RETHINKING CROSS-TOKENIZER ON-POLICY DISTILLATION: FROM ALIGNMENT COVERAGE TO SUPERVISION RELIABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

On-Policy Distillation (OPD) trains a student on its own generations using teacher feedback. With different tokenizers, comparing teacher and student predictions requires alignment at both sequence and vocabulary levels. In this paper, we examine whether expanding this alignment coverage improves learning. Across three heterogeneous teacher–student pairs on mathematical reasoning and code generation, strict 1:1 groups already cover most student-generated tokens despite substantial vocabulary mismatch. On responses sampled from the students before distillation, the shared vocabulary retains nearly all teacher and student probability mass at strictly aligned positions on average. Restricting reverse KL to a student-selected top-16 subset of the shared vocabulary at each strict position achieves accuracy comparable to full shared-vocabulary OPD, outperforming the evaluated cross-tokenizer baselines. Adding mean squared error supervision on span log-probabilities in mismatch groups gives complete supervision coverage, yet reduces accuracy. At checkpoints from training with only the strict loss, the span gradients show weak or negative directional agreement with the strict gradients and grow in magnitude relative to them. These diagnostics may help explain the accuracy drop from adding span supervision. Our findings motivate a shift from maximizing *alignment coverage* to prioritizing *supervision reliability*: compact supervision at strict positions can be more effective than broader coverage that introduces weakly aligned or conflicting training signals.

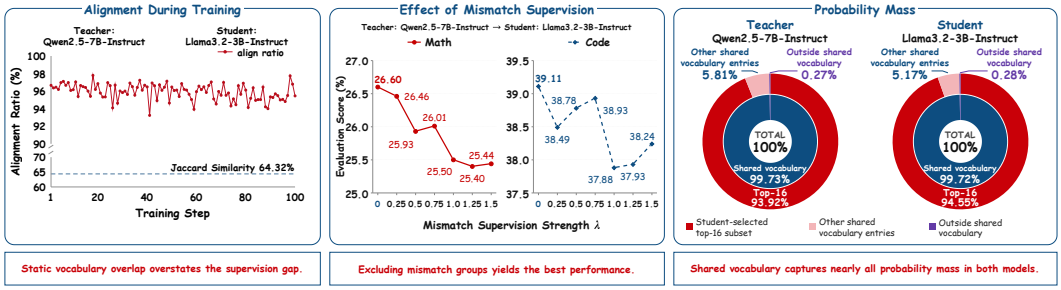


Figure 1: Three diagnostics of Cross-Tokenizer OPD for Qwen2.5-7B-Instruct→Llama-3.2-3B-Instruct. **Left:** The logged student-side alignment ratio remains high during Strict full training despite substantially lower static vocabulary overlap. **Middle:** Adding supervision on mismatch groups lowers mathematics and code performance. **Right:** At strictly aligned positions in student-generated responses before distillation, the shared vocabulary retains nearly all teacher and student probability mass on average, and a student-selected top-16 subset retains most of it on both sides.

1 INTRODUCTION

On-Policy Distillation (OPD) trains a language model on its own generations using feedback from a teacher (Agarwal et al., 2024; Gu et al., 2024). By supplying supervision at the prefixes reached by

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.

the current student, OPD connects teacher feedback to the student's evolving behavior. Extending this method across model families requires comparing predictions from models with different tokenizers. The same response can have different token boundaries, and the next token distributions are defined over different vocabularies. Cross-Tokenizer OPD must therefore account for alignment at both sequence and vocabulary levels.

Recently, Cross-Tokenizer distillation addresses these obstacles through rank-based distribution matching, learned mappings, likelihood matching, and byte-level interfaces (Boizard et al., 2025; Zhang et al., 2024; Minixhofer et al., 2025; Singh et al., 2026). For on-policy training, SimCT recovers supervision through aligned multi-token units and reports gains over shared-vocabulary OPD (Sun et al., 2026). Byte-Prefix Marginalization maps teacher probability mass onto student tokens, while also identifying harmful supervision at whitespace positions (Wang et al., 2026). These developments make additional teacher signals accessible and motivate examining their learning value. We ask: *how much useful supervision does strict matching already retain, and what does recovering the excluded targets contribute to learning?*

Alignment coverage describes how much of the response and the models' vocabularies can be compared across tokenizers. *Supervision reliability* concerns whether the resulting teacher targets provide useful guidance for student learning at the states it visits. Static vocabulary overlap counts vocabulary entries equally, whereas their frequency and predictive probability on student trajectories can be highly uneven. A large vocabulary gap can therefore coexist with frequent strict alignment and concentrated probability mass on shared vocabulary entries. For spans that lack strict alignment, multi-token grouping makes additional supervision possible. Whether this additional supervision improves learning depends on the objective applied to the aligned groups.

Prior OPD studies show that concentrated prediction supports can retain substantial distillation benefits (Fu et al., 2026; Li et al., 2026). Under heterogeneous tokenizers, evaluating such supports also requires accounting for which response positions and vocabulary entries admit direct comparison. We evaluate the learning value of this supervision through training comparisons, using probability-mass measurements and gradient diagnostics to interpret the outcomes.

Our study covers three heterogeneous teacher–student pairs on mathematical reasoning and code generation, including instruction-tuned and base students. We start from strict Cross-Tokenizer OPD, which applies reverse KL at strictly aligned positions. At these positions, one student token and one teacher token cover the same span of the response. Both predictive distributions are restricted and renormalized on the shared vocabulary. We then vary the weight of a mean squared error (MSE) loss on span log-probabilities and the number of shared vocabulary entries used at each strict position. Figure 1 shows the coverage, mismatch-supervision, and probability-mass observations. Our main findings are:

- **Closing the coverage gap with span MSE reduces accuracy.** Strict student-token coverage is 85.57–96.98% across the studied runs, despite static vocabulary Jaccard overlap of 39.49–64.87%. Holding the strict objective fixed, we add MSE supervision on span log-probabilities. Every positive weight gives complete structural supervision coverage, yet all 18 tested positive-weight settings lower the full-average accuracy (Section 3).

- **A student-selected top-k subset retains most of the learning benefit.** On responses sampled from the students before distillation, the shared vocabulary retains nearly all teacher and student probability mass at strict positions on average. The selected subsets also retain most of this mass for both models. With $k = 16$, training preserves at least 96% of the full-average improvement achieved by full shared-vocabulary OPD over the undistilled student. The full and compact strict objectives attain higher full averages than the evaluated cross-tokenizer baselines on all three model pairs (Section 4).

- **Span gradients have weak directional agreement and growing relative magnitude.** At checkpoints from training with only the strict loss, the span gradients show weak or negative agreement with strict gradients. Their agreement is consistently below a reference formed by splitting the positions contributing to the strict loss into two subsets, and the span-to-strict gradient norm ratio increases across the measured training checkpoints. The growing relative scale of these weakly aligned gradients may help explain the lower accuracy under span supervision (Section 5).

A student-selected top-k subset of the shared vocabulary retains most of the observed distillation gains, while the span objective expands structural supervision coverage at the cost of accuracy. These findings motivate a shift from *alignment coverage* to *supervision reliability*: the value of additional supervision should be assessed through its interaction with strict distribution matching and its effect on downstream performance.

2 PRELIMINARIES

2.1 NOTATION

Let $x \sim \mathcal{D}$ denote a *prompt* and $y = (y_1, \ldots, y_L)$ denote a *response* token sequence, where $y_{<i} = (y_1, \ldots, y_{i-1})$ is the *prefix* of y. We consider two LLMs: a student model π_θ and a frozen teacher model π_{T}, where π_θ defines a distribution $\pi_\theta(\cdot|x)$ over vocabulary $\mathcal{V}_\theta$, and π_{T} defines a distribution $\pi_{\mathrm{T}}(\cdot|x)$ over vocabulary $\mathcal{V}_{\mathrm{T}}$. A *trajectory* is a pair (x, y) with y sampled autoregressively from the policy given x.

