Rethinking Selective Knowledge Distillation

Anonymous Authors[1]

Abstract

Growing efforts to improve knowledge distillation (KD) in large language models (LLMs) replace dense teacher supervision with selective distillation, which uses a subset of token positions, vocabulary classes, or training samples for supervision. However, it remains unclear which importance signals, selection policies, and their interplay are most effective. In this work, we revisit *where* and *how* to distill in autoregressive LLMs. We disentangle selective KD along the position, class, and sample axes and systematically compare importance signals and selection policies. Then, guided by this analysis, we identify underexplored opportunities and introduce student-entropy-guided position selection (SE-KD). Across a suite of benchmarks, SE-KD often improves accuracy, downstream task adherence, and memory efficiency over dense distillation. Extending this approach across the class and sample axes (SE-KD$_{3X}$) yields complementary efficiency gains that make offline teacher caching feasible. In practice, this reduces wall time by 70% and peak memory by 18%, while cutting storage usage by 80% over prior methods without sacrificing performance.

1. Introduction

Large language models (LLMs) achieve state-of-the-art results across diverse tasks, but their size makes them expensive to serve and difficult to adapt. Knowledge distillation (KD; Hinton et al., 2015) addresses this by training a smaller student model to imitate a larger teacher. For autoregressive LLMs, this is typically done by matching the teacher's next-token distribution at every position of the training sequence.

However, applying knowledge distillation at every token position is often suboptimal due to the uniform supervision across all positions. Recent studies demonstrate that performance can be improved by selecting or reweighting positions for KD based on signals such as student cross-entropy (Wang et al., 2021), teacher uncertainty (Zhong et al., 2024; Huang et al., 2025), and teacher-student discrepancy (Xie et al., 2025). Yet, it remains unclear which token-importance signals most reliably identify positions that benefit from logit-based distillation in LLMs, and how different position-selection policies interact with these signals to shape an effective distillation curriculum.

In this work, we revisit *where* and *how* to apply teacher supervision in knowledge distillation for autoregressive LLMs. We first disentangle selective KD into five design axes: the alignment criterion, positions, classes, samples, and features. Within this framework, we focus on 3 key selection axes (Fig. 1)—positions, classes, and samples—and systematically analyze: (i) the choice of position-importance signal, comparing uncertainty- and discrepancy-based measures such as entropy and teacher–student KL; (ii) the policy used to convert these signals into selective supervision, e.g., top-k selection, curriculum learning, and stochastic allocation under fixed budgets; and (iii) how position selection interacts with sparsification along the class and sample axes.

Motivated by gaps revealed by this analysis, we identify two underexplored opportunities: (1) the use of *student entropy* as a position-importance signal, and (2) joint selection across multiple axes. We address these gaps by introducing a student-entropy-guided position-selective KD method, called SE-KD, and its 3-axis variant SE-KD$_{3X}$, which applies selection over samples, positions, and classes (Fig. 1D).

Through experiments across a broad suite of benchmarks, covering 9 importance signals, 5 selection policies, and 6 KD baselines, we find that student-uncertainty-based position selection reliably identifies high-value tokens for distillation. Selecting the top-20% positions based on student-entropy yields a consistent improvement in average evaluation accuracy (64.8 vs. 64.4 for Full KD) and perplexity (6.9 vs. 7.3), while requiring supervision on only a fraction of token positions. These gains come with some reduction in calibration (0.273 $\rightarrow$ 0.276), yet substantially reduce computational and memory overhead, as fewer teacher and student logits need to be computed.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

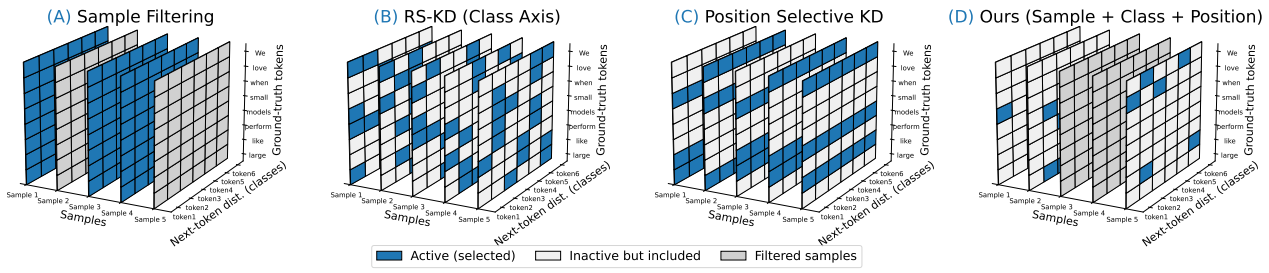

Figure 1. Illustration of three key selection axes for knowledge distillation: **(A)** sample selection, **(B)** class sampling (RS-KD), **(C)** position-selective KD, and **(D)** our combined approach, SE-KD$_{3X}$, which integrates sample, class, and position selection. Blue cells denote active (selected) supervision, light gray indicates inactive but included elements, and dark gray denotes filtered samples.

Next, we evaluate SE-KD and SE-KD$_{3X}$ on two additional settings of on-policy distillation (Agarwal et al., 2024), where supervision is applied to student-generated trajectories, and task-specific distillation, focusing on math reasoning. We find that our approach remains competitive in both settings, suggesting that our method generalizes beyond a single distillation regime.

Finally, we analyze the efficiency gains of position-selective KD when combined with sample filtering and class sampling. On 80M token distillation, student-entropy-based sample selection reduces total wall time by 70%, and class-sampled offline caching becomes feasible in practice, cutting storage by 99.96%, while maintaining performance.

In conclusion, our work makes the following contributions:

- We propose a general, theoretical framework for selective KD that organizes prior methods and highlights unexplored variants.
- We introduce new selective KD variants, SE-KD and SE-KD$_{3X}$, guided by *student entropy*. We show that these variants provide an often best-performing signal for KD, outperforming prior position-importance metrics across position-selective, and yielding the strongest gains in accuracy, while preserving downstream task adherence.
- We show that student-entropy-guided position selection, combined with class and sample selection, improves distillation accuracy while enabling efficient training via an offline teacher cache and substantially reduced runtime, memory usage, and storage.

We will release our code upon publication.

2. Related Work

KD with Position Selection A prominent line of work has explored ways to improve KD by selectively apply supervision at only part of the sequence positions. Wang et al. (2021) selected the top $k\%$ positions with the highest student cross-entropy using both batch-local selection and global-level selection (GLS). More recently, Huang et al. (2025) proposed down-weighting positions whose student proposals are not supported by the teacher. In parallel, token-adaptive frameworks dynamically adjust token-level supervision based on teacher-student distribution discrepancy (Xie et al., 2025). These approaches focus on a single selection heuristic or setting, broadly following an "80/20" intuition: a small fraction of high-entropy "fork" positions may carry much of the distillation signal (Wang et al., 2025). Our work extends these efforts by providing a unified comparison of position-selection strategies and metrics under a common distillation setup, isolating which signals reliably identify informative positions across tasks.

Curriculum Learning For chain-of-thought distillation, Feng et al. (2024) learned position-importance weights and used a curriculum that expands supervision from easier to harder reasoning steps under a given budget. Inspired by this work, we incorporate curriculum in two ways: (1) our student-entropy selection induces an implicit curriculum as supervised positions adapt during training; and (2) we evaluate an explicit curriculum-style position-selection method.

Uncertainty-Guided Position-Weighting KD Previous work showed that uncertainty-weighted distillation can improve reliability and calibration (Guo et al., 2024). Recently, Adaptive-Teaching KD (AT-KD; Zhong et al., 2024) built on Decoupled KD (Zhao et al., 2022) and routes token-level supervision using the teacher's gold-label probability, $1 - p_t(y_t)$, where $p_t(y_t)$ is the teacher probability assigned to the ground-truth next token. Per batch, AT-KD ranks positions by this uncertainty score and splits them into easy and hard tokens, skipping the target-class KL term on easy tokens while emphasizing diversity on hard tokens. Unlike prior approaches that incorporate uncertainty through position-wise loss reweighting, our method uses uncertainty solely as a ranking signal for explicit selection.

KD with Class Sampling A complementary line of work has focused on reducing distillation cost by sparsifying the teacher's output distribution. Deterministic top-k or percentile truncation of teacher logits (Raman et al., 2023;

Shum et al., 2024) reduces compute and storage costs but discards tail mass, inducing biased gradient estimates and miscalibrated students. Random-Sampling KD (RS-KD; Anshumann et al., 2025) replaced truncation with importance sampling to provide unbiased gradient estimates and improved calibration. These works focus on class sampling, which is one of the selection axes we study.

KD with Sample Selection & Weighting Distillation efficiency can also be improved by reducing the number of teacher queries. For example, UNIX (Xu et al., 2023) uses uncertainty-aware sampling to focus distillation on informative samples. Other work focused on the sample selection to improve accuracy. Entropy-based adaptive KD reweighs the KD loss by prioritizing *samples* according to the entropy of the teacher and student (Su et al., 2023). More recently, Difficulty-Aware Knowledge Distillation (DA-KD) (He et al., 2025) explicitly measures sample (or position) difficulty via the discrepancy between teacher and student cross-entropy losses, defined as the CE ratio, $\mathcal{L}^{\mathrm{CE}}_{\text{student}}(x)/\mathcal{L}^{\mathrm{CE}}_{\text{teacher}}(x)$, and utilizes this score for difficulty-aware stratified sampling, so that distillation focuses on hard but informative examples while maintaining data diversity.

