FROM GRADIENTS TO CAPABILITIES: UNDERSTANDING MULTI-TEACHER ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-teacher on-policy distillation (MOPD) aims to combine the strengths of RL-trained teachers in a single student, but how teacher signals affect parameter changes remains underexplored. We study Qwen3-1.7B with four domain teachers trained with RL from the same initialization as the student, comparing gradients, optimizer updates, and task learning curves, with additional SmolLM3-3B diagnostics. We find that several factors influence teacher signals. First, loss averaging implicitly weights responses: token averaging favors longer responses, and equalizing domain contributions retains this weighting within domains. Second, Adam's first moment reduces differences in parameter updates: the cosine similarity is 0.83 between teachers and 0.96 between averaging rules, despite differences in raw gradients. Third, BF16 rounding hides small changes: about 97% of FP32 master weights differ from initialization, but only 7–11% of BF16 weights do. Finally, the top-64 intersection KL gradient closely matches Qwen's full-vocabulary gradient, but the effect on task performance depends on averaging: mathematics accuracy is 2.6 points higher than with sampled-token policy-gradient (PG) under response averaging and 2.1 points lower under global token averaging.

1 INTRODUCTION

Reinforcement learning produces strong language-model specialists in mathematics, coding, and instruction following, but combining their strengths remains challenging (Ma et al., 2026). Multi-teacher on-policy distillation (MOPD) trains a single student using next-token probabilities from domain teachers on student-generated responses (Team et al., 2026; NVIDIA, 2026; Kimi Team et al., 2026; DeepSeek-AI et al., 2026). Yet strong teachers do not ensure that the student acquires every teacher's strengths (Gao et al., 2026). Because all teachers influence shared student parameters, we study how teacher losses combine and how the optimizer converts their gradients into parameter changes, connecting supervision to updates and task performance.

Existing MOPD recipes average losses within responses (Ma et al., 2026), normalize at the response and domain levels (Bai et al., 2026), or correct unequal domain token shares (Gao et al., 2026). These choices change the weights assigned to teachers and responses, but when does correcting unequal loss weights improve capability integration? Prior work also compares losses computed from sampled tokens and top-k vocabulary (Ma et al., 2026). However, Nemotron 3 Ultra reports no performance advantage from broader vocabulary supervision in its preliminary experiments (NVIDIA, 2026, Section 3.3.5). When does using more vocabulary candidates improve task performance? Separately, parameter analyses show concentrated, sparse changes (Yu et al., 2026; Shen et al., 2026). We therefore also explore how the optimizer and numerical precision affect observed parameter changes.

We use Qwen3-1.7B and RL teachers for mathematics, code, instruction following, and science, with additional diagnostics on SmolLM3-3B. As in existing MOPD recipes, these teachers are trained by RL from the student's initialization (Ma et al., 2026; Gao et al., 2026). To isolate loss choices, we compare gradients and optimizer steps from identical student parameters, responses, and optimizer states. We also track per-task learning curves as training progresses. Three findings show how supervision choices affect parameter updates and task performance (Figure 1):

1. Loss averaging acts as implicit response weighting, and its effect depends on the task and objective. Global token averaging gives longer-response domains more loss weight. Equalizing

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.

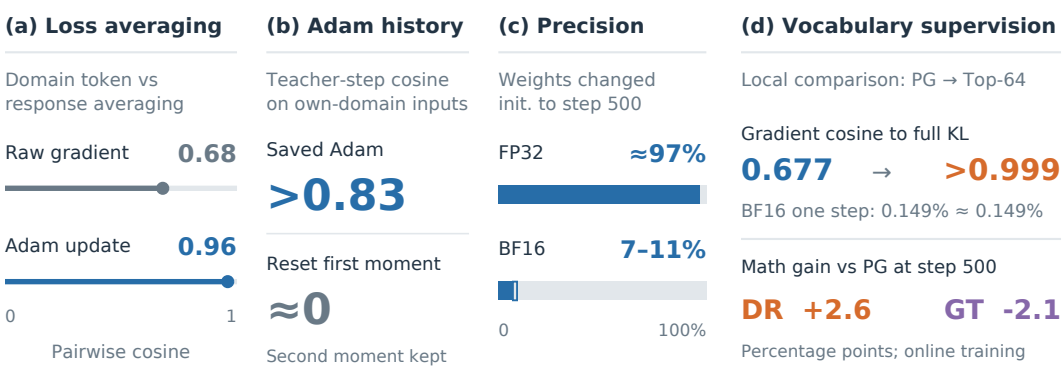

Figure 1: **From supervision choices to parameter updates and task gains.** (a) Averaging changes gradients more than Adam updates. (b) Shared momentum aligns teacher updates. (c) BF16 rounding hides widespread FP32 changes. (d) Local gradient fidelity and one-step BF16 changes, followed by mathematics gains in separate 500-step training runs. DR/GT denote domain response/global token averaging.

domain weights removes this imbalance, but a covariance identity shows that it still shifts each domain's gradient toward longer responses; domain response averaging weights responses equally. No averaging rule performs best on every task, and the distillation loss changes how loss weights translate into task scores.

2. Adam's first moment aligns updates, and BF16 rounding hides widespread changes. Updates computed from different teachers' losses point in similar directions even when their current gradients are nearly orthogonal. Using the PG loss, SGD achieves higher four-task mean scores than Adam with all loss averaging rules. After MOPD training, most student parameters differ from initialization in FP32, but only a small fraction differ after BF16 rounding.

3. Better gradient approximation does not guarantee larger capability gains. Gradients from shared teacher and student top-64 vocabulary closely match Qwen's full-vocabulary direction. Final mathematics accuracy with the top-64 intersection KL loss is 2.6 points higher than with the PG loss under response averaging, but 2.1 points lower under global token averaging. Although the PG loss is unbiased for the full-vocabulary gradient, a single sampled token gives a noisy estimate, and neither unbiasedness nor closer approximation predicts which tasks improve. In SmolLM, top-64 vocabulary retain most student probability on average but still leave gradient error.

2 Preliminaries

Multi-teacher on-policy distillation (MOPD). For a prompt $x \sim \mathcal{D}_d$ from domain d, the student generates $y \sim p_\theta(\cdot \mid x)$. The domain's frozen teacher q_d provides next-token probabilities at each prefix $h_t = (x, y_{<t})$. Batch $\mathcal{B}$ pools responses from different domains; each response i uses its domain's teacher and averages T_i valid token losses: $\ell_i = T_i^{-1} \sum_t \ell_{i,t}$. We minimize

$$\mathcal{L}(\theta) = \sum_{i \in \mathcal{B}} w_i \ell_i(\theta), \qquad w_i \geq 0, \quad \sum_{i \in \mathcal{B}} w_i = 1. \tag{1}$$

Each training step uses fresh responses, whose text stays fixed while we differentiate the loss with respect to student parameters θ.

Distillation losses. Write $p_\theta(v) = p_\theta(v \mid h_t)$ and $q_d(v) = q_d(v \mid h_t)$. The full-vocabulary KL loss computes reverse KL using all tokens in vocabulary $\mathcal{V}$; the sampled-token policy-gradient (PG) loss uses the token y_t sampled by the student:

$$\ell_{\text{full}} = \sum_{v \in \mathcal{V}} p_\theta(v) \log \frac{p_\theta(v)}{q_d(v)}, \qquad \ell_{\text{PG}}(y_t) = -\,\text{sg}[\log q_d(y_t) - \log p_\theta(y_t)] \log p_\theta(y_t) \tag{2}$$

Here sg treats its argument as constant when differentiating. The PG loss is a surrogate: its expected gradient under $y_t \sim p_\theta$ equals the full reverse-KL gradient (Appendix E); individual samples can differ. Let S_k be the student's k most probable tokens under p_θ and T_k the teacher's k most probable tokens under q_d. Top-k OPD usually restrict the KL to one of these sets (student-top-k on S_k or teacher-top-k on T_k). Our top-k intersection KL loss instead uses only the tokens in both sets, $I = I_k = S_k \cap T_k$ (Appendix D.2):

$$\begin{aligned} \bar{p}_\theta^I(v) &= \frac{p_\theta(v)}{\sum_{u \in I} p_\theta(u)}, \qquad \bar{q}_d^I(v) = \frac{q_d(v)}{\sum_{u \in I} q_d(u)}, \quad v \in I, \\ \ell_{\text{top-}k} &= \sum_{v \in I} \bar{p}_\theta^I(v) \log \frac{\bar{p}_\theta^I(v)}{\bar{q}_d^I(v)}. \end{aligned} \tag{3}$$

Full-vocabulary softmax probabilities are rescaled to sum to one within I_k, comparing candidates' relative probabilities; empty intersections give zero loss. Because the renormalized probabilities depend only on logits in I_k, the loss has zero gradient with respect to the logits of tokens outside I_k. The *supervision density* counts vocabulary candidates: 1 for PG, $|I_k| \le k$ for Top-k, and $|\mathcal{V}|$ for full-vocabulary KL.

