DWA-KD: Dual-Space Weighting and Time-Warped Alignment for Cross-Tokenizer Knowledge Distillation

Duc Trung Vu[1]*, Pham Khanh Chi[1]*, Dat Phi Van[1]*, Linh Ngo Van[1], Sang Dinh[1]†, Trung Le[2]
[1] Hanoi University of Science and Technology, Vietnam,
[2] University of Monash, Australia,

Abstract

Knowledge Distillation (KD) has emerged as a crucial technique for compressing Large Language Models (LLMs). Although existing cross-tokenizer KD methods have made notable progress, their effectiveness remains constrained by suboptimal alignment across sequence and vocabulary levels. To address these limitations, we introduce Dual-Space Weighting and Time-Warped Alignment (DWA-KD), a novel cross-tokenizer distillation framework that enhances token-wise distillation through dual-space entropy-based weighting and achieves precise sequence-level alignment by leveraging both lexical and semantic information. At the token level, DWA-KD maps teacher representations into the student space and vice versa, performing dual-space KD via Kullback–Leibler divergence (KL). The process is modulated by dual-space weights that up-weight tokens where the student is uncertain and the teacher is confident, thereby focusing learning on informative tokens rather than treating all positions equally. At the sequence level, DWA-KD applies Soft Dynamic Time Warping (Soft-DTW) to both the embedding and final hidden-state layers, enabling robust alignment of lexical and contextual semantics between teacher and student sequences. Extensive experiments across diverse NLP benchmarks demonstrate that DWA-KD outperforms state-of-the-art KD baselines, while ablation studies confirm the complementary contributions of entropy-based token weighting and embedding and final hidden state layer Soft-DTW alignment.

1 Introduction

Large language models deliver strong results but incur high compute, memory, and latency (Kaplan et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023; OpenAI, 2023). This creates a need for smaller models that remain efficient in practice. Knowledge distillation (KD) (Hinton et al., 2015) has emerged as a promising solution to this challenge by transferring knowledge from a large teacher model to a compact student model, maintaining essential performance while reducing inference overhead (Ko et al., 2024; Chen et al., 2025; Gu et al., 2024). Conventional KD is effective for LLMs but is often constrained by a *same-tokenizer* setting, i.e., requiring shared vocabulary between teacher and student, which limits applicability across heterogeneous models (Wen et al., 2023; Gu et al., 2024; Ko et al., 2024).

Cross-tokenizer KD (CTKD) removes the shared-tokenizer assumption but introduces main challenges: (i) *tokenization mismatch* and (ii) *sequence misalignment*. Recent CTKD methods address these challenges by leveraging Optimal Transport (OT) (Boizard et al., 2024; Cui et al., 2025), learning projectors and uses cross-model attention to reconcile hidden spaces (Zhang et al., 2024) or aligning strings with dynamic programming on token sequences (Wan et al., 2024). While OT-based sequence alignment (Boizard et al., 2024; Cui et al., 2025; Truong et al., 2025) does not inherently preserve temporal order (Su and Hua, 2017), Wan et al. (2024) and Zhang et al. (2024) only captures surface form or logit-level and overlook semantics (Chen et al., 2025). Meanwhile, a novel method - Contextual Dynamic Mapping (CDM) (Chen et al., 2025) - introduces entropy-weighted dynamic programming based on Dynamic Time Warping (DTW) to adapt token alignment. However, CDM is computationally expensive. Its string-level DP uses CPU-side loops each forward pass, which leads to GPU under-utilization. Recent work has also explored cross-tokenizer distillation under preference/alignment signals (Nguyen et al., 2026).

DTW-based objectives also appear in broader settings (e.g., Fu et al. (2023); Wan et al. (2024)), which is used to align at sequence-level. Despite

*Equal contribution.
†Corresponding author: sangdv@soict.hust.edu.vn

Findings of the Association for Computational Linguistics: EACL 2026, pages 3513–3527
March 24-29, 2026 ©2026 Association for Computational Linguistics

recent progress in cross-tokenizer knowledge distillation, current approaches exhibit significant limitations. These methods assume all tokens contribute equally, neglecting the varying importance or informativeness of individual tokens. Moreover, sequence-level context is often under-weighted in CTKD, weakening robustness when tokenizers diverge substantially.

To address these limitations, we propose **Dual-Space Weighting and Time-Warped Alignment (DWA-KD)**, a CTKD framework that aligns teacher and student at both token and sequence levels. At the token level, we adopt an asymmetric scheme: on the *student side*, each position is weighted by the product of the student's normalized entropy and the teacher's *projected* confidence (the maximum probability after mapping the teacher distribution into the student vocabulary), emphasizing updates where the student is uncertain and the teacher target is reliable; on the *teacher side*, we use *entropy-only* weighting, assigning larger weights to low-entropy (confident) teacher tokens. This dual mechanism concentrates learning on the most consequential tokens while avoiding noisy or redundant updates. At the sequence level, we employ normalized Soft-DTW (Cuturi and Blondel, 2017) with an attention-informed soft band over both embeddings and final hidden states, reducing input-level mismatch and encouraging the student to track the teacher's semantic trajectory before producing logits.

We evaluate our framework across multiple benchmarks under cross-tokenizer setting. The experimental results demonstrate that DWA-KD exhibits superiority over existing cross-tokenizer distillation approaches among different datasets. Furthermore, ablation study confirms the effectiveness of our main components. In summary, the main contributions of this work are:

- We present **Dual-Space Weighting and Time-Warped Alignment (DWA-KD)**, a cross-tokenizer distillation framework that couples selective *token-level* transfer with *sequence-level* Soft-DTW, applied to both embeddings and final hidden states, with an *attention-informed soft band* that keeps alignments plausible and gradients smooth.
- We introduce **a dual-space token weighting scheme**: on the student side, weights scale with the product of student uncertainty and teacher → student projected confidence; on the teacher side, weights increase with teacher confidence (low entropy). This focuses learning on tokens that matter while suppressing noisy updates.
- We conduct comprehensive experiments across diverse datasets to evaluate our approach, showing that it enhances the performance of student models compared to standard knowledge distillation baselines.

