CSD: COMMON-SUPPORT DISTILLATION FOR CROSS-TOKENIZER KNOWLEDGE TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-tokenizer logits distillation enables knowledge transfer across model families, allowing students to learn from stronger or more specialized teachers. However, differences in vocabulary and segmentation make teacher and student next-token distributions difficult to align. Existing logits-based methods are easy to integrate into standard distillation pipelines, but often rely on restrictive heuristics or tokenizer-pair-specific choices. They may restrict supervision to shared vocabulary subsets, require manually selecting different objectives for different tokenizer pairs, or match only the observed chunk and its complement rather than individual alternative continuations. We introduce Common-Support Distillation (CSD), a unified logits-based framework that constructs a shared support between the teacher and student at each aligned input chunk and matches the induced distributions with a single objective. For one-to-one alignments, CSD uses explicit common-token pairs as the shared support. For other alignments, it constructs a virtual support whose probabilities are computed using the autoregressive chain rule. A residual event captures probability mass outside the explicit support, ensuring that both induced distributions remain normalized. Across four teacher–student settings comprising Llama-3.2-1B and Gemma3-1B-PT students with Qwen and Phi teachers, CSD consistently achieves the highest average performance among the evaluated cross-tokenizer methods, outperforming the strongest alternative by up to 1.8 points and surpassing both continued pretraining and same-tokenizer distillation. For Llama-3.1-8B, CSD improves average accuracy over the strongest evaluated baseline by up to 4.2 points for the base student and 1.6 points for the instruction-tuned student.

1 INTRODUCTION

Knowledge distillation (KD) transfers knowledge from a teacher to a smaller student, typically by matching their predictive distributions (Hinton et al., 2015). In language models, standard token-level KD assumes a shared vocabulary and aligned prediction positions. When teacher and student use different tokenizers, the same text is segmented differently (requires token level alignment), and their distributions assign probability mass to different token outcomes. These mismatches prevent direct token-level distribution matching and restrict the choice of teachers.

We tackle the issue of logits distillation between models of different families and different tokenizers. Existing approaches differ in how they handle vocabulary and segmentation mismatches. GOLD (Patiño et al., 2025) directly matches teacher and student probabilities for tokens with identical strings in both vocabularies. For tokens without such a match, it sorts each model's remaining probabilities and compares them by rank, ignoring token identity. X-Token identifies two limitations of this partition: loss of task-critical token identity and suppressive gradients on unmatched tokens (Sreenivas et al., 2026). Its P-KL and H-KL modes use projection and relaxed hybrid matching, respectively. However, selecting between them relies on a manual coverage audit for each tokenizer pair. Approximate Likelihood Matching (ALM) provides a uniform formulation across tokenizer pairs by matching the probability assigned to each observed aligned chunk and aggregating all remaining probability mass into a single residual event (Minixhofer et al., 2025). While broadly applicable, this aggregation discards fine-grained dark knowledge about how the teacher distributes probability among alternative continuations.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We introduce Common-Support Distillation (CSD), which makes teacher and student predictions comparable by organizing them over a shared support of text-equivalent continuation events. We first align token chunks that decode to the same observed text span. For one-to-one chunks, the support consists of common-token pairs from the two vocabularies. For one-to-many or many-to-one chunks, each event pairs a single token from one model with a token sequence from the other. For example, given the observed alignment (`120` $\leftrightarrow$ `[1][2][0]`), fixing the prefix `[1][2]` and varying the final token yields `120` $\leftrightarrow$ `[1][2][0]`,...,`129` $\leftrightarrow$ `[1][2][9]`. A sequence event is assigned the product of its constituent conditional token probabilities. For accepted many-to-many chunks, the observed alignment forms the sole explicit event. Finally, a residual event captures probability mass outside the selected support, yielding normalized distributions that CSD matches with a single objective across all chunk types and tokenizer pairs.

Our contributions are summarized as follows:

- **A uniform cross-tokenizer objective.** CSD matches teacher and student distributions over a shared support of paired token sequences plus a residual event. The same distribution-matching objective applies across chunk shapes and tokenizer pairs without pair-specific loss selection.
- **Consistent gains across model families.** Across four settings with 1B Llama and Gemma students and Qwen and Phi teachers, CSD achieves the highest five-task average, outperforming X-Token by 0.66–5.38 points and ALM by 1.33–3.30 points, while also surpassing continued pretraining and same-tokenizer KD.
- **Efficient top-k distillation.** In the Qwen–Llama setting, using only the top 16 or 128 teacher tokens reduces the five-task average by just 1.74% or 0.74% relative to full logits, substantially lowering training-step time with minimal accuracy loss.
- **Gains at larger scales and training stages.** For Llama-3.1-8B base students, CSD outperforms continued pretraining by 2.17–4.43 points and ALM by 2.61–4.23 points across two teachers. For an instruction-tuned student, it improves over continued training and ALM by 3.90 and 1.60 points.

2 Related Work

Existing cross-tokenizer distillation methods differ in how they make teacher and student predictions comparable. We group these approaches into representation-based methods and logits-based methods.

Representation-based methods. These methods access hidden states to align different output spaces. Dual-Space Knowledge Distillation (DSKD) learns projections between teacher and student representations and uses cross-model attention to align sequences before comparing predictions (Zhang et al., 2024). Byte-Level Distillation (BLD) attaches a byte-level prediction head to student hidden states and guides it with teacher probabilities converted to bytes (Singh et al., 2026).

Logits-based methods. These methods derive supervision from existing output logits, using tokenizer information for alignment. Several approaches transfer probabilities through token correspondences: MinED uses edit-distance alignment (Wan et al., 2024); Contextual Dynamic Mapping (CDM) refines sequence and vocabulary mappings using contextual predictions (Chen et al., 2025); and SeDi combines graph-based token-span alignment, vocabulary re-encoding, and entropy alignment (Wang et al., 2026). Other approaches change the quantities being compared. Universal Logit Distillation (ULD) matches sorted probability vectors, discarding token identity (Boizard et al., 2024). GOLD utilizes a hybrid loss formulation where KL is used for common tokens in both vocabularies and ULD loss for tokens without such a match (Patiño et al., 2025). X-Token addresses limitations of GOLD through two loss modes: projection (P-KL) and relaxed hybrid (H-KL) matching. However, it requires a manual coverage audit to decide the loss mode for each tokenizer pair (Sreenivas et al., 2026). Approximate Likelihood Matching (ALM) instead uses a common chunk-level formulation across tokenizer pairs, matching distributions over the observed aligned chunk and its complement (Minixhofer et al., 2025). However, it does not individually match the probabilities of alternative tokens or token sequences.

CSD combines a common objective across tokenizer pairs with finer-grained matching. It constructs normalized distributions over a common support and retains the remaining probability mass in a residual event, without pair-specific loss-mode selection, hidden-state mappings, or auxiliary prediction heads.

3 COMMON-SUPPORT DISTILLATION

CSD matches teacher and student predictions of corresponding textual outcomes, conditioned on the same preceding text. It comprises three stages. First, sequence alignment identifies paired token ranges covering the same substring using character offsets and decoded-text checks. Second, common-support construction maps each chunk's predictions onto shared events defined by common-token pairs or token sequences representing the same text (Figure 1). Event probabilities are computed directly from token predictions or through the autoregressive chain rule. A residual event retains all remaining mass, ensuring normalization. Finally, distribution matching applies a uniform objective across chunk shapes and teacher–student tokenizer pairs. Algorithm 1 in Appendix C summarizes the computation of the CSD training objective for one batch.

3.1 SEQUENCE ALIGNMENT

Token-level predictions. Let a text sequence x be tokenized by the student and teacher as $\mathbf{s} = (s_1, \ldots, s_{n_S})$, $\mathbf{t} = (t_1, \ldots, t_{n_T})$, with vocabularies $\mathcal{V}_S$ and $\mathcal{V}_T$, respectively. Their next-token distributions are denoted by $p_S(\cdot \mid \mathbf{s}_{<i})$ over $\mathcal{V}_S$ and $p_T(\cdot \mid \mathbf{t}_{<j})$ over $\mathcal{V}_T$. In this case, even when $i = j$, the histories may represent different text. Moreover, their distributions assign probabilities to different vocabulary tokens. Thus, direct token-level matching is not suitable here.

Aligned chunks. We first define (i, j) as a *common boundary* if the histories $\mathbf{s}_{<i}$ and $\mathbf{t}_{<j}$ cover the same prefix of the original text. Then, given two consecutive common boundaries (i, j) and $(i+M, j+N)$, with $M, N \geq 1$, we can define an *aligned chunk* $c = (\mathbf{s}^c, \mathbf{t}^c)$ covering the intervening substring, where $\mathbf{s}^c = (s_i, \ldots, s_{i+M-1}) = (s_1^c, \ldots, s_M^c)$, $\mathbf{t}^c = (t_j, \ldots, t_{j+N-1}) = (t_1^c, \ldots, t_N^c)$. The shape $M{:}N$ records the student and teacher token counts in the aligned chunk. We refer to chunks with $M = N = 1$ as *one-to-one* and to chunks with $M > 1$ or $N > 1$ as *non-one-to-one*. In cross-tokenizer distillation, the aim is to match the probabilities of corresponding student and teacher realizations within each aligned chunk.