2.2 ON-POLICY DISTILLATION

On-Policy Distillation (OPD) samples trajectories from the student and aligns the student to the teacher on the prefixes the student actually visits (Agarwal et al., 2024). OPD can also be viewed as a special case of dense KL-constrained reinforcement learning, where the teacher distribution induces a token-level reward and the KL regularizer has a fixed relative weight (Yang et al., 2026). With a shared tokenizer, OPD minimizes the following objective:

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}\left[\sum_{i=1}^{L} \mathrm{KL}(\pi_\theta(\cdot|x, y_{<i}) \| \pi_{\mathrm{T}}(\cdot|x, y_{<i}))\right]. \tag{1}$$

2.3 CROSS-TOKENIZER ON-POLICY DISTILLATION

Different tokenizers can assign different token boundaries and vocabulary entries to the same text. Cross-Tokenizer OPD must therefore account for alignment at both sequence and vocabulary levels (Wang et al., 2026).

Token-group alignment. For scoring and alignment, we tokenize the decoded student response with both models' tokenizers. We denote the resulting student and teacher response-token sequences by $y = (y_1, \ldots, y_L)$ and $v = (v_1, \ldots, v_n)$, respectively. The implementation preserves the positions of these response tokens in each model's encoded input, as detailed in Appendix A. We then retain the token offsets shared by both sequences, including the start and end of the response. For the r-th interval between consecutive shared boundaries, let S_r^θ and S_r^{T} contain the student and teacher token indices within that interval, respectively. The tokens in these two aligned token groups concatenate to the same span. Applying this construction to every interval partitions the response into R pairs of aligned token groups:

$$\mathcal{S}(y) = \left\{\left(S_r^\theta, S_r^{\mathrm{T}}\right)\right\}_{r=1}^{R}. \tag{2}$$

We divide the group indices into those of strict 1:1 groups and mismatch groups:

$$\mathcal{A}_{1:1}(y) = \left\{r : |S_r^\theta| = |S_r^{\mathrm{T}}| = 1\right\}, \quad \mathcal{A}_{\mathrm{mis}}(y) = \{1, \ldots, R\} \setminus \mathcal{A}_{1:1}(y). \tag{3}$$

A strict 1:1 group consists of one student token and one teacher token spanning the same interval of the response. We refer to their paired prediction positions as *strict positions*. A mismatch group requires multiple tokens on at least one side to construct the same span.

Shared vocabulary. We identify vocabulary entries across models by their underlying tokens, excluding special tokens and ambiguous matches. Under this convention, the shared vocabulary is $\mathcal{V}_\cap = \mathcal{V}_\theta \cap \mathcal{V}_{\mathrm{T}}$. For each $r \in \mathcal{A}_{1:1}(y)$, write $S_r^\theta = \{i_r\}$ and $S_r^{\mathrm{T}} = \{j_r\}$. The student and teacher predict according to $\pi_\theta(\cdot \mid x, y_{<i_r})$ and $\pi_{\mathrm{T}}(\cdot \mid x, v_{<j_r})$, respectively. For $w \in \mathcal{V}_\cap$, we use w to

index its corresponding vocabulary entry in each distribution. Restricting and renormalizing both distributions over the shared vocabulary gives

$$\bar{\pi}_\theta(w \mid x, y_{<i_r}) = \frac{\pi_\theta(w \mid x, y_{<i_r})}{\sum_{w' \in \mathcal{V}_\cap} \pi_\theta(w' \mid x, y_{<i_r})}, \bar{\pi}_{\mathrm{T}}(w \mid x, v_{<j_r}) = \frac{\pi_{\mathrm{T}}(w \mid x, v_{<j_r})}{\sum_{w' \in \mathcal{V}_\cap} \pi_{\mathrm{T}}(w' \mid x, v_{<j_r})}. \quad (4)$$

Strict Cross-Tokenizer objective. Following the OPD objective above, we sum reverse KL over strictly aligned positions using the shared-vocabulary distributions:

$$\mathcal{L}_{1:1}(\theta) = \mathbb{E}_{x \sim \mathcal{D},\, y \sim \pi_\theta(\cdot|x)} \left[\sum_{r \in \mathcal{A}_{1:1}(y)} \mathrm{KL}\left(\bar{\pi}_\theta(\cdot \mid x, y_{<i_r}) \| \bar{\pi}_{\mathrm{T}}(\cdot \mid x, v_{<j_r})\right) \right]. \quad (5)$$

3 Revisiting Supervision Coverage

Figure 1 presents three observations for Qwen→Llama: strict alignment remains high during training, adding span supervision lowers the reported downstream scores, and most predictive mass at strict positions lies on shared vocabulary entries. In this section, we first measure strict token coverage, then test the learning effect of supervising mismatch groups, and finally measure the probability mass retained by the shared vocabulary.

Experimental setting. The experiments cover four model series: Qwen (Yang et al., 2025), Llama (Grattafiori et al., 2024), Granite[1], and Phi (Abouelenin et al., 2025). Throughout, arrows point from teacher to student. We study Qwen2.5-7B-Instruct→Llama-3.2-3B-Instruct, Granite-4.1-8B→Phi-4-mini-instruct, and Granite-4.1-8B→Qwen2.5-7B-Base, abbreviated as Qwen→Llama, Granite→Phi, and Granite→Qwen, respectively. Training starts from a common source pool of 20,000 prompts, comprising 10,000 mathematics prompts from DAPO-Math-17K (Yu et al., 2025) and 10,000 code prompts from CodeForces[2]. We evaluate mathematics on MATH500 (Lightman et al., 2024), GSM8K (Cobbe et al., 2021), AIME-2024, AIME-2025, AIME-2026, AMC23, and Minerva-Math (Lewkowycz et al., 2022), and code on HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), and LiveCodeBench (Jain et al., 2025). We report accuracy on all benchmarks, averaging over 32 sampled responses per problem for mathematics (mean@32) and 8 for code (mean@8). The *full* gives equal weight to the arithmetic means within the two domains. Appendix A provides experimental details.

3.1 Strict Alignment on Student Trajectories

Static overlap and dynamic coverage. We compare static vocabulary Jaccard overlap J_{vocab} with strict token coverage on student rollouts. Appendix B gives the static overlap definition and complete vocabulary statistics. For the response y, strict student-token coverage is the fraction of non-special student tokens belonging to strict 1:1 groups:

$$C^\theta_{1:1}(y) = \frac{\sum_{r \in \mathcal{A}_{1:1}(y)} |S^\theta_r|}{\sum_{r=1}^{R} |S^\theta_r|}. \quad (6)$$

Strict teacher-token coverage $C^{\mathrm{T}}_{1:1}(y)$ replaces S^θ_r with S^{T}_r in both sums.

Strict supervision remains prevalent. All 512 student rollouts generated at each of the 100 iterations are saved, giving 51,200 responses per run. We aggregate coverage over the complete run and three disjoint windows: steps 1–20, 41–60, and 81–100. Table 1 contrasts Jaccard overlap of 39.49–64.87% with strict token coverage of 85.57–96.98% of student tokens and 82.82–97.26% of teacher tokens over the complete runs. Granite→Phi has the lowest Jaccard overlap but the highest coverage under both tokenizations. Within each pair, the range across the three reported windows is at most 1.43 percentage points for strict student-token coverage and 3.75 percentage points for strict teacher-token coverage. These results show that substantial static vocabulary mismatch can coexist with strict alignment at most positions on student-generated trajectories. High strict token coverage alone, however, does not establish whether the remaining mismatch groups offer useful supervision.