3. A Framework for Selective Knowledge Distillation

We propose a general framework for selective KD, which encapsulates existing approaches and highlights opportunities for extending them. We then outline key design choices involved in the implementation of our framework.

Problem Setup In knowledge distillation, a student model is trained to imitate a teacher model by minimizing the divergence between their next-token distributions over a set of inputs. Let $x = (x_1, \ldots, x_L)$ be an input sample of L tokens and $\mathcal{V}$ the shared vocabulary of the teacher and student. At each position $t \in \{1, \ldots, L-1\}$, the teacher and student define next-token distributions $p_t(\cdot) = p(\cdot \mid x_{\le t})$ and $q_t(\cdot) = q(\cdot \mid x_{\le t})$ over $\mathcal{V}$.

The standard non-selective form, dubbed Full KD, optimizes a mixture of the teacher–student KL divergence and the ground-truth cross-entropy (CE), averaged over token positions and training samples. For a given sample, the distillation loss at position t is defined as

$$\ell_{\mathrm{KD}}(t) = \lambda\,\mathrm{KL}(p_t \,\|\, q_t) + (1-\lambda)\,\mathrm{CE}(y_t, q_t), \quad (1)$$

and for a training set $\mathcal{D}$ the overall objective is

$$\mathcal{L}_{\mathrm{KD}} = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\frac{1}{L_i - 1}\sum_{t=1}^{L_i-1}\ell^{(i)}_{\mathrm{KD}}(t), \quad (2)$$

where $\ell^{(i)}_{\mathrm{KD}}(t)$ is the loss at position t of sample i, and L_i is the length of sample i.

Selection therefore can be applied over three different axes: *classes* at a specific position, *positions* of a given sample, and *samples* in the training set. Let $\mathrm{KL}_{\mathcal{C}_t^{(i)}}$ denote the KL divergence computed over a subset of classes $\mathcal{C}_t^{(i)} \subseteq \mathcal{V}$ (where $\mathcal{C}_t^{(i)} = \mathcal{V}$ for Full KD). Moreover, let $m_t^{(i)} \in \{0,1\}$ indicate whether position t of the i-th sample receives supervision and $s_i \in \{0,1\}$ whether sample i is selected for distillation. The objective of selective KD can be written as:

$$\ell^{(i)}_{\mathrm{SKD}}(t) = \lambda\,\mathrm{KL}_{\mathcal{C}_t^{(i)}}(p_t \,\|\, q_t) + (1-\lambda)\,\mathrm{CE}(y_t, q_t) \quad (3)$$

$$\mathcal{L}^{(i)}_{\mathrm{SKD}} = \frac{1}{\sum_{t=1}^{L_i-1} m_t^{(i)}}\sum_{t=1}^{L_i-1} m_t^{(i)}\,\ell^{(i)}_{\mathrm{SKD}}(t) \quad (4)$$

$$\mathcal{L}_{\mathrm{SKD}} = \frac{1}{\sum_{i=1}^{|\mathcal{D}|} s_i}\sum_{i=1}^{|\mathcal{D}|} s_i\,\mathcal{L}^{(i)}_{\mathrm{SKD}} \quad (5)$$

The primary question is *how to choose classes, positions, and samples* for distillation, namely, how to construct $\mathcal{C}_t$, $m_t^{(i)}$, and s_i.

Key Choices for Selective Distillation We decompose selective KD into five orthogonal design choices that determine how teacher information is transferred to the student:

1. *Alignment criterion*: the objective used for teacher–student alignment, e.g., KL-based or Decoupled KD.
2. *Position axis*: which token positions receive distillation, i.e., how to choose $m_t^{(i)}$. We study this axis via (i) the position-importance metric $u(t)$, which quantifies the importance of each position t, and (ii) the position-selection policy, namely, a rule that maps the scores $u(t)$ for a given sample to the values $m_t^{(i)}$.
3. *Class axis*: how the teacher distribution over the vocabulary is sparsified at each position, choosing $\mathcal{C}_t^{(i)}$.
4. *Sample axis*: which training examples are distilled, i.e., how to choose s_i.
5. *Feature axis* (*not explored here*): which teacher and student representations are being aligned. Beyond next-token distributions, KD can align intermediate features, such as hidden states or attention maps (Romero et al., 2015; Jiao et al., 2020). While selection can be applied on this axis as well (e.g., choosing layers or heads), we leave this direction for future work.

Table 1 summarizes prior methods for selective KD in terms of our framework. Notably, we observe that no prior work has exploited selection across more than a single axis. Moreover, student entropy as a distillation signal is underexplored, despite evidence for its effectiveness in training (Wang et al., 2025). We tackle these gaps next.

Table 1. **Overview of selective KD methods with selection-axis membership.** The columns **Pos**, **Cls**, and **Smp** indicate whether a method applies selection/sparsification along the position, class, or sample axes, respectively. ✓ denotes that the method explicitly acts on that axis, while ✗ indicates it does not. We highlight our proposed student-entropy variants in green.

	Method	Description	Pos	Cls	Smp
Alignment criterion	Full KD (Hinton et al., 2015)	KL/CE on all positions (Eq. 1)	✗	✗	✗
	Decoupled KD (Zhao et al., 2022)	Reweighs target vs. non-target terms in the KL loss	✗	✗	✗
	AT-KD (Zhong et al., 2024)	Routes positions into easy/hard buckets with separate KL terms using teacher's gold-label (y_t) probability $1 - p_t(y_t)$	✓	✗	✗
	Weighted KD (Guo et al., 2024)	Reweighs per-position KLD in the loss using $w_t \propto u(t)$	✓	✗	✗
Position-importance metric	Student CE (Wang et al., 2021)	Student cross-entropy $\mathrm{CE}(y_t, q_t)$	✓	✗	✗
	Teacher CE (Zhong et al., 2024)	Teacher cross-entropy $\mathrm{CE}(y_t, p_t)$	✓	✗	✗
	Student entropy	Student entropy $H(q_t)$	✓	✗	✗
	Teacher entropy	Teacher entropy $H(p_t)$	✓	✗	✗
	KL / reverse-KL	Teacher–student discrepancy $\mathrm{KL}(p_t \Vert q_t)$ / $\mathrm{KL}(q_t \Vert p_t)$	✓	✗	✗
	KL + student entropy	Combined discrepancy and uncertainty ranking $\mathrm{KL}(p_t \Vert q_t) + H(q_t)$	✓	✗	✗
	CE ratio (He et al., 2025)	Teacher–student CE ratio $r(t) = \mathrm{CE}(y_t, q_t)/\mathrm{CE}(y_t, p_t)$	✓	✗	✗
	CE ratio + student entropy	Combined difficulty and uncertainty $r(t) + H(q_t)$	✓	✗	✗
Position-selection policy	Top-k%	Chooses the top-k% positions according to $u(t)$	✓	✗	✗
	GLS (Wang et al., 2021)	Top-k% normalized across batches with global thresholds τ	✓	✗	✗
	Curriculum learning (Feng et al., 2024)	Shifts supervision from easy to hard positions with scheduled window	✓	✗	✗
	Pos RS-KD / Pos RS-KD*	Stochastic estimator $q(t) \propto w_t$ of Weighted/Full KD	✓	✗	✗
Class sampling	RS-KD (Anshumann et al., 2025)	At position t, sample with repetition U indices $v_k \propto p_t(v)$. Let $\mathcal{C}_t = \{v_k\}_{k=1}^{U}$ be the unique sampled indices. Build a sparse teacher target $\tilde{p}_t$ on $\mathcal{C}_t$ (from sampled counts) and minimize $\sum_{v \in \mathcal{C}_t} \tilde{p}_t(v) \log(\tilde{p}_t(v)/q_t(v))$.	✗	✓	✗
Sample selection	Top-ℓ% avg. student entropy (Xu et al., 2023)	Selects samples using student entropy $U_i = \frac{1}{L_i - 1} \sum_t H(q_t)$	✗	✗	✓

4. Student Entropy Guided Selective KD

Given the gaps in prior work, we introduce a selective KD method that leverages student entropy as a position-importance signal and employ selection across axes.

Student Entropy-based Position Selection (SE-KD) We use student entropy to score position importance, i.e., $u(t) = H(q_t)$. Given a sample i of length L, SE-KD selects the top-k% most uncertain positions for distillation:

$$m_t^{(i)} = \mathbb{I}[u(t) \geq \tau], \tag{6}$$

where τ is chosen such that exactly $\lceil k(L-1) \rceil$ positions satisfy $m_t^{(i)}=1$. We additionally use a per-sequence normalization in the loss to ensure a fixed supervision budget.

Cross-Axis Selection In addition to position selection, we extend SE-KD to operate across the three axes of classes, positions, and samples. Specifically, we apply class selection via per-token class sampling ($\mathcal{C}_t^{(i)}$) using RS-KD, and sample selection via top-ℓ% ranking by average student entropy computed in a single forward-pass preprocessing step using a frozen student, then distilling on the top-ℓ% samples. We call this variant SE-KD$_{3X}$. These extensions are orthogonal to position selection and enable a unified multi-axis KD that improves accuracy, efficiency, and storage cost.