Domain weighting and loss normalization. Let $\mathcal{B}_d$ contain domain-d responses and let $\lambda_d \ge 0$, $\sum_d \lambda_d = 1$. For nonempty responses and domains, response $i \in \mathcal{B}_d$ has weights

$$w_i^{\text{GT}} = \frac{T_i}{\sum_{j \in \mathcal{B}} T_j}, \qquad w_i^{\text{DT}} = \frac{\lambda_d T_i}{\sum_{j \in \mathcal{B}_d} T_j}, \qquad w_i^{\text{DR}} = \frac{\lambda_d}{|\mathcal{B}_d|}. \tag{4}$$

Global token averaging (GT) weights all response tokens equally, so each domain's weight follows its token share. *Domain token averaging* (DT) fixes each domain's weight at λ_d and weights responses by length. *Domain response averaging* (DR) keeps these domain weights but weights responses equally within each domain. Equal prompt counts can still give unequal domain weights under GT.

3 Loss Averaging Implicitly Weights Domains and Responses

Open-MOPD reports a large instruction-following deficit relative to single-domain students, with math/code responses averaging about 10,500 tokens and IF responses 409 tokens (Gao et al., 2026). In our less extreme length regime, we compare the three averaging rules (Section 2) to separate weighting between and within domains.

3.1 Loss averaging changes gradients and per-task scores, but barely the mean

With equal prompt counts, mathematics supplies about three times as many tokens as instruction following, giving it three times the loss weight under global token averaging (GT; Figure 2a). Domain token averaging (DT) fixes domain weights but still weights longer responses more. Domain response averaging (DR) also weights responses equally. For response gradients $g_i = \nabla_\theta \ell_i$ and unweighted domain means $\bar{g}_d, \bar{T}_d$, DT's within-domain gradient is

$$\frac{\sum_{i \in \mathcal{B}_d} T_i g_i}{\sum_{i \in \mathcal{B}_d} T_i} = \bar{g}_d + \frac{\text{Cov}_d(T, g)}{\bar{T}_d}. \tag{5}$$

The covariance term measures how length weighting shifts the domain gradient when longer responses have systematically different gradients. DR removes this term. Open-MOPD's token-share correction recovers DT (Gao et al., 2026): it fixes domain weights but retains length weighting within each domain. Appendix B extends this identity to arbitrary response and position weights.

To measure the change in gradient direction, we compare DT and DR on the same responses at fixed student parameters. Their mean gradient cosine is 0.68 in six batches (Figure 2b). In full training runs, the range of four-task mean scores among GT, DT, and DR after 500 steps is 0.28 percentage points under Adam with the PG loss and 0.06 with the top-64 intersection KL loss. Under SGD with the PG loss, this range is 0.68 points (Table 1).

Table 1: **MOPD scores by loss averaging and distillation loss.** Scores (%) are mean ± standard deviation across five training seeds; Mean equally weights the four task means. Domain responses and Domain tokens assign equal domain weights, averaging responses equally or by length, respectively. Global token averaging weights all response tokens equally.

Loss averaging	Math	Code	IF	Science	Mean
Initial student	68.80	16.40	18.33	32.20	33.93
Adam, PG loss					
Domain responses	73.36 ± 0.79	18.26 ± 0.71	28.07 ± 1.19	35.94 ± 1.42	38.91 ± 0.69
Domain tokens	73.28 ± 0.69	19.24 ± 0.83	27.07 ± 1.01	34.93 ± 1.35	38.63 ± 0.65
Global tokens	76.32 ± 0.73	17.91 ± 0.66	27.13 ± 1.24	33.81 ± 1.54	38.79 ± 0.70
Adam, top-16 intersection KL loss					
Domain responses	75.48 ± 0.89	17.73 ± 0.62	26.80 ± 1.12	35.98 ± 1.31	39.00 ± 0.67
Adam, top-64 intersection KL loss					
Domain responses	75.96 ± 0.80	18.29 ± 0.75	27.33 ± 1.00	34.68 ± 1.48	39.07 ± 0.68
Domain tokens	73.80 ± 0.97	18.86 ± 0.69	28.33 ± 1.35	35.19 ± 1.27	39.05 ± 0.72
Global tokens	74.20 ± 0.80	19.77 ± 0.88	28.67 ± 1.20	33.38 ± 1.61	39.01 ± 0.59
SGD, PG loss					
Domain responses	76.48 ± 0.73	19.53 ± 0.79	28.20 ± 1.12	35.56 ± 1.36	39.94 ± 0.52
Domain tokens	75.20 ± 0.87	20.28 ± 0.92	28.47 ± 1.12	33.88 ± 1.45	39.46 ± 0.55
Global tokens	74.96 ± 1.05	19.24 ± 0.73	27.53 ± 1.35	35.32 ± 1.52	39.26 ± 0.60

GT assigns mathematics about 44% of the loss weight (Figure 2a). Relative to DR, GT raises mathematics accuracy by 3.0 points and lowers science by 2.1 points with the PG loss. With top-64, mathematics falls by 1.8 points despite its larger weight, while code rises by 1.5 points despite its smaller weight (Table 1). Thus, the loss changes how weighting affects task performance (Section 5.3).

3.2 Adam reduces differences in update direction

For each of these six batches, we apply the DT and DR gradients separately using the same saved Adam state. The resulting one-step parameter updates have a mean cosine of 0.96 (Figure 2b). Adam mixes current gradients with past gradients through its first moment and rescales coordinates through its second moment (Kingma & Ba, 2015).

DR triggers more clipping, which rescales gradients without changing direction. Resetting Adam's first moment shrinks steps; resetting both moments enlarges them (Figure 3a,b). For each rule, we scale a single SGD step to match its saved-state Adam step norm. The held-out KL change differs more between rules under SGD than under saved-state Adam (panel c). During training, Adam step norms become similar despite distinct gradient norms, but cumulative parameter changes remain different (panels d–f).

3.3 Raising the response cap shifts token-averaged objectives most

We compare maximum training response lengths of 4,096 (4K) and 8,192 (8K) generated tokens per response. Unlike RL with final-answer rewards (Yu et al., 2025), on-policy distillation can supervise incomplete responses using teacher probabilities at each prefix (Agarwal et al., 2024). Increasing this limit can add supervised token positions, redistributing even DR's fixed response weight. At three checkpoints, paired batches generated under the 4K and 8K limits share student and Adam states. Relative raw-gradient differences, $\|g_{8K} - g_{4K}\|/\|g_{4K}\|$, follow GT > DT > DR (Figure 4b); clipping and Adam can alter this ranking. In training, the 8K-minus-4K mean-score differences at step 100 are GT $+1.05$, DT $+0.32$, and DR -0.83 percentage points (panel c).