Finding aligned chunks with character offsets. For student tokens s_r and teacher tokens t_r, their tokenizers provide character spans $[a_r^S, b_r^S)$ and $[a_r^T, b_r^T)$ in the original text. Thus, we use a two-pointer scan that starts from tokens with the same start offset and extends whichever range ends earlier until the end offsets first coincide. That is, for an aligned chunk $c = (\mathbf{s}^c, \mathbf{t}^c)$, the two token ranges share the same character span $[a_c, b_c)$, with $a_c := a_i^S = a_j^T$, $b_c := b_{i+M-1}^S = b_{j+N-1}^T$, and $a_c < b_c$. The retained set $\mathcal{C}$ excludes unmatched ranges and candidates that fail the decoded-chunk checks, which allow Unicode normalization and specified whitespace equivalences. We provide the full procedure for offset-based alignment in Appendix B.

3.2 COMMON-SUPPORT CONSTRUCTION

Alignment identifies the observed token ranges and their histories. We next define the events whose probabilities are matched within each aligned chunk. This event set forms the common support $\mathcal{E}_c$, constructed as shown in Figure 1. For one-to-one chunks, each explicit event pairs student and teacher realizations of the same decoded text. Each retained token-pair set contains no repeated student or teacher token IDs. For non-one-to-one chunks, we build virtual events whose probabilities can be computed for both models. These events, together with a residual event, define the common support for the distributions P_S^c and P_T^c compared in distillation.

3.2.1 ONE-TO-ONE CHUNKS

Common-token pairs. For one-to-one chunks (Figure 1(a)), the common support can be built by checking the token pairs in the student vocabulary $\mathcal{V}_S$ and teacher vocabulary $\mathcal{V}_T$ that decode to the same text fragment. For $u \in \mathcal{V}_S$ and $v \in \mathcal{V}_T$, we define the common-token pairs as: $\mathcal{M} = \{(u_k, v_k)\}_{k=1}^K \subseteq \mathcal{V}_S \times \mathcal{V}_T$, where K denotes the total number of valid token pairs. This relation is shared across one-to-one chunks for a tokenizer pair and includes alternatives beyond the observed tokens.

Normalized event distributions. Given the token pairs, each event e_k^c pairs student token u_k with teacher token v_k, while e_{res}^c collects unpaired tokens. The common support is defined as: $\mathcal{E}_c = \{e_1^c, \ldots, e_K^c, e_{\text{res}}^c\}$. In this event order, their distributions can be represented as $P_S^c = \left(p_S(u_1 \mid h_S^c), \ldots, p_S(u_K \mid h_S^c), 1 - \sum_{k=1}^K p_S(u_k \mid h_S^c)\right)$, and $P_T^c =$

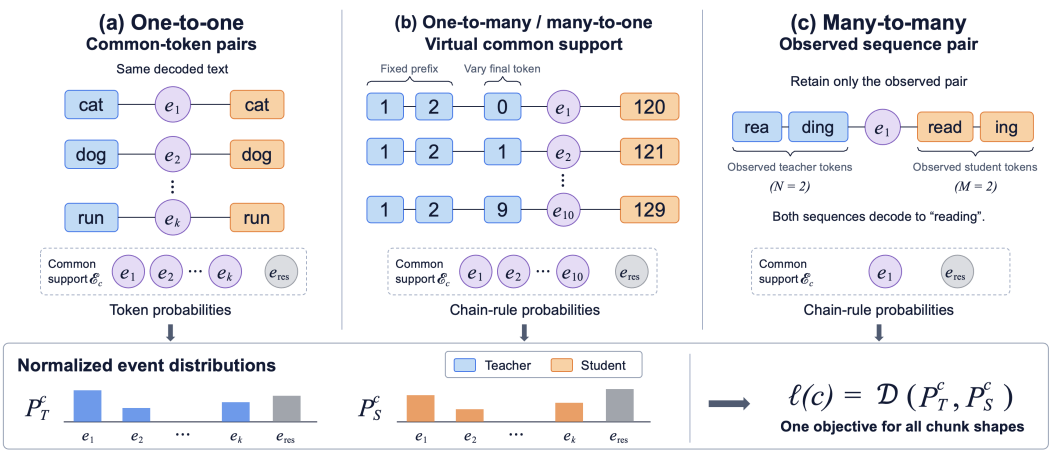

Figure 1: Common-support construction in CSD, illustrated with example tokenizations. (a) One-to-one chunks use common-token pairs. (b) In the illustrated one-to-many chunk, the teacher prefix `[1][2]` is fixed while the final token varies, matching single student tokens `120`–`129`. Swapping the model roles gives the many-to-one case. (c) Many-to-many chunks retain only the observed sequence pair. A residual event preserves all remaining mass, and the same objective matches the resulting normalized distributions. Tokenizations and probability bars are illustrative.

$\left(p_T(v_1 \mid h_T^c), \ldots, p_T(v_K \mid h_T^c), 1 - \sum_{k=1}^{K} p_T(v_k \mid h_T^c)\right)$. Here, $h_S^c = \mathbf{s}_{<i}$ and $h_T^c = \mathbf{t}_{<j}$ denote the student and teacher token histories preceding chunk c.

The residual incorporates unmatched tokens through their aggregate probability mass, avoiding potentially inaccurate token-level correspondences as in GOLD (Patiño et al., 2025). X-Token (Sreenivas et al., 2026) identifies suppression in GOLD's original common-token loss: a common-token-only loss computed from full-vocabulary softmax probabilities pushes unmatched student logits downward. For that loss, including the residual makes the gradient depend on the teacher–student residual-mass mismatch, allowing either upward or downward pressure. Thus, even a task-critical token is not always penalized for lacking a common-token counterpart. ALM uses the observed chunk as its sole explicit event (Minixhofer et al., 2025), whereas CSD expands this to include alternative common tokens (Appendix D).

3.2.2 Non-one-to-one chunks

Virtual common support. When segmentation differs, corresponding text requires multiple tokens on at least one side. We therefore construct a virtual common support from paired token sequences. For a chunk of shape $M{:}N$ with $(M, N) \neq (1, 1)$, we fix the observed prefixes $\mathbf{s}_{<M}^c$ and $\mathbf{t}_{<N}^c$ and allow only the final tokens to vary. The prefixes contain $M-1$ and $N-1$ tokens, respectively, and are empty on a single-token side ($M = 1$ or $N = 1$).

Let Dec_S and Dec_T denote the decoding functions and $\|$ token-sequence concatenation. A final-token pair $(u, v) \in \mathcal{V}_S \times \mathcal{V}_T$ defines a *paired completion* if $\mathrm{Dec}_S(\mathbf{s}_{<M}^c \,\|\, u) = \mathrm{Dec}_T(\mathbf{t}_{<N}^c \,\|\, v)$. The paired sequences must decode to the same text, which need not equal the observed chunk text; their token counts remain M and N. Let the retained valid final-token pairs form the candidate set $\mathcal{A}_c = \{(u_k^c, v_k^c)\}_{k=1}^{K_c}$. Then we can define each shared event e_k^c to pair $\mathbf{s}_{<M}^c \,\|\, u_k^c$ and $\mathbf{t}_{<N}^c \,\|\, v_k^c$. We define shared events this way, with fixed prefixes of $M-1$ and $N-1$ tokens, so that their probabilities can be computed from the observed-prefix predictions using the chain rule:

$$
\begin{aligned}
q_S^c(k) &= \left[\prod_{r=1}^{M-1} p_S(s_r^c \mid h_S^c \,\|\, \mathbf{s}_{<r}^c)\right] p_S(u_k^c \mid h_S^c \,\|\, \mathbf{s}_{<M}^c), \\
q_T^c(k) &= \left[\prod_{r=1}^{N-1} p_T(t_r^c \mid h_T^c \,\|\, \mathbf{t}_{<r}^c)\right] p_T(v_k^c \mid h_T^c \,\|\, \mathbf{t}_{<N}^c).
\end{aligned}
\tag{1}
$$

An empty product equals one. Each $q_S^c(k)$ and $q_T^c(k)$ is the probability of a specific token-sequence realization, computed by the chain rule without length normalization.