2 Related Work and Background

Knowledge Distillation Knowledge distillation (KD) is a widely adopted technique for transferring knowledge from a teacher to a student model by aligning soft targets such as logits or intermediate representations (Hinton et al., 2015; Kim and Rush, 2016). With the rise of large language models (LLMs), many conventional KD methods (Park et al., 2021; Gu et al., 2023; Wu et al., 2024; Ko et al., 2024), while effective, are often limited to same-tokenizer setting. Cross-tokenizer KD (Zhou et al., 2022; Zhang et al., 2023; Liu et al., 2022) addresses this challenge but introduce other problem, mainly tokenization mismatch and sequence mismatch. To address these issues, DSKD (Zhang et al., 2024) aligns tokens by learning projectors that map student and teacher hidden states into a shared space, but it treats all positions equally and ignores token-wise importance. MultiLevelOT (Cui et al., 2025) and MCW-KD (Vuong et al., 2026) leverage Wasserstein/OT-style objectives for distillation. Beyond token-level matching, MultiLevelOT (Cui et al., 2025) and CDM (Chen et al., 2025) further extend alignment to token and sequence levels: MultiLevelOT measures discrepancies within and across tokens via optimal transport but does not inherently preserve temporal order, while CDM applies entropy-weighted DTW in the logit space, which can underuse contextual information encoded in hidden states. In contrast, our approach employs asymmetric token weighting and applies normalized Soft-DTW at both the embedding and final hidden-state layers. This combination reduces input-level mismatch and better captures semantic trajectories, enabling more precise and efficient knowledge transfer.

Dynamic Time Warping Dynamic Time Warping (DTW) (Sakoe and Chiba, 1978) is a classical algorithm for computing optimal alignments between temporal sequences, accommodating differ-

ences in length or variations in speed. Its original formulation, however, is non-differentiable, limiting use in gradient-based learning. Soft Dynamic Time Warping (Soft-DTW) (Cuturi and Blondel, 2017) addresses this by replacing the hard minimum with a smooth approximation, enabling end-to-end training. Soft-DTW has demonstrated improved stability and performance in domains such as time series regression, speech, handwriting, and motion trajectory prediction. Building on this, several DTW-based methods leverage soft alignment to capture temporal or structural correspondences between teacher and student representations, aligning token spans before pooling logits for divergence computation (Fu et al., 2023; Wan et al., 2024; Chen et al., 2025). In contrast, our approach applies Soft-DTW directly at the embedding and hidden-state levels, providing differentiable alignment that captures both lexical and semantic dependencies for more effective knowledge transfer.

2.1 Background

2.1.1 Dual-Space Knowledge Distillation

DSKD (Zhang et al., 2024) addresses tokenization mismatch by learning soft correspondences between student and teacher tokens via a cross-model attention (CMA) layer. Concretely, student embeddings of input and target tokens are concatenated and projected to query vectors $Q \in \mathbb{R}^{S\times 2D}$; teacher embeddings and output hidden states are then normalized to form key vectors $K \in \mathbb{R}^{T\times 2D}$ and projected to value vectors $V \in \mathbb{R}^{T\times d}$. Scaled dot-product attention is formalized as:

$$a^{t\to s} = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{2D}}\right) \in \mathbb{R}^{S\times T}. \quad (1)$$

produces teacher-to-student alignments and being used to align teacher hidden state to student hidden state $\tilde{h}^{t\to s} = a^{t\to s}V \in \mathbb{R}^{n\times d}$. A reverse alignment matrix is defined as:

$$a^{s\to t} = \mathrm{softmax}\left(\frac{KQ^\top}{\sqrt{2D}}\right) \in \mathbb{R}^{T\times S} \quad (2)$$

and also be used to project and align student's hidden states to teacher's hidden state $\tilde{h}^{s\to t} = a^{s\to t}P^{s\to t}\left(h^s;\theta_P^{s\to t}\right) \in \mathbb{R}^{T\times D}$. where h^s is the output hidden states of the whole sequence from the student model.

Dual-Space KD. (i) *Student space:* $\tilde{h}^{t\to s}$ is mapped by the student head to produce output distribution $p^{t\to s}$. Because the projectors start at random, an auxiliary cross-entropy is used to warm them up; then a KL term matches $p^{t\to s}$ to the student distribution p^s. (ii) *Teacher space:* reverse-aligned hidden state $\tilde{h}^{s\to t}$ are passed through the teacher head to obtain $p^{s\to t}$, which is aligned to the teacher distribution p^t via KL.

Objective. The final loss sums KD in both spaces plus the auxiliary CE:

$$\mathcal{L}_{\mathrm{DSKD}} = \mathcal{L}_{\mathrm{kd}}^{\mathrm{stu}} + \mathcal{L}_{\mathrm{kd}}^{\mathrm{tea}} + \mathcal{L}_{\mathrm{ce}}^{t\to s}. \quad (3)$$

2.1.2 Dynamic Time Warping (DTW)

DTW (Sakoe and Chiba, 1978) is a dynamic-programming method that aligns two time sequences by allowing non-linear time warping. This property makes DTW particularly useful for sequences with varying lengths or speed variations. Formally, given sequences $X = (x_1, \ldots, x_N)$, and $Y = (y_1, \ldots, y_M)$, DTW constructs a cost matrix $\mathbf{C} \in \mathbb{R}^{N\times M}$ with entries:

$$C_{ij} = d(x_i, y_j), \quad (4)$$

where $d(\cdot,\cdot)$ is a local distance function (e.g., squared Euclidean distance), and computes the accumulated cost matrix $\mathbf{R}$ recursively via:

$$R_{ij} = C_{ij} + \min\left(R_{i-1,j},\, R_{i,j-1},\, R_{i-1,j-1}\right). \quad (5)$$

While DTW preserves the temporal order of elements, the non-differentiability of the $\min$ operator prevents its direct integration into gradient-based learning frameworks. Soft Dynamic Time Warping (Soft-DTW) (Cuturi and Blondel, 2017) extends DTW by introducing a differentiable relaxation of the alignment cost. The non-differentiable $\min$ in the recurrence is replaced with a smooth softmin operator:

$$\min{}^{\gamma}(a_1, \ldots, a_n) = -\gamma \log \sum_i e^{-a_i/\gamma} \quad (6)$$

where $\gamma > 0$ controls the smoothness and as $\gamma \to 0$, it converges to the standard minimum. Incorporating this operator into the DTW recurrence yields a differentiable cost that supports backpropagation, enabling end-to-end training in sequence modeling tasks such as knowledge distillation, speech modeling, and time series prediction.

