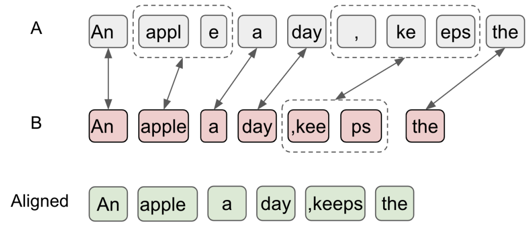
Enhancing Cross-Tokenizer Knowledge Distillation with Contextual Dynamical Mapping

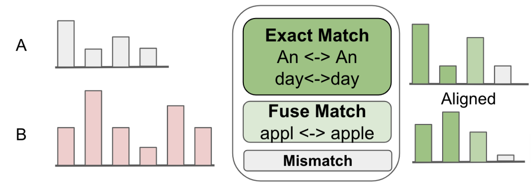
Yijie Chen[1,2*], **Yijin Liu**[3], **Fandong Meng**[3], **Yufeng Chen**[1,2], **Jinan Xu**[1,2†], **Jie Zhou**[3]
[1] Key Laboratory of Big Data & Artificial Intelligence in Transportation
(Beijing Jiaotong University), Ministry of Education
[2] School of Computer Science and Technology, Beijing Jiaotong University, Beijing, China
[3]Pattern Recognition Center, WeChat AI, Tencent Inc, China
{22120354, chenyf, jaxu}@bjtu.edu.cn
{yijinliu, fandongmeng, withtomzhou}@tencent.com

Abstract

Knowledge Distillation (KD) has emerged as a prominent technique for model compression. However, conventional KD approaches primarily focus on homogeneous architectures with identical tokenizers, constraining their applicability in cross-architecture scenarios. As for the cross-tokenizer KD, the differences in the tokenizers give rise to two fundamental challenges: (1) sequence misalignment caused by divergent tokenization strategies, and (2) mismatched vocabulary size and composition. While existing probability-matching methods attempt to address these issues, their efficacy remains limited due to suboptimal alignment in both the sequence and vocabulary aspects. To overcome these limitations, we propose Contextual Dynamic Mapping (CDM), a novel cross-tokenizer distillation framework that employs contextual information to enhance sequence alignment precision and dynamically improves vocabulary mapping. We evaluated the effectiveness of our approach across five advanced and widely-used model families (*i.e,* LLama3, Phi3, Gemma2, OPT and Qwen2), which were configured into three distinct teacher-student pairs. Our method shows significant advantages over existing cross-tokenizer distillation baselines across diverse benchmarks, including instruction-following, code generation and math. Notably, our analysis reveals that combining conventional same-tokenizer distillation and cross-tokenizer distillation through CDM yields further performance improvements.[1]



(a) First, the token sequences of teacher and student models are aligned to spans consisting of similar text.



(b) The vocabulary distribution from the teacher and student models should be aligned via token mapping.

Figure 1: The illustration of the alignment process of cross-tokenizer knowledge distillation. A and B mean the tokenizers of the student or teacher models.

1 Introduction

Knowledge distillation (KD) (Hinton, 2015; Wen et al., 2023; Gu et al., 2024; Ko et al., 2024; Guo et al., 2025) has emerged as a promising methodological framework for enhancing the performance of compact models through knowledge transfer from larger, more powerful teacher models. Nevertheless, conventional KD approaches, which aim to minimize the distribution difference between the logits of teachers and students, are still restricted by the requirement for tokenizer consistency between teachers and students. To address this limitation, Cross-Tokenizer KD (CTKD) (Fu et al., 2023; Boizard et al., 2024) has emerged as a critical research frontier. As illustrated in Figure 1, the core challenges of CTKD arise from two fundamental aspects: (1) divergent tokenization strategies induce sequence misalignment during text processing, and (2) vocabulary discrepancies create dimensional and semantic mismatches in output probability spaces. Both the sequence and vocabulary level misalignments create significant barriers to effective knowledge transfer between heteroge-

[*] Work done when Yijie was interning at Pattern Recognition Center, WeChat AI, Tencent Inc, China.
[†] Jinan Xu is the corresponding author.
[1]The code is available at https://github.com/pppa2019/ContexualDynamicMapping

Findings of the Association for Computational Linguistics: ACL 2025, pages 8005–8018
July 27 - August 1, 2025

neous architectures. Current approaches attempt to bridge these gaps through two primary strategies:

- Tokens mapping based on text character similarity (Fu et al., 2023; Wan et al., 2024a,b), which risks semantic misalignment (*e.g.,* confusing "denoted" and "devoted")
- Optimal transport methods (Boizard et al., 2024; Cui et al., 2024) that compute full distribution distances but lack explicit semantic alignment.

To quantify the discrepancies in sentence tokenization and vocabulary among mainstream large language models (LLMs) equipped with different tokenizers, we conducted a comprehensive analysis of alignment rates[2] across five mainstream LLMs. The results demonstrate the wide range of sequence alignment rates (around 30%–90%) and vocabulary alignment rates (around 10%–90%), which collectively highlight the substantial room for improvement in cross-tokenizer knowledge transfer. To further enhance the accuracy of model output alignment, we introduce Contextual Dynamic Mapping (CDM), a novel CTKD framework that introduces:

- **Sequence-level**: Entropy-weighted dynamic programming, which improves the precision of sequential token alignment by dynamically adjusting the alignment process utilizing entropy-based measure of tokens.
- **Vocabulary-level**: Context-aware candidate matching dynamically constructs semantic-preserving token mappings, achieving an improved balance between token-level precision and minimization of unmapped token pairs.

Building on this mapping framework, we futher prioritize Top-K tokens based on their contextual significance, which effectively suppresses noise while enhancing reasonable token mapping rates further.