Selective LM Head and Chunked Entropy Computation Selective KD enables two simple, selection-aware optimizations for reducing the logit-related memory footprint. Let B denote the batch size, L the sequence length, and $V = |\mathcal{V}|$ the vocabulary size.

First, *chunked entropy computation* computes per-position entropy without materializing the full $[B, L, V]$ sized logits tensor: the student hidden states are projected through the LM head in small chunks with gradients disabled, reduced to $O(BL)$ entropy scalars, and discarded.

Second, a *selective LM head* computes logits only at the positions across the batch N_{select}: teacher logits shrink from $[B, L, V]$ to $[N_{\text{select}}, V]$, and for the student it computes logits *with gradients enabled* only at selected positions, so the KL loss backpropagates through N_{select} positions rather than all BL, reducing both forward and backward cost.

5. Experiments

We conduct comprehensive experiments to assess selective KD methods along the axes defined in §3. Notably, the design space is large even under conservative choices, spanning position-importance metrics, position-selection policies, and class/sample selection, which yields hundreds of configurations and makes exhaustive evaluation infeasible. We therefore use a controlled evaluation protocol in which we fix all but one axis at a time. This allows us to isolate the effect of each design choice.

Methods We evaluate all the position importance metrics and selection policies in Table 1. Except for GLS, position selection is always normalized per sequence length. Unless stated otherwise, all models are trained with the same hyperparameters described in §B. Below are additional details on the position selection policies:

- **GLS**: Maintains a queue of recent entropy values and sets τ to the empirical $(100-k)$-th percentile of this global distribution to stabilize top-k selection across batches.
- **Pos RS-KD**: A stochastic position-selection policy inspired by RS-KD, sampling positions with probability $q(t) = \frac{H(q_t)}{\sum_j H(q_j)}$. While treated here as a selection policy, repeated sampling induces implicit loss reweighting, yielding an unbiased estimator of weighted KD (see §A).
- **Pos RS-KD***: Importance-corrected variant: after sampling positions with probability $q(t)$, each sampled position loss is reweighted by $1/q(t)$, yielding an unbiased estimator of Full KD.
- **Curriculum**: A curriculum-style position-selection method with a fixed budget of k=20% positions per sequence, gradually shifting supervision from low to high-student-entropy tokens over training.

Baselines and Ablations We compare against the following baselines and component ablations:

- Off-the-shelf student without distillation, and the teacher as an upper bound.
- **Full KD**: Supervised KD applied densely over all classes, positions, and samples.
- **AT-KD**: As a representative uncertainty-guided position-weighting method.
- **RS-KD**: Class-axis selective distillation using importance sampling over teacher logits.
- **RandomPos** k%: Random position selection supervising a fixed fraction k% of positions per sample.
- **TopSmp** ℓ%: Student entropy-based sample selection. This is an ablation of SE-KD$_{3X}$ that removes class sampling (RS-KD) and position selection.
- **RandomSmp** ℓ%: Random sample selection supervising a fixed fraction ℓ% of training samples.

We separate global configuration selection from final evaluations. All methods share the same distillation setup: KD hyperparameters (e.g. temperature $T = 1.0$ and loss weighting $\alpha_{CE} = 0.0$) are selected once on validation data and then fixed, with no method-specific tuning (see §C for details). Supervision budgets for top-k% position selection and top-ℓ% sample selection are chosen via a search on validation splits using a single run per setting, and then fixed for all main comparisons. We report the validation-split ablations used for configuration selection in F.

Evaluation We consider two distillation setups:

(1) *General-purpose distillation* on a large pretraining-style corpus. We train all models on 80 million tokens from FineWeb-Edu (Penedo et al., 2024) and evaluate them in a zero-shot setting. Documents are packed into sequences of up to 512 tokens. We measure performance on HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2019), and Arc-E (Clark et al., 2018) (multiple-choice reasoning); GSM8K (Cobbe et al., 2021) (math reasoning); and LAMBADA (Paperno et al., 2016) (long-range prediction), reporting average accuracy. For LAMBADA, we additionally report perplexity and expected calibration error (ECE; Guo et al., 2017). We also evaluate instruction-following on IFEval (Zhou et al., 2023) reporting Pass@1 according to the official verifier[1]. All results are averaged over three random seeds, with standard deviations reported in §G.

(2) *Task-specific distillation* on a downstream reasoning task. We apply KD directly on the GSM8K training set (Cobbe et al., 2021) and report exact-match accuracy on the GSM8K test set. In addition to standard off-policy distillation, we evaluate on-policy distillation (Agarwal et al., 2024). We exclude SE-KD$_{3X}$ from this evaluation, as class-level sampling relies on an offline teacher cache that is incompatible with dynamic student text generation.

Models We follow prior work (Chen et al., 2025; Lu & Lab, 2025) and use Qwen3-1.7B as a student and Qwen3-8B as a teacher (Yang et al., 2025).

6. Results

Comparing Position-Importance Metrics We begin by fixing the selection policy and budget to top-20% and comparing the position-importance metrics. Table 2 presents the results, showing that student entropy based signals and teacher–student discrepancy metrics (CE ratio, KL and reverse KL) most reliably identify informative positions: Top-20% student entropy achieves strong performance (64.8

[1] https://github.com/google-research/google-research/tree/master/ifeval

Table 2. Evaluation results of various **position-importance metrics with Top-20% hard selection.** The best student method is in **bold**, the second best is underlined, and ***bold italic*** denotes the teacher (upper bound). Standard deviation values are in §G.

Method	Acc. ↑	IFEval ↑	PPL ↓	ECE ↓
Qwen3 1.7B	61.9	19.4	12.2	30.5
Qwen3 8B	***73.8***	***28.9***	***4.6***	***23.5***
Full KD	64.4	20.5	7.3	27.3
RandomPos 20%	64.2	20.2	7.7	27.2
AT-KD	63.8	19.8	7.3	**26.7**
Position selection policy: Top 20%				
Student entropy (SE-KD)	64.8	21.4	6.9	27.6
Teacher entropy	63.2	20.5	9.4	27.3
Student CE	63.8	20.4	8.1	27.8
Teacher CE	63.4	19.4	9.3	27.8
KL	64.5	21.0	7.2	**26.7**
Reverse KL	64.7	20.7	6.8	27.0
CE ratio	64.6	**22.5**	**6.5**	27.7
CE ratio + student entropy	64.6	21.4	6.7	27.5
Student entropy + KL	**65.1**	20.9	6.8	26.9

Table 3. Evaluation results of **position-selection policies, applied with student-entropy** as position-importance metric and distillation budget of 20%, against baselines.

Method	Acc. ↑	IFEval ↑	PPL ↓	ECE ↓
Qwen3 1.7B	61.9	19.4	12.2	30.5
Qwen3 8B	***73.8***	***28.9***	***4.6***	***23.5***
Full KD	64.4	20.5	7.3	27.3
RandomPos 20%	64.2	20.2	7.7	27.2
AT-KD	63.8	19.8	7.3	**26.7**
Position importance metric: Student entropy, $k = 20\%$				
Top 20% (SE-KD)	**64.8**	**21.4**	**6.9**	27.6
Top 20% GLS 30K	64.5	20.7	7.5	27.6
Curriculum 20%	64.6	20.7	**6.9**	27.7
Pos RS-KD* 20%	63.6	20.6	8.3	27.6
Pos RS-KD 20%	63.0	20.1	9.9	27.0

accuracy, 6.9 perplexity), beating Full KD, and RandomPos while top-20% KL/reverse-KL/CE-ratio remain competitive (64.5–64.7 accuracy, with best perplexity at 6.5). In contrast, ranking by teacher entropy/CE underperforms in both accuracy and perplexity. Notably, calibration differences are small; although AT-KD, KL, and reverse KL achieve the best ECE, the gaps are limited, suggesting that gains mainly stem from better supervision allocation rather than changes in confidence behavior.

Comparing Position Selection Policies We compare position-selection policies at a fixed importance metric and budget. As shown in Table 3, Top-20% selection by student entropy (SE-KD) yields the strongest overall performance, improving accuracy (64.4 → 64.8), perplexity (7.3 → 6.9), and instruction-following (20.5 → 21.4). It outperforms Full KD, random selection, GLS, curriculum scheduling, and AT-KD in accuracy and IFEval, though AT-KD achieves the best calibration, followed by Pos RS-KD and only then

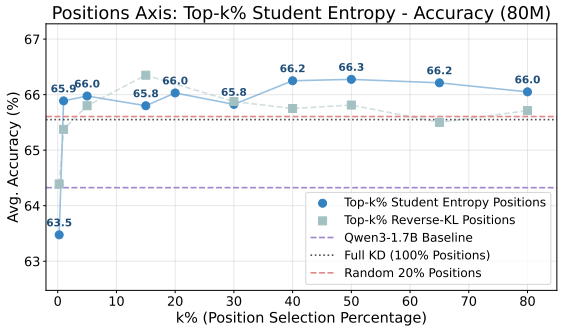

Figure 2. **Position-axis budget sweep.** Average validation accuracy after distilling on 80M FineWeb-Edu tokens as a function of the supervised position budget $k\%$. We compare Top-$k\%$ student-entropy (SE-KD) and Top-$k\%$ reverse-KL, with Full KD and RandomPos as reference. The teacher accuracy is 77.0.