For $M = 1$ or $N = 1$ (Figure 1(b)), one side represents each candidate text with a single token. We can therefore precompute bidirectional mappings between student and teacher tokens, enabling $O(1)$ lookup per mapping entry when retrieving valid final-token pairs. In contrast, for $M, N > 1$ (Figure 1(c)), valid pairs depend on the observed prefixes on both sides, making exhaustive precomputation impractical. Constructing a larger support therefore requires additional chunk-specific matching. To avoid this cost, we retain only the observed final-token pair (s_M^c, t_N^c). This is a computational restriction on the support, not a requirement of the objective; richer many-to-many support is possible but is not constructed here.

Normalized event distributions. Each pair defines an event e_k^c with probabilities $q_S^c(k)$ and $q_T^c(k)$. In the order $(e_1^c, \ldots, e_{K_c}^c, e_{\text{res}}^c)$, $P_m^c = \left(q_m^c(1), \ldots, q_m^c(K_c), 1 - \sum_{k=1}^{K_c} q_m^c(k)\right),\ m \in \{S, T\}$. Within each model, equal-length chains with distinct final tokens define mutually exclusive events, ensuring a nonnegative residual. For accepted many-to-many chunks, we have $K_c = 1$, resulting in a binary distribution similar to ALM (Minixhofer et al., 2025). For one-to-one, one-to-many, and many-to-one with $K_c > 1$, the expanded support provides finer-grained supervision by preserving the relative probabilities of multiple alternative events. Appendix A.4 reports the empirical support-size distributions.

3.2.3 TOP-k SUPPORT SELECTION

CSD is compatible with top-k selection on either or both models, or directly over paired events. We can treat top-k selection as restricting the candidate vocabulary used for common-support construction at each prediction position. For student position i and teacher position j, define $\mathcal{V}_{S,k}^i = \text{TopK}_{u \in \mathcal{V}_S}\, p_S(u \mid \mathbf{s}_{<i}),\ \ \mathcal{V}_{T,k}^j = \text{TopK}_{v \in \mathcal{V}_T}\, p_T(v \mid \mathbf{t}_{<j})$, where TopK returns the set of k highest-probability tokens. Either model may be truncated.

The reduced vocabularies restrict which token pairs enter the common support. For one-to-one chunks, only common-token pairs whose tokens remain in the candidate vocabularies contribute. For non-one-to-one chunks, virtual support uses the reduced set of valid final-token pairs, with the observed prefixes fixed and their probabilities available. Thus, top-k selection shrinks the explicit support without changing the text-matching conditions.

Retained probabilities come from the original full-vocabulary distributions without renormalization. Excluded mass is absorbed by the existing residual event on each side, so both event distributions remain normalized and the matching objective retains the same form. Our experiments evaluate teacher-side top-k selection in Section 4.4.

3.3 DISTRIBUTION MATCHING

Both single-token and multi-token constructions yield normalized event distributions P_T^c and P_S^c, allowing us to use one matching rule. Although support sizes may differ across chunks, the teacher and student distributions refer to the same events within each chunk. A retained chunk is *accepted* for KD if its support contains at least one explicit event ($K_c \geq 1$). Chunks that do not meet this criterion contribute no KD term.

Per-chunk loss. For an accepted chunk c, we use $\ell(c) = \mathcal{D}(P_T^c, P_S^c)$, where $\mathcal{D}$ can be any suitable divergence, including forward KL (FKL), reverse KL (RKL), or Jensen–Shannon divergence (JSD). The loss compares the probabilities of all explicit events together with the residual mass, retaining aggregate supervision for outcomes outside the explicit support. The same selected $\mathcal{D}$ applies across chunk shapes and tokenizer pairs, without pair-specific loss selection.

Batch loss. Let $\mathcal{C}_{1:1}$ and $\mathcal{C}_{\neq}$ be the accepted one-to-one and non-one-to-one chunks in a batch. Each accepted chunk contributes one loss term. For a nonempty accepted set, we aggregate these terms as $\mathcal{L}_{\text{KD}}^{\beta,\mathcal{D}} = \frac{\sum_{c \in \mathcal{C}_{1:1}} \ell(c) + \beta \sum_{c \in \mathcal{C}_{\neq}} \ell(c)}{|\mathcal{C}_{1:1}| + |\mathcal{C}_{\neq}|}$. The configurable weight $\beta \geq 0$ controls the relative contribution of non-one-to-one chunks. It weights their losses without changing the underlying support or event probabilities. The denominator counts accepted chunks independently of β. If none is accepted, we set $\mathcal{L}_{\text{KD}}^{\beta,\mathcal{D}} = 0$.

Dynamic joint training. Following X-Token (Sreenivas et al., 2026), we combine the student's next-token cross-entropy $\mathcal{L}_{\text{CE}}$ with dynamically rescaled KD. Cross-entropy supervises the observed student tokens, while KD transfers the teacher's probabilities over the shared events. For $\mathcal{L}_{\text{KD}}^{\beta,\mathcal{D}} > 0$,

$\mathcal{L}_{\text{train}} = \text{sg}\Big(\mathcal{L}_{\text{CE}}/\mathcal{L}_{\text{KD}}^{\beta,\mathcal{D}}\Big)\,\mathcal{L}_{\text{KD}}^{\beta,\mathcal{D}} + \mathcal{L}_{\text{CE}}$, where sg stops gradients through the entire ratio. The scaling factor is computed from the current batch losses and treated as a constant during backpropagation. This balances scalar loss values, not necessarily gradient magnitudes. Student gradients flow through all retained chain factors and explicit and residual probabilities; the teacher remains fixed. If KD is zero, training uses $\mathcal{L}_{\text{CE}}$ alone.

4 Experiments

4.1 Experimental Setup

Models and data. Our experiments use Llama-3.2-1B (Grattafiori et al., 2024) and Gemma3-1B-PT (Gemma Team, 2025) as students and four teachers: Llama-3.2-3B and Gemma3-4B-PT are tokenizer-matched teachers for their respective students, while Qwen3-4B-Base (Yang et al., 2025) and Phi-4-mini (Abouelenin et al., 2025) are cross-tokenizer teachers. Separately, we evaluate scalability and different training stages using Llama-3.1-8B: we distill the base checkpoint from Qwen3-14B-Base and Phi-4-14B, and the instruction-tuned checkpoint from Qwen3-14B. We use the NVIDIA CLIMB corpus (Diao et al., 2025) [1] for all base-student experiments and Nemotron-Cascade-SFT (Yang et al., 2026) [2] for the instruction-tuned experiment.

Training. The 1B experiments run for $30{,}000$ steps with an effective batch size of 768 and a context length of 4096. We use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, gradient clipping at 1.0, an initial student learning rate of 5×10^{-5}, and a weight decay of 0.001. The 8B scaling experiments use the same settings and run for $2{,}000$ steps. The default CSD configuration uses $\mathcal{D} = \text{JSD}$, $\beta = 5$.

Evaluation. Our primary evaluation for base models comprises MMLU (Hendrycks et al., 2020), GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), WinoGrande (Sakaguchi et al., 2021), and MBPP (Austin et al., 2021). We use 3-shot prompting, greedy decoding for generative tasks, and the same lm-evaluation-harness configuration for every method. For the instruction-tuned student, we evaluate 0-shot GSM8K, MATH, IFEval (Zhou et al., 2023), MBPP, and MBPP+ (Liu et al., 2023), following task-specific instruction-tuned protocols held fixed across methods.

Baselines. We compare against the unmodified base model, continued pretraining (CPT) without a teacher, and same-tokenizer KD. Cross-tokenizer baselines include ALM (Minixhofer et al., 2025), ULD (Boizard et al., 2024), GOLD (Patiño et al., 2025), and X-Token (Sreenivas et al., 2026). For X-Token, P-KL and H-KL denote its projection and hybrid configurations, respectively. We select P-KL or H-KL for each teacher–student pair following the coverage analysis in the original work. The selected configuration is indicated in each table. Appendix D compares the differences between CSD and baselines. In our experiments, all cross-tokenizer methods use the same offset-based alignment, and GOLD, X-Token, and CSD use JSD in the main comparisons. For each student setting, all continued-training runs use the same data and update budget. Standalone teacher scores are reported in Table 11 for reference and are excluded from comparisons among student models.

4.2 Main Results at 1B Scale

Llama-3.2-1B student. GOLD can suffer from an *uncommon-token failure*: task-critical tokens, including numeral tokens important for mathematical reasoning, may fall outside its matched set and receive harmful supervision, as discussed by X-Token (Sreenivas et al., 2026). X-Token handles the two teacher–student pairs through different loss modes: P-KL with the Qwen teacher and H-KL with the Phi teacher, both outperforming their corresponding GOLD baselines. In contrast, CSD uses the same event-matching objective for both pairs and further improves over X-Token by 1.38 and 0.66 average points, reaching the highest five-task averages of 33.84 and 35.84, respectively. Across the two teachers, CSD also improves over ULD by 3.38–4.66 points, over ALM by 1.92–3.30 points, and over GOLD by 0.84–2.80 points. It further exceeds update-matched CPT by 3.05–5.05 points and same-tokenizer KD from Llama-3.2-3B by 2.85–4.85 points. These results show that a single cross-tokenizer objective can exploit stronger out-of-family teachers more effectively than either additional pretraining or distillation from a same-family teacher in these settings.