The experiments are based on five encompassing open-source model series, and the training and evaluation contain various tasks, including instruction following, code generation, and math. The experimental results demonstrate that CDM exhibits consistent and substantial superiority over current mainstream cross-tokenizer distillation approaches among different tasks (*e.g.,* for Qwen2-0.5B model, CDM has the improvement of 4.27% on instruction following tasks, 12.19% on code tasks and 3.34% on math tasks) and even surpasses the performance of same-tokenizer distillation in some settings. Our main contributions are as follows:

[2]The detailed statistical process is shown at Section 2

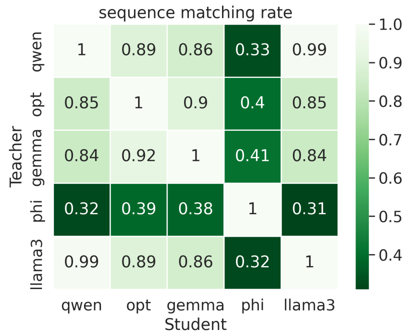


Figure 2: Matching rate of sequence alignment results.

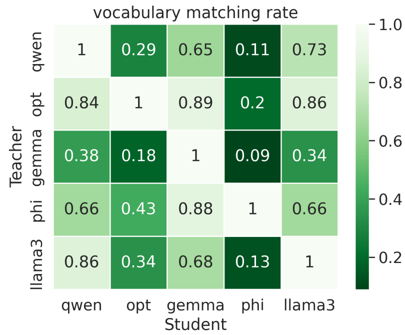


Figure 3: Matching rate of vocabulary alignment results.

- We propose the CDM method, which facilitates cross-tokenizer distillation through contextual information to improve the matching accuracy in sequence and vocabulary aspects.
- Extensive experiments are conducted on various backbone models and datasets across three tasks, and the experimental results indicate the effectiveness of CDM consistently.
- We provide a detailed analysis of the alignment rate improvement in CDM to illustrate the correlation between alignment and distillation effectiveness.
- We observe that the model performance can be further improved by combining the conventional same-tokenizer distillation and CDM.

2 Preliminary Analysis

To systematically investigate the fundamental challenges in CTKD, we analyze the alignment rates in the sequence and vocabulary levels separately. On

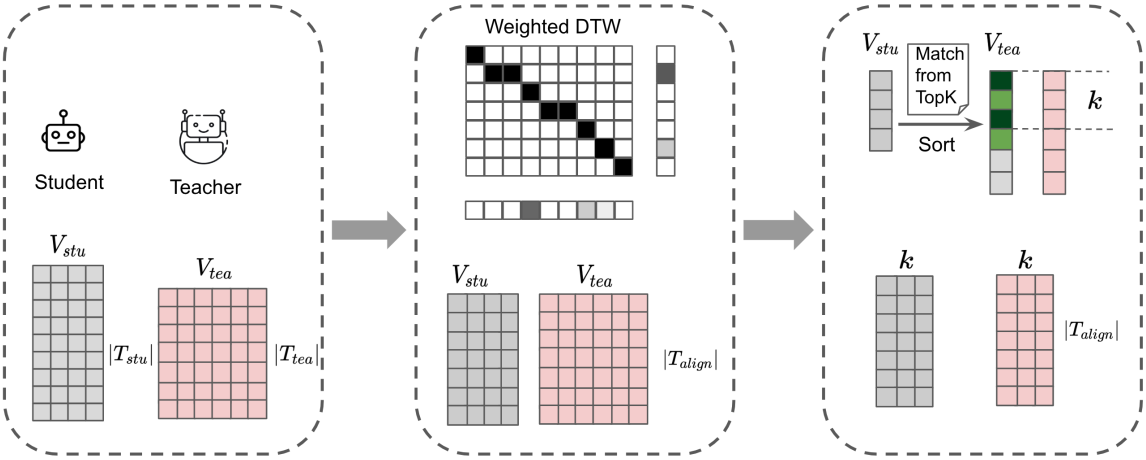


Figure 4: The architecture of CDM consists of two key components: an entropy-weighted Dynamic Time Warping (DTW) sequence alignment algorithm and a dynamic Top-K vocabulary mapping algorithm. Following the mapping procedure, the output representations from both the teacher and student models are aligned to ensure consistency in both dimensional structure and semantic space.

the sequence level, we calculate the tokenization result of the two selected tokenizers. Concretely, for a certain sentence, we statistically analyze the set of tokens overlap and average on the sample number as final results. The statistic is based on 3000 examples sampled from the training corpus of Dolly-15K (Ouyang et al., 2022) On the vocabulary level, we calculate the ratio of tokens that exactly matches the two selected tokenizers.

The statistical results presented in Figure 2 and Figure 3 highlight two critical limitations: (1) vocabulary-level alignment rates, which reach as low as 9%, and (2) sequence-level alignment rates, with minimum values as low as 31%. These findings underscore the critical need for supplementary mapping mechanisms within existing methodologies. Furthermore, the intersection of these two mismatch dimensions creates compounding effects, namely, sequence alignment errors amplify vocabulary mismatches through error propagation. This issue motivates our proposed contextual dynamic mapping framework, which mitigates alignment errors by utilizing contextual information.

3 Contextual Dynamic Mapping

3.1 Formal Definition

In this section, we take subscripts stu and tea to denote the student and teacher models, respectively. The key symbols used in the algorithm are defined in Table 1. Given an input sentence, the tokenization process yields token sequences T_{stu} and T_{tea}, along with the vocabulary size $|V_{stu}|$ and $|V_{tea}|$ ($|V_{stu}| \neq |V_{tea}|$ and $|T_{stu}| \neq |T_{tea}|$). The model output logits are then expressed as $O_{stu} \in \mathbb{R}^{|V_{stu}| \times |T_{stu}|}$ and $O_{tea} \in \mathbb{R}^{|V_{tea}| \times |T_{tea}|}$.