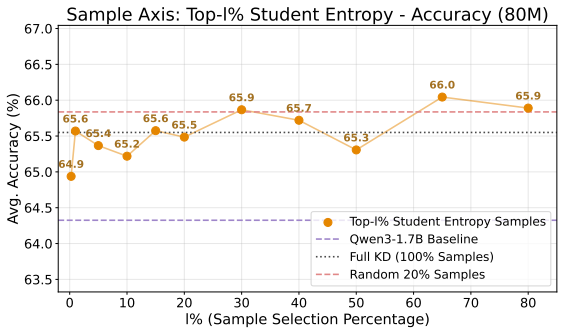

Figure 3. **Sample-axis budget sweep.** Average validation accuracy after distilling on 80M FineWeb-Edu tokens as a function of the sample-selection budget $\ell\%$. Only the top-$\ell\%$ samples ranked by average student entropy are distilled; Full KD and RandomSmp are shown for reference. The teacher accuracy is 77.0.

SE-KD. Pos RS-KD and Pos RS-KD* underperform Top-$k\%$, suggesting that naive entropy-proportional sampling can be suboptimal without additional smoothing or coverage constraints (see §D). Overall, student-entropy-guided selection is the most reliable position-selection policy at k=20%, supporting the view that dense supervision is suboptimal.

The Effect of Distillation Budget on Performance Fig. 2 and 3 report the average accuracy on the validation sets (ArcEasy, GSM8K, HellaSwag and PIQA), averaged over multiple runs. We show the performance of SE-KD and reverse-KL, as a representative student–teacher discrepancy metric, across varying selection budgets. The best performance for both methods is obtained for k=20% (Fig. 2), consistent with recent findings that roughly 20% of high-entropy tokens disproportionately drive learning (Wang et al., 2025). Both methods are robust across a wide range of k values, with a shallow optimum at intermediate budgets; notably, supervising as little as $\sim 1\%$ of positions already matches or exceeds Full KD, while extremely small budgets (e.g., $\sim 0.25\%$) remain closer to the undistilled

Table 4. **Multi-axis selective KD**, comparing SE-KD$_{3X}$ against baselines and mixes of position selection (SE-KD), class sampling (RS-KD), and sample selection (TopSmp) on general-purpose distillation (test split, 80M tokens). We report average accuracy across benchmarks (Acc.), instruction-following performance (IFEval), LAMBADA perplexity (PPL), and expected calibration error (ECE). Standard deviations over three seeds are in §G.

Method	Acc. ↑	IFEval ↑	PPL ↓	ECE ↓
Qwen3 1.7B	61.9	19.4	12.2	30.5
Qwen3 8B	73.8	29.0	4.6	23.5
AT-KD	63.8	20.7	7.4	**26.7**
Full KD	64.4	20.5	7.3	27.3
RandomPos 20%	64.1	19.8	7.6	27.1
RandomSmp 20%	64.0	20.6	8.2	27.5
SE-KD	**64.8**	21.4	**6.9**	27.6
RS-KD	64.7	20.9	7.4	27.3
TopSmp 20%	64.2	20.8	7.4	27.8
TopSmp 20% + RS-KD	64.1	20.9	7.4	27.7
SE-KD + TopSmp 20%	64.6	**22.0**	6.9	28.0
SE-KD$_{3X}$	64.4	20.7	7.3	27.9

baseline. We use k=20% in subsequent experiments as a strong accuracy–compute trade-off (near the plateau) and to stay consistent with prior "small-fraction" findings (Wang et al., 2025). We also vary the *sample-axis* budget by distilling only the top-ℓ% samples ranked by average student entropy (Fig. 3). Accuracy changes little with ℓ, while compute scales roughly linearly, so we use ℓ=20% in multi-axis experiments.

Selection Across Positions, Classes, and Samples Table 4 compares selective KD across the position, class, and sample axes. Position selection is the dominant performance contributor; our student-entropy SE-KD improves average accuracy from 64.4 (Full KD) to 64.8, improves instruction-following (21.4 vs. 20.5) and reduces PPL (6.9 vs. 7.3), with a modest ECE increase (27.6 vs. 27.3). RS-KD improves accuracy and preserves calibration, while TopSmp remains close overall to Full KD but degrades calibration. Combining all axes, SE-KD$_{3X}$ achieves competitive performance (64.4 accuracy, 20.7 IFEval, 7.3 PPL) and slightly worse calibration, while substantially reducing runtime, memory, and storage (see §7).

General-Purpose vs. Task-Specific Distillation Table 5 reports task-specific distillation results on GSM8K, which differ qualitatively from general-purpose distillation on FineWeb-Edu. In the off-policy regime, Full KD achieves the best GSM8K accuracy (71.6), while entropy-based Top-20% position selection degrades performance (69.5). Our strongest method, SE-KD + TopSmp, remains close to Full KD (70.9) despite substantially reduced supervision. In the on-policy regime, SE-KD + TopSmp attains the highest GSM8K accuracy (71.2), outperforming Full KD (70.6), while average accuracy differences remain small.

Table 5. **Results for task-specific distillation on GSM8K.** We compare off-policy and on-policy KD methods, reporting GSM8K exact-match accuracy, average evaluation suite accuracy, and LAMBADA OAI perplexity. For on-policy distillation, we used the reverse-KL alignment criterion. Standard deviations are in §G.

	Method	GSM8K Acc. ↑	Acc. ↑	PPL ↓
	Qwen3 1.7B	68.2	61.9	12.2
	Qwen3 8B	***87.8***	***73.8***	***4.6***
Off Policy Distill.	Full KD	**71.6**	**64.5**	**7.8**
	RandomPos 20%	70.2	64.0	8.0
	SE-KD	69.5	63.9	8.0
	Pos RS-KD 20%	70.5	63.5	8.9
	Pos RS-KD* 20%	69.1	63.3	9.2
	TopSmp 20%	69.0	63.6	8.6
	SE-KD + TopSmp 20%	70.9	64.0	8.6
	SE-KD$_{3X}$	70.2	63.9	8.6
On Policy Distill.	Full KD	70.6	**63.7**	10.0
	RandomPos 20%	69.3	63.3	10.0
	SE-KD	70.0	**63.7**	**9.5**
	Pos RS-KD 20%	70.5	63.2	10.5
	Pos RS-KD* 20%	69.7	63.3	10.0
	TopSmp 20%	70.4	**63.7**	10.1
	SE-KD + TopSmp 20%	**71.2**	63.4	10.4

In the on-policy setting, combining entropy-guided position selection with sample filtering yields the strongest results. However, in the off-policy regime, and unlike general-purpose distillation, entropy-guided position selection alone does not consistently outperform Full KD on GSM8K. Instead, our methods remain close to Full KD after a single epoch despite using substantially less supervision. We attribute this in part to GSM8K's limited size, which may constrain the benefits of selective distillation and allow them to emerge more clearly with larger datasets or multi-epoch training. We leave this hypothesis for future work.

7. Distillation Efficiency

A major motivation for selective KD is reducing computational costs. We therefore analyze distillation efficiency in terms of *offline storage* for teacher supervision and *runtime compute* during distillation. We show that while position selection primarily improves accuracy, sample-level selection yields prominent efficiency gains, and class-level sampling enables orders-of-magnitude reductions in storage.

7.1. Storage Efficiency

We follow the formulation of Anshumann et al. (2025), focusing on savings from class- and sample-selection. Position selection is excluded since it would require storing dynamic uncertainty masks (see §E).

Storage is measured in bytes per token and reported in decimal terabytes (TB) for a dataset of N=100B tokens. Storing

Table 6. **Offline cache footprint** in terabytes (TB) for N=100B training tokens and vocabulary size $|\mathcal{V}|$=100,000. RS-KD uses importance sampling over classes; SE-KD$_{3X}$ further reduces storage via sample-level selection with ℓ=20%.

Method	Classes	TB (U=12)	TB (U=64)
Full KD	$\lvert\mathcal{V}\rvert = 100K$	10,000.0	10,000.0
RS-KD	U	3.6	19.2
SE-KD$_{3X}$	$U \times 0.2$	0.72	3.84
Vanilla CE	1	0.3	0.3

Table 7. **Runtimes and test accuracy** for sample-selection methods (80M tokens, top-20%, single runs) on GeForce RTX 3090.

Method	Sample Selection	Total Wall Time	Acc.
Full KD (100% positions)	0h00m	22h52m	**64.6**
RandomPos 20%	0h00m	18h38m	64.1
TopSmp CE ratio	8h50m	13h36m	64.3
TopSmp KL	9h37m	14h42m	64.2
TopSmp student entropy (ours)	**2h01m**	**7h01m**	64.2
SE-KD$_{3X}$ (cache construction)	2h11m	8h46m	64.4
SE-KD$_{3X}$ (reuse offline cache)	0h00m	3h58m	64.4

a single sampled teacher class requires 24 bits (3 bytes): 17 bits for the vocabulary index and 7 bits for a quantized probability, so caching $U = |\mathcal{C}_t|$ sampled classes costs $3U$ bytes per position.

