual calculation. We empirically observe that using  $\beta$  at sequence level (average of token-wise norm ratios) can stabilize training and yield better results.

Then, we train the student model to make accurate predictions based on the residual hidden states using the student's LM head  $W^S$  with the cross-entropy loss:

$$\mathcal{L}_{\text{res}} = -\sum_{i=1}^{n_{\mathcal{S}}} \log \tilde{\mathbb{P}}\left(x_{i+1} | x_{:i}\right), \quad \text{where} \quad \tilde{\mathbb{P}}\left(x_{i+1} | x_{:i}\right) = \operatorname{softmax}\left(\mathbf{W}^{\mathcal{S}} \tilde{\mathbf{h}}_{i}^{\mathcal{S}}\right)$$
(6)

This objective fundamentally changes the learning dynamics: instead of passively replicating the teacher's distributions, the student learns to actively identify the differences between its current understanding and the teacher's knowledge. This prevents the superficial pattern matching that characterizes teacher imperfection, ultimately leading to better generalization of the student model.

#### 3.3 EXPERT KNOWLEDGE FUSION

For teacher models with MoE structures, we propose an efficient expert knowledge fusion mechanism to effectively utilize the knowledge of all experts for better knowledge distillation. Given E experts in the teacher model, we denote  $\mathbf{h}_i^{(m)}$  as the outputs of the m-th expert at the token position i for the last MoE layer. We enrich the expert outputs through scaled dot-product attention:

$$\tilde{\mathbf{h}}_{i}^{(m)} = \sum_{j=1}^{E} \alpha_{mj} \mathbf{h}_{i}^{(j)}, \quad \text{where} \quad \alpha_{mj} = \operatorname{softmax} \left( \frac{\mathbf{h}_{i}^{(m)} (\mathbf{h}_{i}^{(j)})^{\top}}{\sqrt{d_{\mathcal{T}}}} \right)$$
(7)

where  $\alpha_{mj}$  captures the relevance between the outputs of the m-th and j-th experts. This self-attention mechanism allows each expert to incorporate complementary information from similar

experts, resulting in more informative representations. Finally, the enriched expert outputs are aggregated to form the final states following the original MoE workflow:

$$\mathbf{h}_i^{\mathcal{T}} = \sum_{j \in \text{top-}k} g_j \tilde{\mathbf{h}}_i^{(j)} \tag{8}$$

where  $g_j$  denotes the router probability assigned to the j-th expert, and top-k denotes the k experts selected by the router. Unlike previous methods that rely on stochastic sampling or router probability adjustment (Kim et al., 2025), our attention-based fusion can directly leverage the complementary knowledge of all experts in a single forward pass.

#### 3.4 Cross-Model Attention Mechanism

In Section 3.2, we introduce our residual learning design in the same-tokenizer setting where the input sequences for the teacher and student are exactly the same. In the cross-tokenizer setting, since teacher and student have different vocabularies, their tokenizers will output sequences of different lengths for the same text input. This hinders the application of residual learning.

To address this challenge, we propose a cross-model attention mechanism to align the compressed teacher hidden states  $\mathbf{h}_i^{\mathcal{T} \to \mathcal{A}}$  for the input tokens of the student. We first map the student hidden states  $\mathbf{h}_i^{\mathcal{S}}$  to the same low-dimensional space  $\mathcal{A}$  (denoted as  $\mathbf{h}_i^{\mathcal{S} \to \mathcal{A}}$ ) via a trainable projector. Then, for each pair of student token i and teacher token i, we compute their semantic similarity via dot-products using the normalized hidden states, followed by row-wise softmax normalization:

$$\mathbf{A}_{ij} = \left[ \frac{\mathbf{h}_{i}^{\mathcal{S} \to \mathcal{A}}}{\operatorname{std}\left(\mathbf{h}_{i}^{\mathcal{S} \to \mathcal{A}}\right)} \right]^{\top} \left[ \frac{\mathbf{h}_{j}^{\mathcal{T} \to \mathcal{A}}}{\operatorname{std}\left(\mathbf{h}_{j}^{\mathcal{T} \to \mathcal{A}}\right)} \right], \quad \text{then} \quad \mathbf{A}[i,:] = \operatorname{softmax}\left( \frac{\mathbf{A}[i,:]}{\sqrt{d_{\mathcal{A}}}} \right) \tag{9}$$