Symbol	Description
T_{stu}, T_{tea}	Token sequences from the student and teacher models
D_{DTW}	Minimized alignment cost in DTW
$EditDistance(\cdot)$	Edit distance between two tokens
Π	Set of all possible alignment paths
$H(O)$	Position-wise entropy vector
$\phi(X)$	MinMax normalization function
$W(O)$	Alignment weight vector
C	Constant controlling the weight range

Table 1: Symbol definitions for CDM

3.2 Sequence Alignment with Contextual Weight

Fu et al. (2023) introduced a method based on Dynamic Time Warping (DTW) for sequence token alignment, later adopted in (Wan et al., 2024a,b). The DTW is a dynamic programming method that can minimize the cost D_{DTW} between the two token lists for the same sentence tokenized by different tokenizers. The algorithms minimize:

$$D_{\text{DTW}}(T_{stu}, T_{tea}) = \min_{\pi \in \Pi} \sum_{(i,j) \in \pi} cost(t_i^{stu}, t_j^{tea}) \tag{1}$$

In existing works (Fu et al., 2023; Wan et al., 2024b), the cost function of DTW $cost(\cdot)$ use edit distance $EditDistance(t_i^{stu}, t_j^{tea})$. However, this approach demonstrates suboptimal performance with occasionally generating non-sensical alignments, for the edit distance metric introducing misalignment.

To overcome these constraints, we propose an entropy-enhanced DTW to incorporate the context information into the alignment process. The entropy-weight prioritizes the more informative tokens while suppressing noisy alignments, mitigating the sequence token mismatch problem. To illustrate, consider the example in Figure 1. For the original DTW, the edit distance between *"e"* (from A, denote with *italics*) and "apple" (from B) is larger than the edit distance between *"e"* and "a"(from B), which leads to the mistaken matching between the *"e"* and "a". However, the entropy in "a" and *"e"* both have higher entropy compared with "apple" for their ambiguity, so after adding entropy-based weight the distance metric between *"e"* and "apple" will be lower, leading to a correct mapping (*"appl"*, "apple") and (*"e"*, "apple").

For a tokenizer with vocabulary size V, let m denote any model in which both can be a teacher or student, $O_m \in \mathbb{R}^{|T_m| \times |V_m|}$ represent the output logits where $|T_m|$ denotes the length of the token sequence and $o_i \in R^{|V_m|}$ presents the logits vector of the i-th vector in O_m. We first compute position-wise entropy $H \in \mathbb{R}^{|T_m|}$ through:

$$\mathrm{H}(o_i) = -\sum_{j=1}^{|V_m|} p(o_i^j) \log p(o_i^j) \tag{2}$$

Subsequently, we apply the MinMax as the normalization function $\phi(X) = \frac{x - min(x)}{max(x) - min(x)}$ and conduct a linear mapping to obtain alignment weights.

$$\mathrm{W}(O) = \lceil \mathrm{Sigmoid}(\phi(H(O))) \cdot C + C \rceil \tag{3}$$

The hyperparameter C controls the weighting range $[C, 2C]$ to ensure both flexibility and computational efficiency in the weighting process. After calculate the weight as Equation 3, let $cost(t_i^{stu}, t_j^{tea}) = w_i^{stu} \cdot w_j^{tea} \cdot EditDistance(t_i^{stu}, t_j^{tea})$ be the cost function of DTW. With weighted DTW, we obtain span-level token mapping lists T_{stu}^{seq} and T_{tea}^{seq}. The original logits O_{tea} and O_{stu} are merged according to these mapping lists using mean pooling. After that, the outputs $O_{stu}^{seq} \in \mathbb{R}^{|T_{align}| \times |V_{stu}|}$ and $O_{tea}^{seq} \in \mathbb{R}^{|T_{align}| \times |V_{tea}|}$ are obtained and have been aligned in the sequence-level.

Algorithm 1 Algorithm of token alignment

Input: $O_{stu}^{seq}, O_{tea}^{seq}, F_{EM}, \theta$
Output: $O_{stu}^{align}, O_{tea}^{align}$
1: initialize $F_{dynamic} := F_{EM}$
2: gets the tokens $T_{stu}^{topk}, T_{tea}^{topk}$ with Top-K logits in each position utilizing.
3: **for** each position $i \in [1, |T_{align}|)$ **do**
4:   **for** each $tok_a \in T_{stu}^{topk}[i]$ **do**
5:     **if** $tok_a \notin F_{EM}$ **then**
6:       $best = \emptyset; min\_dist = \infty$
7:       **for** each $tok_b \in T_{tea}^{topk}[i]$ **do**
8:         $d = dist\_func(tok_a, tok_b)$
9:         **if** $d < \theta$ **and** $d < min\_dist$ **then**
10:           $best = tok_a, min\_dist = d$
11:         **end if**
12:       **end for**
13:       **if** $best \neq \emptyset$ **then**
14:         $F_{dynamic} := F_{dynamic} \cup \{tok_b \rightarrow best\}$
15:       **end if**
16:     **end if**
17:   **end for**
18: **end for**
19: $O_{stu}^{align} = \mathrm{Mask}(F_{dynamic}(O_{tea}^{topk}))$
20: $O_{tea}^{align} = \mathrm{Mask}(O_{tea}^{topk})$
Return: $O_{stu}^{align}, O_{tea}^{align}$

3.3 Dynamic Vocabulary Mapping with Contextual Candidates

After the sequence alignment, the model logits are still not aligned on the vocabulary dimension, *i.e.,* $|V_{stu}| \neq |V_{tea}|$. CDM uses contextual information to improve the vocabulary mapping accuracy, and the core process and is described as Algorithm 1. First, tokens that can be exactly matched between the two tokenizers are stored in a mapping dictionary F_{EM} to facilitate efficient mapping operations. For tokens that remain unmapped, we introduce a dynamic mapping dictionary $F_{dynamic}$, which is initialized as a copy of F_{EM} (line 1). The process involves utilizing the aligned model logits O_{tea}^{seq} and O_{stu}^{seq}. In order to preserve the most relevant information in the context for the training instance and avoid mapping between irrelevant tokens, we select the top k logits at each position according to Equation 4.

$$\text{Top-}K(o_i) = \mathrm{argsort}\left(\sum_{j=1}^{|V|} o_i^j, \downarrow\right)[:k] \tag{4}$$

Then, Top-$K(O_{stu}^{seq})$, Top-$K(O_{tea}^{seq})$ yields both the corresponding logits value $O_{tea}^{topk} \in \mathbb{R}^{|T_{align}|\times k}, O_{stu}^{topk} \in \mathbb{R}^{|T_{align}|\times k}$ and token sets $T_{stu}^{topk}, T_{tea}^{topk}$. At each token position $i \in [0, |T_{align}|)$, there are three possible situations for the tokens during mapping:

- The tokens can be exactly matched keep the exact match mapping results(line 5)
- For tokens in $T_{tea}^{topk}[i]$ that cannot be exactly matched, search for the most similar token from the tokens in $T_{stu}^{topk}[i]$. To evaluate the similarity, we employ edit distance with length normalization with a similarity threshold θ to mitigate noisy matches (lines 6-15)
- For the tokens that cannot find a similar token to match, their corresponding logits will be masked before distillation. (lines 19-20)

During the training phase, $F_{dynamic}$ is continuously updated through this iterative process. After the alignment of student output to teacher output, the vocabulary distribution of O_{stu}^{seq} is projected via $F_{dynamic}$ to establish a shared probability space with O_{tea}^{seq}. To prevent noise from the mismatch vocabulary, we employ a masking operator Mask$(\cdot)$ that masks the logits for unmatched positions in the vocabulary dimension. This alignment mechanism yields the refined outputs O_{tea}^{align} and O_{stu}^{align}.

To enhance the alignment of crucial tokens for student modeling, we implement a reverse-direction alignment from teacher to student. This process generates a reverse mapping dictionary $F_{dynamic}^{reverse}$ through analogous computational procedures. The alignment yields outputs: $O_{tea}^{reverse} = F_{dynamic}^{reverse}(O_{stu}^{topk})$ and $O_{stu}^{reverse} = \text{Mask}(O_{stu}^{topk})$. The final representations are constructed through vector concatenation (denoted by $\oplus$), formulated as: $O_{stu}^{f} = O_{stu}^{align} \oplus O_{stu}^{reverse}$ and $O_{tea}^{f} = O_{tea}^{align} \oplus O_{tea}^{reverse}$. $O_{stu}^{f} \in \mathbb{R}^{|V_{align}|\times 2k}$ and $O_{tea}^{f} \in \mathbb{R}^{|V_{align}|\times 2k}$ have the same dimension and the same meaning correspond to concrete tokens or spans in sequence and vocabulary dimension.

3.4 Aligned Logits Distillation

After performing contextual alignment in both sequence and vocabulary dimensions, the distribution differences between the teacher and student logits are computed using the KL divergence, as shown in Equation 5:

$$L_{\text{KL}}(O_{stu}^{f}||O_{tea}^{f}) = \sum_{i=1}^{k} O_{stu}^{f}[i] \log \left(\frac{O_{stu}^{f}[i]}{O_{tea}^{f}[i]}\right) \tag{5}$$

Meanwhile, the language modeling target uses standard cross-entropy loss for a sentence $T = \{t_0, t_1, \ldots\}$ and is defined as Equantion 6.

$$L_{lm} = -\sum_{i}^{|T|} \log P(t_i|t_{i-1}, \cdots, t_1) \tag{6}$$

Integrated with the language modeling loss, weighted by α, the final objective function is formulated in a manner analogous to classical model distillation, as presented in Equation 7.

$$L = \alpha \cdot L_{\text{KL}} + (1-\alpha) \cdot L_{lm} \tag{7}$$

4 Experiments

In our experiments, we select five widely used open-source models with different architectures including Llama-3 (8B) (Dubey et al., 2024), OPT (1.3B/6.7B) (Zhang et al., 2022), Gemma-2 (2B/9B) (Team et al., 2024), Phi3 (mini-3.8B) (Abdin et al., 2024) and Qwen2 (0.5B/7B)(Yang et al., 2024). As listed in Table 2, according to our statistic results in Section 2, we select the combinations of the teacher model and student models from the pairs with relatively poor matching rates on vocabulary or sequence level.

teacher	student	SMR	VMR
Llama-3-8B	Gemma-2-2B	85.52%	67.79%
Llama-3-8B	OPT-1.3B	89.03%	34.46%
Phi3-mini-3.8B	Qwen2-0.5B	31.54%	65.65%

Table 2: The settings on teacher and student models, where the SMR means the sequence matching rate and VMR means the vocabulary matching rate.

4.1 Experiment Settings

Baseline Methods In our experiments, the primary baseline constitutes supervised fine-tuning (SFT) applied to both teacher and student models. To provide a comprehensive comparison, our baselines include the following methods for same-tokenizer model distillation (the teacher model maintains an identical architecture to the student model but with scaled-up parameters, *e.g,* Qwen2-7B serves as the teacher model for distilling knowledge into Qwen2-0.5B):

Type	Name	Number Train	Number Test
Instruction Following	Dolly	1100	500
	Self-Inst	-	242
	Vicuna	-	80
	S-NI	-	1694
	UnNI	-	1000
Code Generation	CodeM	9600	-
	HumanEval+	-	164
	MBPP	-	500
Math	GSM-8B	7473	1000

Table 3: Data statistic of training and evaluation data.

- Forward KL divergence (FKL): the standard distillation loss function. Let $p(x)$ be the distribution of the student model, and $p(s)$ be the distribution of the teacher model, then the loss function is $L_{fkl}(p(x)||q(x)) = \mathbb{E}_{q(x)}[log(\frac{q(x)}{p(x)}]$.

- Reverse KL divergence (RKL): reverse the distribution of teacher and student in KL divergence calculation.

And the following methods are for cross-tokenizer distillation:

- Unified Logits Distance (ULD) (Boizard et al., 2024): this cross-tokenizer distillation method leverages Optimal Transport to enable a unified distillation.

- MinED (Wan et al., 2024b): on the sequence level, the DTW uses edit distance as a cost function. On the vocabulary level, it uses exact match first and then uses edit distance to search the most similar token from the full vocabulary to get supplemental mappings.

Training Settings We use the learning rate 2e-5 and batch size 32 for Supervised Fine-Tuning (SFT) to train 10 epochs. For distillation methods, we follow the setting in MiniLLM (Gu et al., 2024), the details of hyperparameters are appended in Section A.1. We evaluate the methods on three types of tasks: instruction-following, code generation, and math. The training datasets include Dolly-15K (Ouyang et al., 2022) for instruction following, CodeM (Zan et al., 2024) for coding tasks, and GSM-8K for math tasks. The training is conducted on 8 Ascend 910Bs and using DeepSpeed[3] ZeRO stage 1 for model parallel.