Table 6 summarizes the storage footprint for Full KD, RS-KD, and SE-KD$_{3X}$. As a baseline, we add vanilla CE training without teacher logits. Unlike Anshumann et al. (2025) who used U=12, we use U=64 for improved stability, yielding $64 \times 3 = 192$ bytes/position. Caching full teacher logits over a vocabulary of size $|\mathcal{V}|$=100,000 requires 200 kB per position in float16, making RS-KD with U=64 roughly 10^3–$2{\times}10^3$ times more storage-efficient, or 19.2 TB for N=100B tokens. With sample selection, we distill only on the top-ℓ% samples ranked by average student entropy from a single forward pass of a frozen student before distillation. This reduces storage linearly with ℓ. For ℓ=0.2, this yields: $\ell \cdot U \cdot 3 = 38.4$ bytes/position, or 3.84 TB in total.

Overall, RS-KD reduces storage from 10,000 TB to 19.2 TB (99.8%, $\sim 520\times$) and SE-KD$_{3X}$ further reduces this to 3.84 TB (99.96% vs. Full KD and 80% vs. RS-KD). Sample indices are also cached but incur negligible storage.

7.2. Runtime Efficiency

Runtime Speedups SE-KD$_{3X}$ achieves substantial efficiency gains through sample selection, which directly reduces the number of sequences requiring teacher supervision. As shown in Table 7, this leads to a pronounced reduction in total wall-clock time. Sample selection incurs an upfront scoring cost: teacher–student metrics require full passes (8h50m for CE ratio, 9h37m for KL), while student-only entropy is cheaper (2h01m). Reusing an offline cache of selected indices removes this step, reducing SE-KD$_{3X}$ runtime to 3h58m (Table 7). Concretely, for a training set of N samples with average length L, Full KD requires $\mathcal{O}(NL)$ teacher queries, whereas selecting only the top-ℓ% most uncertain samples reduces this cost to $\mathcal{O}(\ell NL)$. This makes sample filtering the dominant efficiency lever, while position selection provides the main accuracy gains by concentrating supervision on high-value tokens within selected samples.

Memory Savings of SE-KD Position selection primarily reallocates the KD signal *within* a sequence. Since the student forward/backward pass already processes all tokens, position selection alone does not yield a proportional wall-clock speedup beyond reducing KL computation on non-selected positions. Yet, selection enables memory-oriented implementations that substantially reduce the peak logit-related memory footprint.

In our setting (B=2, L=512), selective LM heads with chunked entropy at k=20% reduce the sum of per-GPU peak memory allocations by 18.3% (33.18 GB → 27.10 GB): student peak drops by 28.1% (15.88 GB → 11.42 GB) and teacher peak by 9.4% (17.30 GB → 15.68 GB). The gains come from avoiding full $[B, L, V]$ logit materialization during selection and restricting KD logits/backprop to the N_{sel} selected positions. See §H for ablations and memory traces.

8. Conclusion and Discussion

We revisit selective knowledge distillation for autoregressive LLMs through a unified framework that disentangles where and how teacher supervision is applied. Across a systematic study, we find that dense, uniform logit supervision is often unnecessary: for general-purpose distillation, concentrating supervision on a small subset of high-uncertainty positions consistently matches or outperforms Full KD.

Student-entropy-guided Top-20% selection is the most reliable overall strategy, while curriculum learning, CE-ratio ranking, and teacher–student KL are promising alternatives. We also show that position selection integrates effectively with class- and sample-level sparsification, yielding favorable accuracy–efficiency trade-offs; in particular, SE-KD$_{3X}$ enables substantial speedups via sample filtering and offline teacher caching, and can be implemented with reduced peak memory through a selective LM head.

Limitations and Future Work The selective KD design space is large; to keep comparisons controlled, we study a single, widely used teacher–student pair and a fixed supervision budget. Validating the trends across additional model families, scales, and longer contexts is an important next step. Selective policies may also interact with alternative alignment criteria (e.g., feature-based KD), and the smaller performance degragation we observe in task-specific distillation suggest further optimizations are needed.

Impact Statement

This paper aims to advance knowledge distillation for large language models. We do not identify societal impacts specific to this work beyond the general considerations associated with training and deploying language models.

References

Agarwal, R., Vieillard, N., Zhou, Y., Stanczyk, P., Garea, S. R., Geist, M., and Bachem, O. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Anshumann, A., Zaidi, M. A., Kedia, A., Ahn, J., Kwon, T., Lee, K., Lee, H., and Lee, J. Sparse logit sampling: Accelerating knowledge distillation in LLMs. *arXiv preprint arXiv:2503.16870*, 2025. URL https://arxiv.org/abs/2503.16870.

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. Piqa: Reasoning about physical commonsense in natural language, 2019. URL https://arxiv.org/abs/1911.11641.

Chen, W.-R., Kothapalli, V., Fatahibaarzi, A., Sang, H., Tang, S., Song, Q., Wang, Z., and Abdul-Mageed, M. Distilling the essence: Efficient reasoning distillation via sequence truncation, 2025. URL https://arxiv.org/abs/2512.21002.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018. URL https://aclanthology.org/P18-1260.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Feng, K., Li, C., Zhang, X., Zhou, J., Yuan, Y., and Wang, G. Keypoint-based progressive chain-of-thought distillation for LLMs. *arXiv preprint arXiv:2405.16064*, 2024. URL https://arxiv.org/abs/2405.16064.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Guo, Z., Wang, D., He, Q., and Zhang, P. Leveraging logit uncertainty for better knowledge distillation. *Scientific Reports*, 14(31249), 2024. doi: 10.1038/s41598-024-82647-6. URL https://www.nature.com/articles/s41598-024-82647-6.

He, C., Ding, Y., Guo, J., Gong, R., Qin, H., and Liu, X. DA-KD: Difficulty-aware knowledge distillation for efficient large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=NCYBdRCpw1.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Huang, H., Song, J., Zhang, Y., and Ren, P. Selectkd: Selective token-weighted knowledge distillation for llms, 2025. URL https://arxiv.org/abs/2510.24021.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., and Liu, Q. TinyBERT: Distilling BERT for natural language understanding. In Cohn, T., He, Y., and Liu, Y. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4163–4174, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.372. URL https://aclanthology.org/2020.findings-emnlp.372/.

Lu, K. and Lab, T. M. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, Q., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016. URL https://aclanthology.org/P16-1144.

Penedo, G., Kydlíček, H., Ben Allal, L., Lozhkov, A., Mitchell, M., Raffel, C., Von Werra, L., and Wolf, T. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*, 2024. URL https://arxiv.org/abs/2406.17557.

Raman, N., Vare, S., Srinivasan, A., Chandra, V., and Khandelwal, K. For distillation, tokens are not all you need. OpenReview, 2023. URL https://openreview.net/pdf?id=2fc5GOPYip.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets.

In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6550.

Shum, K., Xu, M., Zhang, J., Chen, Z., Diao, S., Dong, H., Zhang, J., and Raza, M. O. FIRST: Teach a reliable large language model through efficient trustworthy distillation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 12646–12659. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.emnlp-main.703.pdf.

Su, C.-W., Tseng, S.-H., Martins, J. V., Ichimura, N., Seiji, Y., and Chou, C.-H. EA-KD: Entropy-based adaptive knowledge distillation. *arXiv preprint arXiv:2311.13621*, 2023. URL https://arxiv.org/abs/2311.13621.

Wang, F., Yan, J., Meng, F., and Zhou, J. Selective knowledge distillation for neural machine translation. *arXiv preprint arXiv:2105.12967*, 2021. URL https://arxiv.org/abs/2105.12967.

Wang, S., Yu, L., Gao, C., Zheng, C., Liu, S., Lu, R., Dang, K., Chen, X., Yang, J., Zhang, Z., Liu, Y., Yang, A., Zhao, A., Yue, Y., Song, S., Yu, B., Huang, G., and Lin, J. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025. URL https://arxiv.org/abs/2506.01939.

Xie, X., Xue, Z., Wu, J., Li, J., Wang, Y., Hu, X., Liu, Y., and Zhang, J. Llm-oriented token-adaptive knowledge distillation, 2025. URL https://arxiv.org/abs/2510.11615.

Xu, G., Liu, Z., and Loy, C. C. Computation-efficient knowledge distillation via uncertainty-aware mixup. *Pattern Recognition*, 138:109338, 2023. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2023.109338. URL https://www.sciencedirect.com/science/article/pii/S0031320323000390.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zellers, R., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019. URL https://aclanthology.org/P19-1472.

Zhao, B., Cui, Q., Song, R., Qiu, Y., and Liang, J. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11953–11962, 2022. URL https://arxiv.org/abs/2203.08679.

Zhong, Q., Ding, L., Shen, L., Liu, J., Du, B., and Tao, D. Revisiting knowledge distillation for autoregressive language models. *arXiv preprint arXiv:2402.11890*, 2024. URL https://arxiv.org/abs/2402.11890.

Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., and Hou, L. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A. Proof: Positional Random Sampling Selection is an Unbiased Estimator of Weighted KD

In this section, we prove that a knowledge distillation using the positional RS selection method matches the weighted KD in expectation. It is important to note that one can easily transform such a selection method to match full KD in expectation, but we deliberately do not do so, since we aim to match a weighted KD that emphasizes tokens according to their entropy.