[1] https://huggingface.co/blog/ibm-granite/granite-4-1

[2] https://huggingface.co/datasets/open-r1/codeforces

Table 1: Static vocabulary Jaccard overlap J_{vocab} and strict token coverage on student trajectories from Strict full training. $C^{\theta}_{1:1}$ and $C^{\text{T}}_{1:1}$ denote student and teacher strict token coverage over the indicated training steps.

Teacher→Student	J_{vocab}	Coverage	Training steps			
			1–100	1–20	41–60	81–100
Qwen→Llama	64.32	$C^{\theta}_{1:1}$	93.56	94.63	93.20	93.43
		$C^{\text{T}}_{1:1}$	85.91	88.65	84.90	85.36
Granite→Phi	39.49	$C^{\theta}_{1:1}$	96.98	96.65	96.95	97.12
		$C^{\text{T}}_{1:1}$	97.26	95.83	97.50	97.96
Granite→Qwen	64.87	$C^{\theta}_{1:1}$	85.57	85.23	85.06	85.88
		$C^{\text{T}}_{1:1}$	82.82	81.15	82.77	83.86

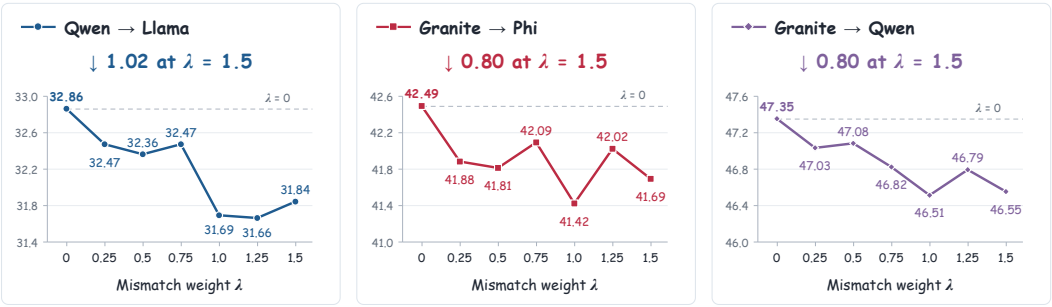


Figure 2: Full-average accuracy (%) across mismatch weights. Dashed lines mark strict supervision ($\lambda = 0$), and annotations give the decrease at $\lambda = 1.5$ in percentage points.

3.2 Learning Value of Mismatch Supervision

We test the learning value of the remaining mismatch groups by adding span supervision to the strict objective in Equation 5.

Span supervision. Each mismatch group in Equation 2 pairs student and teacher token sequences for the same text span, potentially involving different numbers of autoregressive decisions. For $r \in \mathcal{A}_{\text{mis}}(y)$, we compute the probabilities of these observed token paths as

$$q_{\theta}^{(r)} = \prod_{i \in S_r^{\theta}} \pi_{\theta}(y_i \mid x, y_{<i}), \quad q_{\text{T}}^{(r)} = \prod_{j \in S_r^{\text{T}}} \pi_{\text{T}}(v_j \mid x, v_{<j}). \tag{7}$$

We match these probabilities with a mean squared error (MSE) loss over mismatch groups, using a log-probability formulation:

$$\mathcal{L}_{\text{span}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_{\theta}(\cdot|x)} \left[\sum_{r \in \mathcal{A}_{\text{mis}}(y)} \left(\log q_{\theta}^{(r)} - \log q_{\text{T}}^{(r)} \right)^2 \right]. \tag{8}$$

The total loss is

$$\mathcal{L}_{\lambda}(\theta) = \mathcal{L}_{1:1}(\theta) + \lambda \mathcal{L}_{\text{span}}(\theta). \tag{9}$$

The baseline $\lambda = 0$ applies supervision only to strict 1:1 groups. Every $\lambda > 0$ additionally supervises all mismatch groups. All positive weights therefore give 100% complete supervision coverage. Varying λ changes the influence of the span loss while keeping the set of included groups fixed for any given response. We sweep $\lambda \in \{0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5\}$ for all three teacher–student pairs, as shown in Figure 2.

Strict supervision performs best throughout the weight sweep. All three pairs attain their highest full average at $\lambda = 0$. The 18 positive-weight settings score 0.27–1.20 percentage points below their respective strict baselines. The detailed results in Appendix C show that $\lambda = 0$ also gives the highest math and code averages for each pair. Figure 2 shows an overall downward trend in full-average accuracy as λ increases across all three pairs. With span log-probability MSE, achieving complete supervision coverage reduces downstream accuracy across the tested weights. This extends the distinction between coverage and learning value in Figure 1 to all three model pairs.

3.3 Predictive Mass Concentrates on the Shared Vocabulary

We now measure the probability mass that each model assigns to the shared vocabulary at strict positions. Static vocabulary overlap counts vocabulary entries, but their contribution to the predictive distributions depends on the probability assigned to them in context.

Measuring probability mass. For each strict 1:1 group $r \in \mathcal{A}_{1:1}(y)$, abbreviate the original full-vocabulary predictions at its strict positions as $\pi_\theta^{(r)}(w) = \pi_\theta(w \mid x, y_{<i_r})$ and $\pi_{\mathrm{T}}^{(r)}(w) = \pi_{\mathrm{T}}(w \mid x, v_{<j_r})$. For $a \in \{\theta, \mathrm{T}\}$, the probability mass on the shared vocabulary is

$$M_{\cap,a}^{(r)} = \sum_{w \in \mathcal{V}_\cap} \pi_a^{(r)}(w). \tag{10}$$

To measure concentration within the shared vocabulary, we select top-k shared vocabulary entries to which it assigns the highest probabilities. These entries form the *student-selected top-k subset*:

$$\mathcal{K}_k^{(r)} = \mathrm{TopK}_{w \in \mathcal{V}_\cap}\left(\pi_\theta^{(r)}(w), k\right), \quad M_{k,a}^{(r)} = \sum_{w \in \mathcal{K}_k^{(r)}} \pi_a^{(r)}(w). \tag{11}$$

Unlike the shared vocabulary, this subset depends on the student distribution at each strict position. When used as the support of the distillation loss, we also refer to it as the *top-k support*.

For each model pair, the probability-mass probe uses all 20k prompts. We use the student before distillation to generate one rollout per prompt, scoring both models at each measured strict position. Both models are evaluated on the same student-selected top-k subset using their original softmax probabilities. We average each statistic over all measured strict positions and evaluate $k \in \{4, 8, 16, 32, 64, 128\}$ alongside the full shared vocabulary. Appendix B reports the detailed results.

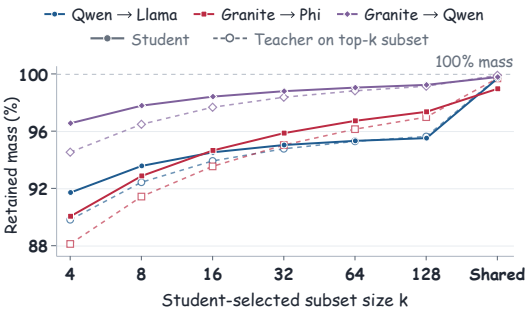


Figure 3: Probability mass retained by student-selected top-k subsets and the full shared vocabulary at strict positions before distillation.

Shared vocabulary captures nearly all predictive mass. The "Shared" endpoints in Figure 3 show that the full shared vocabulary retains 99.69–99.90% of teacher mass and 98.99–99.81% of student mass on average across the three pairs. Strict alignment makes supervision available at most generated token positions, while shared vocabulary entries carry almost all predictive probability at the measured strict positions. The vocabulary entries excluded from comparison collectively receive little probability on average under either model. Therefore, static vocabulary mismatch alone does not determine how much probability mass the shared vocabulary retains.