[3]https://github.com/microsoft/DeepSpeed

Evaluation Settings For instruction-following task, we follow the existing work (Gu et al., 2024; Zhang et al., 2024) and evaluate Rouge-L (Lin, 2004) on a series of instruction-following test sets including Dolly , Self-Instruction (Wang et al., 2023) (Self-Inst), Vicuna-Evaluation (Chiang et al., 2023) (Vicuna), Super-Natrul Instructions (Wang et al., 2022) (S-NI), and Unnatural Instruction (Honovich et al., 2022) (UnNI). For code generation tasks, the test set contains HumanEval+ (Chen et al., 2021) and MBPP+ (Austin et al., 2021) using EvalPlus (Liu et al., 2024) for an evaluation with more test cases and stricter. The evaluate metric is Pass@1, meaning the ratio of generated code can pass all test cases in one shot, and the decoding setting is greedy search. For math tasks, the test set is GSM-8K (Cobbe et al., 2021). The evaluate metric is Test@1, which has the same meaning as Pass@1 for math problems, and the decoding setting is also greedy search. The data statistic of training and evaluation data can be referred to Table 3.

4.2 Main Results

We conducted the fine-tuning and distillation experiments in the three settings of teacher-student. The main experiments of instruction-following are shown in Table 4, and there are two main findings. First, among all settings, the performance of CDM is significantly higher than other cross-tokenizer distillation baselines (*e.g.*, CDM outperforms ULD by around 0.88 average Rouge-L). Second, compared with the same-tokenizer distillation methods, including FKL and RKL, CDM achieves better performance in OPT and significantly exceeds the related method of cross-tokenizer distillation[4].

For code generation and mathematical reasoning tasks, we excluded OPT models due to their inadequate pre-training in these specialized domains, which fundamentally limits knowledge distillation efficacy. Both Gemma and Qwen models achieved varying degrees of performance gains through distillation, with the CDM method consistently demonstrating the most stable and superior effectiveness among cross-tokenizer approaches. Particularly, Qwen2-0.5B delivers notable average improvements of 12.19% on code generation and 3.34% on math tasks. These consistent im-

[4]Noting that the performance between cross-tokenizer KD and the same-tokenizer KD methods cannot be compared strictly for the difference in their teacher capacity. However, this finding indicates the potential of the CDM.

Type	Setting	Dolly	Self-Inst	Vicuna	S-NI	UnNI	#Avg IF	HumanEval+	MBPP+	GSM-8K
Student Model: Gemma-2-2B										
SFT	Gemma-2-2B	25.12	14.94	16.89	25.29	30.07	22.46	21.34	21.34	29.95
	Gemma-2-9B	26.72	18.01	18.85	27.74	34.83	25.23	24.39	27.51	45.34
	Llama-3-8B	27.01	21.90	17.00	30.66	35.23	26.36	34.76	50.26	44.20
Same Tokenizer KD *(teacher: Gemma-2-9B)*	FKL	26.51	14.30	18.64	27.61	32.06	23.82	18.90	23.00	34.80
	RKL	25.26	13.80	18.64	23.70	29.79	22.24	18.90	21.42	27.37
Cross Tokenizer KD *(teacher: Llama-3-8B)*	MinED	25.83	**16.16**	16.40	25.99	28.60	22.60	20.12	**22.22**	28.43
	ULD	26.11	14.58	17.25	**27.69**	30.53	23.23	20.40	17.70	26.38
	CDM	**26.13**	14.89	**18.33**	26.40	**32.00**	**23.55**	**23.78**	21.69	**30.40**
Student Model: OPT-1.3B										
SFT	OPT-1.3B	25.48	14.26	14.81	25.88	31.93	22.47	–	–	–
	OPT-6.7B	28.40	15.71	15.82	26.87	33.56	24.07	–	–	–
	Llama-3-8B	27.01	21.90	17.00	30.66	35.23	26.36	–	–	–
Same Tokenizer KD (*OPT-6.7B*)	FKL	25.36	15.24	16.16	26.47	31.38	22.92	–	–	–
	RKL	25.03	13.24	15.42	23.86	31.27	21.77	–	–	–
Cross Tokenizer KD *(teacher: Llama-3-8B)*	MinED	25.21	12.60	15.60	24.51	30.52	21.69	–	–	–
	ULD	25.45	13.69	**15.88**	25.82	30.07	22.18	–	–	–
	CDM	**26.15**	**14.39**	15.77	**26.33**	**32.33**	**23.00**	–	–	–
Student Model: Qwen2-0.5B										
SFT	Qwen2-0.5B	24.66	15.17	15.22	30.31	35.00	24.07	15.85	22.22	27.22
	Qwen2-7B	29.07	22.69	21.42	37.31	41.04	30.31	39.02	39.42	59.14
	Phi3-mini-3.8B	29.19	25.39	21.81	37.97	41.07	31.09	51.83	48.68	64.67
Same Tokenizer KD (*Qwen2-7B*)	FKL	27.41	19.68	19.24	32.67	37.46	27.29	17.07	23.38	27.67
	RKL	26.15	16.15	16.62	30.32	37.53	25.35	20.73	22.75	26.38
Cross Tokenizer KD *(teacher: Phi3-mini-3.8B)*	MinED	25.55	16.26	15.37	30.76	35.69	24.72	17.10	22.20	24.41
	ULD	**26.43**	16.15	15.34	30.63	36.07	24.93	17.07	22.49	26.38
	CDM	25.45	**16.55**	**16.38**	**30.66**	**36.47**	**25.10**	**18.90**	**23.81**	**28.13**

Table 4: Main results of comparing CDM and the baseline models, where"#AVG IF" means the average score of the instruction-following tasks). The **blod** text means the best performance in comparable cross-tokenizer distillation settings. The table consists of three sections, each labeled with the student models in distillation experiments.

provements across three task categories substantiate CDM's effectiveness, with comprehensive analyses provided in Section 5.1.