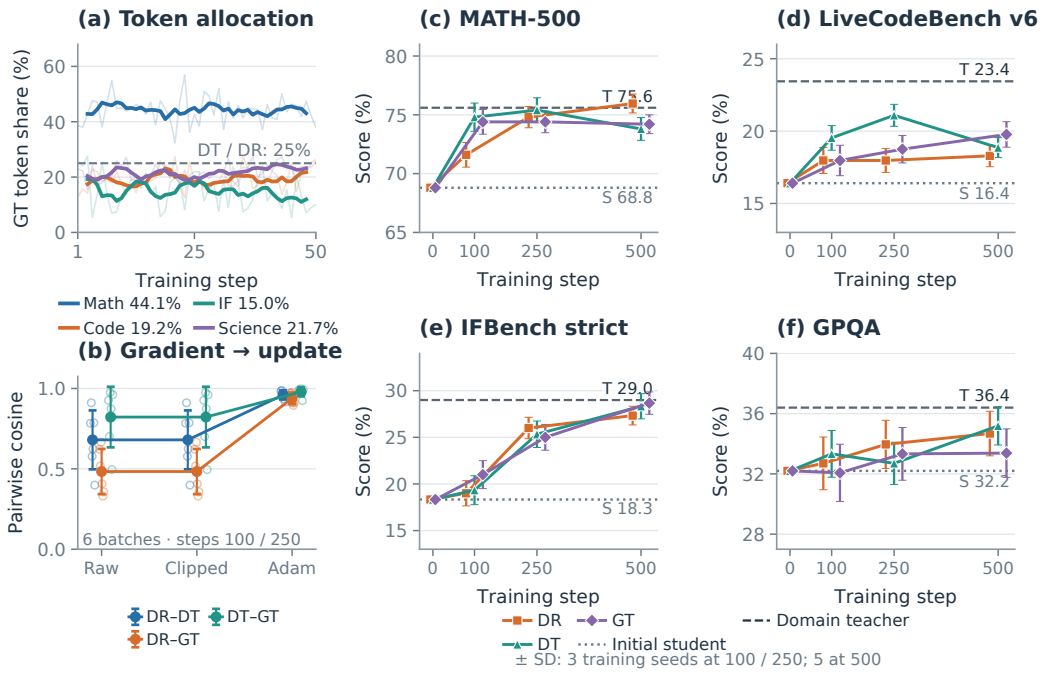

Figure 2: **Loss averaging changes gradients, while shared Adam state aligns updates.** (a) Domain token shares under global token averaging. (b) Gradient and update cosines for six batches (three per checkpoint). (c–f) Per-task learning curves with the top-64 intersection KL loss. Error bars show standard deviations from three training seeds at steps 100 and 250, and five at step 500. GT/DT/DR denote global token, domain token, and domain response averaging. S/T mark the initial student and domain teacher.

4 BF16 Rounding and Adam Momentum Obscure How Teachers Change the Student

4.1 BF16 rounding hides widespread FP32 changes

Under domain response averaging, about 97% of the MOPD student's FP32 master weights differ from initialization, versus 7–9% after rounding them to BF16 (7–11% including DT and GT; Figure 5a). The master weights retain higher precision for optimization. For each parameter, change means its final value minus its initial value. We rank parameters by squared change. About one third account for 90% of the total in FP32, versus about 4% in BF16. Rounding therefore hides small changes and makes widespread parameter movement appear sparse.

4.2 Momentum hides differences between teachers

We fix parameters and Adam state from the MOPD student trained with the PG loss. For each teacher, we compute the corresponding student loss gradient and simulate one FP32 parameter update. Adam's first moment mixes past gradients into each update. When teachers score student responses from their own domains, the mean cosine, computed using all teacher pairs and batches, exceeds 0.83 with the PG and top-64 intersection KL losses. Resetting the first moment while retaining the second, or using norm-matched SGD, reduces it to near zero (Figure 5b). Shared momentum therefore explains most of this alignment. Momentum can move parameters even with zero current gradient. We subtract this baseline to measure how the current gradient changes Adam's update, including its second-moment scaling. Let g_d be the student loss gradient from teacher d's supervision, H the saved Adam moments and step counter, and $U(g; H)$ the resulting parameter update:

$$\delta U_d = U(g_d; H) - U(0; H). \tag{6}$$



Figure 3: **Optimizer processing attenuates differences between averaging rules.** (a) Gradient retention and clipping rates. (b) Adam step norms with saved or reset moments (step count retained). (c) Held-out reverse KL after minus before one step; SGD matches each rule's saved-Adam step norm. Open points: batches; filled points/bars: mean/SD. (d–f) Raw gradient norms, FP32 step norms, and cumulative FP32 displacement under Adam with the top-64 intersection KL loss. Faint/solid traces show individual steps/25-step medians; dotted line: clipping threshold. GT/DT/DR denote global token, domain token, and domain response averaging.

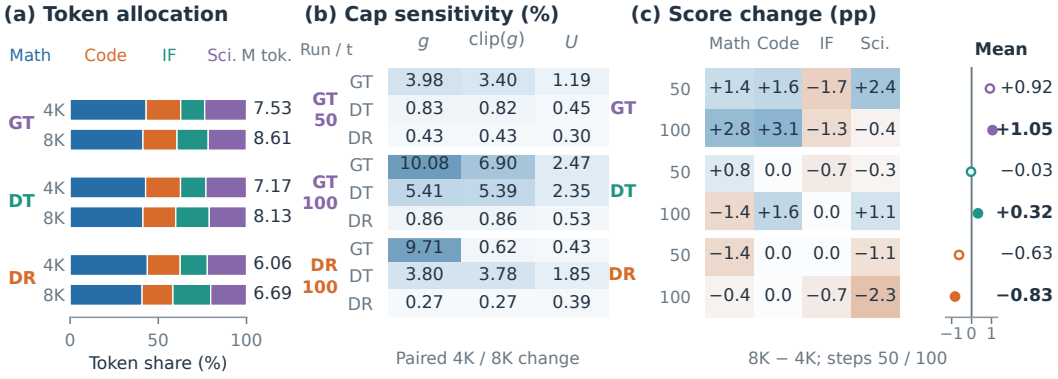

Figure 4: **Loss averaging influences the effect of doubling the maximum training response length from 4,096 (4K) to 8,192 (8K) tokens.** (a) Domain token shares and total response tokens (millions) during 100 steps. (b) Relative 4K/8K changes in gradients (g), clipped gradients, and Adam steps (U); groups name source 4K run and training step, rows mark averaging rules. Each group averages three paired batches. (c) Task score differences (8K minus 4K, percentage points); adjacent points show equal-domain means at steps 50 (open) and 100 (filled). GT/DT/DR denote global token, domain token, and domain response averaging.

The mean pairwise cosine of these differences is below 0.01 on domain-specific inputs. Having all teachers score identical student prefixes holds inputs fixed and raises it to about 0.24–0.35 (Figure 5c). Teacher and input differences can therefore remain hidden within aligned Adam updates.

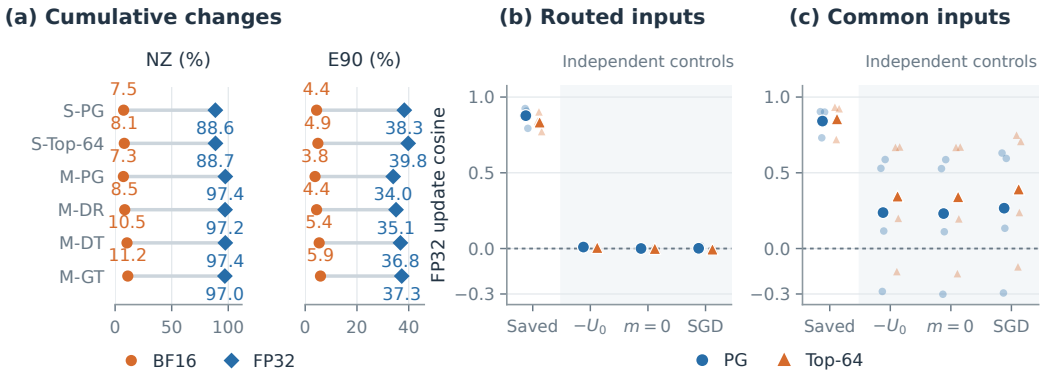

Figure 5: **BF16 rounding hides small changes; shared Adam momentum aligns student updates.** (a) FP32/BF16 parameter fractions: changed from initialization (NZ), or carrying 90% of squared change (E90). S/M: math-only/MOPD training; PG: sampled-token loss; Top-64: intersection KL loss. M-DR/M-DT/M-GT: Top-64 with response/domain-token/global-token averaging; other rows average responses. (b,c) Pairwise update cosines: own-domain responses (Routed) or shared prefixes (Common). Saved retains trained Adam state; $-U_0$ subtracts the zero-gradient step; $m = 0$ resets the first moment. Small points average teacher pairs within batches; large points average batches.