3 Methodology

3.1 Notations

Let the student and teacher models produce token sequences $\mathbf{x} = (x_1, x_2, \ldots, x_S)$ and $\mathbf{y} =$

$(y_1, y_2, \ldots, y_T)$, with respective vocabularies of sizes V^s and V^t. The student's predictive distribution over its vocabulary is denoted as $p^s(x_i \mid x_{<i}) \in \mathbb{R}^{V^s}$, and the teacher's predictive distribution over its own vocabulary is denoted as $p^t(y_j \mid y_{<j}) \in \mathbb{R}^{V^t}$. The entropy of the student model at position i is defined as

$$H^s(x_i) = -\sum_{i=1}^{V^s} p^s(x_i \mid x_{<i}) \log p^s(x_i \mid x_{<i}), \tag{7}$$

and the entropy of the teacher model at position j is given by

$$H^t(y_j) = -\sum_{j=1}^{V^t} p^t(y_j \mid y_{<j}) \log p^t(y_j \mid y_{<j}). \tag{8}$$

Additionally, let the student and teacher produce embedding sequences $E^s = (e_1^s, e_2^s, \ldots, e_S^s) \in \mathbb{R}^{S \times d_S}$ and $E^t = (e_1^t, e_2^t, \ldots, e_T^t) \in \mathbb{R}^{T \times d_T}$, as well as final hidden-state representations $H^s = (h_1^s, h_2^s, \ldots, h_S^s) \in \mathbb{R}^{S \times d_S}$ and $H^t = (h_1^t, h_2^t, \ldots, h_T^t) \in \mathbb{R}^{T \times d_T}$. Since teacher and student models typically differ in dimensionality, two lightweight linear projections $W_e^{t \to s}, W_h^{t \to s} \in \mathbb{R}^{d_S \times d_T}$ are introduced to map teacher representations into the student space:

$$\tilde{E}^t = E^t (W_e^{t \to s})^\top, \tag{9}$$

$$\tilde{H}^t = H^t (W_h^{t \to s})^\top. \tag{10}$$

Moreover, as in DSKD, the teacher to student distribution $p^{t \to s}(x_i \mid x_{<i})$ is aligned to the student distribution $p^s(x_i \mid x_{<i})$ via $D(\cdot \mid \cdot)$. Conversely, student to teacher distribution $p^{s \to t}(y_j \mid y_{<j})$ is aligned to the teacher distribution $p^t(y_j \mid y_{<j})$ via $D(\cdot \mid \cdot)$. Here $D(\cdot \mid \cdot)$ denotes KL divergence.

3.2 Weighted Dual-Space Knowledge Distillation

In sequence modeling, not all tokens contribute equally to the learning process. Certain tokens carry more informative or uncertain content, whereas others are predictable or less critical (Zhong et al., 2024; Akhauri et al., 2025). To more effectively guide the student model during knowledge distillation, it is beneficial to assign distinct weights to individual tokens, thereby emphasizing those of greater significance (Le et al., 2025). Our approach enables the learning process to concentrate on the most certain teacher token elements of the sequence, enhancing both training efficiency and overall model performance. In this work, we propose a mechanism for assigning weights to each token in both the student and teacher spaces. To ensure stability during training, we scale the token weights so that their sum equals the sequence length. This normalization makes the KD process robust to both sequence length variability and shifting entropy distributions as the model learns, avoiding the need for constant retuning of loss coefficients.

Token weight in student space. Inspired by Wang et al. (2025) and Furlanello et al. (2018), in the student space, we weight each token by combining the student's uncertainty with the teacher's confidence projected into the student vocabulary. Formally, we define peak confidence from teacher at student sequence position i as:

$$g^{t \to s}(x_i) = \max_{i \in V_s} p^{t \to s}(x_i \mid x_{<i}). \tag{11}$$

The weight $w_i^{(s)}$ of student token i is defined as:

$$w_i^{(s)} = H^s(x_i) \cdot g^{t \to s}(x_i). \tag{12}$$

By weighting each student token as Eq. 12, we consider a token to be important only when the student needs help (high uncertainty), and that token has a reliable teacher target in the student's vocabulary (high peak confidence). Consequently, the knowledge distillation loss in the student space is formulated as:

$$\widetilde{L}_{\text{KD}}^{\text{stu}} = \sum_i w_i^{(s)} D(p^{t \to s}(x_i \mid x_{<i}) \parallel p^s(x_i \mid x_{<i})), \tag{13}$$

This weighting mechanism prevent wasting capacity on tokens the student already knows (low entropy) or where the teacher is unsure (low max-probability).

Token weight in teacher space. In the teacher space, we assign token weights based on the teacher's confidence, computed as a normalized complement of the token-level entropy.

$$w_j^{(t)} = 1 - \frac{H^t(y_j)}{\log V^t} \tag{14}$$

This weighting mechanism ensures that the student primarily learns from tokens where the teacher is confident (low entropy), while ignoring uncertain

teacher tokens, thereby reducing potential noise. Consequently, the knowledge distillation loss in the teacher space is formulated as:

$$\widetilde{L}_{\text{KD}}^{\text{tea}} = \sum_j w_j^{(t)} D\big(p^t(x_j \mid x_{<j}) \,\|\, p^{s\to t}(x_j \mid x_{<j})\big) \tag{15}$$

Token alignment loss. The overall loss of token level alignment combines the KD losses in both spaces with the cross-entropy loss:

$$L_{\text{W-KD}} = \widetilde{L}_{\text{KD}}^{\text{stu}} + \widetilde{L}_{\text{KD}}^{\text{tea}} + L_{\text{CE}}^{t\to s}. \tag{16}$$

3.3 Dynamic Time Warping Alignment

We align teacher and student embeddings and hidden states in shared latent spaces via DTW loss to transfer representational and temporal knowledge across models with different architectures.

Pairwise costs. For each representation type (embedding or hidden), we measure pairwise dissimilarity between the student sequence and the projected-teacher sequence using cosine distance.* Let S and T be the student/teacher sequence lengths. The cost matrices $C \in \mathbb{R}^{S\times T}$ are

$$C\big(E^{\text{stu}}, \tilde{E}^{\text{tea}}\big) = 1 - \cos\big(E^{\text{stu}}, \tilde{E}^{\text{tea}}\big), \tag{17}$$

$$C\big(H^{\text{stu}}, \tilde{H}^{\text{tea}}\big) = 1 - \cos\big(H^{\text{stu}}, \tilde{H}^{\text{tea}}\big), \tag{18}$$

where $\cos(\cdot,\cdot)$ denotes cosine similarity applied row-wise to produce an $S \times T$ matrix.