5 Analysis

Our analysis experiments adopts Phi3-mini-3.8B and Qwen2-0.5B as a default configuration, mainly including quantitative measurements of sequence-level and vocabulary-level alignment improvements (Section 5.1), comparative analysis with sequence-level knowledge distillation (Section 5.2), and systematic exploration of dual-teacher distillation paradigms (Section 5.3). We introduce "Average IF" as a composite metric aggregating performance across five instruction-following tasks to streamline result interpretation. Moreover, the supplementary analysis, including time cost analysis (Section A.2), evaluation on general instruction-following testset (Section A.3), ablation studies (Section A.4), sensitivity analyses (Section A.5), and comparative case studies of alignment outcomes (Sections A.6-A.7), are comprehensively documented in the Appendix.

5.1 The Statistic on Alignment Rate

In this section, we analyze the extent of the sequence-level and vocabulary-level alignment improvement separately.

Sequence Level We define the sequence alignment rate through the following procedure. When aligning two sequences, adjacent tokens may merge into contiguous spans for correspondence mapping. Similar to the statistic in Section 2, the correspondent span pairs that cannot exactly match will be regarded as a mismatch. We sampled 3,000 sentences from the training set of Dolly. The alignment rate of the results in the pure edit distance cost function was 78.31%, which improved to 82.20% after adding entropy weight, demonstrating our method's efficacy. Detailed case studies supporting these findings are presented in SectionA.6.

Vocabulary Level On the vocabulary level, CDM optimizes noise suppression and coverage in vocabulary mapping at the same time. Exact Match (EM) provides unambiguous alignment but leads to limited mapping coverage, while similarity-based fuzzy matching inevitably introduces erroneous mappings that negatively impact model distillation effectiveness. On the mapping coverage, the basic EM between Qwen2 and Phi3 achieves 65% on the coverage rate, and CDM significantly improves to 87%. There are two representative offline vocabulary matching approaches, including mapping using edit distance (ED) (Wan et al., 2024b) and edit distance with prefix constrain(PrefixED) (Wu et al., 2024), and both of them achieve over 99% coverage rates. To ensure rigorous comparison with existing approaches, we conduct controlled experiments maintaining identical experimental conditions except for the vocabulary mapping strategy. As shown in Table 5, outperforms these high-coverage methods by substantial margins, confirming its superior noise suppression capability. Extended case studies with detailed pattern analyses are available in Appendixn A.7.

Setting	Average IF
ED	24.32
PrefixED	24.12
CDM	**25.10**

Table 5: Comparison between different vocabulary mapping methods.

5.2 Comparison with Sequence-Level KD

In this section, we provide a comparison between our method and sequence-level KD (Kim and Rush, 2016) (SeqKD). Unlike the probability-based methods discussed in the main experiments, SeqKD uses the generated text of the teacher model to enhance the student model's performance. In the most advanced models, SeqKD is also applied for cross-tokenizer distillation scenarios (Guo et al., 2025).[5] According to Table 6, the results demonstrate that SeqKD and SFT have close performance, which indicates the necessity of logit-based distillation.

Setting	Average IF
SFT	24.07
SeqKD	24.05
CDM	**25.10**

Table 6: The comparison between CDM and SeqKD.

[5]Our implementation employs sampling decoding (Temperature=0.2) and integrates teacher-generated data with original training data, conducting 5-epoch training to maintain equivalent training iterations.

5.3 Dual-teacher Distillation

To investigate the impact of knowledge distillation across different model architectures, we adopted an integrated approach that combines both same-tokenizer and cross-tokenizer knowledge distillation based on OPT-1.3B. Although the results in Table 7 indicate limited improvement over SFT for both FKD and CDM, a significant improvement (10.46% in instruction-following tasks) was observed when simultaneously leveraging the knowledge of two distinct teacher models through an average loss of distillation. The findings in dual-teacher settings imply that different tokenizer strategies may provide complementary information that is effectively utilized to enhance model performance.

Settings	Average IF
SFT	22.47
w/ OPT-6.7B (FKD)	22.92
w/ Llama-3-8B (CDM)	23.00
Dual Teacher	**24.82**

Table 7: The comparison for distillation using dual-teacher and single-teacher settings.

6 Related Work

In the context of cross-tokenizer distillation, several methods have been proposed to align the probability distributions of models before performing distillation. This alignment typically involves both sequence and vocabulary dimensions. Fu et al. (2023) aligns sequences through dynamic programming and aligns vocabularies through exact matching. To improve vocabulary alignment, the (Wan et al., 2024a,b) series introduced methods such as MinED and statistical matching for fuzzy matching to supplement vocabulary alignment. Despite these advancements, their effectiveness remains constrained by the prevalence of numerous mismatches. Beyond alignment strategies based on text character similarity, Boizard et al. (2024) proposed the use of optimal transport to quantify the distance between model logits. Furthermore, Cui et al. (2024) refine this approach by optimizing the cost function at both sequence and vocabulary levels, effectively integrating both local and global information to improve overall performance. Additionally, DSKD (Zhang et al., 2024) introduces a dual alignment framework that simultaneously aligns hidden states and model logits. However, both ULD and DSKD methods suffer from inefficiencies due to underutilization of the vocabulary.

7 Conclusion and Future Work

In this work, we propose Contextual Dynamic Mapping (CDM), a novel approach to CKTD. CDM enhances the CTKD by improving the alignment of model outputs on both sequence and vocabulary levels through the use of online context information. Specifically, the method incorporates entropy-based weights in the sequential alignment process and employs contextual Top-K token pairs to dynamically map vocabulary probabilities. Through extensive experiments across three groups of teacher-student configurations and three types of tasks (instruction-following, code generation and math), our method demonstrates significant advantages over existing approaches and shows further potential in dual-teacher scenarios. Furthermore, statistical analyses and case studies are presented to demonstrate how the method enhances model alignment accuracy in both sequence and vocabulary. In future work, we plan to further scale the application of CDM by assessing its performance on more diverse training datasets and larger student and teacher models to enhance scalability. In particular, the dual-teacher distillation will be extended to multi-teacher settings for further observation.