By computing similarity scores between all token pairs, we construct a cross-model attention matrix  $\mathbf{A} \in \mathbb{R}^{n_S \times n_T}$ , where  $n_S$  and  $n_T$  are the sequence lengths for student and teacher respectively. Each element  $\mathbf{A}_{ij}$  represents the semantic similarity between student token i and teacher token j. Using the cross-model attention matrix  $\mathbf{A}$ , we calculate the aligned teacher hidden states  $\hat{\mathbf{h}}_i^{T \to \mathcal{A}}$  for each student token by taking a weighted sum of the compressed teacher hidden states:

$$\hat{\mathbf{h}}_i^{\mathcal{T} \to \mathcal{A}} = \sum_{j=1}^{n_{\mathcal{T}}} \mathbf{A}_{ij} \mathbf{h}_j^{\mathcal{T} \to \mathcal{A}},\tag{10}$$

This mechanism allows each student token to attend to the most semantically relevant teacher tokens, effectively bridging the tokenization gap and preserving semantic information. Then, we can use  $\hat{\mathbf{h}}_i^{\mathcal{T} \to \mathcal{A}}$  instead to perform residual learning under the cross-tokenizer settings.

#### 3.5 Training Objective

We incorporate the residual learning objective from Eqn. 6 with a standard supervised fine-tuning (SFT) objective where the student model learns to predict the next token from the ground-truth data. The final loss function is:

$$\mathcal{L} = \lambda_{\text{res}} \mathcal{L}_{\text{res}} + \lambda_{\text{SFT}} \mathcal{L}_{\text{SFT}}, \tag{11}$$

where  $\lambda_{\rm res}$  and  $\lambda_{\rm SFT}$  are hyperparameters that balance the contribution of each loss component during student model training. In our experiments, we simply use  $\lambda_{\rm res} = \lambda_{\rm SFT} = 0.5$ .

#### 4 EXPERIMENTS

#### 4.1 EXPERIMENTAL SETUP

**Models.** We evaluate our framework under both the same-tokenizer and cross-tokenizer settings. For the same-tokenizer setting, we conduct experiments on knowledge distillation from Mixtral-8×7B-Instruct (Jiang et al., 2024) to Mistral-7B and distillation from LLaMA2-7B (Touvron et al., 2023) to TinyLLaMA-1.1B (Zhang et al., 2024). To optimize LLaMA2-7B's performance as a teacher model, we further fine-tune it on the training dataset before knowledge distillation. For the cross-tokenizer

Table 1: Performance comparison using Rouge-L scores (%) for same-tokenizer KD. We report the average scores over 3 random seeds.

Methods	Dolly	SelfInst	VicunaEval	S-NI	UnNI	Avg.		
$Mixtral-8 \times 7B-Instruct \rightarrow Mistral-7B$								
Teacher zero-shot	25.55	24.12	26.62	40.84	36.21	30.67		
Student SFT	25.85	22.17	17.18	33.41	30.26	25.77		
ULD	28.64	21.99	21.31	35.71	34.42	28.41		
MultiLevelOT	30.55	23.07	20.62	34.79	36.78	29.16		
DSKD	25.35	20.43	18.27	35.39	31.45	26.18		
Ours	31.39	24.27	20.48	38.82	38.42	30.68		
LLaMA2-7B → TinyLLaMA-1.1B								
Teacher SFT	26.42	17.90	17.65	26.82	29.77	23.71		
Student SFT	24.26	16.41	16.13	25.70	27.41	21.98		
ULD	23.15	15.72	15.73	28.52	28.01	22.23		
MultiLevelOT	21.80	14.60	14.82	26.62	25.35	20.64		
DSKD	21.66	15.21	14.87	25.51	27.90	21.03		
Ours	25.70	20.50	16.84	32.50	33.64	25.84		

setting, we use Mixtral-8×7B-Instruct as the teacher model and use TinyLLaMA-1.1B and GPT2-120M (Radford et al., 2019) as the student models. We perform full fine-tuning on GPT2-120M and LoRA fine-tuning on TinyLLaMA-1.1B and Mistral-7B. In this way, our evaluation covers diverse knowledge distillation settings in practice.