Banding via Attention Entropy. We start from the Sakoe–Chiba (SC) idea (Sakoe and Chiba, 1978) and discourage off-diagonal warps by adding a fixed penalty to cross-cost entries outside a diagonal band. Rather than a hard mask, this soft penalty keeps the objective differentiable. The band is adapted per student position using DSKD cross-model attention matrix A between student and teacher tokens. Let $A \in \mathbb{R}^{\text{S}\times\text{T}}$ be the row-normalized cross-model attention from student tokens to teacher tokens (rows sum to 1). We apply a row-wise softmax over teacher tokens, and use that distribution to set the band. For each student index, we form a soft center as the attention-weighted average teacher index, a linear center from the proportional diagonal, and blend the two to obtain the row's band center; the band width widens with the row's attention entropy. Formally, for each row i:

$$c_i = \alpha \sum_{j=1}^{T} j A_{i,j} + (1-\alpha) i \cdot \frac{T}{S}. \tag{19}$$

The band width is then set from the attention entropy - high entropy (uncertain attention) yields a wider band, while low entropy tightens it:

$$H_i = -\sum_{j=1}^{T} A_{i,j} \log A_{i,j}, \tag{20}$$

And the adaptive width is the combination of base width b (the minimum band size even under low uncertainty) and attention entropy H_i:

$$w_i = b + \beta H_i. \tag{21}$$

The final cross-cost is penalized outside the adaptive band:

$$\tilde{C}_{i,j}^{s,t} = C_{i,j}^{s,t} + \lambda_{\text{band}} \cdot \mathbf{1}_{|j-c_i|>w_i}. \tag{22}$$

Additional details of formulation of attention matrix A are provided in Appendix C.

Sequence-level Alignment via Soft-DTW Let $s_\gamma(X,Y) \equiv \text{sDTW}_\gamma\big(\widetilde{C}(X,Y)\big)$ denote the Soft-DTW score on the cost matrix with adaptive band $\widetilde{C}(X,Y)$. Following prior work on *Soft-DTW divergence* (Cuturi and Blondel, 2017; Blondel et al., 2021), we remove self-similarity bias and enforce symmetry by

$$\text{nDTW}_\gamma(X,Y) = s_\gamma(X,Y) - \Delta_\gamma(X,Y) \tag{23}$$

$$\Delta_\gamma(X,Y) = \tfrac{1}{2}\big[s_\gamma(X,X) + s_\gamma(Y,Y)\big]. \tag{24}$$

We apply these formulas to the pairs $(X,Y) \in \{(E^{\text{stu}}, \tilde{E}^{\text{tea}}),\ (H^{\text{stu}}, \tilde{H}^{\text{tea}})\}$. This normalization makes distances comparable across sequences of different lengths or embedding magnitudes and stabilizes gradients. We sum the divergences at the hidden and embedding levels:

$$\begin{aligned} L_{\text{DTW}} &= nDTW_\gamma\big(H^{stu}, \tilde{H}^{tea}\big) \\ &\quad + nDTW_\gamma\big(E^{stu}, \tilde{E}^{tea}\big) \end{aligned} \tag{25}$$

The first term promotes temporal agreement between hidden-state trajectories; the second anchors lexical correspondences via token embeddings. Together, they encourage the student to absorb both contextual (semantic) and lexical structure under tokenizer and length mismatches.

*We share the projector for hidden-state representations with the projector used in DSKD and introduce an independent projector for embeddings.

3.4 Overall Loss

The final training objective is formulated as a weighted combination of complementary components:

$$L = \lambda_{\text{CE}} L_{\text{CE}} + \lambda_{\text{W-KD}} L_{\text{W-KD}} + \lambda_{\text{DTW}} L_{\text{DTW}}$$

Here, $\mathcal{L}_{\text{CE}}$ denotes the standard cross-entropy loss for supervised learning, $\mathcal{L}_{\text{W-KD}}$ represents the knowledge distillation objective that transfers soft distributions from the teacher, and $\mathcal{L}_{\text{DTW}}$ captures temporal alignment between student and teacher via dynamic time warping. The coefficients $\lambda_{\text{CE}}, \lambda_{\text{W-KD}}, \lambda_{\text{DTW}}$ balance the contribution of each term.

4 Experiments

Our experiments demonstrate that DWA-KD outperforms existing cross-tokenizer distillation methods across multiple benchmarks. Ablation studies further verify the effectiveness of entropy-based token weighting and DTW with attention-informed banding in enhancing alignment and knowledge transfer. Additional experiments are in appendix D.

4.1 Settings

Baselines and LLMs.

We evaluate our method in cross-tokenizer settings using both small- and large-scale teacher-student pairs. For the small-scale setup, we consider Qwen 1.5–1.8B as the teacher and GPT-2 120M as the student, and Qwen 1.5–1.8B as the teacher and GPT-2 340M as the student. For the large-scale setup, we use Mistral 7B as the teacher with TinyLlama 1.1B as the student, and Qwen 2.5–7B as the teacher with GPT-2 1.5B and OPT 2.7B as the student. We benchmark our approach against several cross-tokenizer knowledge distillation baselines, including ULD (Boizard et al., 2024), MinED (Wan et al., 2024), MultiLevelOT (Cui et al., 2025), and DSKD (Zhang et al., 2024) *.

Datasets

We perform experiments on multiple instruction-following datasets. Following the approach of (Gu et al., 2024), we employ the DATABRICKS-DOLLY-15K dataset for the knowledge distillation process. For evaluation purposes, beyond this primary dataset, we incorporate three instruction-tuning benchmarks as supplementary test sets for out-of-distribution assessment: Super-Natural-Instructions (**S-NI**) (Wang et al., 2022), Vicuna Evaluation (VicunaEval) (Chiang et al., 2023), and Self-Instruct (Self-Inst) (Wang et al., 2023). This comprehensive evaluation enables us to examine the model's generalization performance across a broad spectrum of instruction domains.

Training and Evaluation

For GPT-2, we fully fine-tune both teacher and student models, while for TinyLLaMA we apply LoRA fine-tuning. Specifically, the temperature τ is set to 2.0 based on validation set performance. In addition, all projectors in our approach are implemented as linear layers, which introduce only a small number of additional parameters during training. For evaluation, we generate responses from each model using five different random seeds, and the final performance is quantified by Rouge-L (Lin, 2004) between the generated outputs and the human-annotated references. Additional details on model configurations, along with the training and evaluation setup, are provided in Appendix A.