4.3 Momentum-free SGD achieves higher mean scores

In MOPD, the student generates new responses as it learns, changing the prefixes teachers supervise. Momentum may retain directions favored by earlier responses and delay adaptation to current supervision, a concern also raised for reinforcement learning (Mukherjee et al., 2026). In full training runs with the PG loss, momentum-free SGD, which keeps no optimizer history, achieves higher four-task mean scores than Adam on all three averaging rules (DR: 39.94 vs. 38.91; DT: 39.46 vs. 38.63; GT: 39.26 vs. 38.79; Table 1).

5 Closer Vocabulary Gradients Do Not Consistently Improve Capabilities

5.1 Top-64 intersections nearly recover the full-vocabulary gradient

With fixed domain weights, vocabulary truncation can change the student's combined loss gradient (Anshumann et al., 2025; Huang & Wei, 2026). We measure retained student probability, $c_k(h) = \sum_{v \in I_k} p_\theta(v \mid h)$, and agreement with the full-vocabulary KL gradient. In Qwen, the top-64 intersection retains, on average, more than 99.9% of total student probability and more than 99.97% of the student's own top-64 mass. Its gradient direction nearly matches the full-vocabulary KL gradient (Figure 6b). Averaging gradients from 1, 16, or 64 samples per fixed prefix raises cosine to this reference from 0.54 to 0.87 and 0.97 (Figure 8a).

SmolLM3-3B illustrates the limits of mean coverage: top-64 retains 97.42% initially and 98.56% after training, yet 8.42% and 3.65% of weighted prefixes, respectively, retain less than 90% of student probability. After training, relative gradient error spans 2.16–24.54% among diagnostic batches (Figure 7).

5.2 Supervision breadth barely affects the fraction of changed BF16 weights

Starting from identical student parameters and Adam state, the PG, top-64 intersection KL, teacher-top-64, and full-vocabulary KL losses each change nearly the same fraction of BF16 weights in one step (Figure 6a). Adam history strongly affects these counts: even with zero current gradient, about 0.14% of BF16 weights change, close to the supervised fractions. Resetting only the first moment lowers the fraction; fully resetting Adam raises it (Figure 8b,c). Therefore, the fraction of BF16

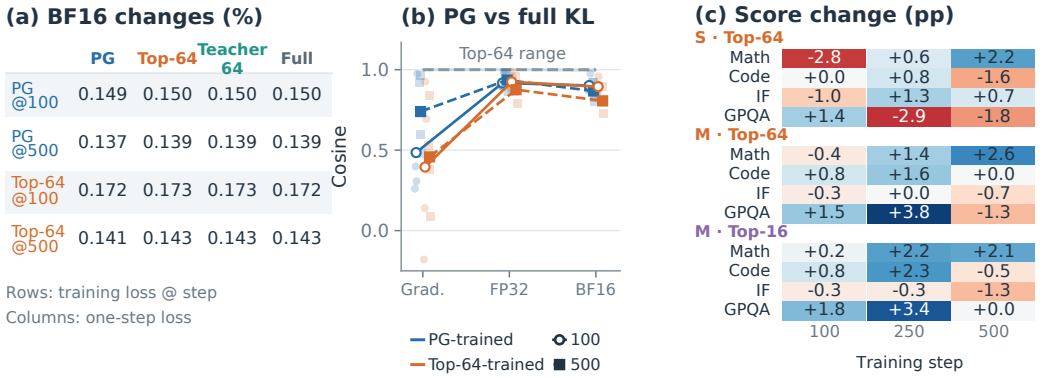

Figure 6: **Top-64 closely matches full-vocabulary gradient directions in Qwen.** (a) Rows: training loss@step. Columns compare one-step BF16 changed fractions for different losses at fixed parameters, responses, and Adam state. (b) PG-to-full gradient and update cosines across training snapshots; gray shows Top-64-to-full ranges. (c) Score differences from PG under response averaging. PG: sampled-token loss; Top-16/Top-64: intersection KL losses; Teacher-64: teacher-supported loss without renormalization (Appendix D); Full: full-vocabulary KL loss. S/M: mathematics-only/MOPD training.

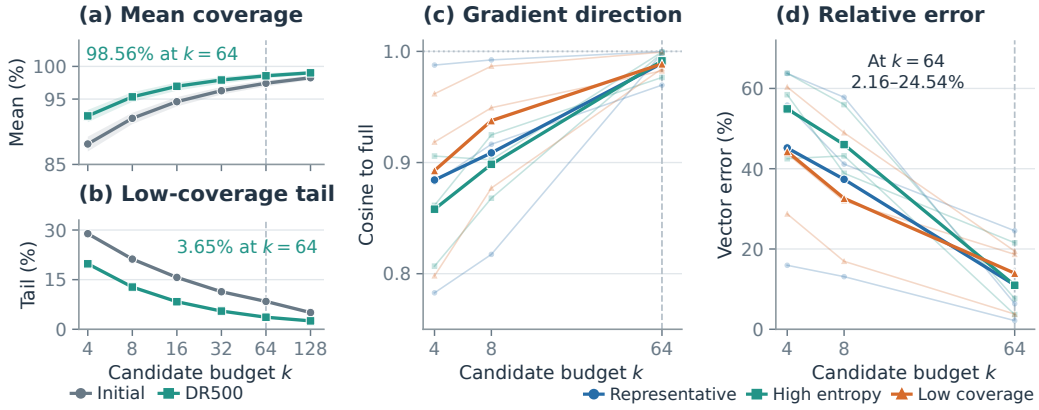

Figure 7: **High mean coverage can hide gradient error in SmolLM.** (a,b) Mean intersection coverage and fraction below 90%; domains, responses within domains, and positions within responses are equally weighted. Bands: response-cluster 95% intervals. (c,d) Gradient cosine and relative error against full-vocabulary KL after training; relative error is the gradient-difference norm divided by the full-gradient norm. Dashed lines: $k = 64$. Thin/bold lines: batches/means. Initial: initialization; DR500: 500 steps with top-64 intersection KL and response averaging.

weights that change reflects optimizer history and rounding (Section 4). Coverage and gradient agreement assess the truncation approximation.

5.3 Task gains from top-k depend on the averaging rule

Under response averaging, top-64 exceeds PG in mathematics accuracy by 2.6 percentage points, while instruction following and science scores decrease; the four-task mean rises by 0.16 points. Under global token averaging, mathematics accuracy falls by 2.1 points, while code and instruction following improve; the mean rises by 0.21 points (Table 1). Close gradient agreement therefore coexists with task-specific gains and losses, and response weighting changes which tasks benefit.

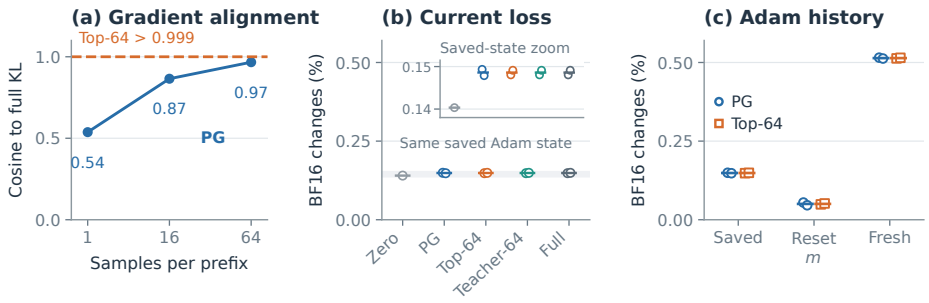

Figure 8: **Sampling improves gradient alignment; Adam history affects BF16 changes.** (a) Mean gradient cosine to full KL. (b,c) Changed BF16 fractions with shared axes; inset repeats (b)'s categories. PG: sampled-token loss; Top-64: intersection KL; Teacher-64: teacher-supported loss; Full: full-vocabulary KL. Saved retains Adam state; reset-m clears its first moment; fresh Adam clears both moments and step count. In (b,c), open points show measurements; bars show means.