Consider a sequence of length N token positions, indexed by $t \in \{1, \dots, N\}$. Let L_t denote the per-token distillation loss at position t, and let $w(t)$ be a non-negative importance weight assigned to that token. We sample K token indices t_k i.i.d. from the following distribution:

$$q(t) = \frac{w(t)}{\sum_{j=1}^{N} w(j)}.$$

Here, $q(t)$ denotes the sampling distribution over token positions, $\widehat{\mathcal{L}}$ is the empirical loss estimator, and $\mathbb{E}[\cdot]$ denotes expectation over the sampling process.

Using this notation, we have

$$\begin{aligned}\mathbb{E}[L_{t_k}] &= \sum_{t=1}^{N} q(t) L_t = \sum_{t=1}^{N} \frac{w(t)}{\sum_{j=1}^{N} w(j)} L_t \\ &= \mathcal{L}_{\text{weighted}}.\end{aligned}$$

Hence, the probability of sampling token t is proportional to its contribution in the weighted KD objective.

$$\begin{aligned}\mathbb{E}[\widehat{\mathcal{L}}] = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} L_{t_k}\right] &= \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}[L_{t_k}] \\ &= \frac{1}{K} \cdot K \cdot \mathcal{L}_{\text{weighted}} = \mathcal{L}_{\text{weighted}}.\end{aligned}$$

It can also be viewed by denoting c_t as how many times token t was sampled:

$$c_t = \sum_{k=1}^{K} \mathbf{1}_{t_k = t}, \quad \widehat{\mathcal{L}} = \frac{1}{K}\sum_{t} c_t L_t,$$
$$\mathbb{E}[c_t] = K q(t).$$

So,

$$\begin{aligned}\mathbb{E}[\widehat{\mathcal{L}}] = \frac{1}{K}\sum_{t} \mathbb{E}[c_t] L_t = \frac{1}{K}\sum_{t} K q(t) L_t \\ = \sum_{t} q(t) L_t.\end{aligned}$$

Hence, $\widehat{\mathcal{L}}$ is an unbiased estimator of the **weighted KD objective**:

$$\mathcal{L}_{\text{weighted}} := \sum_{t=1}^{N} q(t)\, L_t = \frac{\sum_t w(t)\, L_t}{\sum_j w(j)}.$$

Importance-corrected positional random sampling. For completeness, we note that positional random sampling can also be made an unbiased estimator of the full (unweighted) KD objective via importance correction. Specifically, if each sampled loss is reweighted by the inverse sampling probability, $\widehat{\mathcal{L}}_{\text{IC}} = \frac{1}{KN}\sum_{k=1}^{K} \frac{L_{t_k}}{q(t_k)}$, then $\mathbb{E}[\widehat{\mathcal{L}}_{\text{IC}}] = \frac{1}{N}\sum_{t=1}^{N} L_t$, recovering Full KD in expectation. We referred to this variant as *importance-corrected positional random sampling* and evaluated it separately in our experiments.

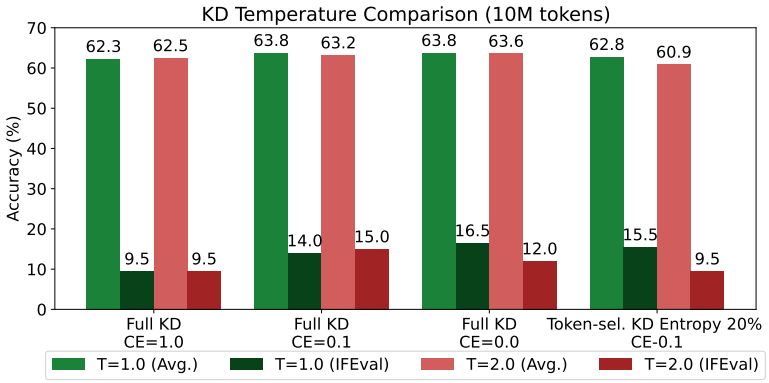

Figure 4. **Temperature ablation for Full KD.** We compare T=2.0 vs. T=1.0 and report average accuracy over five benchmarks (ArcEasy, GSM8K, HellaSwag, PIQA, and LAMBADA OpenAI).

B. Hyperparameters

Tables 8 and 9 list the hyperparameter choices shared across all runs and the settings that differ between distillation variants.

Component	Value
Teacher model	Qwen/Qwen3-8B (online, no quantization)
Student model	Qwen/Qwen3-1.7B
Dataset	FineWeb-Edu stream (80M tokens)
Sequence length	512 tokens (`max_seq_len=512`)
Epochs	1 pass over the streamed subset
Mini-batch	batch size 2 $\times$ 8 gradient accumulation steps (effective batch 16)
Optimizer	bitsandbytes Adam8bit (lr 1×10^{-5})
KD temperature	1.0
CE mixing weight	$\alpha_{\text{CE}} = 0.0$ (pure KL divergence loss)
Offline cache	Enabled with $U = 64$ cached classes
Seeds	1337, 1338, 1339 (or 1340, 1341, 1342 for the GSM8K setup)

Table 8. Shared hyperparameters across all experiments.

Variant	Additional settings
Full KD	Distills all tokens ($k = 100\%$).
SE-KD (student entropy top-k)	$k = 20\%$; selection normalized by sequence length
Curriculum Learning	`SELECTION_CURRICULUM_STEPS`= 4000.
Random token selection	$k = 20\%$; uniform random token selection, normalized by length.
Pos-RS-KD	$k = 20\%$; student entropy scoring; `POS_RS_MATCH_FULL_KD`= 1 for corrected variant.

Table 9. Settings specific to each distillation variant reported in the main tables.

C. Additional Results

This section reports auxiliary experiments that motivate the hyperparameter choices used throughout the paper. We compare temperature settings and cross-entropy mixing weights for the Full KD baseline. Across these ablations, temperature T=1.0 mostly outperforms higher temperatures, and the cross-entropy component provides negligible benefit; moreover, including CE would prevent some of our selection-based efficiency optimizations (e.g., restricting gradient-carrying logits to selected positions). Accordingly, we use T=1.0 and set λ=1 in Eq. 1 (pure KL) in all main experiments.

D. Positional Random Sampling Underperformance

Fig. 6 visualizes the difference between deterministic Top-$k\%$ position selection and positional random sampling (Pos RS-KD) at the same budget. While Pos RS-KD is attractive because it introduces stochasticity according to an uncertainty-

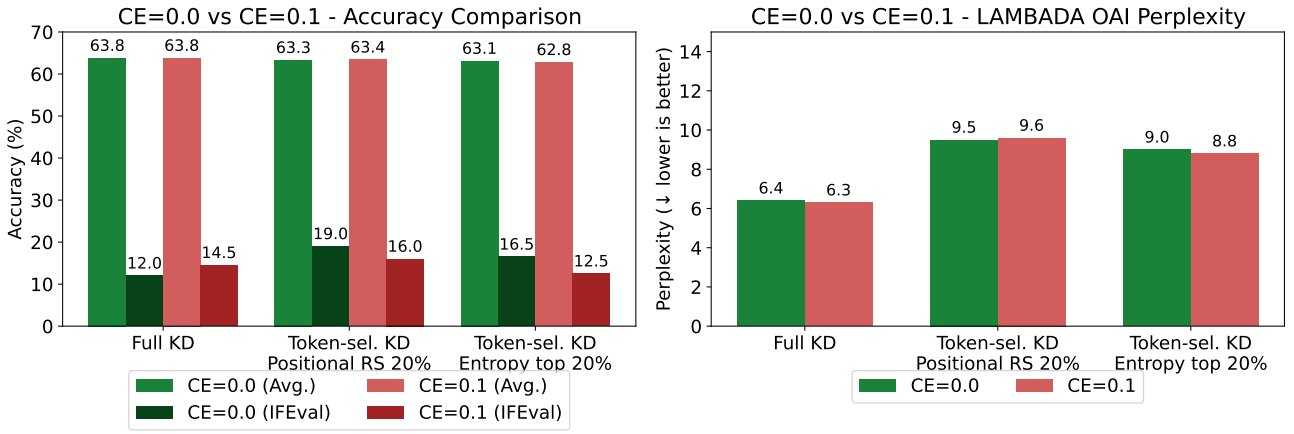

Figure 5. **Cross-entropy mixing ablation for Full KD.** We compare α_{CE}=0.1 vs. α_{CE}=0.0 and report the same average accuracy metric. This study uses a smaller 10M-token run and is included as a sanity check rather than a fully converged comparison.

derived weight, it underperformed Top-k in both accuracy and calibration in our general-purpose setting (Table 3).

A possible explanation is entropy-mass concentration within a sequence: if the per-sequence entropy distribution is highly peaked, then sampling proportionally to entropy can allocate a large fraction of the budget to a small set of extreme-entropy positions (often near the beginning of the sequence).This can reduce coverage of other informative positions that Top-k would deterministically include, and may increase variance across updates.

There are several simple mitigations that may improve Pos RS-KD in future work: (i) *temperature smoothing* of the sampling distribution (e.g., sampling from $\propto H(q_t)^{1/T}$ with $T > 1$) to flatten overly-peaked sequences; (ii) lightweight *heuristics* such as excluding the first few eligible positions or clipping extreme entropies; and (iii) combining entropy-proportional sampling with a small deterministic "coverage" component (e.g., reserving part of the budget for Top-$k\%$ and sampling the remainder). We leave a systematic study of these variants to future work.