A student-selected top-k subset also captures most teacher mass. At $k = 16$, the selected shared vocabulary entries retain at least 93.54% of teacher mass and 94.55% of student mass on average in each of the three pairs. From $k = 4$ to $k = 128$, successive doublings of the subset size yield progressively smaller gains in all six curves. Expanding the subset from 16 to 128 entries increases the retained mass by at most 3.44 percentage points for the teacher and 2.70 points for the student. Selection uses only the student distribution, yet the selected shared vocabulary entries also carry most of the teacher's predictive mass.

Table 2: Cross-tokenizer distillation accuracy (%). Base denotes the student before distillation. Bold marks the best distilled score per column.

Method	Qwen→Llama			Granite→Phi			Granite→Qwen		
	Math	Code	Full	Math	Code	Full	Math	Code	Full
Base	18.32	35.60	26.96	33.06	40.04	36.55	24.61	34.51	29.56
ULD	20.51	37.74	29.12	35.24	42.01	38.63	33.36	20.87	27.12
Extended ULD	23.01	**39.26**	31.13	35.28	41.42	38.35	33.14	32.00	32.57
GOLD	21.21	33.10	27.16	36.22	47.29	41.75	38.26	54.84	46.55
SimCT	25.31	37.86	31.59	36.67	46.70	41.68	38.64	53.56	46.10
Strict full	26.60	39.11	**32.86**	**37.49**	47.49	42.49	**39.85**	54.84	47.35
Strict top-16	**26.86**	38.42	32.64	37.23	47.29	42.26	39.07	55.06	47.06
Strict top-128	26.31	38.44	32.38	37.27	**47.82**	**42.54**	39.67	**55.62**	**47.65**

4 Learning with Student-Selected Subsets

In this section, we test whether training on this subset retains the learning gains of distillation on the full shared vocabulary.

We restrict the strict objective in Equation 5 to the student-selected top-k subset $\mathcal{K}_k^{(r)}$, renormalizing both models' distributions on this same set before computing reverse KL. We compare $k = 16$ and $k = 128$ with the full shared vocabulary across all three model pairs and report evaluation results on mathematical reasoning and code generation benchmarks. We also compare strict supervision with ULD (Boizard et al., 2025), its span-grouped variant Extended ULD, GOLD[3], and SimCT (Sun et al., 2026). Table 2 reports domain and full averages for all methods, and Appendix C provides benchmark-level results. All four baselines are implemented in KDFlow (Zhang et al., 2026). Appendix A details their position alignment, vocabulary handling, EOS settings, and loss reductions.

Strict supervision improves both domains. Strict full and both compact variants improve math and code accuracy over Base in every model pair, providing the reference for assessing how much learning benefit is retained on the student-selected top-k subsets.

Top-k subsets retain most of the performance gains. Top-16 retains at least 96% of the reported full-average improvement from Base to strict full in each pair. Most of the observed learning benefit can therefore be retained on a small student-selected top-k subset, with no consistent further gain from increasing its size.

Compact supervision achieves higher performance. All three strict variants exceed all four alternatives in both math and full averages on every model pair, with top-16 ahead of the strongest alternative in the full average by 0.51–1.05 percentage points. Appendix D reports additional results for distillation from a larger 235B teacher to an 8B student, where Strict top-16 achieves the highest Math, Code, and Full averages among the evaluated distillation methods. On ALFWorld, Strict top-16 also attains the highest success rate among the evaluated distillation methods on both seen and unseen splits, as shown in Appendix E.

5 Optimization Diagnostics of Span Supervision

In this section, we examine the gradient direction and relative scale between $\mathcal{L}_{1:1}(\theta)$ and $\mathcal{L}_{\text{span}}(\theta)$ in Equation 9.

[3] https://huggingface.co/docs/trl/gold_trainer

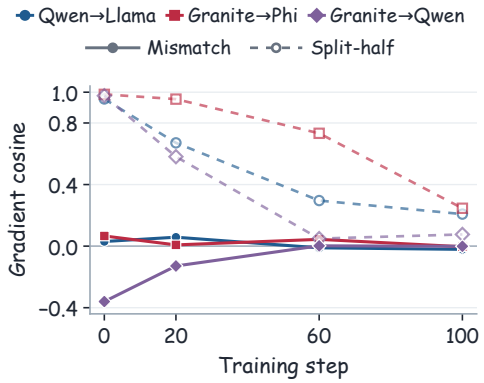


Figure 4: Mismatch–strict gradient cosine c_{mis} (solid) and split-half reference c_{ctrl} (dashed) at checkpoints from training with only the strict loss.

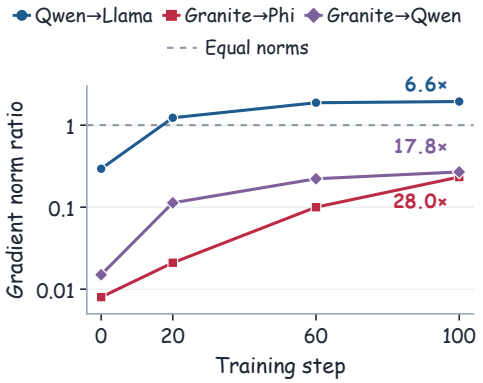


Figure 5: Mismatch-to-strict gradient norm ratio ρ, at checkpoints from training with only the strict loss, with growth factors from step 0 to 100 computed from unrounded checkpoint medians.

Gradient diagnostics. We compare the gradients of strict distribution matching and the span loss, evaluating both losses on the same sampled responses held fixed during differentiation:

$$g_{1:1} = \nabla_\theta \mathcal{L}_{1:1}, \quad g_{\text{mis}} = \nabla_\theta \mathcal{L}_{\text{span}}. \tag{12}$$

We measure directional agreement using gradient cosine (Yu et al., 2020) and relative magnitude using the norm ratio:

$$c_{\text{mis}} = \frac{\langle g_{1:1}, g_{\text{mis}} \rangle}{\|g_{1:1}\|_2 \|g_{\text{mis}}\|_2}, \quad \rho = \frac{\|g_{\text{mis}}\|_2}{\|g_{1:1}\|_2}. \tag{13}$$

We analyze checkpoints at steps 0, 20, 60, and 100 from the Strict full run for each model pair, generating student responses to the same 512 prompts at each checkpoint. We compute the diagnostics from the raw accumulated gradients over all trainable student parameters, and report the median of each statistic across replay units at each checkpoint. Within each batch, we randomly split the positions contributing to the strict loss into two subsets, A and B, and accumulate their gradients separately across the replay unit. Their cosine $c_{\text{ctrl}} = \cos(g_A, g_B)$ provides a reference for gradient agreement between position subsets under the same objective. Table 6 in Appendix B reports the complete statistics and replay setting.

Span gradients show weak agreement with strict supervision. Figure 4 shows mismatch cosines near zero for Qwen→Llama and Granite→Phi. Granite→Qwen instead starts with negative agreement and approaches zero later in training. Although the split-half strict cosine decreases overall during training, it remains higher than the mismatch–strict cosine at every measured checkpoint in each pair.

Relative gradient scale changes during training. Figure 5 shows an increasing span-to-strict gradient norm ratio across the measured checkpoints in all three pairs. At these checkpoints, a fixed positive weight on the span loss would give it an increasing gradient norm relative to the strict component. The growing relative magnitude of these weakly aligned span gradients may contribute to the lower accuracy observed with supervision on mismatch groups in Section 3.2.