Gemma3-1B-PT student. Gemma3-1B-PT provides a challenging transfer setting with its SentencePiece-based tokenizer. As shown in Table 2, GOLD yields zero MBPP accuracy with both teachers and weak mathematical reasoning results, consistent with the uncommon-token failure

[1] https://huggingface.co/datasets/nvidia/Nemotron-ClimbMix

[2] https://huggingface.co/datasets/nvidia/Nemotron-Cascade-2-SFT-Data

Table 1: Main results for the Llama-3.2-1B student. † X-Token configurations are selected using tokenizer-pair coverage analysis.

Setting	Method	MMLU	WG	GSM8K	MATH	MBPP	Avg.-5
No distillation	Llama-3.2-1B (base)	32.05	61.48	5.69	5.48	26.40	26.22
	Continued pretraining	42.14	62.27	14.71	7.82	27.00	30.79
Same tokenizer	Llama-3.2-3B	43.94	62.75	12.28	7.76	28.20	30.99
Cross-tokenizer Qwen3-4B	ULD	39.19	61.96	12.74	6.40	25.60	29.18
	ALM	42.81	**62.98**	16.07	**8.74**	29.00	31.92
	GOLD	**45.71**	62.43	11.68	5.56	29.80	31.04
	X-Token (P-KL)†	45.63	62.19	15.69	7.78	31.00	32.46
	CSD (ours)	45.22	61.80	**21.30**	8.26	**32.60**	**33.84**
Cross-tokenizer Phi-4-mini	ULD	41.64	**63.38**	19.94	6.54	30.80	32.46
	ALM	44.41	62.35	18.88	8.68	28.40	32.54
	GOLD	46.42	62.98	26.46	9.34	29.80	35.00
	X-Token (H-KL)†	**46.65**	63.22	26.08	**9.36**	30.60	35.18
	CSD (ours)	46.47	62.98	**26.91**	9.22	**33.60**	**35.84**

Table 2: Transfer to the Gemma3-1B-PT student. † X-Token configurations are selected using tokenizer-pair coverage analysis.

Setting	Method	MMLU	WG	GSM8K	MATH	MBPP	Avg.-5
No distillation	Gemma3-1B-PT (base)	26.44	61.72	1.59	3.58	8.20	20.31
	Continued pretraining	36.00	58.96	10.77	5.10	15.00	25.17
Same tokenizer	Gemma3-4B-PT	38.51	61.25	9.25	5.30	18.40	26.54
Cross-tokenizer Qwen3-4B	ULD	25.99	60.85	4.47	2.96	13.20	21.49
	ALM	37.25	60.46	11.30	5.08	17.80	26.38
	GOLD	**41.23**	60.69	4.32	4.62	0.00	22.17
	X-Token (P-KL)†	39.40	**60.93**	12.59	5.10	17.20	27.04
	CSD (ours)	39.56	**60.93**	**13.04**	**5.40**	**19.60**	**27.71**
Cross-tokenizer Phi-4-mini	ULD	31.63	61.07	7.88	4.08	14.00	23.73
	ALM	38.12	61.33	12.36	5.02	19.00	27.17
	GOLD	41.21	60.93	1.36	3.92	0.00	21.48
	X-Token (P-KL)†	39.72	**62.35**	1.74	3.40	10.80	23.60
	CSD (ours)	**41.28**	61.64	**15.62**	**5.76**	**20.60**	**28.98**

discussed above (see Appendix A.2). As a result, X-Token uses P-KL for both teacher–student pairs. This configuration is effective with Qwen, reaching a five-task average of 27.04, but remains below CPT with Phi: its average is 23.60, with GSM8K and MATH scores of only 1.74 and 3.40.

In contrast, CSD uses the same common-support objective for both teachers and achieves the highest five-task averages of 27.71 with Qwen and 28.98 with Phi, exceeding X-Token by 0.67 and 5.38 points, respectively. Across the two teachers, CSD also improves over ULD by 5.25–6.22 points and over ALM by 1.33–1.81 points, while outperforming GOLD in both settings. It further exceeds CPT by 2.54–3.81 points and same-tokenizer KD from Gemma3-4B-PT by 1.17–2.44 points. Together with the Llama results, these findings show that one uniform objective achieves the best aggregate performance across all evaluated teacher–student pairs.

4.3 Support Selection for Non-One-to-One Chunks

Table 3 compares three support policies for non-one-to-one chunks: realized-only support for all chunks, the proposed shape-dependent support, and full support for all chunks. Realized-only support retains the observed sequence pair, whereas full support includes all unambiguous final-token pairs with the observed prefixes fixed. The proposed policy uses full support for one-to-many and many-to-one chunks and realized-only support for many-to-many chunks. All variants include a residual event.

Table 3: Support-selection ablation for non-one-to-one chunks with the Llama-3.2-1B student. Realized-only retains the observed sequence pair; full retains all admissible final-token pairs with fixed observed prefixes. All variants include a residual event. Avg.-5 averages MMLU, WinoGrande, GSM8K, MATH, and MBPP. **Bold** marks the best average for each teacher.

Support policy	$1{:}N$ / $M{:}1$	$M{:}N$ $(M, N > 1)$	**Qwen3-4B Avg.-5**	**Phi-4-mini Avg.-5**	**Avg. time/step (s) Qwen / Phi**
Realized-only for all	Realized-only	Realized-only	33.58	35.22	7.45/6.99
CSD (proposed)	Full	Realized-only	**33.84**	**35.84**	7.45/7.06
Full support for all	Full	Full	33.69	35.53	21.31/27.64

For each teacher, the one-to-one construction, divergence, β, student, and training settings remain unchanged across variants.

CSD achieves the highest Avg.-5 with both teachers, improving over realized-only support by 0.26 points with Qwen and 0.62 points with Phi at nearly unchanged step time. These gains suggest that matching alternative realizations in one-to-many and many-to-one chunks provides useful additional supervision. Extending full support to many-to-many chunks yields lower averages while requiring $2.86\times$ and $3.92\times$ as much time per step, respectively, compared with CSD. Thus, expanding the support does not necessarily improve accuracy in the evaluated settings. These results support using full support for one-to-many and many-to-one chunks while retaining only the observed pair for many-to-many chunks, balancing distillation quality and computational cost.

4.4 Top-k Robustness and Training Efficiency

Figure 2 compares five-task accuracy and average training-step time with the Qwen3-4B teacher and Llama-3.2-1B student. We evaluate CSD with different teacher top-k settings alongside the baselines.

Robustness to teacher-logit truncation. CSD largely preserves its accuracy under top-k truncation. Compared with the full-logit Avg.-5 of 33.84, retaining only 16 or 128 teacher tokens yields 33.25 and 33.59, respectively, corresponding to relative reductions of 1.74% and 0.74%. Increasing k to 8192 raises Avg.-5 to 33.66, only 0.18 points below full logits. These results show that a substantially reduced teacher support remains effective in this setting.

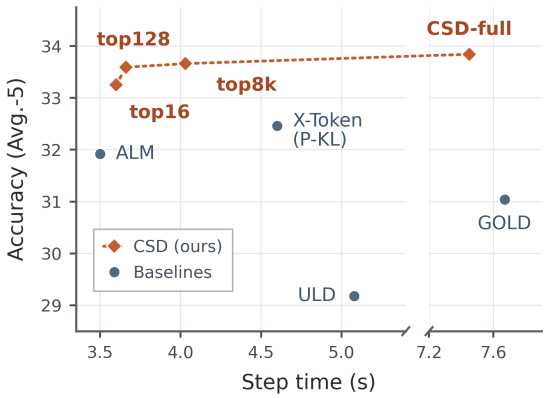

Figure 2: Accuracy–efficiency trade-off with Qwen3-4B as teacher and Llama-3.2-1B as student. Lower step time and higher Avg.-5 are better.

Accuracy–efficiency trade-off. As shown in Figure 2, CSD requires 3.66 seconds per step with $k = 128$, compared with 7.45 seconds for full logits, giving a $2.04\times$ speedup with only a 0.25-point reduction in Avg.-5. All three evaluated top-k configurations are both faster and more accurate than ULD, GOLD, and X-Token (P-KL). In particular, $k = 128$ improves over the strongest baseline (X-Token) by 1.13 average points while reducing step time by 20.4%. ALM remains slightly faster at 3.50 seconds per step, but its Avg.-5 is 1.67 points lower than CSD at $k = 128$. Thus, top-k truncation gives CSD a favorable accuracy–efficiency trade-off in the evaluated setting.