4.2 Main Results

Methods	Dolly	SelfInst	Vicuna	S-NI	Avg.
Qwen-1.5−1.8B → GPT2-120M					
Teacher	28.23	19.58	19.59	34.36	25.44
SFT	23.78	6.78	**17.04**	7.81	13.85
ULD	23.77	9.30	14.33	14.04	15.36
DSKD	24.26	10.07	15.25	17.15	16.68
MinED	24.21	10.02	14.96	16.40	16.40
MultiLevelOT	23.02	8.41	13.79	12.26	14.37
DWA-KD	**24.29**	**11.15**	15.64	**19.63**	**17.68**
Qwen-1.5−1.8B → GPT2-340M					
Teacher	28.23	19.58	19.59	34.36	25.44
SFT	23.11	9.09	14.89	13.03	15.03
ULD	23.90	9.96	15.04	16.26	16.29
DSKD	25.43	11.29	15.08	17.18	17.25
MinED	24.48	11.21	15.56	15.69	16.74
MultiLevelOT	23.95	10.21	14.80	15.87	16.21
DWA-KD	**26.64**	**12.10**	**16.99**	**21.67**	**19.35**

Table 1: Rouge-L scores (in %) averaged over five random seeds across multiple benchmarks for two teacher–student pairs.

*We exclude CDM (Chen et al., 2025) from baselines because its string-level dynamic programming requires CPU-side loops at each forward pass, making large-scale GPU training impractical.

Methods	Dolly	SelfInst	Vicuna	S-NI	Avg
Qwen2.5-7B → GPT2-1.5B					
Teacher	28.49	24.67	20.48	39.87	28.38
SFT	21.83	13.62	15.95	21.66	18.27
ULD	24.52	15.11	15.94	26.18	20.44
MinED	25.52	15.39	16.15	26.25	20.83
MultiLevelOT	24.40	14.53	15.97	23.94	19.71
DSKD	25.38	16.10	16.84	25.82	21.04
DWA-KD	**26.64**	**16.54**	**17.86**	**30.34**	**22.85**
Qwen2.5-7B-Instruct → OPT-2.7B					
Teacher	28.49	24.67	20.48	39.87	28.38
SFT	27.10	13.90	16.60	24.90	20.63
ULD	26.65	15.37	16.97	25.44	21.11
MinED	26.89	14.98	17.04	25.94	21.21
MultiLevelOT	26.76	15.51	16.56	24.84	20.92
DSKD	26.93	**16.22**	17.86	27.33	22.09
DWA-KD	**28.45**	15.65	**18.16**	**30.54**	**23.20**
Mistral-7B → TinyLLaMA-1.1B					
Teacher	31.56	25.10	20.50	36.07	28.31
SFT	23.20	14.88	16.42	27.79	20.57
ULD	25.48	17.72	17.31	32.54	23.26
DSKD	26.28	17.19	**18.74**	31.93	23.54
MinED	25.54	18.23	17.02	31.42	23.05
MultiLevelOT	24.56	15.61	16.84	27.91	21.23
DWA-KD	**26.38**	**19.62**	17.63	**36.59**	**25.06**

Table 2: Rouge-L (%) on multiple teacher–student pairs. Bold indicates the best *student* score per block.

Table 1 and table 2 summarizes the ROUGE-L performance of DWA-KD compared with standard supervised fine-tuning (SFT) and recent state-of-the-art cross-tokenizer knowledge distillation (CTKD) baselines, including ULD, DSKD, MinED, and MultiLevelOT. Evaluations were conducted across four teacher–student model pairs with distinct tokenization schemes on multiple benchmarks (Dolly, Self-Inst, Vicuna, and S-NI). Overall, DWA-KD achieves the best or competitive ROUGE-L scores across all teacher–student model pairs. For the Mistral-7B → TinyLLaMA-1.1B setting, DWA-KD attains the highest scores on four benchmarks, while remaining close to the best method on Dolly. In the Qwen-1.5B → GPT2-340M pair, it again delivers the best overall results, showing clear improvements on Vicuna and S-NI. For the Qwen2.5-7B-Instruct → GPT2-1.5B configuration, DWA-KD outperforms all baselines, being only slightly below DSKD on SelfInst. These results demonstrate that DWA-KD maintains strong and stable performance across diverse model scales and datasets. These results confirm that DWA-KD provides consistent improvements over existing CTKD approaches, validating the effectiveness of its dual-space weighting and time-warped alignment mechanisms.

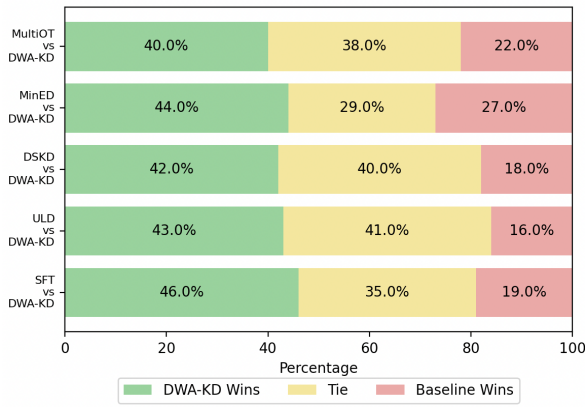

Figure 1: Win rates (%) for distilling Mistral-7B to TinyLLaMA-1.1B , evaluated by GPT-4.1 on response quality.

4.3 Evaluation via GPT-4

To complement ROUGE-L, we also use the `GPT-4.1` API as an automatic judge for pairwise comparisons between our method and each baseline. With a fixed evaluation prompt (Figure 4), the model selects the response that better follows the instruction; to reduce position bias (Zheng et al., 2023), outputs are randomly assigned to Responses A and B for every example. We report win rates over 100 pairs together with tie rates, and Figures 1 and 2 show that DWA-KD consistently outperforms all baselines in instruction-following quality.

4.4 Representation Similarity between Teacher and Student Models

To conduct an empirical study to compare token-level response patterns between teacher and student models despite differing hidden state dimensions, we adopt the approach from (Zhang et al., 2024). For a given sentence with n tokens, we construct n × n structure matrices using cosine similarity and normalized inner products of the models' hidden states. Lower values indicate that the student's representations are closer to the teacher's. We compute these distances over 100 training sentences and

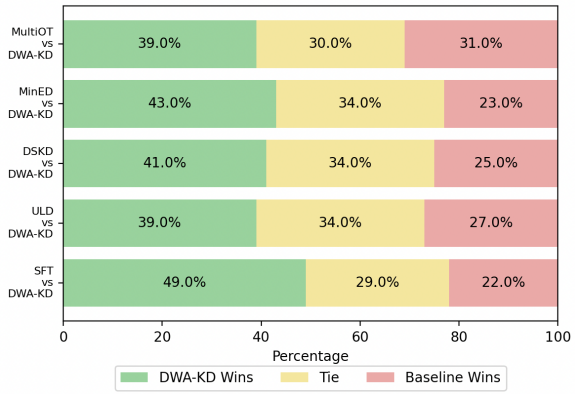

Figure 2: Win rates (%) for distilling Qwen2.5-7B-Instruct to OPT-2.7B , evaluated by GPT-4.1 on response quality.

present results as box plots for the teacher–student pair Mistral-7B → TinyLlama-1.1B. Across Figure 3, DWA-KD attains the smallest median distances, significantly smaller than others', on both structure matrices—cosine and normalized inner product—indicating that DWA-KD more faithfully preserves the teacher's token-to-token representation structure than all other methods.