8 Limitation

Despite its effectiveness in cross-tokenizer knowledge distillation, our proposed Contextual Dynamic Mapping (CDM) method has certain limitations. First, CDM's performance depends on the quality and diversity of the training data. Although we evaluated its effectiveness across multiple tasks, its applicability to other domains or data types, such as less structured or noisy text, remains unexplored. The method's robustness to variations in data distribution and complexity requires further investigation. Second, our experiments are limited to student models ranging from 0.5B to 2B parameters and teacher models from 3.8B to 8B parameters. Due to computational constraints, we did not conduct experiments with larger-scale models using full-parameter fine-tuning, which is left for future research.

Acknowledgments

The research work described in this paper has been supported by the National Nature Science Foundation of China (No. 62376019, 62476023, 61976015, 61976016, 61876198, and 61370130), and the National Key R&D Program of China (2020AAA0108001). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions for improving this paper.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

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*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Xiao Cui, Mo Zhu, Yulei Qin, Liang Xie, Wengang Zhou, and Houqiang Li. 2024. Multi-level optimal transport for universal cross-tokenizer knowledge distillation on language models. *arXiv preprint arXiv:2412.14528*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Geoffrey Hinton. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

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

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

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2024. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024a. Knowledge fusion of large language models. In *The Twelfth International Conference on Learning Representations*.

Fanqi Wan, Longguang Zhong, Ziyi Yang, Ruijun Chen, and Xiaojun Quan. 2024b. Fusechat: Knowledge fusion of chat models. *arXiv preprint arXiv:2408.07990*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508.

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

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

Jiayi Wu, Hao Sun, Hengyi Cai, Lixin Su, Shuaiqiang Wang, Dawei Yin, Xiang Li, and Ming Gao. 2024. Cross-model control: Improving multiple large language models in one-time training. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Daoguang Zan, Ailun Yu, Wei Liu, Bo Shen, Shaoxin Lin, Yongshun Gong, Yafen Yao, Yan Liu, Bei Guan, Weihua Luo, et al. 2024. Codem: Less data yields more versatility via ability matrix. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 714–729.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. 2024. Dual-space knowledge distillation for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18164–18181.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Appendix

A.1 Details of Training Settings in Main Experiemnts

For all SFT baselines, the last checkpoint of 10 epochs is applied to be the final result. For distillation settings, three-epoch SFT is conducted first, and then seven epochs of continual distillation are applied to keep the total steps the same. The hyperparameters about distillation are as Table 8.

Notation	Value
θ	0.3
K	100
α	0.5
T	2.0
C	3

Table 8: Hyperparameters of the distillation methods.

A.2 Time Cost Analysis

In CDM, the vocabulary and sequential alignment components require additional computational overhead, with both operations exhibiting quadratic time complexity relative to sequence length ($O(n^2)$). However, since the maximum sequence length configured during training is typically set to 512-2048 tokens in fine-tuning scenarios, this design enables containment of the overall training duration within practical limits.

In Table 9, we compare the time cost of DTW and the two baseline methods:

- ULD: the distillation method only aligns on the final loss, regarded as nearly a minimal time-consuming in cross-tokenizer training.
- MinED: align both on the sequential and vocabulary dimensions.
- CDM: using contextual information to improve the sequential and vocabulary dimension.

The experiment settings in this section are distilling Qwen2-0.5B by Phi-3-mini-3.8B in the code dataset for 10 epochs, and all the hyperparameters are the same as the main experiments.

train method	time(hour:min:s)
ULD	6:16:18
MinED	7:53:04
CDM	8:31:04

Table 9: Time cost for three CTKD methods.

According to the time cost of the three methods, the time increase of CDM is around 35% for ULD and around 7% for MinED, which is acceptable for the training process.

A.3 Evaluation on General Instruction-following

To strengthen the empirical validation, we augmented our evaluation framework through rigorous experimentation on IFEval(Zhou et al., 2023), a standardized benchmark for assessing instruction-following capabilities. Under the trained settings where Gemma2-2B is employed as the student model and Llama-3-8B as the teacher, with the Dolly training set, we evaluate the IFEval performance for the models trained by different methods in Table 10. The experimental results demonstrate that CDM significantly improves over the baseline methods. Specifically, CDM outperforms conventional distillation approaches by around 2%-7% in the average of accuracies under IFEval's multi-dimensional criteria, confirming the superior capability of CDM.

A.4 Ablation Study

To verify the effectiveness of the modules of the method independently, we conducted ablation studies comparing three configurations: (1) removing sequence-level dynamic alignment, (2) using exact match directly and disabling dynamic vocabulary matching and (3) eliminating the dual mapping strategy (*i.e.*, cancel the reverse process in Section 3.3, and only calculate distillation loss based on the Top-K logits of the teacher model and its correspondent logits in the student model outputs). As evidenced by Table 11, each component removal adversely affects model convergence. Specifically, disabling sequence-level alignment reduces performance by 1.23 points on average, while removing vocabulary-level matching and dual mapping cause 0.86-point and 1.25-point degradations, respectively. These results quantitatively demonstrate the necessity of simultaneous optimization across both sequence and vocabulary dimensions and dual mapping.

A.5 On the Sensitivity of Hyper-Parameters

This section systematically examines the parameter sensitivity of two critical hyperparameters in our CDM framework: the similarity threshold (θ) and k in the Top-K selection.

On Similarity Threshold The similarity threshold θ is the key to controlling the selection of candidate tokens for mapping, which can be referred to in Algorithm 1. In this section, we conduct a comprehensive sensitivity analysis by evaluating

setting	inst_level_loose_acc	inst_level_strict_acc	prompt_level_loose_acc	prompt_level_strict_acc	Average
SFT	23.74	22.66	14.23	13.49	18.53
ULD	21.82	21.22	11.46	10.91	16.35
MinED	26.98	25.54	15.71	14.60	20.71
CDM	**28.66**	**27.34**	**18.85**	**17.38**	**23.06**

Table 10: The experiment results of IFEval.