4.5 Scaling to Larger and Instruction-Tuned Students

To evaluate whether the gains extend beyond 1B students, we study both Llama-3.1-8B base and instruction-tuned checkpoints. Given the higher training cost, all continued-training runs use an update-matched budget of $2{,}000$ steps.

Base student. We distill the Llama-3.1-8B base student from Qwen3-14B-Base and Phi-4-14B. Table 4 shows that CSD improves over both update-matched CPT and ALM on all four tasks with either teacher. With Qwen, CSD reaches an Avg.-4 of 56.72, exceeding CPT and ALM by 2.17 and 2.61 points, respectively. With Phi, it reaches 58.98, with corresponding gains of 4.43 and 4.23

Table 4: Scaling results for the Llama-3.1-8B base student. Teacher scores serve as standalone references. Avg.-4 is the unweighted mean over the four displayed tasks.

Setting	Method	MMLU	GSM8K	MATH	MBPP	Avg.-4
No distillation	Llama-3.1-8B (base)	64.98	54.13	21.16	48.60	47.22
	Continued pretraining	66.19	68.39	32.42	51.20	54.55
Teachers (reference)	*Phi-4-14B*	*80.37*	*91.96*	*50.98*	*72.20*	*73.88*
	Qwen3-14B-Base	*80.71*	*86.73*	*54.84*	*74.20*	*74.12*
Cross-tokenizer	Qwen3-14B-Base, ALM	65.91	69.45	29.08	52.00	54.11
	Qwen3-14B-Base, CSD	**66.35**	**71.19**	**35.74**	**53.60**	**56.72**
	Phi-4-14B, ALM	66.12	72.48	29.58	50.80	54.75
	Phi-4-14B, CSD	**67.74**	**75.59**	**37.00**	**55.60**	**58.98**

Table 5: Results for the Llama-3.1-8B-Instruct student. IFEval-p and IFEval-i denote prompt- and instruction-level strict accuracy. Evaluation follows each task's standard instruction-tuned protocol. Avg.-6 is the unweighted mean over the six displayed metric columns.

Setting	Method	GSM8K	MATH	IFEval-p	IFEval-i	MBPP	MBPP+	Avg.-6
No distillation	Llama-3.1-8B-Instruct	75.36	46.18	74.49	82.01	64.02	52.65	65.79
	Continued training	72.48	40.72	71.72	78.30	64.20	55.90	63.89
Teacher (reference)	*Qwen3-14B*	*78.62*	*78.76*	*84.47*	*89.57*	*83.79*	*72.34*	*81.26*
Cross-tokenizer	Qwen3-14B, ALM	**79.76**	46.72	70.06	78.06	65.66	**56.89**	66.19
	Qwen3-14B, CSD	76.50	**47.76**	**76.71**	**83.45**	**65.91**	56.39	**67.79**

points. The largest gains over ALM occur on MATH: 6.66 points with Qwen and 7.42 points with Phi. These results support the effectiveness of CSD with a larger student under a limited budget.

Instruction-tuned student. We further evaluate CSD with Llama-3.1-8B-Instruct as student and Qwen3-14B as teacher. Table 5 shows that CSD achieves the highest student Avg.-6 of 67.79, outperforming the initial checkpoint, continued training, and ALM by 2.00, 3.90, and 1.60 points, respectively. It exceeds ALM on four of six metrics, including gains of 6.65 and 5.39 points on IFEval-p and IFEval-i, both of which degrade under ALM relative to the initial checkpoint.

Together, these results show that the benefits of CSD extend beyond 1B students to both base and instruction-tuned 8B models. Across the evaluated settings, CSD improves average performance over teacher-free continued training and ALM, supporting its effectiveness at different training stages.

5 Conclusion

We introduced CSD, a cross-tokenizer KD method that matches normalized teacher and student distributions over a shared support of common-token and multi-token events, with a residual event capturing the remaining probability mass. Its unified objective applies across chunk shapes and tokenizer pairs without pair-specific loss selection, matching teacher probabilities over alternative events where available.

CSD achieves the highest average score among the evaluated baselines across the four 1B teacher–student settings and outperforms continued training and ALM for both base and instruction-tuned 8B students. For the instruction-tuned student, it also improves all six reported metrics over the initial checkpoint, including instruction-following accuracy. The ablations support using full support for one-to-many and many-to-one chunks and realized-only support for many-to-many chunks, balancing distillation quality and computational cost. In the evaluated Qwen–Llama setting, teacher top-k truncation closely preserves full-logit accuracy while substantially reducing training-step time.

Our evaluation focuses primarily on English tasks, and the residual event does not preserve preferences among unmatched outcomes. Future work will explore richer common support, particularly for many-to-many chunks, task-specific weighting for β during training, and multilingual evaluation to further assess robustness across languages and tokenizer families.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods and interpret results. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, clean and reformat dataset, and support qualitative and thematic data analysis.

Additionally, we used generative AI tools to create or modify scientific figures or images, create or edit software code, draft parts of a research paper, summarize or analyse existing literature, edit a research paper to improve readability, identify relevant literature, suggest a structure for a research paper, and propose a title or keywords for a research paper.

The authors have reviewed all AI-assisted material, checking citations against the original sources and figures and quantitative claims against the experimental results. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with AI assistance.

REFERENCES

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025.

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

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

Yijie Chen, Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. Enhancing cross-tokenizer knowledge distillation with contextual dynamical mapping. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8005–8018, 2025.

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

Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan Su, Markus Kliegl, Zijia Chen, Peter Belcak, Yoshi Suhara, Hongxu Yin, et al. Nemotron-climb: Clustering-based iterative data mixture bootstrapping for language model pre-training. *arXiv preprint arXiv:2504.13161*, 2025.

Gemma Team. Gemma 3 Technical Report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

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

Carlos Miguel Patiño, Kashif Rasul, Quentin Gallouédec, Ben Burtenshaw, Sergio Paniego, Vaibhav Srivastav, Thibaud Frere, Ed Beeching, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Unlocking on-policy distillation for any model family, 2025.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Avyav Kumar Singh, Yen-Chen Wu, Alexandru Cioba, Alberto Bernacchia, and Davide Buffelli. Cross-tokenizer llm distillation through a byte-level interface. *arXiv preprint arXiv:2604.07466*, 2026.

Sharath Turuvekere Sreenivas, Adithyakrishna Venkatesh Hanasoge, Mingyu Yang, Ali Taghibakhshi, Saurav Muralidharan, Ashwath Aithal, and Pavlo Molchanov. X-token: Projection-guided cross-tokenizer knowledge distillation. *arXiv preprint arXiv:2605.21699*, 2026.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. Knowledge fusion of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 18303–18322, 2024.

Huazheng Wang, Yongcheng Jing, Haifeng Sun, Jingyu Wang, Jianxin Liao, Leszek Rutkowski, and Dacheng Tao. Bridging the tokenizer gap: Semantics and distribution-aware knowledge transfer for unbiased cross-tokenizer distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33494–33502, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, Sheng-Chieh Lin, Chankyu Lee, Yangyi Chen, Dongfu Jiang, Jiafan He, et al. Nemotron-cascade 2: Post-training llms with cascade rl and multi-domain on-policy distillation. *arXiv preprint arXiv:2603.19220*, 2026.

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

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

A TOKENIZER AND ALIGNMENT STATISTICS

We characterize vocabulary overlap, task-relevant mapping gaps, and aligned-chunk statistics across the four 1B teacher–student pairs. Pair labels use teacher→student order, whereas chunk shapes use student:teacher token counts.

A.1 TOKENIZER OVERLAP

Table 6 summarizes the common-token map for each pair. The common-pair count, $|\mathcal{M}|$, is the number of retained one-to-one token correspondences used to construct explicit events for one-to-one chunks. Because each token appears in at most one pair, this count also gives the number of covered tokens on each side. Student and teacher coverage divide this count by their respective text-distillation vocabulary sizes.

Table 6: Vocabulary overlap for the evaluated teacher–student pairs. Common pairs are retained one-to-one token correspondences. Student and teacher coverage divide the common-pair count by the respective text-distillation vocabulary size.

Teacher → student	**Student vocab. size[1]**	**Teacher vocab. size[1]**	**Common pairs**	**Student covered**	**Teacher covered**
Qwen3-4B → Llama-3.2-1B	128,256	151,669	108,581	84.66%	71.59%
Phi-4-mini → Llama-3.2-1B	128,256	200,029	102,149	79.64%	51.07%
Qwen3-4B → Gemma3-1B-PT	262,144	151,669	94,932	36.21%	62.59%
Phi-4-mini → Gemma3-1B-PT	262,144	200,029	130,704	49.86%	65.34%

[1]Vocabulary sizes count entries represented in the text-distillation mappings, excluding model-output padding. The Llama common-pair counts include EOS. Gemma additionally contains one image-only token, giving 262,145 total tokenizer entries.