4.5 Ablation study

Impact of Each Component To understand the contribution of each module we introduced in our DWA-KD framework, we conducted an ablation study comparing 4 configurations: (1) using only entropy weighting, (2) using only the DTW alignment loss, (3) using only the DTW alignment loss with banding and (4) removing the gating mechanism we introduced in (11). Table 3 shows that each component contributes positively to the overall performance. Incorporating the DTW loss consistently improves alignment quality, while entropy-based weighting enhances token-level selectivity. Adding the adaptive banding further refines sequence matching, leading to the best overall performance when all modules are combined in the full DWA-KD framework.

5 Conclusion

We propose **DWA-KD**, a knowledge distillation framework that aligns teacher and student at both the token and sequence levels. At the token level, DWA-KD extends DSKD with importance weights to emphasize informative tokens during distillation. At the sequence level, it applies DTW to both embedding and final hidden-state sequences, aligning lexical and semantic trajectories across tokenizers. Experiments show that DWA-KD improves teacher–student alignment and consistently boosts student performance across diverse evaluation settings. Future work will explore multi-teacher dis-

Methods	Dolly	SelfInst	Vicuna	S-NI	Avg.
Qwen-1.5–1.8B → GPT2-120M					
$\mathcal{L}_{DSKD}$	24.26	10.07	15.25	17.15	16.68
w/ EW	23.99	10.83	15.65	18.44	17.23
w/ DTW	23.76	**11.26**	15.25	18.93	17.30
w/ bDTW	24.05	10.77	15.75	18.42	17.25
w/o gate	23.94	10.89	**15.82**	19.36	17.50
DWA-KD	**24.29**	11.15	15.65	**19.63**	**17.68**
Mistral-7B → TinyLLaMA-1.1B					
$\mathcal{L}_{DSKD}$	26.28	17.19	**18.74**	31.93	23.54
w/ EW	26.36	18.01	17.97	34.77	24.28
w/ DTW	26.36	18.35	17.72	32.39	23.71
w/ bDTW	**26.67**	18.84	17.76	33.17	24.11
w/o gate	26.47	18.77	16.91	**37.98**	25.03
DWA-KD	26.38	**19.62**	17.63	36.59	**25.06**

Table 3: Ablation results on instruction-following benchmarks. "Avg." denotes the mean over the listed datasets. EW: entropy-only token weights; DTW: Soft-DTW loss; bDTW: Soft-DTW with attention-entropy banding; gate: applying the teacher→student max-probability gate.

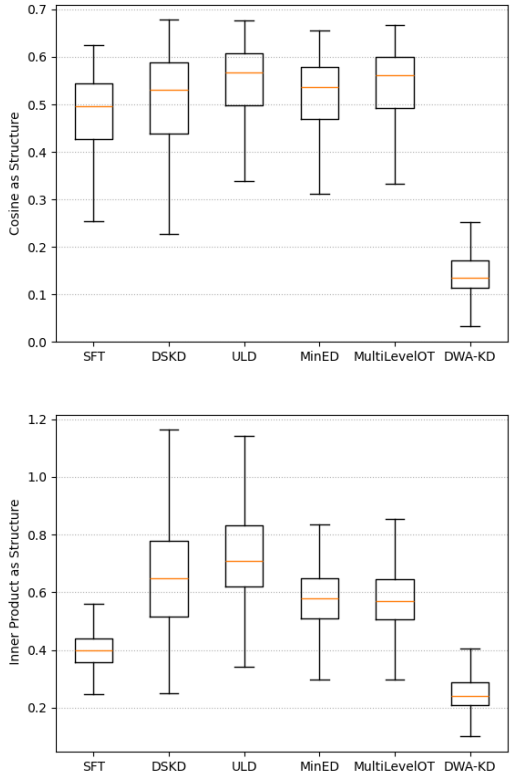

Figure 3: Teacher–student representation structure distance (cosine and normalized inner product; lower is better) for the Qwen-1.5–1.8B → GPT2-120M pair.

tillation for stronger knowledge integration.

6 Limitation

While DWA-KD effectively improves knowledge transfer across heterogeneous tokenizers, several limitations remain. First, the Soft-DTW loss introduces quadratic complexity in sequence length, which can increase computational and memory overhead. Second, the approach relies on a stable teacher–student projection, and poor initialization may weaken early-stage supervision. Moreover, our work was conducted under limited computational budgets, which constrained the scale and diversity of experimentation. We view these limitations as opportunities for further development and broader evaluation, particularly with stronger projection/calibration schemes under tight compute.

Acknowledgments

This project is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2025.16.

References

Yash Akhauri, Ahmed F AbouElhamayed, Yifei Gao, Chi-Chih Chang, Nilesh Jain, and Mohamed S. Abdelfattah. 2025. Tokenbutler: Token importance is predictable. *arXiv preprint arXiv:2503.07518*.

Mathieu Blondel, Arthur Mensch, and Jean-Philippe Vert. 2021. Differentiable dynamic programming for structured prediction and attention. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 130 of *Proceedings of Machine Learning Research*, pages 3853–3861. PMLR.

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*.

Yijie Chen, Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2025. Enhancing cross-tokenizer knowledge distillation with contextual dynamical mapping. *arXiv preprint arXiv:2502.11104*.

Wen-Lin Chiang, Zekun Li, Zhiqiang Lin, Yining Sheng, Zhe Wu, Hao Zhang, Lei Zheng, Shiyu Zhuang, Yu Zhuang, Joseph E. Gonzalez, and 1 others. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90 https://lmsys.org/blog/2023-03-30-vicuna. March 2023, 3(5).

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Xiao Cui, Mo Zhu, Yulei Qin, Liang Xie, Wengang Zhou, and Houqiang Li. 2025. Multi-level optimal transport for universal cross-tokenizer knowledge distillation on language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Oral presentation.

Marco Cuturi and Mathieu Blondel. 2017. Soft-dtw: A differentiable loss function for time-series. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 894–903, Sydney, Australia. PMLR.

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

Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 1602–1611. PMLR.

Yizhe Gu, Liwei Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

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

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*.

Jaewoo Ko, Seungwon Kim, Tian Chen, and Seong-Young Yun. 2024. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*.