**Datasets.** For training, we use the Dolly dataset (Ouyang et al., 2022), processed by Gu et al. (2023), which comprises approximately 11,000 training samples and 1,000 validation samples. We evaluate our method on five standard instruction-following benchmarks: Dolly (500 test samples), Self-Instruction (Wang et al., 2023), Vicuna-Eval (Chiang et al., 2023), Super-Natural Instructions (Wang et al., 2022), and Unnatural Instruction (Honovich et al., 2023). We adopt Rouge-L scores (%) as the primary metric across all datasets.

**Baselines.** We select the following approaches as baselines: supervised fine-tuning (SFT), Universal Logits Distillation (ULD, Boizard et al. (2024)), Multi-Level Optimal Transport (MultiLevelOT, Cui et al. (2025)), Dual-space KD (DSKD, (Zhang et al., 2025b)), and Approximate Likelihood Matching (ALM, (Minixhofer et al., 2025)). Although these baselines were specifically designed for cross-tokenizer KD, they are also capable of achieving state-of-the-art performance in the same-tokenizer setting, except for ALM which can only be applied for cross-tokenizer KD.

Implementation Details. We conduct experiments using PyTorch Distributed on NVIDIA A100 GPUs (80GB). We use bfloat16 precision and gradient accumulations to manage memory constraints. The dimension of the space  $\mathcal A$  is set as  $d_{\mathcal A}=64$  across all experiments, and all projector modules are linear weights without bias. Teacher projectors are pretrained for 10 epochs as described in Section 3.1 and we use the checkpoint at the final step. For final evaluation, we select the checkpoint with the best validation Rouge-L score during distillation. The detailed hyperparameters can be found in Appendix A.

#### 4.2 RESULTS

**Same-tokenizer KD results.** Table 1 reports the Rouge-L scores for same-tokenizer KD across multiple benchmarks. For  $Mixtral-8 \times 7B$ -Instruct  $\rightarrow$  Mistral-7B, our method achieves the highest average score (30.68), which matches the teacher zero-shot performance (30.67) and significantly outperforms the best baseline MultiLevelOT (29.16). For  $LLaMA2-7B \rightarrow TinyLLaMA-1.1B$ , our method surpasses all baselines and teacher SFT, reaching an average of 25.84, which is +2.13 above the teacher SFT performance. These results demonstrate the effectiveness of our proposed framework in improving student's generalization beyond the teacher capacity.

Table 2: Performance comparison using Rouge-L scores (%) for cross-tokenizer KD. We report the average scores over 3 random seeds.

Methods	Dolly	SelfInst	VicunaEval	VicunaEval S-NI		Avg.		
Teacher zero-shot	25.55	24.12	26.62	40.84	36.21	30.67		
Mixtral-8×7B-Instruct → TinyLLaMA-1.1B								
Student SFT ULD MultiLevelOT ALM DSKD Ours	24.26 23.61 22.52 21.93 <b>26.16</b> 25.84	16.41 15.45 14.48 14.11 18.02 <b>19.38</b>	16.13 17.06 14.97 15.37 <b>18.15</b> 17.35	25.70 29.01 27.06 25.75 27.06 <b>30.86</b>	27.41 28.45 25.77 25.49 30.08 <b>32.04</b>	21.98 22.71 20.96 20.53 23.89 <b>25.09</b>		
$\frac{\text{Mixtral-8} \times 7B\text{-Instruct} \rightarrow GPT2-120M}{\text{Mixtral-8} \times 7B\text{-Instruct} \rightarrow GPT2-120M}$								
Student SFT ULD MultiLevelOT ALM DSKD Ours	21.45 22.12 21.40 21.16 <b>23.11</b> 22.62	9.45 10.54 9.57 9.77 11.20 <b>11.66</b>	14.89 15.03 15.56 15.46 <b>15.81</b> 15.12	16.86 18.16 15.88 16.09 20.33 <b>24.51</b>	19.14 20.12 18.05 18.26 21.56 <b>26.15</b>	16.36 17.19 16.09 16.15 18.40 <b>20.01</b>		