Common pairs cover 79.64%–84.66% of the Llama student vocabulary, compared with 36.21%–49.86% for Gemma. These fractions reflect both the available correspondences and vocabulary size: Gemma's larger vocabulary means that lower coverage does not necessarily imply fewer common pairs.

These statistics measure vocabulary-entry coverage, not probability mass or aligned-chunk frequency. Unpaired tokens contribute through the residual event in the one-to-one construction, while virtual support for non-one-to-one chunks is constructed separately. Despite differences in vocabulary overlap, CSD uses the same distribution-matching objective for all four pairs.

A.2 TASK-RELEVANT TOKENS OUTSIDE THE COMMON MAP

Tables 7 and 8 list the three most frequent entries in each pair's exported audit example list. An *uncommon* token lies outside the common-token map used in our experiments, which differs from GOLD's native literal-string mapping. *Occurrences* counts token appearances, while *Examples* counts distinct examples containing the token. GSM8K references include reasoning and final answers; MBPP references are programs with line endings normalized to LF. Spaces, tabs, and newlines are shown as ␣, `\t`, and `\n`, respectively.

GSM8K. The examples include numerals for Qwen–Llama and Phi–Gemma, and unit or rate fragments for the other pairs. Thus, tokens relevant to mathematical reasoning can fall outside the common map, with the affected tokens depending on the tokenizer pair.

MBPP. The Gemma examples frequently involve whitespace and code syntax, whereas the Llama examples include numerals and identifier fragments. These audits illustrate task-relevant mapping gaps, but frequency alone does not establish that a token is critical or that its exclusion causes the observed accuracy differences.

Table 7: Uncommon-token examples in 1,319 GSM8K reference solutions.

Token	Occurrences	Examples
Qwen3-4B → Llama-3.2-1B		
10	1,228	359
20	1,222	334
12	909	256
Phi-4-mini → Llama-3.2-1B		
/week	40	14
/hour	28	17
Subtract	14	13
Qwen3-4B → Gemma3-1B-PT		
/day	55	21
/h	29	18
%\n	24	18
Phi-4-mini → Gemma3-1B-PT		
10	1,228	359
20	1,222	334
12	909	256

Table 8: Uncommon-token examples in 500 MBPP reference programs.

Token	Occurrences	Examples
Qwen3-4B → Llama-3.2-1B		
10	35	18
100	7	6
26	5	3
Phi-4-mini → Llama-3.2-1B		
␣heapq	21	9
rlen	11	1
␣odd	10	6
Qwen3-4B → Gemma3-1B-PT		
␣\n	908	216
):\n	431	314
\t	314	71
Phi-4-mini → Gemma3-1B-PT		
␣\n	908	216
):\n	433	314
\t	314	71

Table 9: Recorded chunk-shape frequencies (%). The final column combines all non-one-to-one chunks. Totals may differ slightly from component sums due to rounding.

Teacher → student	1:1	1:N	M:1	M:N	All non-1:1
Qwen3-4B → Llama-3.2-1B	98.17	1.61	0.16	0.06	1.83
Phi-4-mini → Llama-3.2-1B	97.05	0.62	1.70	0.63	2.95
Qwen3-4B → Gemma3-1B-PT	90.55	3.00	4.87	1.58	9.45
Phi-4-mini → Gemma3-1B-PT	89.75	2.49	6.32	1.44	10.25

A.3 Chunk-Shape Distribution

Table 9 reports recorded chunk-shape frequencies. One-to-many chunks have shape $1{:}N$ with $N > 1$, many-to-one chunks have shape $M{:}1$ with $M > 1$, and many-to-many chunks have shape $M{:}N$ with $M, N > 1$. Many-to-many chunks need not contain equal numbers of tokens on the two sides.

One-to-one chunks dominate for every pair. Non-one-to-one chunks are nevertheless more frequent with Gemma (9.45–10.25%) than with Llama (1.83–2.95%), indicating greater segmentation differences with the evaluated teachers. One-to-many and many-to-one chunks account for most mismatches; many-to-many chunks constitute at most 1.58% of all recorded chunks.

A.4 Common-Support Size Distribution

We measure explicit common-support sizes for one-to-many and many-to-one chunks using 500 samples per teacher–student pair, a context length of 4096, and no top-k truncation. For each recorded chunk, K counts deduplicated explicit events with the observed prefix and multi-token length fixed. The residual event is excluded, so the corresponding event distribution has $K + 1$ entries.

Figure 3 shows empirical cumulative distributions, normalized separately for each pair and chunk shape. Each recorded chunk occurrence has equal weight. The curves include all recorded chunks, including those with no explicit event ($K = 0$), which do not contribute to KD. Chunks with $K = 1$ are accepted and form a binary distribution consisting of one explicit event and the residual.

One-to-many supports are typically small, with median sizes ranging from 3 to 10 across the four pairs. Many-to-one supports vary more substantially: the median sizes are 8 and 6 for the Llama student with Qwen and Phi teachers, respectively, but increase to $1{,}740$ and 175 for the corresponding

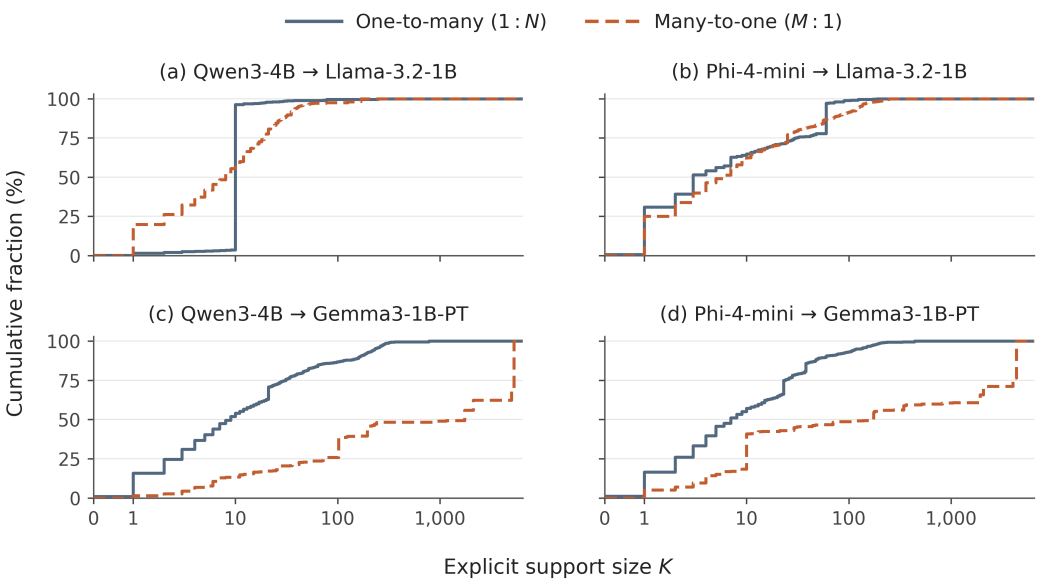

Figure 3: Common-support size distributions for one-to-many and many-to-one chunks. Each panel compares one-to-many and many-to-one chunks for a teacher–student pair. Curves show the percentage of recorded chunks with at most K explicit events, excluding the residual event. The horizontal axis is linear from 0 to 1 and logarithmic above 1.

Gemma pairs. The largest recorded supports also occur in the Gemma many-to-one setting, reaching 5,294 events with Qwen and 4,362 with Phi. Thus, although the same common-support construction applies across pairs, the number of explicit events depends strongly on both the tokenizer pair and the direction of the segmentation mismatch.

B Offset-Based Sequence Alignment

B.1 Inputs and outputs

Our alignment procedure uses the character spans returned by the student and teacher tokenizers for the same text x. Following Section 3.1, these spans are $[a_r^S, b_r^S)$ for student tokens and $[a_r^T, b_r^T)$ for teacher tokens. Offsets refer to character positions in the original text, rather than byte positions or lengths of decoded prefixes. We retain the original token indices to associate each chunk with its model predictions.

The procedure returns a set of valid text chunks $\mathcal{C}$. Each chunk records its starting indices (i, j), the indices immediately after it $(i + M, j + N)$, and its shared source interval $[a_c, b_c)$. These records identify the token ranges, their shape $M{:}N$, and their preceding histories $h_S^c = \mathbf{s}_{<i}$ and $h_T^c = \mathbf{t}_{<j}$. Unmatched ranges and candidates that fail verification are marked invalid.