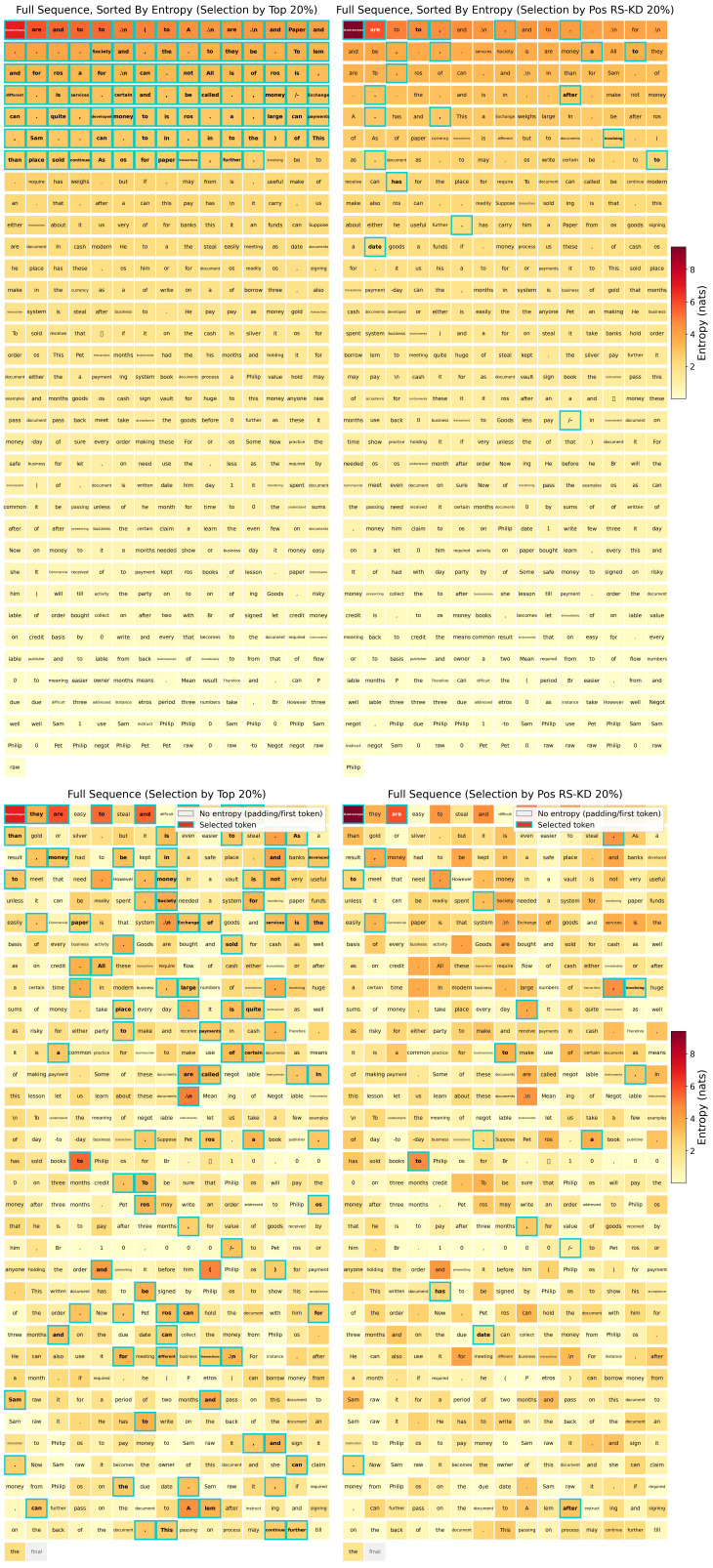

Figure 6. **Top-k vs. positional random sampling at a fixed budget (k=20%).** Tokens are colored by student entropy; teal outlines mark selected positions. **Top row:** the same sequence sorted by entropy, highlighting how each policy allocates its budget across the entropy distribution. **Bottom row:** the original token order (with padding shown as "no entropy"), showing how selections are distributed along the sequence. **Left:** deterministic Top-k; **Right:** entropy-proportional positional RS-KD (Pos RS-KD / Pos RS-KD*).

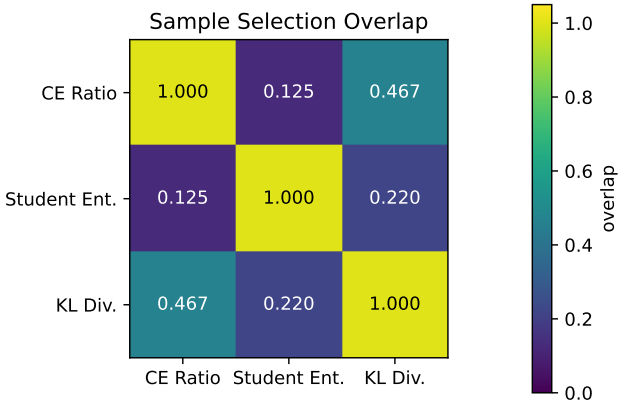

Figure 7. **Sample-selection overlap across metrics.** Pairwise overlap between samples selected by student entropy, CE ratio, and KL divergence. Teacher-based metrics show higher mutual overlap than with student entropy, indicating differing selection stability.

E. The Offline Cache Tradeoff

The cache footprint of SE-KD$_{3X}$ could be further reduced by storing teacher logits only for a fixed subset of selected positions, scaling storage with the position budget $k\%$. However, this introduces a fundamental tradeoff: position-level caching maximizes storage savings but breaks adaptivity, whereas sample-level caching preserves curriculum effects at the cost of a larger cache. We therefore avoid position-level caching, as our default student-entropy selector induces an implicit curriculum—high-entropy positions evolve during training, and freezing a precomputed mask would remove this adaptivity and may degrade distillation quality.

In contrast, we hypothesize that *sample-level* selection is more stable under student learning dynamics, making it suitable for a one-shot prefiltering pass that reduces teacher queries and cache size. This hypothesis is consistent with Xu et al. (2023), who show that samples uncertain for the student are also hard for the teacher, suggesting that sample difficulty is largely data-inherent. However, we do not claim to establish this conclusively. As shown in Fig. 7, an exploratory overlap analysis reveals substantial agreement between teacher-based metrics (KL and CE ratio; 46.7%), but much lower overlap between student entropy and either metric (22.0% and 12.5%), highlighting the need for a more systematic study of sample-selection stability.

An alternative is to construct the cache online during distillation, recording positions or samples selected by the evolving student. While this preserves curriculum effects and may transfer across students, it sacrifices a key benefit of offline caching—the ability to distill while holding only one model in memory—and is less suitable for multi-epoch training. Future work could compare samples selected under online curricula (e.g., GLS) to those from a pre-distillation pass to better characterize selection stability.

F. Validation Split Tables

This appendix reports validation-split results used for model/metric selection and ablations during development. All comparisons in the main paper are based on the held-out test split; the validation split is not used for final evaluation. We evaluated each validation experiment using three seeds (1337, 1338, 1339) and report the average results. The validation benchmark suite is constructed from the average accuracy across the validation splits of ArcEasy, GSM8K, HellaSwag, and PIQA.

Table 10. **Position-importance metrics with Top-20% selection on the validation set**, averaged across three fixed seeds (1337, 1338, 1339), trained on 80M tokens of FineWeb-Edu. Based on these results, we selected student entropy as our position-importance metric (Table 2). It achieves top validation accuracy and uniquely among the top metrics, requires no teacher-side information, enabling the use of a selective LM head on the teacher that avoids logit computation at non-selected positions.

Method	**Acc.** ↑
Qwen3 1.7B	62.2
Qwen3 8B	***75.1***
AT-KD	65.2
Full KD	65.6
Random 20%	65.5
Position selection policy: Top 20%	
Student entropy	**66.0**
Teacher entropy	65.4
Student CE	65.8
Teacher CE	65.3
KL	**66.0**
Reverse KL	**66.0**
CE ratio	65.9
CE ratio + Student Entropy	65.8
Student entropy + KL	65.7

Table 11. **Position-selection policies with student entropy on the validation set**, averaged across three seeds. We selected Top 20% for our main experiments (Table 3): although GLS and Curriculum achieve slightly higher validation accuracy, the differences are small (0.1–0.2 points) and Top 20% is simpler, avoiding additional hyperparameters (queue size for GLS, schedule for Curriculum).

Method	**Acc.** ↑
Qwen3 1.7B	62.2
Qwen3 8B	***75.1***
AT-KD	65.2
Full KD	65.6
Random 20%	65.5
Position selection policy: Top 20%	
Top 20%	66.0
Curriculum Learning 20%	66.1
GLS 30K 20%	**66.2**
Pos RS-KD 20%	64.9
Pos RS-KD* 20%	65.5

G. Standard Deviations

We report standard deviations over three fixed random seeds to quantify run-to-run variance under an otherwise identical training setup.

Table 12. Standard deviations for Table 2 (position-importance metrics with Top-20% selection), computed over three fixed seeds (1337, 1338, 1339).