Tue Le, Hoang Vuong Tran, Quyen Tran, Linh Ngo Van, Mehrtash Harandi, and Trung Le. 2025. Token-level self-play with importance-aware guidance for large language models. In *Advances in Neural Information Processing Systems*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81. Association for Computational Linguistics.

Cheng Liu, Cheng Tao, Jian Feng, and Deyi Zhao. 2022. Multigranularity structural knowledge distillation for language model compression. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pages 1001–1011.

Benjamin Minixhofer, Ivan Vulić, and Edoardo Maria Ponti. 2025. Universal cross-tokenizer distillation via approximate likelihood matching. *Preprint*, arXiv:2503.20083.

Truong Nguyen, Phi Van Dat, Ngan Nguyen, Linh Ngo Van, Trung Le, and Thanh Hong Nguyen. 2026. Ctpd: Cross tokenizer preference distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

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

Gibeom Park, Gunhee Kim, and Eunho Yang. 2021. Distilling linguistic context for language model compression. *arXiv preprint arXiv:2109.08359*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

Bing Su and Gang Hua. 2017. Order-preserving wasserstein distance for sequence matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1049–1057. IEEE.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Minh-Phuc Truong, Hai An Vu, Tu Vu, Nguyen Thi Ngoc Diep, Linh Ngo Van, Thien Huu Nguyen, and Trung Le. 2025. EMO: Embedding model distillation via intra-model relation and optimal transport alignments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7594–7606, Suzhou, China. Association for Computational Linguistics.

Hoang Tran Vuong, Tue Le, Quyen Tran, Linh Ngo Van, and Trung Le. 2026. Mcw-kd: Multi-cost wasserstein knowledge distillation for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

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*.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. 2025. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*. 25 pages, 17 figures, 2 tables.

Yizhong Wang, Yasaman Kordi, Sanyam Mishra, Alan Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Yizhong Wang, Sanyam Mishra, Pouya Alipoormolabashi, Yasaman Kordi, Amirhossein Mirzaei, Anish Arunkumar, Ashwin Ashok, Anand S. Dhanasekaran, Abhinav Naik, David Stap, and 1 others. 2022. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *arXiv preprint arXiv:2204.07705*, 2(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. Association for Computational Linguistics.

Tianyi Wu, Cheng Tao, Jie Wang, Rui Yang, Zhen Zhao, and Nelson Wong. 2024. Rethinking kullback-leibler divergence in knowledge distillation for large language models. *arXiv preprint arXiv:2404.02657*.

Jiaxing Zhang, Asim Muhamed, Arun Anantharaman, Guolin Wang, Chuan Chen, Kai Zhong, Qi Cui, Yiming Xu, Bowen Zeng, Trishul Chilimbi, and 1 others. 2023. Reaugkd: Retrieval-augmented knowledge distillation for pre-trained language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Volume 2: Short Papers*, pages 1128–1136.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. 2024. Dual-space knowledge distillation for large language models. *arXiv preprint arXiv:2406.17328*.

Xue Zhang, Songming Zhang, Yunlong Liang, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2025. A dual-space framework for general knowledge distillation of large language models. *Preprint*, arXiv:2504.11426.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. 2024. Revisiting knowledge distillation for autoregressive language models. *arXiv preprint arXiv:2402.11890*.

Weijie Zhou, Chang Xu, and Julian McAuley. 2022. Bert learns to teach: Knowledge distillation with meta learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7037–7049.

A Experimental Details

A.1 Data

All datasets in our experiments are preprocessed following Wan et al. (2024). For training, we use the Databricks-Dolly- 15k dataset, which contains human-written instruction–response pairs and consists of 11435 samples for training, 1000 for validation, and 500 for testing. To evaluate the generalization ability of our models beyond the Dolly domain, we employ 3 additional datasets:

- SelfIns (Wang et al., 2023): A user-oriented instruction-following set contains 252 samples, covering a diverse range of practical tasks such as email composition, social-media postings, entertainment prompts or programming assistance.
- VicunaEval (Chiang et al., 2023): The 80 challenging questions used in the Vicuna evaluation, spanning 9 categories such as writing, roleplay, math, coding, and knowledge.
- S-NI (Wang et al., 2022): A large benchmark of NLP tasks and their natural language instructions. The test set of SuperNatural-Instructions consisting of 9K samples ranging from 119 tasks. Following Peng et al. (2023), we split the set into 3 subsets whose ground response lengths lie in [0, 5], [6, 10] and [11, $+\infty$], and use the [11, $+\infty$] subset comprising 1694 samples.

A.2 Baselines

We compare our method against state-of-the-art knowledge distillation techniques specifically designed to address discrepancies in tokenizers and vocabularies. These baselines provide a comprehensive framework for evaluating our method's effectiveness to handle tokenizer and vocabulary disparities:

- ULD (Boizard et al., 2024): employs a Wasserstein distance-based loss to replace KL divergence, enabling distillation across different architectures and tokenizers.
- MinED (Wan et al., 2024): introduces a method based on Dynamic Time Warping (DTW) for sequence alignment, which aligns the logits between two token lists of the same sentence tokenized by different tokenizers.
- DSKD (Zhang et al., 2024): unifies the output spaces of the teacher and student distributions by projecting the output hidden states of teacher/student to the representation spaces of student/teacher, overcome limitations in similarity caused by their differing output spaces.
- MultiLevelOT (Cui et al., 2025): a cross-tokenizer knowledge distillation approach that leverages both sequence-aware token-level and sequence-level optimal transport for comprehensive distribution matching.

A.3 Training and Evaluation

Settings	**KD with Qwen1.5-1.8B**			**KD with Mistral7B**		**KD with Qwen2.5**		
	GPT2-120M	GPT2-340M	Qwen1.5	TinyLLaMA	Mistral7B	GPT2-1.5B	OPT2.7B	Qwen2.5
Epoch	20	20	15	15	15	15	15	15
LR	5×10^{-4}	5×10^{-4}	2×10^{-5}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}
Projector LR	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}
Batch Size	8	8	8	8	8	8	8	8
LR Scheduler	Cosine	Cosine	Cosine	Cosine	Cosine	Cosine	Cosine	Cosine
Fine-Tuning Method	Full	Full	Full	LoRA	LoRA	LoRA	LoRA	LoRA
LoRA Rank	N/A	N/A	N/A	256	256	256	256	256
LoRA Alpha	N/A	N/A	N/A	8	8	8	8	8
LoRA Dropout	N/A	N/A	N/A	0.1	0.1	0.1	0.1	0.1

Table 4: Detailed training configurations of KD with Qwen1.5-1.8B, Mistral7B, and Qwen2.5.