We scan both token sequences in order, maintaining one pointer per sequence:

1. **Locate a shared start.** A candidate begins when the current student and teacher tokens have the same start offset. Denoting their indices by i and j, we set $a_c = a_i^S = a_j^T$. Ranges without corresponding source coverage are marked as *orphans*, and the scan continues to the next matchable ranges.
2. **Initialize the ranges.** Include one token from each side, setting $M = N = 1$. The current end offsets are b_{i+M-1}^S and b_{j+N-1}^T.
3. **Extend the earlier-ending range.** If $b_{i+M-1}^S < b_{j+N-1}^T$, include the next student token and increment M. Otherwise, if $b_{j+N-1}^T < b_{i+M-1}^S$, include the next teacher token and increment N.

Compare the end offsets after each extension; the side being extended may change as the scan proceeds.

4. **Close the candidate.** Stop at the first coincident end offset for which both ranges cover the same source interval, and set $b_c = b^S_{i+M-1} = b^T_{j+N-1}$. The index pairs (i, j) and $(i + M, j + N)$ delimit the candidate. If matching coverage cannot be established, mark the unmatched range invalid.
5. **Verify the decoded text.** Apply the checks below to the candidate token sequences $\mathbf{s}^c$ and $\mathbf{t}^c$. Add the chunk to $\mathcal{C}$ only if verification succeeds.
6. **Continue the scan.** Resume after the processed ranges. Pointers never move backward, and processed tokens are not reconsidered. Any trailing range without a counterpart is marked as an orphan.

Every candidate passed to verification satisfies

$$a^S_i = a^T_j = a_c, \qquad b^S_{i+M-1} = b^T_{j+N-1} = b_c, \qquad a_c < b_c. \tag{2}$$

The intervening token sequences are $\mathbf{s}^c$ and $\mathbf{t}^c$, as defined in Section 3.1. Their lengths determine whether the chunk is one-to-one, one-to-many, many-to-one, or many-to-many; no shape is prescribed in advance. Extending to a shared end offset accommodates different segmentations without requiring correspondence between internal token positions.

Invalid ranges are excluded from aligned KD, but their tokens remain in the original sequences and in subsequent conditioning histories.

B.2 Decoded-text verification

Matching source coverage identifies a candidate but does not automatically validate it. We verify each complete candidate range rather than repeatedly decoding the preceding histories. Each range is decoded with its own tokenizer and normalized using Unicode NFC:

$$d^c_S = \mathrm{NFC}(\mathrm{Dec}_S(\mathbf{s}^c)), \qquad d^c_T = \mathrm{NFC}(\mathrm{Dec}_T(\mathbf{t}^c)).$$

The comparison uses decoded text, not vocabulary token spellings. Decoding the complete range also avoids requiring individual tokens to represent complete text fragments.

A candidate passes verification if any of the following conditions holds:

- **Exact match:** $d^c_S = d^c_T$.
- **Whitespace-stripped match:** the whitespace-stripped forms of d^c_S and d^c_T are identical.
- **Pure-whitespace match:** both decoded strings consist entirely of whitespace.

These checks accommodate tokenizer-specific whitespace and decoding artifacts. Candidates that fail all three checks are marked invalid.

A valid text chunk therefore requires both matching source coverage and an accepted decoded-text comparison. NFC normalization and whitespace equivalence are used only for verification: they do not modify the original offsets, token ranges, or histories used to compute model probabilities.

B.3 Special tokens and the KD interface

BOS, EOS, and padding tokens are matched separately by semantic role, rather than by character spans. EOS and padding may be treated as equivalent when permitted by the tokenizer-role configuration; their token IDs need not coincide across models.

Alignment supplies the valid observed token ranges, their histories, and their chunk shapes. Common-support construction then determines the explicit events and their probabilities, as described in Section 3.2. The whitespace allowances used during alignment do not replace the text-matching conditions used to construct that support. A valid aligned chunk with no retained explicit event contributes no KD term, as defined in Section 3.3.

Algorithm 1 Common-Support Distillation: one training batch

Require: Text batch $\mathcal{X}$; student S and frozen teacher T; maps $\mathcal{M}, \mathcal{B}$
Require: Divergence $\mathcal{D}$; non-one-to-one weight $\beta \geq 0$; teacher support limit k ($k = \infty$ disables truncation)
Ensure: Joint training loss $\mathcal{L}_{\text{train}}$

Tokenize $\mathcal{X}$ with both tokenizers, retaining character offsets
Compute full-vocabulary probabilities $p_{S,r}$ and $p_{T,r}$ using the observed token histories
$\mathcal{L}_{\text{CE}} \leftarrow$ student next-token cross-entropy on the batch
$L_\Sigma \leftarrow 0$; $n_{\text{acc}} \leftarrow 0$
for each text $x \in \mathcal{X}$ with token sequences $\mathbf{s}, \mathbf{t}$ **do**
  $\mathcal{C}(x) \leftarrow$ ALIGNOFFSETS$(\mathbf{s}, \mathbf{t}, \text{offsets})$
  for each chunk $c = (i, j, M, N) \in \mathcal{C}(x)$ **do**
    $\mathbf{s}^c \leftarrow \mathbf{s}_{i:i+M-1}$; $\mathbf{t}^c \leftarrow \mathbf{t}_{j:j+N-1}$
    if $M = N = 1$ **then** ▷ One-to-one chunks
      $\mathcal{A}_c \leftarrow \mathcal{M}$
    else if $\min(M, N) = 1$ **then** ▷ One-to-many or many-to-one chunks
      $\mathcal{A}_c \leftarrow$ PAIREDCOMPLETIONS$(\mathcal{B}, \mathbf{s}^c_{<M}, \mathbf{t}^c_{<N}, M, N)$
    else ▷ Many-to-many chunks
      $\mathcal{A}_c \leftarrow \{(s^c_M, t^c_N)\}$ if $\text{Dec}_S(\mathbf{s}^c) = \text{Dec}_T(\mathbf{t}^c)$; otherwise $\mathcal{A}_c \leftarrow \varnothing$
    end if
    if $k < \infty$ **then** ▷ Optional teacher top-k selection
      $V^c_{T,k} \leftarrow \text{TopK}_v\, p_{T,j+N-1}(v)$
      $\mathcal{A}_c \leftarrow \{(u, v) \in \mathcal{A}_c : v \in V^c_{T,k}\}$
    end if
    if $\mathcal{A}_c = \varnothing$ **then** ▷ Skip chunks with no explicit event
      continue
    end if
    Enumerate $\mathcal{A}_c = \{(u_a, v_a)\}_{a=1}^{K_c}$
    if $M = N = 1$ **then** ▷ Direct next-token probabilities
      $q^c_S(a) \leftarrow p_{S,i}(u_a)$; $q^c_T(a) \leftarrow p_{T,j}(v_a)$, for $a = 1, \ldots, K_c$
    else ▷ Chain-rule probabilities
      $\alpha_S \leftarrow \prod_{r=1}^{M-1} p_{S,i+r-1}(s^c_r)$; $\alpha_T \leftarrow \prod_{r=1}^{N-1} p_{T,j+r-1}(t^c_r)$
      $q^c_S(a) \leftarrow \alpha_S p_{S,i+M-1}(u_a)$; $q^c_T(a) \leftarrow \alpha_T p_{T,j+N-1}(v_a)$, for $a = 1, \ldots, K_c$
    end if
    ▷ Collect all remaining mass in a residual event
    $P^c_m \leftarrow \big(q^c_m(1), \ldots, q^c_m(K_c), 1 - \sum_{a=1}^{K_c} q^c_m(a)\big)$, for $m \in \{S, T\}$
    $w_c \leftarrow 1$ if $M = N = 1$; otherwise $w_c \leftarrow \beta$
    $L_\Sigma \leftarrow L_\Sigma + w_c \mathcal{D}(P^c_T, P^c_S)$; $n_{\text{acc}} \leftarrow n_{\text{acc}} + 1$
  end for
end for
▷ Average over accepted chunks, with an unweighted count
$\mathcal{L}_{\text{KD}} \leftarrow L_\Sigma / \max(1, n_{\text{acc}})$
if $\mathcal{L}_{\text{KD}} > 0$ **then** ▷ Dynamically balance KD and cross-entropy
  $\mathcal{L}_{\text{train}} \leftarrow \text{sg}(\mathcal{L}_{\text{CE}} / \mathcal{L}_{\text{KD}})\, \mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{CE}}$
else
  $\mathcal{L}_{\text{train}} \leftarrow \mathcal{L}_{\text{CE}}$
end if
return $\mathcal{L}_{\text{train}}$

C CSD TRAINING ALGORITHM

Algorithm 1 summarizes the computation of the CSD training objective for one batch. For each tokenized sequence, let $p_{S,r}(u) = p_S(u \mid \mathbf{s}_{<r})$ and $p_{T,r}(v) = p_T(v \mid \mathbf{t}_{<r})$ denote the next-token probabilities at position r. The common-token map $\mathcal{M}$ and bidirectional virtual-support index $\mathcal{B}$ are precomputed once for each tokenizer pair. Each retained pair set is one-to-one in token IDs on both sides.