6 Related Work

On-Policy Distillation. On-Policy Distillation trains the student on its own rollouts with teacher supervision (Song & Zheng, 2026). MiniLLM (Gu et al., 2024) uses reverse KL to discourage the student from assigning excessive probability to regions unlikely under the teacher. GKD (Agarwal et al., 2024) supports different divergences and mixtures of student-generated sequences and fixed training data. Yang et al. (2026) cast OPD as dense KL-constrained RL with an implicit per-token reward given by the teacher-to-reference log-probability ratio. Subsequent work examines which

teacher signals are effective at student-visited prefixes. Fu et al. (2026) analyze unreliable guidance and uneven supervision in sampled-token OPD, and propose distribution matching on teacher-selected local top-k supports. Entropy-aware OPD (Jin et al., 2026) augments reverse KL with forward KL at positions of high teacher entropy, while TIP (Xu et al., 2026) studies trajectory-position selection using student entropy and teacher–student disagreement. Li et al. (2026) relate distillation success to teacher–student compatibility and show that the intersection of their top-k predictions contains most probability mass and can retain the benefits of student top-k supervision. Their shared support consists of high-probability predictions within a common vocabulary. Our comparison of student-selected top-k subsets tests whether concentrated supervision remains sufficient when heterogeneous tokenizers constrain which response positions and vocabulary entries admit direct comparison, using the shared vocabulary at strict positions.

Cross-Tokenizer Distillation. Different tokenizers introduce discrepancies in both token boundaries and vocabularies. ULD (Boizard et al., 2025) compares sorted output probabilities without requiring matching token identities. DSKD (Zhang et al., 2024) and CDM (Chen et al., 2025) construct mappings between model representations or vocabularies, while DWA-KD (Vu et al., 2026) combines representation alignment with entropy-based weighting. Other methods match likelihoods across tokenizations (Minixhofer et al., 2025) or aggregate tokens into span representations before distillation (Dao et al., 2026). Byte-Level Distillation converts teacher predictions to byte probabilities and equips the student with a byte-level decoder head (Singh et al., 2026). For on-policy training, GOLD[4] merges text-equivalent token groups and uses rank-based or hybrid identity-based distribution matching. SimCT (Sun et al., 2026) adds minimal aligned multi-token units to the shared supervision space and reports improvements over shared-vocabulary OPD from recovering excluded supervision. Byte-Prefix Marginalization (Wang et al., 2026) maps teacher probability mass onto student tokens through byte-prefix relations and retains unmatched mass in a residual category. It also identifies harmful supervision at all-whitespace positions and prevents the resulting training collapse by masking them. We test the learning value of structural recovery by adding an span log-probability MSE loss while holding strict supervision fixed. Probability-mass and gradient analyses, together with comparisons against alternative objectives, distinguish alignment coverage from supervision reliability in the objectives and transfers we evaluate.

7 Conclusion

We re-examined the role of alignment coverage in Cross-Tokenizer On-Policy Distillation across three heterogeneous teacher–student pairs on mathematical reasoning and code generation. Strict 1:1 groups already cover most student-generated tokens despite substantial static vocabulary mismatch. On responses sampled from the students before distillation, the shared vocabulary retains nearly all teacher and student probability mass at strict positions on average. A compact student-selected top-k subset of the shared vocabulary retains most of this mass on both sides. Restricting reverse KL to a student-selected top-16 subset of the shared vocabulary at each strict position retains at least 96% of the full-average improvement achieved by full shared-vocabulary OPD over the undistilled student, with higher performance than the evaluated cross-tokenizer baselines.

Adding span log-probability MSE on mismatch groups achieves complete supervision coverage but reduces downstream accuracy across all tested positive weights. Across the measured checkpoints from training with only the strict loss, the span gradients grow in magnitude relative to the strict gradients, while their directional agreement remains weak or negative. This combination may help explain the accuracy drop from adding span supervision. These findings motivate a shift from *alignment coverage* to *supervision reliability*: increasing coverage alone does not guarantee better distillation, and the quality of the added supervision must be assessed by its contribution to student learning.

AI use statement

We use generative AI tools solely for language polishing to improve the grammar, clarity, and readability of this paper. These tools were not used for experimental design, implementation, execution, or analysis. We take full responsibility for the final content of this work.

[4] https://huggingface.co/docs/trl/gold_trainer

Ethics statement

Use of Human Annotations Human annotations are only used in methodological research at the beginning of the work, to assist in analyzing the feasibility of the proposed solution. Annotators consented to the use of data for research purposes. We ensure that the privacy of all annotators is protected throughout the annotation process, and all of them are adequately paid according to local standards. Human annotations are not applied during the evaluation of our method.

Risks In this paper, all datasets are obtained from official sources. The datasets adopted have been anonymized and do not contain offensive information. However, we cannot guarantee that the datasets do not contain socially harmful or toxic language.

Reproducibility statement

We describe the alignment rules, distillation objectives, and diagnostic measures in Sections 2–5. Appendix A details the data preparation, training configurations, baseline implementations, and evaluation procedures. Appendix B provides the protocols for probability-mass measurements and gradient diagnostics. Appendices C–E report the benchmark-level results and additional experiments with a larger teacher and on ALFWorld. Our code is available at `https://anonymous.4open.science/r/Cross-Tokenizer-OPD`.

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.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=3zKtaqxLhW`.

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. *Trans. Mach. Learn. Res.*, 2025, 2025. URL `https://openreview.net/forum?id=bwRxXiGO9A`.

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

Yijie Chen, Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. Enhancing cross-tokenizer knowledge distillation with contextual dynamical mapping. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pp. 8005–8018. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-ACL.419. URL `https://doi.org/10.18653/v1/2025.findings-acl.419`.

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.

Quoc Phong Dao, Hoang Son Nguyen, Pham Khanh Chi, Tung Nguyen, Linh Ngo Van, Nguyen Thi Ngoc Diep, and Trung Le. SRA: span representation alignment for large language model

distillation. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 32961–32975. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.1522. URL https://doi.org/10.18653/v1/2026.acl-long.1522.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. *arXiv preprint arXiv:2605.00674*, 2026.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=chfJJYC3iL.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/18abbeef8cfe9203fdf9053c9c4fe191-Abstract-Conference.html.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

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. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract-Conference.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Benjamin Minixhofer, Ivan Vulic, and Edoardo Maria Ponti. Universal cross-tokenizer distillation via approximate likelihood matching. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/720f9f5dc751eb56952ae4fee2398f73-Abstract-Conference.html.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=0IOX0YcCdTn.

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.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

Jie Sun, Mao Zheng, Mingyang Song, Qiyong Zhong, Yilin Cheng, Bichuan Feng, Pengfei Liu, Junfeng Fang, and Xiang Wang. Simct: Recovering lost supervision for cross-tokenizer on-policy distillation. *arXiv preprint arXiv:2605.07711*, 2026.

Duc Trung Vu, Chi Pham Khanh, Phi Van Dat, Ngo Van Linh, Dinh Viet Sang, and Trung Le. DWA-KD: dual-space weighting and time-warped alignment for cross-tokenizer knowledge distillation. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Findings of the Association for Computational Linguistics: EACL 2026, Rabat, Morocco, March 24-29, 2026*, Findings of ACL, pp. 3513–3527. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.FINDINGS-EACL.181. URL https://doi.org/10.18653/v1/2026.findings-eacl.181.

Hao Wang, Kun Yuan, Wenlin Zhong, Minglei Zhang, Han Xiao, Ming Sun, and Honggang Qi. Cross-tokenizer on-policy distillation via byte-prefix marginalization. *arXiv preprint arXiv:2607.22334*, 2026.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. Tip: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 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.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: an open-source LLM reinforcement learning system at scale. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. Dual-space knowledge distillation for large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 18164–18181. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.1010. URL https://doi.org/10.18653/v1/2024.emnlp-main.1010.

Songming Zhang, Xue Zhang, Tong Zhang, Bojie Hu, Yufeng Chen, and Jinan Xu. Kdflow: A user-friendly and efficient knowledge distillation framework for large language models. *arXiv preprint arXiv:2603.01875*, 2026.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark W. Barrett, and Ying Sheng. Sglang: Efficient execution of structured language model programs. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/724be4472168f31ba1c9ac630f15dec8-Abstract-Conference.html.