**Cross-tokenizer KD results.** Table 2 presents the results for cross-tokenizer KD settings. For  $Mixtral-8 \times 7B$ -Instruct  $\rightarrow$  TinyLLaMA-1.1B, our method again achieves the best average performance (25.09), outperforming the best baseline DSKD (23.89) by +1.20. For  $Mixtral-8 \times 7B$ -Instruct  $\rightarrow$  GPT2-120M, our approach reaches 20.01 average Rouge-L, significantly outperforming all baselines. We also note that the performance gap between the teacher and the fine-tuned student is larger in the cross-tokenizer settings, which is due to the the compounded difficulty of cross-tokenizer alignment and smaller student model size.

In summary, for both same- and cross-tokenizer KD, our method outperforms the existing baselines. The improvements are robust across datasets and teacher-student pairs, highlighting the effectiveness of our approach in preserving and transferring knowledge.

#### 4.3 ABLATION STUDY

To understand the effectiveness of each design choice in our framework, we conduct comprehensive ablation studies under the cross-tokenizer setting where we use Mixtral- $8\times7B$ -Instruct as the teacher and GPT2-120M as the student.

Effectiveness of each technique. We consider the following variants of our method: 1) "w/o  $\beta$ ", in which we remove  $\beta$  for computing the residual hidden states; 2) "w/o accuracy mask", in which we remove the indicator function for computing the residual hidden states; 3) "w/o pretrained  $P^{\mathcal{T} \to \mathcal{A}}$ ", in which we remove the pretraining stage and train  $P^{\mathcal{T} \to \mathcal{A}}$  directly during distillation; 4) "w/o MoE expert fusion", in which we remove our expert knowledge fusion design. We further consider another variant to leverage the knowledge of all experts which simply activates and aggregates all the experts, i.e. using top-k as top-k for Mixtral-k0. Table 3 compares the results of our method with these variants. We can find that all these variants perform significantly worse than the original method, which demonstrates the effectiveness of each design. Among them, removing the coefficient k1 leads to the most significant performance drop, highlighting its role in balancing the teacher–student contributions within residual learning.

Impact of the low-dimensional space. Under the cross-tokenizer settings, the low-dimensional space  $\mathcal{A}$  is used to compress teacher hidden states and perform cross-model attention for alignment. Here, we further study the impact of the dimension  $d_{\mathcal{A}}$  on the distillation performance. Figure 2(a) shows the distillation performance when varying  $d_{\mathcal{A}}$ . We find that increasing  $d_{\mathcal{A}}$  does not provide additional benefit and a very large dimension (e.g.  $d_{\mathcal{A}} = 768$  or 1024) can even significantly harm

Table 3: Ablation study results for different variants under the knowledge distillation from Mixtral- $8 \times 7B$ -Instruct (teacher) to GPT2-120M (student).

Description	Dolly	SelfInst	VicunaEval	S-NI	UnNI	Avg.	
Mixtral-8×7B-Instruct $\rightarrow$ GPT2-120M							
Ours	22.62	11.66	15.12	24.51	26.15	20.01	
w/o β	22.31	9.59	14.43	16.01	18.93	16.25	
w/o accuracy mask	22.63	11.04	14.79	24.02	25.45	19.59	
w/o pretrained $P^{\mathcal{T} \to \mathcal{A}}$	21.52	10.07	15.24	18.81	21.31	17.39	
w/o MoE expert fusion							
- sparse experts (2/8 activated)	22.68	11.84	14.82	23.36	25.61	19.66	
– all experts (8/8 activated)	23.09	10.52	16.10	23.88	24.82	19.68	

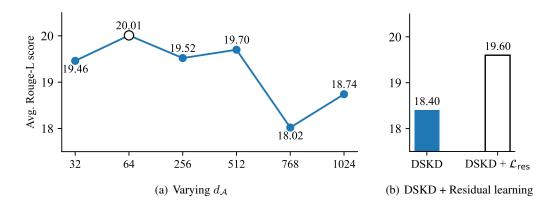


Figure 2: Cross-tokenizer distillation results on (a) varying the dimensionality of the space  $\mathcal{A}$  (b) incorporating residual learning into DSKD.

the performance. It proves that compressing teacher hidden states into a low dimension is not only beneficial but essential for effective knowledge transfer. In our case,  $d_A=64$  yields the best result (20.01), indicating a balance between compression and expressivity.