Setting	Average IF
CDM	**25.10**
- w/o Entropy-based Weight	23.87
- w/o Dynamic Vocabuaray Mapper	24.24
- w/o Dual Mapping	23.85
SFT	24.07

Table 11: Ablation study on CDM.

four representative similarity thresholds: 0.0, 0.1, 0.3, and 0.5, as detailed in Table 12. To ensure consistent evaluation across tokens of varying lengths, we employed an edit-distance score normalized by token length. This approach ensures that a threshold of 0.0 corresponds to exact string matching, while a threshold of 0.5 permits a broader range of fuzzy matches, albeit at the expense of precision. Our experimental results demonstrate that an intermediate threshold strikes a better balance between accuracy and coverage, with thresholds of 0.1 and 0.3 emerging as particularly effective in this context.

θ	Average IF
0.0	24.24
0.1	24.98
0.3	**25.10**
0.5	24.31

Table 12: Sensitivity test on hyperparameter similarity threshold, where 0.0 means exact matching, and 0.5 means a relatively low requirement on similarity.

On the selection of K for Top-K The determination of k in Top-K sampling constitutes a pivotal design consideration for Cross-Domain Mapping (CDM), as it directly influences the number and diversity of candidate tokens during vocabulary mapping. To systematically evaluate how this hyperparameter affects model convergence, we conducted controlled experiments across multiple k values. Empirical results demonstrate that while all Top-K configurations outperform supervised fine-tuning baselines, optimal model performance occurs at $k = 100$. Conversely, performance degradation emerges at both extremes: undersized candidate pools (K=50) restrict mapping flexibility through excessive token exclusion, while oversized pools (K=200) introduce noise token mappings. This non-monotonic relationship is quantitatively validated in Table 13.

k	Average IF
50	24.37
100	**25.10**
200	24.70
500	24.66

Table 13: Sensitivity test on hyperparameter Top-K

A.6 Case Study on Sequence Alignment

The following cases in Table 14 demonstrate the effectiveness of the entropy-based weight in enhancing the DTW alignment algorithm. In the original DTW alignment results, misalignments occur frequently at positions with ambiguous meanings, such as commas or short alphabet spans at the beginning of sentences. In contrast, the entropy-weighted DTW approach in our method (CDM) achieves more accurate span alignment. By assigning higher weights to these ambiguous positions, which carry more information, the underestimation of their importance is reduced, resulting in improved alignment accuracy. This case study demonstrates that context-aware weight calibration substantially improves alignment robustness for linguistically ambiguous elements.

A.7 Case Study on Vocabulary Alignment

Table 15 provides case studies demonstrating how contextual information optimizes vocabulary mapping accuracy in CTKD. Our contextual alignment mechanism successfully resolves lexical ambiguities by integrating character-similarity and semantic dependencies. In contrast, conventional edit-

id	type	content
1	A	Moon ||| Knight ||| is ||| Marvel ||| , ||| Batman ||| is ||| DC
	B	Moon ||| Knight ||| is ||| Marvel ||| , ||| Bat ||| man ||| is ||| DC
	A’	Moon ||| Knight ||| is ||| Marvel |||, ||| Batman ||| is ||| DC
	B’	Moon ||| Knight ||| is ||| Marvel |||,Bat ||| man ||| is ||| DC
	CDM A’	Moon ||| Knight ||| is ||| Marvel ||| , ||| Batman ||| is ||| DC
	CDM B’	Moon ||| Knight ||| is ||| Marvel ||| , ||| Batman ||| is ||| DC
2	A	Ant ||| -Man ||| is ||| Marvel ||| , ||| Ray ||| Palmer ||| is ||| DC
	B	Ant ||| - ||| Man ||| is ||| Marvel ||| , ||| Ray ||| Pal ||| mer ||| is ||| DC
	A’	Ant ||| -Man ||| is ||| Marvel ||| , |||Ray ||| Palmer ||| is ||| DC
	B’	Ant- ||| Man ||| is ||| Marvel ||| , |||RayPal ||| mer ||| is ||| DC
	CDM A’	Ant ||| -Man ||| is ||| Marvel ||| , ||| Ray ||| Palmer ||| is ||| DC
	CDM B’	Ant- ||| Man ||| is ||| Marvel ||| , ||| Ray ||| Palmer ||| is ||| DC
3	A	D ||| odge ||| is ||| American ||| , ||| Volkswagen ||| is ||| German
	B	D ||| odge ||| is ||| American ||| , ||| Volks ||| wagen ||| is ||| German
	A’	D ||| odge ||| is ||| American |||, ||| Volkswagen ||| is ||| German
	B’	D ||| odge ||| is ||| American |||,Volks ||| wagen ||| is ||| German
	CDM A’	D ||| odge ||| is ||| American ||| , ||| Volkswagen ||| is ||| German
	CDM B’	D ||| odge ||| is ||| American ||| , ||| Volkswagen ||| is ||| German

Table 14: Three representative examples of sequence alignment outcomes demonstrate the impact of incorporating contextual information. Noting that although CDM is also not fully correct on the second case, its misalignment parts do not affect the main token meaning.

distance approaches exhibit fundamental limitations: (1) inability to capture semantic relationships beyond surface-form similarity, and (2) lack of dynamic adaptation to contextual variations. The contrast reveals that context-agnostic methods relying solely on character-level edit operations systematically neglect higher-order semantic associations crucial for knowledge transfer.

id	type	content
1	full sentence	There isn't any one bicycle that would be ideal for all people. Bike shops have experts who can advise the right model and size for you and your main uses. You could also look at product results from online bike shops and read reviews to supplement the advice from the shop. A meetup or ride with a local cycling group would be another great source of advice and targeted knowledge for making a decision which bike is right for you.
	w/o context	"_publishers"→"gepubliceerd"
	w/ context	"_publishers"→ ∅
2	full sentence	The English army fought for King Harold Godwinson.
	w/o context	"_fights"→"weights"
	w/ context	"_fights→"fight"
3	full sentence	Championship rowing races are conducted over 2 kilometers (1.2 miles) with dedicated lanes delineated by bouys.
	w/o context	"kilom" → ".iloc";"denoted"→"_devoted"
	w/ context	"kilom"→ " kilomet"; "denoted"→"_defined"

Table 15: Three representative examples of token mapping are presented, where the prefix '_' denotes a space. The tokens are from the Top-K tokens, so they may not appear in the original full text.