We list the detailed training configurations for all models we trained in Table 4. For evaluation, we used random sampling to decode the responses from all models. We set the decoding temperature and top_p to 1.0. We then generate the responses with 5 random seeds and report the averaged ROUGE-L score (Lin, 2004) of each seed. Besides, we also evaluate models using GPT-4.1 judgments. We randomly sample 100 instructions from the Dolly test set and generate one response per model using the same decoding settings (temperature and top-p = 1.0). The GPT-4.1 API is then prompted with a fixed evaluation template (Figure 4) to perform pairwise comparisons. To reduce position bias (Zheng et al., 2023), the two model outputs are randomly ordered as Response A and Response B for each case. We report the proportion of wins and ties across the 100 comparisons. In all experiments, we use fixed DTW banding hyperparameters without tuning; see Table 5 for values.

Symbol	Description	Value
b	Base band width (in tokens)	5
β	Entropy sensitivity	2.0
α	Blend between soft center and linear diagonal	0.7
λ	Additive cost outside the band	1.0

Table 5: Hyperparameters used for soft-DTW banding.

B Representation Similarity between Teacher and Student Model

Following Zhang et al. (2024), we empirically evaluate how well the student's token-level response patterns align with those of the teacher in a realistic KD setting. As the two models differ in hidden dimensionality, direct token-wise comparison of hidden states is infeasible. Instead, given the same input sentence, we compare the structure of token-to-token responses by constructing $n \times n$ structure matrices—computed via cosine similarity and normalized inner products between the models' output hidden states:

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

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

where $\mathcal{M}_{\text{prod}}$ and $\mathcal{M}_{\text{prod}}$ 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:

$$\mathcal{D}_{\text{cosine}} = \sum_{i=1}^{n}\sum_{j=1}^{n} \left|\mathcal{M}^t_{\text{cosine}}(i,j) - \mathcal{M}^s_{\text{cosine}}(i,j)\right|$$

$$\mathcal{D}_{\text{prod}} = \sum_{i=1}^{n}\sum_{j=1}^{n} \left|\mathcal{M}^t_{\text{prod}}(i,j) - \mathcal{M}^s_{\text{prod}}(i,j)\right|$$

Lower values indicate closer alignment between student and teacher representations. We compute these distances on 1,000 training sentences and visualize them using box plots. Two teacher–student pairs are evaluated: Mistral-7B $\rightarrow$ TinyLLaMA-1.1B. As shown in figures 3, DWA-KD achieves the lowest median L1 distances on both cosine and product-based structure matrices, showing that it better preserves the teacher's token-level structure.

C Attention matrix in DTW banding

Following Zhang et al. (2024), we use a *cross-model attention (CMA)* matrix that automatically learns the alignment between tokens of the student sequence consist of S tokens and teacher sequence consist of T

tokens. Specifically, the student's input embeddings $e^s_{1:S}$ and target embeddings $e^s_{2:S+1}$ are concatenated along the last dimension, and project them through a query projector P_q:

$$Q = P_q([e^s_{1:S}; e^s_{2:S+1}]; \theta^q_P) \in \mathbb{R}^{S\times 2D},$$

For the teacher model, the key and value vectors are obtained from its embeddings and output hidden states:

$$K = N([e^t_{1:T}; e^t_{2:T+1}]) \in \mathbb{R}^{T\times 2D},$$

$$V = P_v(N(e^t_{2:T+1}) + N(h^t_{1:T}); \theta^v_P) \in \mathbb{R}^{T\times d},$$

where $N(x) = x/\mathrm{std}(x)$ denotes normalization by the standard deviation, which promotes faster convergence. The attention matrix is then computed as:

$$A = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{2D}}\right) \in \mathbb{R}^{T\times S}.$$

The attention-based center in Eq. 19 leverages the cross-model attention matrix A to capture semantic alignment between student and teacher tokens. The term $\sum_j jA_{i,j}$ reflects a content-informed correspondence, indicating where the teacher's information most strongly aligns with the student token i. In contrast, the linear center $i \cdot \frac{T}{S}$ encodes an absolute positional prior assuming uniform progression. Blending the two allows the band center to incorporate both semantic and positional alignment cues, yielding a more adaptive and stable alignment.

D Additional Experiments

D.1 Comparing with stronger CTKD baselines

To better position DWA-KD against more recent cross-tokenizer distillation approaches suggested by reviewers, we implemented and evaluated two additional baselines: ALM (Minixhofer et al., 2025) and DSKDv2 (Zhang et al., 2025), following their official descriptions/implementations under the same training recipe used throughout the paper (same data, epochs, optimizer settings, and evaluation protocol). Across representative teacher–student pair (**Qwen-1.5-1.8B → GPT2-120M**), DWA-KD remains consistently competitive and achieves higher average performance.

Method	Dolly	SelfInst	Vicuna	S-NI	Avg.
Qwen-1.5-1.8B → GPT2-120M					
Teacher	28.23	19.58	19.59	34.36	25.44
SFT	23.78	6.78	17.04	7.81	13.85
ALM	20.86	10.65	14.76	17.76	16.01
DSKD	24.26	10.07	15.25	17.15	16.68
DWA-KD	24.29	11.15	15.64	19.63	17.68
DSKDv2	24.37	10.89	15.52	20.37	17.78
DWA-KDv2	24.66	10.96	15.21	20.71	17.89

Table 6: Comparison with DSKDv2 and ALM on a representative teacher-student pair.

D.2 Efficiency and resource overhead

Since DWA-KD includes a sequence-level Soft-DTW objective, we additionally report step time and GPU memory usage measured under the same sequence length regime as our main experiments. Table 7 shows that DWA-KD incurs only a modest overhead compared to common GPU-based CTKD baselines (DSKD, MinED, ULD) at batch size 4, while remaining substantially more efficient than CPU-based CDM in its supported batch-size-1 setting.

E Prompt for GPT-4.1 evaluation

Method	Avg Mem (GB)	Peak Mem (GB)	Step time (s)
Batch size 4			
DSKD	20.11	26.38	0.35
MinED	19.63	22.29	0.42
ULD	19.63	27.17	0.44
DWA-KD	20.17	29.92	0.45
Batch size 1			
CDM*	22.61	24.64	1.01
DWA-KD	20.16	22.36	0.30

Table 7: Efficiency measurements. *CDM uses the authors' original implementation which does not support batch sizes > 1, hence the batch-size-1 comparison reflects its practical regime.

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 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 4: Prompt for GPT-4.1 evaluation.