A Experimental Details

We construct a common source pool of 20,000 prompts for the three teacher–student pairs in Section 3: 10,000 mathematics prompts randomly sampled from DAPO-Math-17K (Yu et al., 2025) and 10,000 code prompts randomly sampled from CodeForces[5]. We perform no deduplication. For each student, we encode the formatted student prompt with its tokenizer and retain samples with at most 2,048 prompt tokens. The retained sample count therefore varies with the student tokenizer. The probability-mass probe uses the complete, unfiltered 20k prompts, as detailed in Appendix B. We use the officially released open-source models named in Section 3. All training prompts are formatted using each model's official chat template.

The current student generates one response per sampled prompt. For loss computation, the student and frozen teacher each re-tokenize the decoded response together with their own prompt encoding. On each side, the response length is measured by separately tokenizing the response without added special tokens, and the prompt-response boundary is then set to the joint input-token count minus that response length. The next-token loss mask selects the resulting response positions. A terminal EOS is appended for responses that were not length-truncated. For the strict objectives, content-token alignment follows Section 2, and paired EOS positions are included in the training loss.

Training runs for 100 on-policy iterations, with checkpoints saved every 10 iterations. Student optimization uses FSDP2 on a single node with 8 NVIDIA H20 GPUs. Sleep mode multiplexes these components on the same workers. We use packed samples, bfloat16 computation, gradient checkpointing, and token-chunked loss evaluation with a chunk size of 2,048 to reduce memory consumption. We use SGLang (Zheng et al., 2024) as the rollout engine. The global optimizer batch contains 512 responses across student workers. Table 3 lists the optimization, rollout, and training settings.

Table 3: Training settings.

Setting	Value
Training and optimization	
Training iterations	100
Batch size	512
Dynamic batching	Disabled
Optimizer	AdamW (Loshchilov & Hutter, 2019)
Learning rate	1×10^{-6}
Adam β_1, β_2	$0.9, 0.98$
Weight decay / grad clip	$0/1.0$
Checkpoint interval	10 steps
Distillation and rollout	
Strict divergence	Reverse KL
KD weight / temperature	$1.0/1.0$
Responses per prompt	1
Rollout sampling	$T = 1.0$, top-$p = 0.95$
Max training prompt length	2,048
Max generation length	8,192

Baseline implementations. All four baselines use the KDFlow (Zhang et al., 2026) implementations. SimCT selects `simct`. All four use unit KD weight without cross-entropy mixing and teacher/student temperature 1.0. ULD, Extended ULD, and GOLD exclude terminal EOS from their distillation losses.

Evaluation. Evaluation uses publicly available benchmarks. All distilled models in the main comparison and span-MSE sweeps are evaluated at step 100. Mathematics evaluation covers MATH500, GSM8K, AIME-2024, AIME-2025, AIME-2026, AMC23, and Minerva-Math. Code evaluation

[5] `https://huggingface.co/datasets/open-r1/codeforces`

Table 4: Static vocabulary overlap across the three teacher–student pairs. Vocabulary counts follow the shared-vocabulary convention in Section 2. Overlap ratios are reported as percentages.

Teacher→Student	Vocabulary size			Overlap (%)		
	$\lvert\mathcal{V}_{\mathrm{T}}\rvert$	$\lvert\mathcal{V}_{\theta}\rvert$	$\lvert\mathcal{V}_{\cap}\rvert$	$\frac{\lvert\mathcal{V}_{\cap}\rvert}{\lvert\mathcal{V}_{\theta}\rvert}$	$\frac{\lvert\mathcal{V}_{\cap}\rvert}{\lvert\mathcal{V}_{\mathrm{T}}\rvert}$	J_{vocab}
Qwen→Llama	151,665	128,256	109,567	85.43	72.24	64.32
Granite→Phi	100,352	200,029	85,034	42.51	84.74	39.49
Granite→Qwen	100,352	151,665	99,163	65.38	98.82	64.87

covers HumanEval, MBPP, and LiveCodeBench. We average correctness over 32 responses per mathematics problem and 8 per code problem, reporting mean@32 and mean@8 accuracy. Math and Code are the arithmetic means over their seven and three benchmarks, respectively, and Full is the equally weighted mean of Math and Code. Appendix C reports benchmark-level scores, including the base model and frozen teacher as references. The evaluation clients submit tokenized prompts to SGLang through its OpenAI-compatible completions interface. Both domains use temperature 1.0, top-$p = 0.95$, a maximum of 8,192 generated tokens. For mathematics, the input is the benchmark file's `problem` field as a single user message, rendered with the model's chat template and generation prompt. For HumanEval and MBPP, a single user message contains the dataset prompt followed by an instruction to think through the problem and place the Python solution in a final fenced code block. For LiveCodeBench, we use its generic system message and question template. Each prompt is encoded without adding further special tokens after template rendering.

Mathematics scoring. The grader extracts the final answer from each response. A missing or malformed final boxed answer is counted as incorrect. If the extracted answer exceeds 300 characters, only its first 300 characters are passed to the grader. Both the reference and prediction are wrapped as boxed expressions and processed using Math-Verify[6]. Parsing or verification exceptions count as incorrect. Failed generation requests yield empty responses and remain in the accuracy denominator.

Code scoring. HumanEval and MBPP are loaded through EvalPlus (Liu et al., 2023). The evaluator computes both original and augmented-test outcomes. The reported HumanEval and MBPP scores use the original base tests. For extraction, it takes the text between the final two code-fence lines, or at most the first 200 lines when no complete fence is found, then applies EvalPlus sanitization with the task entry point. LiveCodeBench uses `release_v6` version, with 1,055 problems, code extractor, and code-generation test runner. The wrapper supplies an execution timeout of 10 seconds to the runner and additionally caps each candidate's total evaluation wait at 120 seconds. Scores are averaged from each problem's fraction of passing candidates.

B Diagnostic Protocols and Statistics

Using the vocabulary convention in Section 2, we measure static vocabulary overlap by

$$J_{\text{vocab}} = \frac{|\mathcal{V}_{\cap}|}{|\mathcal{V}_{\theta} \cup \mathcal{V}_{\mathrm{T}}|} = \frac{|\mathcal{V}_{\theta} \cap \mathcal{V}_{\mathrm{T}}|}{|\mathcal{V}_{\theta} \cup \mathcal{V}_{\mathrm{T}}|}. \tag{14}$$

Table 4 reports the vocabulary sizes and the fractions of the student and teacher vocabularies in the shared vocabulary.

For the probability-mass probe, each student before distillation generates one response for each prompt in the complete, unfiltered 20k prompts, using the rollout sampling and generation-length settings in Table 3 for all three model pairs. The training prompt length filter of 2,048 student tokens is not applied to this probe. For scoring, we exclude empty responses, responses whose teacher and student tokenizations yield different response byte streams, responses without scorable strict positions, and responses whose full encoded input exceeds either model's context limit or an enabled analysis sequence-length cap. We measure both models' retained probability mass on the

[6] `https://github.com/huggingface/Math-Verify`

Table 5: Mean retained probability mass (%) at strict positions before distillation. "Shared" denotes the full shared vocabulary, and bold values mark $k = 16$.

Subset	Qwen→Llama		Granite→Phi		Granite→Qwen	
	Student	Teacher	Student	Teacher	Student	Teacher
$k = 4$	91.74	89.81	90.08	88.12	96.58	94.53
$k = 8$	93.60	92.43	92.90	91.43	97.81	96.48
$k = 16$	**94.55**	**93.92**	**94.68**	**93.54**	**98.44**	**97.67**
$k = 32$	95.06	94.77	95.89	95.03	98.82	98.37
$k = 64$	95.35	95.29	96.75	96.14	99.07	98.82
$k = 128$	95.54	95.63	97.38	96.98	99.26	99.14
Shared	99.72	99.73	98.99	99.69	99.81	99.90

Table 6: Gradient diagnostics at checkpoints from training with only the strict loss. c_{mis} and c_{ctrl} denote the mismatch–strict and split-half strict gradient cosines, respectively. ρ is the unweighted span-to-strict norm ratio. Values are rounded to three decimals; figure growth factors use the unrounded medians.