Generality of residual learning. To demonstrate the effectiveness of residual learning, we further apply our design into other white-box KD methods. Figure 2(b) compares the performance of incorporating the residual loss  $\mathcal{L}_{res}$  with the baseline DSKD. We see an improvement of +1.20 on average after incorporating residual learning. It demonstrates the generality of our residual learning mechanism, which has the potential to improve the performance of other white-box KD methods.

### 5 CONCLUSION

In this work, we introduce a novel two-stage KD framework that addresses key challenges arising from the imperfect teacher knowledge in white-box KD settings. We compress the teacher hidden states into a low-dimension space prior to distillation and guide student training through residual learning, which ensures robust knowledge transfer and effectively mitigates overfitting to the teacher's imperfection. Our framework further provides targeted designs for diverse KD settings including cross-tokenizer distillation and distillation from an MoE model. Extensive experiments on multiple instruction-following benchmarks demonstrate that our method consistently outperforms the existing baselines and significantly improves the capacity of the student model under different settings. In the future, we will further investigate if our method can generalize to other tasks such as reasoning-intensive tasks and code generation.

#### REFERENCES

- Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.
- Nicolas Boizard, Kevin El Haddad, Céline Hudelot, and Pierre Colombo. Towards cross-tokenizer distillation: the universal logit distillation loss for llms. *arXiv preprint arXiv:2402.12030*, 2024.
- Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541, 2006.
- Yijie Chen, Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. Enhancing cross-tokenizer knowledge distillation with contextual dynamical mapping. *arXiv preprint arXiv:2502.11104*, 2025.
- Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113, 2023.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv* preprint arXiv:2507.06261, 2025.
- Xiao Cui, Mo Zhu, Yulei Qin, Liang Xie, Wengang Zhou, and Houqiang Li. Multi-level optimal transport for universal cross-tokenizer knowledge distillation on language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23724–23732, 2025.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2023.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv* preprint arXiv:1503.02531, 2015.
- Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14409–14428, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.806. URL https://aclanthology.org/2023.acl-long.806/.
- Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. Dynabert: Dynamic bert with adaptive width and depth. *Advances in Neural Information Processing Systems*, 33: 9782–9793, 2020.
- Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

- Gyeongman Kim, Gyouk Chu, and Eunho Yang. Every expert matters: Towards effective knowledge distillation for mixture-of-experts language models. arXiv preprint arXiv:2502.12947, 2025.
  - Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.
  - Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*, 2024.
  - Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. Distillm-2: A contrastive approach boosts the distillation of llms. *arXiv preprint* arXiv:2503.07067, 2025.
  - Lillian Lee. On the effectiveness of the skew divergence for statistical language analysis. In *International workshop on artificial intelligence and statistics*, pp. 176–183. PMLR, 2001.
  - Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*, pp. 20852–20867. PMLR, 2023.
  - Benjamin Minixhofer, Ivan Vulić, and Edoardo Maria Ponti. Universal cross-tokenizer distillation via approximate likelihood matching. *arXiv preprint arXiv:2503.20083*, 2025.
  - Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.
  - Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
  - Felipe Cruz Salinas, Kenichi Kumatani, Robert Gmyr, Linquan Liu, and Yu Shi. Knowledge distillation for mixture of experts models in speech recognition. Technical report, Technical Report MSR-TR-2022-6, Microsoft Research, May 2022. https://www..., 2022.
  - Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.
  - Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. Codi: Compressing chain-of-thought into continuous space via self-distillation. *arXiv preprint arXiv:2502.21074*, 2025.
  - Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*, 2019.
  - Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.
  - Daniil Tiapkin, Daniele Calandriello, Johan Ferret, Sarah Perrin, Nino Vieillard, Alexandre Ramé, and Mathieu Blondel. On teacher hacking in language model distillation. *arXiv preprint arXiv:2502.02671*, 2025.
  - Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
    - Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *arXiv preprint arXiv:2204.07705*, 2:2, 2022.

- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754/.
- Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. F-divergence minimization for sequence-level knowledge distillation. *arXiv preprint arXiv:2307.15190*, 2023.
- Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. arXiv preprint arXiv:2304.12244, 2023.
- Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models. *arXiv* preprint arXiv:2402.13116, 2024.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Jianhua Zhang, Yi Gao, Ruyu Liu, Xu Cheng, Houxiang Zhang, and Shengyong Chen. Can students beyond the teacher? distilling knowledge from teacher's bias. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22434–22442, 2025a.
- Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.
- Xue Zhang, Songming Zhang, Yunlong Liang, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. A dual-space framework for general knowledge distillation of large language models. *arXiv preprint arXiv:2504.11426*, 2025b.
- Simiao Zuo, Qingru Zhang, Chen Liang, Pengcheng He, Tuo Zhao, and Weizhu Chen. Moebert: from bert to mixture-of-experts via importance-guided adaptation. *arXiv* preprint arXiv:2204.07675, 2022.

# CONFIGURATIONS AND HYPERPARAMETERS

In this appendix, we provide details of the configurations and hyperparameters for training and evaluation, as shown in Table 4.

Table 4: Training configuration and hyperparameters					
Parameter	Value				
Hardware & Training					
GPU	8× NVIDIA A100 (80GB)				
Global batch size	128				
Max sequence length	512				
Precision	bfloat16				
Optimization					
Optimizer	AdamW				
Learning rate (student)	$10^{-3}$				
Learning rate (projectors)	$10^{-3}$				
Learning rate scheduler	Cosine decay				
Weight decay	$10^{-4}$				
Epochs	10				
Architecture					
Alignment space dim $(d_A)$	64				
Projector layers	Linear (without bias)				
Teacher projector pretraining	10 epochs				
Loss Coefficients					
$\lambda_{ m res}$	0.5				
$\lambda_{ ext{SFT}}$	0.5				
LoRA Parameters					
Modules	Q, O, Up and Down projections				
Rank	256				
lpha	8				
Dropout rate	0.1				
<b>Generation Configurations</b>					
Temperature	1.0				
Top P	1.0				

# B RESIDUAL LEARNING VS. DUAL SPACE KD UNDER DIVERGENCE VARIANTS

To further evaluate the robustness of our framework, we compare residual learning with the strong Dual-Space KD (DSKD) baseline under different divergence loss functions. In particular, we replace the KL divergence in DSKD with alternatives including reverse KL, Jensen–Shannon (JS) divergence, adaptive KL, and skewed KL variants, and measure the student performance in each setting. All experiments are conducted with Mixtral-8×7B-Instruct as the teacher and GPT2-120M as the student. The results in Table 5 show that residual learning consistently outperforms DSKD across all divergence objectives, underscoring its generalization and effectiveness. By leveraging the teacher's errors as learning signals, residual learning prevents the student from imitating the teacher's output distributions in a rigid manner, resulting in stronger generalization.

Table 5: Comparing residual learning vs. DSKD under divergence variants.

Description	Dolly	SelfInst	VicunaEval	S-NI	UnNI	Avg.		
Mixtral-8×7B-Instruct $\rightarrow$ GPT2-120M								
DSKD + forward KL	23.11	11.20	15.81	20.33	21.56	18.40		
DSKD + reverse KL	22.03	9.44	12.95	17.08	18.41	15.98		
DSKD + adaptive KL	23.41	10.65	14.76	18.88	20.46	17.63		
DSKD + JS divergence	22.60	10.59	16.21	18.44	20.37	17.64		
DSKD + skew forward KL	23.63	10.89	15.57	19.42	21.21	18.14		
DSKD + skew reverse KL	23.15	11.10	15.64	20.47	21.67	18.40		
Ours	22.62	11.66	15.12	24.51	26.15	20.01		