These outcomes show that gradient fidelity at fixed prefixes is not a sufficient predictor of task-level gains: PG is unbiased for the full reverse-KL gradient, and top-64 approximates its direction with cosine above 0.999 in Qwen, yet their task effects differ across averaging rules.

6 Related Work

Distillation objectives and vocabulary truncation. MiniLLM uses reverse KL (Gu et al., 2024); GKD distills student-generated responses with alternative divergences (Agarwal et al., 2024); f-DISTILL derives token losses from sequence f-divergences (Wen et al., 2023). Sparse Logit Sampling examines top-k bias (Anshumann et al., 2025); Tail-Aware OPD restores discarded tail-mass supervision (Huang & Wei, 2026). We compare coverage, gradient fidelity, and task gains, accounting for sampling noise and optimizer processing.

Multi-teacher distillation and loss weighting. MOPD combines specialized teachers (Ma et al., 2026). Open-MOPD corrects domain token shares but retains within-domain length weighting (Gao et al., 2026). GradNorm uses gradient norms to balance task training rates (Chen et al., 2018). DAPO averages token losses (Yu et al., 2025), while Dr. GRPO analyzes length and reward-normalization bias (Liu et al., 2025). We derive response weights' effects through their covariance with gradients and measure changes in Adam updates and task scores.

Parameter changes and gradient alignment. Task arithmetic combines parameter differences from a shared initialization (Ilharco et al., 2023). Prior work studies small parameter subsets supporting RL (Mukherjee et al., 2025), concentrated changes among distilled students (Yu et al., 2026), and low-dimensional OPD updates (Shen et al., 2026). PCGrad projects conflicting gradients (Yu et al., 2020); CAGrad uses worst-task improvement (Liu et al., 2021). We separate current supervision from Adam history, find nearly orthogonal teacher increments on routed inputs, and distinguish FP32 movement from BF16 sparsity.

7 Discussion

We study how multi-teacher supervision becomes capability through student gradients and parameter updates. Domain balancing retains within-domain length weighting; Adam history aligns updates; BF16 rounding hides small FP32 changes. Closer full-vocabulary gradient approximation does not consistently improve task scores. These findings support choosing response weighting, vocabulary supervision, and optimizer settings together for the target tasks and training budget. Our findings are derived from limited model families and scales, and the observation may differ in larger-scale settings. Future work may explore whether smaller Adam β_1 improves adaptation to teacher signals, and whether gradient clipping and Adam's second-moment normalization could explain why losses with similar expected gradients produce different task outcomes.

AI USE STATEMENT

We used a large language model for minor editorial assistance, including correcting grammar and typo and refining formatting. It did not contribute to the development of research ideas, the analysis, or the claims.

ETHICS STATEMENT

All datasets, models in this work are publicly available. The usage of these datasets, models, code strictly follows the terms and conditions set by the providers.

REPRODUCIBILITY STATEMENT

To ensure reproducibility of our results, we provide detailed documentation of our experimental setup and methodology in the main paper and appendix. The code for data, evaluation, and training will be open-sourced upon publication.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Anshumann, Mohd Abbas Zaidi, Akhil Kedia, Jinwoo Ahn, Taehwak Kwon, Kangwook Lee, Haejun Lee, and Joohyung Lee. Sparse logit sampling: Accelerating knowledge distillation in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18085–18108. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.885. URL https://aclanthology.org/2025.acl-long.885/.

Lei Bai, Zongsheng Cao, Yang Chen, Zhiyao Cui, Shangheng Du, Yue Fan, Shiyang Feng, Zijie Guo, Haonan He, Liang He, Xiaohan He, Shuyue Hu, Yusong Hu, Songtao Huang, Yichen Jiang, Hao Li, Xin Li, Dahua Lin, Weihao Lin, Fenghua Ling, Dongrui Liu, Zhuo Liu, Runmin Ma, Chunjiang Mu, Haoyang Peng, Tianshuo Peng, Jinxin Shi, Luohe Shi, Boyuan Sun, Zelin Tan, Shengji Tang, Qianyi Wang, Yiming Wu, Yi Xie, Xiangchao Yan, Jingqi Ye, Peng Ye, Fangchen Yu, Jiakang Yuan, Bihao Zhan, Bo Zhang, Chen Zhang, Shufei Zhang, Shuaiyu Zhang, Wenlong Zhang, Yiqun Zhang, Junpeng Zhao, Zhijie Zhong, Bowen Zhou, and Yuhao Zhou. Scaling the horizon, not the parameters: Reaching trillion-parameter performance with a 35b agent, 2026. URL https://arxiv.org/abs/2606.30616.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 794–803, 2018. URL https://proceedings.mlr.press/v80/chen18a.html.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao

Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Huan-ang Gao, Haohan Chi, Yong Yan, Shiyuan Feng, Hanlin Wu, Zheng Jiang, Bingxiang He, Wei-Ying Ma, Ya-Qin Zhang, and Hao Zhou. Open-mopd: Diagnosing and fixing capability imbalance in multi-teacher on-policy distillation, 2026. URL https://arxiv.org/abs/2608.19098.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.08543v4.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems*, 2021. doi: 10.48550/arXiv.2103.03874. URL https://arxiv.org/abs/2103.03874v2.

Huipeng Huang and Hongxin Wei. Tail-aware top-k on-policy distillation, 2026. URL https://arxiv.org/abs/2608.14728.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic, 2023. URL https://arxiv.org/abs/2212.04089.