Method	**Accuracy** ↑	**IFEval** ↑	**PPL** ↓	**ECE** ↓
Full KD	0.20	0.56	0.18	0.07
RandomPos 20%	0.15	1.25	0.21	0.59
AT-KD	0.04	0.78	0.06	0.04
Position selection policy: Top 20%				
Student Entropy (SE-KD)	0.14	0.81	0.26	0.12
Teacher Entropy	0.68	0.24	1.28	0.52
Teacher CE	0.30	0.67	0.38	0.79
Student CE	0.16	0.36	0.35	0.20
KL	0.16	1.19	0.09	0.11
Reverse KL	0.10	0.45	0.09	0.04
CE ratio	0.02	0.12	0.34	0.06
CE ratio + Student Entropy	0.06	0.44	0.04	0.07
Student Entropy + KL	0.13	0.13	0.48	0.49

Table 13. Standard deviations for Table 3 (position-selection policies with student entropy), computed over three fixed seeds (1337,1338,1339).

Method	**Accuracy** ↑	**IFEval** ↑	**PPL** ↓	**ECE** ↓
Full KD	0.20	0.56	0.18	0.07
RandomPos 20%	0.15	1.25	0.21	0.59
AT-KD	0.04	0.78	0.06	0.04
Position importance metric: Student entropy, $k = 20\%$				
Top 20% (SE-KD)	0.14	0.81	0.26	0.12
Top 20% GLS 30K	0.23	0.33	0.55	0.14
Curriculum 20%	0.09	0.41	0.21	0.08
Pos RS-KD* 20%	0.39	0.36	0.44	0.10
Pos RS-KD 20%	0.08	0.99	0.26	0.15

Table 14. Standard deviations for Table 4 (general-purpose distillation; test split, 80M tokens), computed over three fixed seeds.

Method	**Accuracy (%)** ↑	**IFEval (%)** ↑	**PPL** ↓	**ECE (%)** ↓
Full KD	0.20	0.56	0.18	0.07
RandomPos 20%	0.15	1.25	0.21	0.59
RandomSmp 20%	0.27	0.06	0.71	0.03
SE-KD	0.13	0.48	0.26	0.08
RS-KD	0.06	5.33	0.06	0.01
TopSmp 20%	0.58	0.48	0.64	0.05
RS-KD + TopSmp 20%	0.11	0.21	0.00	0.04
SE-KD + TopSmp 20%	0.07	0.77	0.27	0.11
SE-KD_{3X}	0.15	1.61	0.12	0.16

Table 15. Standard deviations for Table 5 (task-specific GSM8K distillation; test split), computed over three fixed seeds (1340, 1341, 1342).

	Method	GSM8K Acc.	Acc.	PPL
Off Policy Distill.	Full KD	0.90	0.10	0.06
	Random 20%	0.10	0.12	0.15
	SE-KD	0.12	0.06	0.10
	Pos-RS-KD 20%	0.80	0.17	0.23
	Pos RS-KD* 20%	1.22	0.36	0.31
	TopSmp 20%	0.12	0.00	0.06
	SE-KD + TopSmp 20%	0.06	0.06	0.06
	SE-KD$_{3X}$	0.31	0.15	0.00
On Policy Distill.	Full KD	0.21	0.00	0.06
	Random 20%	0.72	0.06	0.12
	SE-KD	0.15	0.00	0.00
	Pos-RS-KD 20%	2.25	0.00	0.58
	Pos-RS-KD* 20%	0.58	0.06	0.21
	TopSmp 20%	0.49	0.06	0.12
	SE-KD + TopSmp 20%	1.00	0.31	0.66

Table 16. **Memory and speed comparison of selective LM head configurations.** Experiments use Qwen3-8B (teacher) → Qwen3-1.7B (student) on NVIDIA GeForce RTX 3090 GPUs with batch size B=2, sequence length T=512, and α_{CE}=0. *Default flow* corresponds to the standard KD implementation. *Chunked-streaming flow* restructures the computation to match the selective implementation (e.g., streaming entropy computation and position indexing) while selecting all positions (k=100%), isolating the overhead of code reorganization. At k=20%, only the top 20% of positions (by student entropy) participate in the KD loss. Speedup is reported relative to the default flow baseline.

Configuration	k	Student Peak (GB)	Teacher Peak (GB)	Wall Time	Speedup
1M tokens					
Full KD (default flow)	100%	15.88	17.30	16.3 min	1.00×
Full KD (chunked-streaming flow)	100%	14.15	17.58	14.9 min	1.09×
Teacher selective LM head + chunked-streaming flow	100%	14.15	17.29	14.9 min	1.09×
No selective LM head (default flow)	20%	13.59	17.30	13.6 min	1.20×
No selective LM head (chunked-streaming flow)	20%	**11.42**	15.97	12.6 min	1.29×
Chunked-streaming flow					
Teacher selective LM head	20%	**11.42**	**15.68**	12.6 min	1.29×
Student selective LM head	20%	**11.42**	15.97	12.2 min	1.34×
Teacher + Student selective LM head	20%	**11.42**	**15.68**	**12.0 min**	1.36×
5M tokens					
Full KD (default flow)	100%	15.88	17.30	79.2 min	1.00×
Full KD (chunked-streaming flow)	100%	14.15	17.58	72.0 min	1.10×
Chunked-streaming flow					
Teacher selective LM head	20%	**11.42**	**15.68**	62.5 min	1.27×
Student selective LM head	20%	**11.42**	15.97	61.8 min	1.28×
Teacher + Student selective LM head	20%	**11.42**	**15.68**	**60.1 min**	1.32×

H. Memory Efficiency of Selective LM Head and Chunked Streaming Entropy Computation

Table 16 presents a detailed ablation of memory usage and training speed across different KD implementations. Specifically, we compare:

1. **Default KD implementation**: Standard KD that computes full $[B, L, V]$ logits for both teacher and student.
2. **Chunked-streaming implementation**: Incorporates chunked-streaming entropy computation and the selective code path, while still computing logits at all positions. This isolates the effect of chunked streaming independent of position sparsification.
3. **Selective LM head variants**: Compute KD loss on a subset of positions selected by student entropy, with teacher- and/or student-side selective LM heads restricting logit computation and gradient propagation to selected positions.

Several observations emerge from Table 16. First, even at k=100%, the chunked-streaming flow reduces student peak memory from 15.88 GB to 14.15 GB (11%) and yields a 9% speedup by avoiding materialization of full student logit tensors during backpropagation. Second, reducing k from 100% to 20% provides substantial additional savings: even without a selective LM head, student peak memory drops to 13.59 GB (default flow) or 11.42 GB (chunked-streaming flow), with speedups of 1.20× and 1.29×, respectively. Third, adding a selective LM head at k=20% further reduces teacher peak memory from 15.97 GB to 15.68 GB while maintaining the same student memory footprint; the combined teacher + student selective configuration achieves the best wall time (12.0 min, 1.36× speedup).

Fig. 8 visualizes these effects over time. At k=100% (left panel), memory spikes arise from transient allocation of full $[B, L, V]$ logit tensors during each training step. Reducing to k=20% without a selective LM head (middle panel) already lowers peak memory, as fewer positions participate in the KD loss, though full logits are still materialized. With a selective LM head at k=20% (right panel), the spikes are eliminated entirely, as logits are computed only at the selected ∼20% of positions.

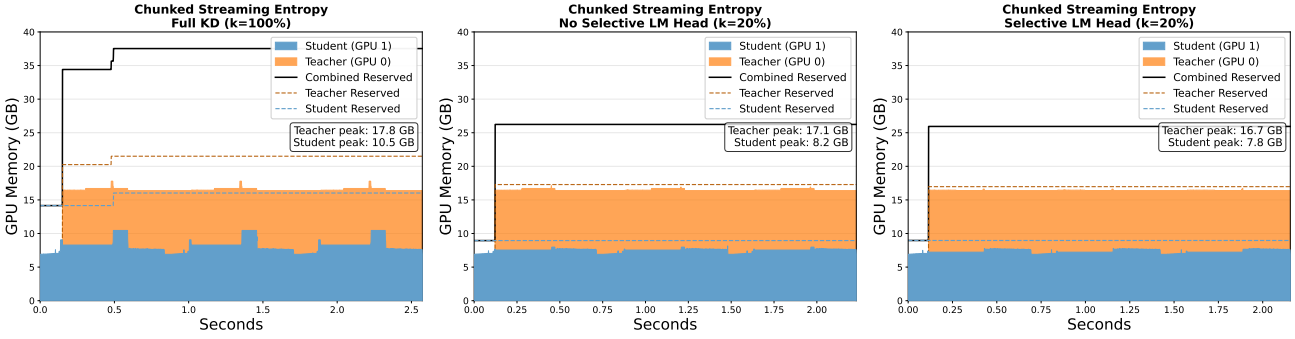

Figure 8. **GPU memory profiles under different selective LM head configurations.** Memory traces from PyTorch profiler over several training steps using chunked-streaming entropy computation. **Left:** Full KD with k=100%, where allocating full $[B, L, V]$ logit tensors induces periodic memory spikes. **Middle:** k=20% without selective LM head, where fewer positions participate in KD but full logits are still materialized. **Right:** k=20% with selective LM head, where logits are computed only at selected positions, eliminating transient spikes and further reducing peak memory. Teacher and student run on separate GPUs; stacked areas show allocated memory and dashed lines indicate reserved memory.