Step	Qwen→Llama			Granite→Phi			Granite→Qwen		
	c_{mis}	c_{ctrl}	ρ	c_{mis}	c_{ctrl}	ρ	c_{mis}	c_{ctrl}	ρ
0	0.030	0.956	0.294	0.066	0.986	0.008	−0.360	0.977	0.015
20	0.058	0.671	1.231	0.008	0.955	0.021	−0.129	0.581	0.113
60	−0.011	0.296	1.877	0.044	0.733	0.100	0.002	0.049	0.222
100	−0.021	0.208	1.942	−0.004	0.246	0.233	−0.001	0.076	0.269

same student-selected top-k subset from their original full-vocabulary probabilities and average over the remaining scorable strict positions.

Gradient diagnostics follow Section 5, using checkpoints from each model pair's Strict full run with responses to the same 512 prompts at each checkpoint. The strict component uses reverse KL on the full shared vocabulary, and the mismatch component uses squared differences between span log-probabilities. Replay matches the training batch settings. We report medians across accumulated replay units, using all trainable student parameters.

Responses are read from saved rollouts and held fixed during differentiation. We measure the raw accumulated parameter gradients, without gradient clipping, optimizer preconditioning, or parameter updates. The split-half reference partitions the strict positions and paired EOS positions within each batch and accumulates the two subsets' gradients separately over the replay unit.

In accumulated replay, samples are excluded if they have neither response content nor an appended EOS, if a mismatch group's teacher and student byte streams differ, or if either encoded input exceeds its model's context limit or an enabled sequence-length cap. Length-truncated responses are retained without an appended EOS. Provided the analysis-unit limit has not been reached, a nonempty final unit is analyzed even when it contains fewer samples than the target batch size. Its losses use its actual token and batch counts, and its finite statistics are included in the summary.

C Complete Downstream Results

All scores are accuracies in percent, using mean@32 for mathematics and mean@8 for code. Distilled model scores are evaluated at step 100. Math, Code, and Full follow the settings in Appendix A.

Table 7: Benchmark-level scores of the undistilled students and frozen teachers for the three main model pairs.

Benchmark	Qwen→Llama		Granite→Phi		Granite→Qwen	
	Base	Teacher	Base	Teacher	Base	Teacher
MATH500	34.32	74.50	62.94	80.81	48.60	80.81
GSM8K	63.44	89.93	86.07	91.51	67.65	91.51
AIME-2024	2.29	11.88	6.35	22.71	5.10	22.71
AIME-2025	0.31	7.50	3.85	18.65	1.77	18.65
AIME-2026	0.52	7.50	4.69	12.81	2.81	12.81
AMC23	15.55	50.94	35.86	64.53	28.05	64.53
Minerva-Math	11.82	35.09	31.69	39.76	18.29	39.76
HumanEval	44.89	72.94	55.34	74.54	44.89	74.54
MBPP	53.27	77.71	49.34	82.97	45.77	82.97
LiveCodeBench	8.63	24.64	15.45	32.46	12.86	32.46
Math	18.32	39.62	33.06	47.25	24.61	47.25
Code	35.60	58.43	40.04	63.32	34.51	63.32
Full	26.96	49.03	36.55	55.29	29.56	55.29

Table 8: Benchmark-level comparison for Qwen2.5-7B-Instruct→Llama-3.2-3B-Instruct.

Benchmark	ULD	Extended ULD	GOLD	SimCT	Strict		
					full	top-16	top-128
MATH500	40.73	47.33	41.70	50.55	52.74	52.92	52.08
GSM8K	62.11	63.96	70.00	78.20	80.21	80.26	80.08
AIME-2024	5.00	5.73	3.33	3.65	4.69	5.00	5.10
AIME-2025	0.52	1.15	0.62	1.56	1.67	1.88	1.98
AIME-2026	0.94	0.73	0.00	0.73	0.94	1.25	0.52
AMC23	19.92	24.77	18.67	25.00	27.66	28.67	26.64
Minerva-Math	14.34	17.37	14.15	17.47	18.31	18.07	17.80
HumanEval	48.09	51.98	43.22	50.91	52.74	51.52	52.90
MBPP	55.06	52.98	45.70	49.67	51.46	50.86	49.77
LiveCodeBench	10.07	12.83	10.38	13.01	13.13	12.89	12.64
Math	20.51	23.01	21.21	25.31	26.60	**26.86**	26.31
Code	37.74	**39.26**	33.10	37.86	39.11	38.42	38.44
Full	29.12	31.13	27.16	31.59	**32.86**	32.64	32.38

D Additional Results with a Larger Teacher

We evaluate strict supervision with Qwen3-235B-A22B-Instruct-2507→Granite-4.1-8B, using Strict top-16 from Section 4. Tables 14 and 15 report performance, with Base and Teacher denoting the undistilled student and frozen teacher. The mathematics domain adds HMMT 2025 February and November (Dekoninck et al., 2026) to the seven benchmarks in Appendix A.

E Additional Results on ALFWorld

We evaluate Cross-Tokenizer On-Policy Distillation on ALFWorld (Shridhar et al., 2021), using Qwen3-4B-Instruct-2507→Llama-3.2-3B-Instruct and the ALFWorld training split. Table 16 compares Strict top-16 from Section 4 with ULD, Extended ULD, GOLD, and SimCT. Base denotes the student before distillation, and Teacher denotes the frozen teacher. We evaluate on the official seen and unseen splits, using 128 episodes per evaluation run and a limit of 50 interaction turns per

Table 9: Benchmark-level comparison for Granite-4.1-8B→Phi-4-mini-instruct.

Benchmark	ULD	Extended ULD	GOLD	SimCT	Strict		
					full	top-16	top-128
MATH500	68.20	68.11	69.06	68.86	70.55	70.10	70.26
GSM8K	86.00	86.25	87.06	86.95	87.53	87.56	87.43
AIME-2024	8.85	8.33	9.48	10.73	11.25	10.83	11.46
AIME-2025	3.44	5.21	6.56	5.83	6.46	7.19	6.98
AIME-2026	5.52	6.15	6.25	6.88	7.81	7.29	6.77
AMC23	43.05	40.47	42.03	44.69	45.86	45.00	45.16
Minerva-Math	31.65	32.46	33.08	32.74	32.96	32.65	32.85
HumanEval	54.42	53.12	67.07	65.17	66.01	65.02	67.00
MBPP	52.65	52.38	53.97	54.17	54.93	55.79	55.75
LiveCodeBench	18.95	18.76	20.83	20.76	21.53	21.07	20.70
Math	35.24	35.28	36.22	36.67	**37.49**	37.23	37.27
Code	42.01	41.42	47.29	46.70	47.49	47.29	**47.82**
Full	38.63	38.35	41.75	41.68	42.49	42.26	**42.54**

Table 10: Benchmark-level comparison for Granite-4.1-8B→Qwen2.5-7B-Base.