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, 2024. URL https://arxiv.org/abs/2403.07974v2.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, M. C., Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Guanduo Chen, Guangyu Chen, Guanzheng Chen, Huarong Chen, Jia Chen, Jianlong Chen, Jun Chen, Kexin Chen, Peng Chen, Ruijue Chen, Wentao Chen, Xin Chen, Yang Chen, Yanru Chen, Yifei Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Dazhi Cheng, Yean Cheng, Jialei Cui, Jingbing Cui, Anqi Dai, Jiaqi Deng, Hao Ding, Rui Ding, Shaofeng Ding, Mengfan Dong, Mengnan Dong, Yuhao Dong, Yuxin Dong, Angang Du, Chenzhuang Du, Dikang Du, Jusen Du, Yulun Du, Yu Fan, Jing Feng, Qiulin Feng, Yichen Feng, Kelin Fu, Qiang Fu, Fuxuan Gao, Hongcheng Gao, Jingyue Gao, Tong Gao, Weijia Gao, Shangyi Geng, Jie Gong, Linhu Gong, Shengao Gong, Xiaochen Gong, Qizheng Gu, Yicheng Gu, Shuhao Guan, Haiqing Guo, Shiqi Guo, Xiang Guo, Zhengyan Guo, Beixi Hao, Wenxin Hao, Xiaoru Hao, Dailan He, Haotian He, Lehan He, Qi He, Weiran He, Xinran He, Xinyi He, Yibo He, Yunjia He, Chao Hong, Tiange Hong, Hao Hu, Jiaxi Hu, Ruikun Hu, Weiming Hu, Yangyang Hu, Zhenxing Hu, Liang Hua, Jinbin Huang, Ke Huang, Ruiyuan Huang, Siying Huang, Weixiao Huang, Yan Huang, Zhengjie Huang, Zhiqi Huang, Yulong Hui, Chaobo Jia, Yutong Jiang, Zhejun Jiang, Zuoyou Jiang, Wenyi Jin, Xinyi Jin, Yu Jing, Huanjun Kong, Guokun Lai, Aidi Li, Cheng Li, Chengyuan Li, Cong Li, Fang Li, Guanyu Li, Haoyang Li, Jia Li, Junxiong Li, Lei Li, Letian Li, Lincan Li, Weihong Li, Wentao Li, Xintong Li, Yang Li, Yishen Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zhaoxi Li, Zheming Li, Zhengxiao Li, Zhiyuan Li, Jiawei Lin, Xiaohan Lin, Yibo Lin, Zichao Lin, Ziyan Lin, Bill Liu, Boxiao Liu, Chuan Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yipeng Liu, Zhengying Liu, Zhiheng Liu, Enzhe Lu, Haoyu Lu, Linqiang Lu, Tingzhan Lu, Zhiyuan Lu, Aotian Luo, G. Luo, Junyu Luo, Yifan Luo, B. Lyu, Wenzhou Lyu, Shaoguang Mao, Yuan Mei, Xin Men, Minqing Ni, Yixuan Niu, Siyuan Pan, Shujun Peng, Zhangyang Qi, Ruoyu Qin, ZeChao Qin, Zeyu Qin, Haiquan Qiu, Jianxin Qiu, Jiezhong Qiu, Bowen Qu, Yuhao Qu, Zeyu Shang, Youbo Shao, Han Shen, Jincheng Shi, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Wingchun Siu, Pengwei Song, Xiaoxi Song, Jianlin Su, Yunfeng Su, Zhaochen Su, Lin Sui, Jingsong Sun, Junyao Sun, Shaoning Sun, Shuzhe Sun, Tongyu Sun, Yujun Sun, Yunpeng Tai, Chuning Tang, Heyi Tang, Sirui Tang, Zecheng Tang, Chaoran Tian, Rongpeng Tian, Yu Tian, Wei Tu, Chensi Wang, Chuang Wang, Chunjie Wang, Dinglu Wang, Feng Wang, Hailong Wang, Haiming Wang, Hao Wang, Hao Wang, Huaqing Wang, Hui Wang, Jiayi Wang, Jinglong Wang, Jinhong Wang, Jiuzheng Wang, Linian Wang, Shaobo Wang, Shenzhi Wang, Shuyi Wang, Si Wang, Siyuan Wang, Tianfu Wang, Wenjue Wang, Xingran Wang, Xinmei Wang, Xinyuan Wang, Xusheng Wang, Yalin Wang, Yangkun Wang, Yao Wang, Yaoyu Wang, Yejie Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhenhao Wang, Zhongsheng Wang, Zifan Wang, Chu Wei, Ming Wei, Shouxin Wei, Zichen Wen, Fan Wu, Haoning Wu, Rucong Wu, Wenhao Wu, Xiaoxue Wu, Yingcong Wu, Yongqi Wu, Yuxin Wu, Zijian Wu, Xinglang Xian, Chenxuan Xiang, Yuye Xiang, Bocheng Xiao, Chenjun Xiao, Xin Xiao, Jin Xie, Xiaotong Xie, Yifeng Xie, Zhe Xie, Bowei Xing, Yiming Xiong, Baosheng Xu, Boyu Xu, Jiale Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Qingtao Xu, Shuyao Xu, Suting Xu, Tiantian Xu, Tianxiang Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ye Xu, Yueni Xu, Ziyao Xu, Haonan Xue, Junjie Yan, Yaoyao Yan, Fan Yang, Guangyao Yang, Hao Yang, Junwei Yang, Ruoyu Yang, Wenjie Yang, Xiaofei Yang, Xinyu Yang, Yi Yang, Yiling Yang, Ying Yang, Yuchen Yang, Zhen Yang, Zhilin Yang, Zian Yang, Zuhao Yang, Haotian Yao, Dan Ye, Haoran Ye, Wenjie Ye, Zhanbo Ye, Bohong Yin, Haoxiang Yin, Xietong Yin, Chengzhen Yu, Haozhen Yu, Longhui Yu, Shengnan Yu, Shuying Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Tongtian Yue, Wei Yue, Yang Yue, Dunyuan Zha, Haobing Zhan, B. H. Zhang, Dehao Zhang, Fei Zhang, Hao Zhang, Haoyuan Zhang, Huanyu Zhang, Jiapei Zhang, Jiaxuan Zhang, Jin Zhang, Kaiyi Zhang, Miaozhen Zhang, Puqi Zhang, Qinglei Zhang, Rong Zhang, Rui Zhang, Shaoshuai Zhang, Shiyi Zhang, Xiaobin Zhang, Xiaoyun Zhang, Y. Zhang, Yangkun Zhang, Ye Zhang, Yichi Zhang, Yikun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Zijing Zhang, Bin Zhao, Chenguang Zhao, Feifan Zhao, Jinglun Zhao, Jinxiang Zhao, Shuai Zhao, Wenshuo Zhao, Xiangyu Zhao, Xuanle Zhao, Yikai Zhao, Zijia Zhao, Haozhi Zheng, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang Zheng, Haofeng Zhong, Lei Zhong, Longguang Zhong, M. Zhou, Qiankang Zhou, Runjie Zhou, Ruozhang Zhou, Xinyu Zhou, Yiqiao Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yangjunfeng Zhu, Yuxuan Zhu, Zhen Zhu, Chen Zhuang, Weiyu Zhuang, and Xinxing Zu. Kimi k3: Open frontier intelligence, 2026. URL https://arxiv.org/abs/2607.24653.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe, 2026. URL https://arxiv.org/abs/2604.13016.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18878–18890, 2021. URL https://arxiv.org/abs/2110.14048.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective, 2025. URL https://arxiv.org/abs/2503.20783.

Ziyuan Liu, Jiao Ou, Jian Liang, Ruiming Tang, and Cheng Luo. Preserving general capabilities during domain specialization with uncertainty-calibrated mopd, 2026. URL https://arxiv.org/abs/2608.26735.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, Jinhao Dong, Zhifang Sui, and Fuli Luo. MOPD: Multi-teacher on-policy distillation for capability integration in llm post-training, 2026. URL https://arxiv.org/abs/2606.30406v1.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tür, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.11711.

Sagnik Mukherjee, Lifan Yuan, Pavan Jayasinha, Dilek Hakkani-Tür, and Hao Peng. Do we need Adam? surprisingly strong and sparse reinforcement learning with SGD in LLMs. In *Forty-third International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.07729.

NVIDIA. Nemotron 3 ultra: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning, 2026. URL https://arxiv.org/abs/2606.15007.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following, 2025. URL https://arxiv.org/abs/2507.02833v3.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022v1.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Zhennan Shen, Yanshu Li, Qingyu Yin, Chak Tou Leong, Zhilin Wang, Yanxu Chen, Rongduo Han, Sunbowen Lee, and Yi R. Fung. On the geometry of on-policy distillation, 2026. URL https://arxiv.org/abs/2606.07082.

Core Team, Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, Gang Xie, Hailin Zhang, Hanglong Lv, Hanyu Li, Heyu Chen, Hongshen Xu, Houbin Zhang, Huaqiu Liu, Jiangshan Duo, Jianyu Wei, Jiebao Xiao, Jinhao Dong, Jun Shi, Junhao Hu, Kainan Bao, Kang Zhou, Lei Li, Liang Zhao, Linghao Zhang, Peidian Li, Qianli Chen, Shaohui Liu, Shihua Yu, Shijie Cao, Shimao Chen, Shouqiu Yu, Shuo Liu, Tianling Zhou, Weijiang Su, Weikun Wang, Wenhan Ma, Xiangwei Deng, Bohan Mao, Bowen Ye, Can Cai, Chenghua Wang, Chengxuan Zhu, Chong Ma, Chun Chen, Chunan