Alignment. ALIGNOFFSETS follows Appendix B: it identifies shared character boundaries and retains chunks that pass the decoded-text checks. Rejected ranges remain in the original sequences and conditioning histories.

Table 10: Comparison of CSD with the published formulations of the cross-tokenizer distillation baselines evaluated in this work. Pair-specific loss choice refers to selecting between loss modes, not constructing tokenizer-dependent mappings.

Method	Matching rule	Handling non-one-to-one chunks	Residual event	Pair-specific loss choice
ULD (Boizard et al., 2024)	L_1 between sorted, zero-padded probability vectors	No chunk construction; index-wise comparison up to the shorter sequence	No	No
GOLD (Patiño et al., 2025)	KL on exact token-string matches + sorted L_1 on unmatched entries	Scales the first-position probability vector by multiplying subsequent realized token probabilities	No	No
X-Token (Sreenivas et al., 2026)	P-KL: vocabulary projection + KL; H-KL: KL on relaxed token matching + sorted L_1 on unmatched entries	Scales the first-position probability vector by multiplying subsequent realized token probabilities	No	Yes
ALM (Minixhofer et al., 2025)	Binary divergence on the observed event and a residual	Chain-rule observed-only probability	Yes	No
CSD (ours)	Divergence between distributions over explicit paired events and a residual	Chain-rule common-support probabilities	Yes	No

Notes. A residual explicitly collects mass outside individually matched events; its absence does not imply unnormalized distributions. X-Token selects P-KL or H-KL through a tokenizer-pair coverage audit.

Precomputed support index. We construct vocabulary-conversion tables in both directions. For each ordinary vocabulary token, we first seek a canonicalized token-surface match; otherwise, we decode the token and retokenize the result with the other tokenizer without adding special tokens. Special tokens are matched by semantic role, including EOS. Each direction stores one deterministic conversion per successfully mapped source token.

For a stored sequence $(u_1, \ldots, u_n)$, the index key consists of its length n and token-ID prefix $(u_1, \ldots, u_{n-1})$. The associated entry pairs the source token with u_n. For one-to-many chunks, PAIREDCOMPLETIONS queries the student-to-teacher index using the observed teacher prefix and token count. For many-to-one chunks, it queries the reverse index using the observed student prefix and token count. Within each bucket, we process source tokens in increasing ID order and retain pairs without repeated token IDs on either side. The indices are built once per process and cached.

Here, full support refers to all retained candidates in the corresponding index bucket, rather than an enumeration of every possible tokenization. Given a constructed prefix key, bucket retrieval takes expected $O(1)$ time. Constructing the key depends on prefix length, while gathering and scoring K candidates requires $O(K)$ work in addition to prefix scoring.

Probability computation and optimization. Empty prefix products equal one, so the same probability computation also applies to one-to-one chunks. Top-k selection only restricts the explicit support: retained probabilities are not renormalized, and all observed prefix-token probabilities remain available even when their tokens are outside the top-k sets. The residual contains all mass outside the retained support. The denominator counts accepted chunks without weighting by β. During backpropagation, sg stops gradients through the entire scaling ratio, while student gradients flow through all retained chain factors and residual probabilities; the teacher remains fixed.

D COMPARISON WITH LOGITS-BASED DISTILLATION METHODS

Table 10 distinguishes distribution matching from the construction of supervision for non-one-to-one chunks. The table follows published formulations; experimental adaptations are described separately in Section 4.1.

Like ALM, CSD includes a residual event without pair-specific loss-mode selection. It additionally supports finer-grained matching for one-to-one, one-to-many, and many-to-one chunks, while retaining binary support for many-to-many chunks.

Table 11: Standalone teacher performance under the same 3-shot evaluation protocol used for the 1B students. Avg.-5 is the unweighted mean over the five displayed tasks.

Teacher	MMLU	WG	GSM8K	MATH	MBPP	Avg.-5
Llama-3.2-3B	55.97	70.01	24.94	8.82	38.20	39.59
Gemma3-4B-PT	59.34	73.16	35.78	15.26	45.80	45.87
Qwen3-4B-Base	72.43	72.30	84.61	27.76	67.60	64.94
Phi-4-mini	68.32	74.90	82.71	19.30	56.00	60.25

Table 12: Divergence and chunk-weight ablations with Qwen3-4B and Llama-3.2-1B. Divergence comparisons fix $\beta = 5$; weight comparisons use JSD. FKL and RKL denote forward and reverse KL. All other settings are fixed. † marks the shared default, reported once. Avg.-5 averages the five displayed tasks. **Bold** marks the highest reported average.

Teacher	Divergence	β	MMLU	WG	GSM8K	MATH	MBPP	Avg.-5
Qwen3-4B	JSD	$5^{\dagger}$	45.22	61.80	21.30	8.26	32.60	33.84
	FKL	5	44.55	62.59	19.71	8.60	27.80	32.65
	RKL	5	45.48	62.43	23.05	7.30	30.60	33.77
	JSD	1	45.24	62.35	19.71	7.12	30.00	32.88
	JSD	20	45.03	62.51	23.20	8.82	30.00	**33.91**

E More Evaluations

E.1 Standalone Teacher Performance

Table 11 reports standalone scores for all teachers used in the 1B experiments under the same 3-shot evaluation protocol. These scores contextualize teacher strength and are excluded from comparisons among student methods.

Both cross-tokenizer teachers achieve higher Avg.-5 than the same-tokenizer teachers, providing an opportunity to transfer knowledge from stronger models outside the student's family. Qwen has the highest average (64.94) and leads on four tasks, while Phi (60.25) performs best on WinoGrande. This ranking does not carry over directly to distilled students: with CSD, Phi produces higher student averages than Qwen for both Llama and Gemma (Tables 1 and 2). Standalone teacher accuracy therefore does not by itself determine transfer effectiveness.

E.2 Divergence and Chunk Weighting

Table 12 examines the divergence and non-one-to-one chunk weight β with the Qwen3-4B teacher and Llama-3.2-1B student. At $\beta = 5$, JSD achieves an Avg.-5 of 33.84, compared with 32.65 for FKL and 33.77 for RKL. JSD has the highest reported average among the three divergences, although its advantage over RKL is only 0.07 points. The common-support construction accommodates all three choices.

With JSD, increasing β from 1 to 5 raises Avg.-5 from 32.88 to 33.84. Increasing it further to 20 yields 33.91, an additional 0.07-point gain. These results suggest that non-one-to-one chunks provide useful supervision, but do not establish a universally optimal weight. We retain $\beta = 5$ as the default and leave it configurable.

E.3 Detailed Teacher Top-k Results

Table 13 provides per-task scores and training-step times for the configurations in Section 4.4. Using Qwen3-4B and Llama-3.2-1B, we compare teacher-side truncation at $k = 16$, 128, and 8192 with full logits. All other settings remain fixed.

Table 13: Teacher top-k results with Qwen3-4B and Llama-3.2-1B. The full-logit setting uses no teacher-side truncation. Avg.-5 is the unweighted mean over the five tasks; time denotes average training-step time in seconds. **Bold** marks the highest average.

Teacher support	MMLU	WG	GSM8K	MATH	MBPP	Avg.-5	Time (s)
$k = 16$	44.91	62.75	19.71	8.30	30.60	33.25	3.60
$k = 128$	45.16	62.60	21.15	8.24	30.80	33.59	3.66
$k = 8192$	44.77	62.83	21.61	8.48	30.60	33.66	4.03
Full logits	45.22	61.80	21.30	8.26	32.60	**33.84**	7.45

All three top-k settings remain within 0.59 average points of full-logit performance while reducing training-step time. At $k = 128$, Avg.-5 is 33.59, only 0.25 points below full logits, and step time falls from 7.45 to 3.66 seconds, a $2.04\times$ speedup. Increasing k to 8192 raises Avg.-5 to 33.66 and step time to 4.03 seconds. Individual task scores are not monotonic in k, but aggregate accuracy remains close to the full-logit result. These observations support teacher top-k truncation as an effective accuracy–efficiency trade-off in this setting.

Timing protocol. The production timing benchmarks use 256 NVIDIA H100 GPUs across 32 nodes, with a per-GPU microbatch size of 1, gradient accumulation of 3, a global batch size of 768, and a sequence length of 4096. Teacher predictions are computed online. Measurements include teacher and student forward passes, loss and support computation, backward propagation, distributed communication, optimizer updates, and data loading and transfer. They exclude offline data preparation and tokenizer-table construction, process and model startup, checkpoint writing, and additional measurement synchronization. Runtime caches remain enabled.