Benchmark	ULD	Extended ULD	GOLD	SimCT	Strict		
					full	top-16	top-128
MATH500	65.84	66.16	73.31	73.73	74.91	75.25	75.31
GSM8K	85.63	83.97	88.03	87.75	88.64	88.81	88.87
AIME-2024	7.60	8.23	10.94	11.67	14.37	14.17	14.17
AIME-2025	3.85	4.38	8.33	9.06	9.27	8.75	9.69
AIME-2026	6.98	5.31	7.29	5.73	6.25	6.46	6.46
AMC23	38.67	39.30	46.88	49.69	52.11	46.95	50.00
Minerva-Math	24.98	24.61	33.04	32.86	33.40	33.08	33.17
HumanEval	34.15	51.07	67.84	68.60	68.67	69.74	69.21
MBPP	20.37	35.02	71.69	71.20	71.56	71.06	73.21
LiveCodeBench	8.10	9.91	24.99	20.88	24.29	24.37	24.45
Math	33.36	33.14	38.26	38.64	**39.85**	39.07	39.67
Code	20.87	32.00	54.84	53.56	54.84	55.06	**55.62**
Full	27.12	32.57	46.55	46.10	47.35	47.06	**47.65**

episode. Scores are averaged over three evaluations. SR is the success rate in percent, and Turns denotes the average number of interaction turns across all episodes, including successes and failures. Overall is the arithmetic mean of the seen and unseen scores for each metric.

Table 11: Benchmark-level span-MSE weight sweep for Qwen2.5-7B-Instruct→Llama-3.2-3B-Instruct.

Benchmark	**Span-MSE weight λ**						
	0	**0.25**	**0.5**	**0.75**	**1**	**1.25**	**1.5**
MATH500	52.74	52.38	51.21	51.19	50.50	50.18	49.95
GSM8K	80.21	79.96	79.35	79.65	79.11	78.94	78.75
AIME-2024	4.69	5.10	5.21	5.21	4.58	4.06	5.62
AIME-2025	1.67	1.77	1.56	1.56	1.35	1.25	1.25
AIME-2026	0.94	0.42	0.62	0.31	0.42	0.52	0.21
AMC23	27.66	27.81	26.02	26.72	25.47	25.78	24.92
Minerva-Math	18.31	17.77	17.52	17.42	17.07	17.04	17.39
HumanEval	52.74	52.36	51.98	52.06	50.08	49.92	50.84
MBPP	51.46	50.46	52.08	51.79	51.12	51.39	51.39
LiveCodeBench	13.13	12.64	12.29	12.95	12.44	12.48	12.50
Math	**26.60**	26.46	25.93	26.01	25.50	25.40	25.44
Code	**39.11**	38.49	38.78	38.93	37.88	37.93	38.24
Full	**32.86**	32.47	32.36	32.47	31.69	31.66	31.84

Table 12: Benchmark-level span-MSE weight sweep for Granite-4.1-8B→Phi-4-mini-instruct.

Benchmark	**Span-MSE weight λ**						
	0	**0.25**	**0.5**	**0.75**	**1**	**1.25**	**1.5**
MATH500	70.55	71.04	70.01	70.44	69.59	70.47	70.08
GSM8K	87.53	87.64	87.31	87.57	87.09	87.31	87.46
AIME-2024	11.25	10.31	10.42	10.31	9.69	10.31	10.00
AIME-2025	6.46	6.67	6.46	6.77	7.08	6.25	6.46
AIME-2026	7.81	5.94	6.77	7.19	6.88	6.77	6.35
AMC23	45.86	43.36	44.06	46.25	44.77	44.92	43.36
Minerva-Math	32.96	32.96	32.73	32.12	31.99	33.50	32.12
HumanEval	66.01	65.09	63.64	64.41	64.48	63.64	63.19
MBPP	54.93	54.79	55.56	55.89	53.21	56.55	56.22
LiveCodeBench	21.53	20.84	21.21	20.51	20.63	20.68	21.11
Math	**37.49**	36.85	36.82	37.24	36.73	37.08	36.55
Code	**47.49**	46.91	46.80	46.94	46.11	46.96	46.84
Full	**42.49**	41.88	41.81	42.09	41.42	42.02	41.69

Table 13: Benchmark-level span-MSE weight sweep for Granite-4.1-8B→Qwen2.5-7B-Base.

Benchmark	**Span-MSE weight λ**						
	0	**0.25**	**0.5**	**0.75**	**1**	**1.25**	**1.5**
MATH500	74.91	75.45	74.97	74.84	74.54	74.72	74.81
GSM8K	88.64	88.79	88.75	88.77	88.83	88.95	88.77
AIME-2024	14.37	15.42	13.02	14.27	12.71	12.19	12.92
AIME-2025	9.27	8.02	9.27	9.69	8.23	8.02	8.02
AIME-2026	6.25	6.15	5.94	6.15	5.83	5.83	5.52
AMC23	52.11	49.45	49.45	50.62	48.44	49.06	49.30
Minerva-Math	33.40	32.97	33.94	33.48	33.66	33.85	33.43
HumanEval	68.67	67.99	68.22	67.45	67.38	67.99	67.99
MBPP	71.56	71.23	71.79	70.57	70.90	71.76	70.57
LiveCodeBench	24.29	24.55	24.44	23.86	24.10	24.15	23.86
Math	**39.85**	39.46	39.33	39.69	38.89	38.95	38.97
Code	**54.84**	54.59	54.82	53.96	54.13	54.63	54.14
Full	**47.35**	47.03	47.08	46.82	46.51	46.79	46.55

Table 14: Benchmark-level scores of the undistilled student and frozen teacher for Qwen3-235B-A22B-Instruct-2507→Granite-4.1-8B.

Benchmark	**Base**	**Teacher**
MATH500	80.81	95.88
GSM8K	91.51	94.24
AIME-2024	22.71	60.94
AIME-2025	18.65	51.98
AIME-2026	12.81	65.52
AMC23	64.53	90.00
Minerva-Math	39.76	48.10
HMMT 2025 February	10.00	41.15
HMMT 2025 November	12.92	46.15
HumanEval	74.54	87.12
MBPP	82.97	94.54
LiveCodeBench	32.46	65.38
Math	39.30	66.00
Code	63.32	82.35
Full	51.31	74.17

Table 15: Benchmark-level distillation results for Qwen3-235B-A22B-Instruct-2507→Granite-4.1-8B. Bold marks the best Math, Code, and Full averages.

Benchmark	ULD	Extended ULD	GOLD	SimCT	Strict top-16
MATH500	81.31	87.29	85.91	86.08	89.75
GSM8K	91.59	93.06	93.12	93.15	93.69
AIME-2024	23.23	37.19	35.52	37.40	43.02
AIME-2025	21.15	26.98	27.19	29.17	35.10
AIME-2026	13.33	30.10	23.02	27.29	38.44
AMC23	67.34	72.66	69.30	74.22	76.80
Minerva-Math	40.44	41.22	43.35	41.58	44.11
HMMT 2025 February	11.56	12.60	12.29	10.94	16.67
HMMT 2025 November	13.12	17.19	17.81	18.75	24.06
HumanEval	76.98	65.40	76.45	82.32	91.54
MBPP	84.09	69.11	76.03	78.90	85.55
LiveCodeBench	32.48	33.01	35.73	36.74	43.76
Math	40.34	46.48	45.28	46.51	**51.29**
Code	64.52	55.84	62.74	65.99	**73.62**
Full	52.43	51.16	54.01	56.25	**62.46**

Table 16: ALFWorld results for Qwen3-4B-Instruct-2507→Llama-3.2-3B-Instruct. Bold marks the highest SR and lowest Turns among distilled models.

Method	Seen		Unseen		Overall	
	SR ↑	Turns ↓	SR ↑	Turns ↓	SR ↑	Turns ↓
Base	12.50	46.60	8.85	47.28	10.68	46.94
Teacher	29.69	41.07	29.43	41.22	29.56	41.15
ULD	1.56	49.61	1.30	49.74	1.43	49.68
Extended ULD	25.52	41.95	23.44	42.61	24.48	42.28
GOLD	3.65	49.30	1.56	49.58	2.61	49.44
SimCT	25.00	42.31	32.03	39.31	28.52	40.81
Strict top-16	**26.04**	**40.73**	**34.64**	**38.82**	**30.34**	**39.78**