Li, Dawei Zhu, Deshan Xiao, Dong Zhang, Duo Zhang, Fangyue Liu, Feiyu Yang, Fengyuan Shi, Guoan Wang, Hao Tian, Hao Wu, Heng Qu, Hongfei Yi, Hongxu An, Hongyi Guan, Xing Zhang, Yifan Song, Yihan Yan, Yihao Zhao, Yingchun Lai, Yizhao Gao, Yu Cheng, Yuanyuan Tian, Yudong Wang, Zhen Tang, Zhengju Tang, Zhengtao Wen, Zhichao Song, Zhixian Zheng, Zihan Jiang, Jian Wen, Jiarui Sun, Jiawei Li, Jinlong Xue, Jun Xia, Kai Fang, Menghang Zhu, Nuo Chen, Qian Tu, Qihao Zhang, Qiying Wang, Rang Li, Rui Ma, Shaolei Zhang, Shengfan Wang, Shicheng Li, Shuhao Gu, Shuhuai Ren, Sirui Deng, Tao Guo, Tianyang Lu, Weiji Zhuang, Weikang Zhang, Weimin Xiong, Wenshan Huang, Wenyu Yang, Xin Zhang, Xing Yong, Xu Wang, Xueyang Xie, Yilin Jiang, Yixin Yang, Yongzhe He, Yu Tu, Yuanliang Dong, Yuchen Liu, Yue Ma, Yue Yu, Yuxing Xiang, Zhaojun Huang, Zhenru Lin, Zhipeng Xu, Zhiyang Chen, Zhonghua Deng, Zihan Zhang, and Zihao Yue. Mimo-v2-flash technical report, 2026. URL https://arxiv.org/abs/2601.02780.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10817–10834. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.605. URL https://aclanthology.org/2023.acl-long.605/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388v1.

Guo Yu, Wenlin Liu, Yulan Hu, Hao-Xuan Ma, Jun-Peng Jiang, and Han-Jia Ye. Dense supervision, sparse updates: On the sparsity and geometry of on-policy distillation, 2026. URL https://arxiv.org/abs/2606.13657v3.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan 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, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020. URL https://arxiv.org/abs/2001.06782.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes, 2026. URL https://arxiv.org/abs/2605.11182.

A Experimental Setup

A.1 Models, data, and training

MOPD training. Our main experiments compare how normalization, loss formulation, and optimizer choice affect learning from the same four domain teachers. The student is Qwen3-1.7B (Yang et al., 2025), and the teachers were trained by reinforcement learning from the same initialization to specialize in mathematics, code, instruction following, and science. The main training comparisons use seeds 42, 43, 44, 45, 46.

AdamW uses learning rate 2.5×10^{-7}, $(\beta_1, \beta_2) = (0.9, 0.98)$, $\epsilon = 10^{-8}$, zero weight decay, a constant learning rate, and global norm-1 clipping. SGD uses learning rate 2.5×10^{-2}, following the setting selected by Mukherjee et al. (2026) through an analysis of AdamW's effective per-parameter learning rates and an SGD learning-rate sweep. Student responses are sampled at temperature 1 and top-$p = 1$, with a 2,048-token prompt cap and a maximum response length of 4,096 tokens (4K), increased to 8,192 tokens (8K) in the response-length comparison (Figure 4). Each training step aggregates one fresh response for each of 64 prompts.

MOPD training allocates 16 prompts to each of the four domains and assigns target domain weights of 0.25 under the domain-balanced rules. Single-teacher training uses all 64 prompts from the teacher's domain. Top-64 and Top-16 denote intersection KL losses using Equation 3 with $k = 64$ and $k = 16$, respectively.

Experiments are conducted using 10 * NVIDIA RTX PRO 6000 Blackwell GPUs.

Domain data. We construct the four domain datasets from Nemotron-3-Nano-RL-Training-Blend. Table 2 lists the sources and the sizes of each dataset. The mathematics pool combines DAPO-Math-17k and the mathematics split of Skywork-OR1-RL-Data with OmniMath excluded. The code, instruction-following, and science pools use the competitive-coding, instruction-following, and STEM multiple-choice components, respectively.

Table 2: Domain data sources and converted pool sizes before stage-specific filtering.

Domain	Upstream dataset	Rows
Mathematics	`BytedTsinghua-SIA/DAPO-Math-17k` and `Skywork/Skywork-OR1-RL-Data`	22,056
Code	`nvidia/Nemotron-RL-coding-competitive_coding`	19,169
Instruction following	`nvidia/Nemotron-RL-instruction_following`	16,575
Science	`nvidia/Nemotron-RL-knowledge-mcqa` (STEM multiple-choice component)	19,670

Teacher construction. We train four domain-specific Qwen3-1.7B teachers independently from the common initialization using GRPO (Shao et al., 2024), implemented in Slime codebase. Each rollout batch contains 16 prompts, with the number of responses per prompt specified in Table 3. Generation uses temperature 1, top-$p = 1$, no top-k sampling truncation, a 2,048-token prompt limit, and an 8,192-token response limit. We apply the chat template with thinking disabled. Training uses AdamW with a constant learning rate of 10^{-6}, $(\beta_1, \beta_2) = (0.9, 0.9987381276)$, $\epsilon = 10^{-8}$, zero weight decay, no learning-rate warmup, and global gradient-norm clipping at 1.

Table 3: Teacher GRPO setup.

Domain	Responses per prompt	Batch Size	Updates
Mathematics	8	16	500
Code	16	16	500
Instruction following	16	16	500
Science	4	16	1000

A.2 Evaluation and scoring

We evaluate mathematics on MATH-500 (500 questions) (Hendrycks et al., 2021), code on LiveCodeBench v6 (1,055 questions) (Jain et al., 2024), instruction following on IFBench strict (300 questions) (Pyatkin et al., 2025), and science on GPQA Diamond (198 questions) (Rein et al., 2023). The first three benchmarks use one greedy response per question. GPQA uses 32 responses per question at temperature 1 and top-$p = 1$; average@32 accuracy first averages response correctness within each question and then averages these question scores. Evaluation permits up to 32,768 response tokens within a 32,768-token context. Reported cross-domain means weight the four benchmarks equally.

B Loss Averaging: Derivation and Controls

Response length changes both a domain's contribution to the loss and the relative weights of responses within it. For a fixed rollout batch, token masks, and domain weights, define the empirical covariance with normalization $1/n_d$, where $n_d = |\mathcal{B}_d|$. Expanding gives $\mathrm{Cov}_d(T, g) = n_d^{-1} \sum_i T_i g_i - \bar{T}_d \bar{g}_d$; dividing by $\bar{T}_d$ gives Equation 5. Reweighting each domain's global-token contribution by its target weight divided by its observed token share gives domain-token averaging exactly. The remaining length weighting within each domain is captured by this covariance.

The same identity holds when T_i is replaced by any nonnegative response weight f_i with positive mean, and analogously for token-level weights: weighting by f shifts the unweighted domain gradient by $\mathrm{Cov}_d(f, g)/\bar{f}_d$. Length is one implicit choice of f. UC-MOPD chooses weights explicitly by retaining rollouts with strong positive learning signal (Liu et al., 2026). Each choice changes the domain gradient through the covariance between f and per-response gradients. Within each domain, averaging rules also reweight positions: late positions occur only in long responses, and DR gives each token of a long response a smaller weight, so DR moves loss weight toward early positions relative to DT and GT.

B.1 Fixed-batch gradient and optimizer comparisons

To measure the effect of averaging alone, we reuse the same responses, token masks, teacher scores, and student state for all three rules. At steps 100 and 250 of the MOPD student trained with the PG loss, we cache three response batches (seeds 1042, 1043, and 1044), each with eight responses per domain, using the training prompt format and a 4,096-token response cap. We compute each response's mean top-64 gradient at the saved BF16 parameter values in FP32, then aggregate under domain-response (DR), domain-token (DT), and global-token (GT) averaging.

In these six batches, the DR–DT raw-gradient cosine ranges from 0.40 to 0.91 (mean 0.68; Figure 2b). Therefore length weighting changes the gradient direction even when domain weights are fixed. We verify the covariance identity in FP64 for all 24 batch-domain groups; the largest relative residual is 1.53×10^{-15}.

The optimizer interventions in Figure 3 ask how much of this gradient difference survives the update. Each aggregate gradient is clipped at norm 1 and applied to the same saved FP32 Adam state. The saved-state branch retains both moments; reset-m zeros the first moment; reset-(m, v) zeros both moments. Both reset branches retain the step counter. Momentum-free SGD follows the negative gradient and is rescaled separately for each averaging rule to match that rule's saved-state Adam step norm. Each branch applies one step to a checkpoint copy, casts the updated parameters to BF16, and measures full-vocabulary reverse KL on one held-out response per domain.

C Parameter and Optimizer Measurements

C.1 Recording gradients, optimizer steps, and weight changes

To distinguish the current learning signal from the change stored in the model, we record gradients before and after clipping, FP32 optimizer steps, and cumulative displacement from initialization in both FP32 master weights and BF16 model weights. Global statistics count unique model coordi-

nates, excluding vocabulary padding and duplicate entries for tied weights. Energy concentration ranks coordinates by absolute change and measures their share of squared Euclidean norm.

Checkpoint measurements preserve FP32 master weights, BF16 model weights, Adam moments, hyperparameters, and step counters. Student and teacher values are checked against their saved models in the same precision. Forward and backward passes use FP32 with TF32 disabled. Diagnostic steps use a non-fused Adam implementation applied to copies of the saved state; online training uses fused Adam.

C.2 Separating the current gradient from optimizer history

The controls in Figure 8(b,c) start from the MOPD student trained with the PG loss at step 100. Two cached batches (seeds 42 and 43) each contain one response per domain, with a 2,048-token prompt cap and a 256-token response cap. All eight responses reach the cap. We sample four positions without replacement per response and weight domains and sampled positions equally; each domain teacher scores its own response.

We compare three Adam states. *Saved* retains both moments and the step counter. *Reset m* zeros the first moment while retaining the second moment and counter. *Fresh* zeros both moments and resets the counter. All gradients are clipped at norm 1, with zero weight decay. A separate zero-gradient control retains the trained moments and is evaluated once. Casting the saved FP32 master weights to BF16 without an optimizer step exactly reproduces the saved model, so this control isolates movement from the retained optimizer state.

D Vocabulary-Restricted Losses

D.1 Teacher-supported reference loss

To compare update concentration for different vocabulary supports, we supplement the PG and top-k intersection KL losses with the teacher-top-k reference loss and the full-vocabulary KL loss. Teacher-top-k uses the fixed teacher top-k set T_k and full-vocabulary probabilities, without renormalizing the probabilities of tokens in T_k:

$$\ell_{\text{Teacher-}k} = \sum_{v \in T_k} \left[p_\theta(v) \log \frac{p_\theta(v)}{q_d(v)} - p_\theta(v) + q_d(v) \right]. \tag{7}$$

Teacher-top-k and full KL serve as single-step references; our comparisons use $k = 64$ (Teacher-64 in figures). End-to-end training compares the PG loss with the top-k intersection KL loss at $k \in \{16, 64\}$.

D.2 Why we use top-k intersections

Our top-k intersection KL loss (Top-k) is motivated by an engineering constraint in teacher scoring. Computing a student-top-k loss requires the teacher's scores at the student's top-k token IDs, which depend on the response position. The SGLang teacher interface used in our pipeline does not directly support these position-dependent queries: `token_ids_logprob` accepts one shared list of token IDs for all positions. Zhu et al. (2026) documents this interface mismatch. A workaround queries the union of all student candidate sets at every position, but for a response of length L this returns $L \times |U|$ scores, where $U = \bigcup_{t=1}^{L} S_k(h_t)$, instead of the desired $L \times k$ scores. The union can grow to $\min(|\mathcal{V}|, Lk)$, substantially increasing memory and data-transfer costs.

We therefore obtain each model's top-k token IDs and scores and match IDs locally at each position, retaining $I_k = S_k \cap T_k$. Both models' scores are available on this intersection without additional teacher queries for student-selected IDs. We renormalize the probabilities of tokens in I_k and apply Equation 3; k is the candidate budget per model, while the supervised support can be smaller.

Li et al. (2026) find that intersection-based supervision recovers nearly all the performance gains of student-top-k OPD in their $k = 16$ reasoning experiments. Their result suggests that restricting supervision to shared candidates need not materially reduce OPD performance in that setting, providing empirical motivation for our intersection-based implementation.

E UNBIASEDNESS OF PG AT A FIXED PREFIX

Fix a domain d and a prefix $h = (x, y_{<t})$. Write $p_\theta(v) = p_\theta(v \mid h)$ and $q_d(v) = q_d(v \mid h)$. Assume a finite vocabulary $\mathcal{V}$, differentiable student probabilities, and $p_\theta(v), q_d(v) > 0$ for every $v \in \mathcal{V}$. The teacher is frozen, and all gradients below hold the prefix fixed.

Proposition 1 (Fixed-prefix unbiasedness)**.** *For a token $Y \sim p_\theta(\cdot \mid h)$, the PG surrogate in Equation 2 satisfies*

$$\mathbb{E}_{Y \sim p_\theta(\cdot|h)} \left[\nabla_\theta \ell_{\mathrm{PG}}(Y) \right] = \nabla_\theta \ell_{\mathrm{full}}, \tag{8}$$

where each sampled token and the argument of sg *are held constant when differentiating the surrogate.*

Proof. For a sampled token v, the stop-gradient operation gives

$$\nabla_\theta \ell_{\mathrm{PG}}(v) = (\log p_\theta(v) - \log q_d(v)) \, \nabla_\theta \log p_\theta(v). \tag{9}$$

Differentiating the full reverse-KL loss instead gives

$$\begin{aligned} \nabla_\theta \ell_{\mathrm{full}} &= \sum_{v \in \mathcal{V}} \nabla_\theta p_\theta(v) \left(\log \frac{p_\theta(v)}{q_d(v)} + 1 \right) \\ &= \sum_{v \in \mathcal{V}} p_\theta(v) \log \frac{p_\theta(v)}{q_d(v)} \, \nabla_\theta \log p_\theta(v) + \underbrace{\nabla_\theta \sum_{v \in \mathcal{V}} p_\theta(v)}_{=0} \\ &= \mathbb{E}_{Y \sim p_\theta(\cdot|h)} \left[\nabla_\theta \ell_{\mathrm{PG}}(Y) \right]. \end{aligned} \tag{10}$$

The second line uses $\nabla_\theta p_\theta(v) = p_\theta(v) \nabla_\theta \log p_\theta(v)$ and $\sum_v p_\theta(v) = 1$. □

F DETAILS OF TEACHER, VOCABULARY, AND OPTIMIZER CONTROLS

F.1 COMPARING TEACHERS AT A SHARED STUDENT STATE

The teacher controls separate differences in supervision from differences in inputs and optimizer history. Figure 5(a) compares cumulative parameter changes under domain-response averaging, with top-64 domain-token and global-token runs included for context; paired points show FP32 and BF16 statistics. Panels (b,c) start from the same MOPD student trained with the PG loss and its saved Adam moments. In the domain-specific comparison, each teacher scores responses from its own domain; in the shared-input comparison, all teachers score the same prefixes. Each of four cached batches contains all six teacher pairs. Small points summarize pairs within a batch, and large points average batches.

To measure the increment attributable to current supervision, Equation 6 subtracts the Adam step obtained with a zero current gradient before computing FP32 update cosines. SGD and reset-m steps are each rescaled to the norm of the saved-state Adam step. Reset-m retains the second moment and step counter.

F.2 COMPARING VOCABULARY SUPPORTS AT MATCHED PREFIXES

Figure 6(a,b) uses the MOPD students trained with the PG or top-64 intersection KL loss at steps 100 and 500, all trained with domain-response averaging. Each checkpoint supplies four cached batches and 96 prefixes. Within a checkpoint, losses share prefixes and Adam moments; responses are generated separately for each checkpoint. Panel (a) reports batch means. In panel (b), points show batches, lines connect PG batch means, and shading spans the top-64 measurements. The direction comparison includes the PG, top-64 intersection KL, and full-vocabulary KL losses. Panel (c) compares top-16 and top-64 training against PG training at the same task, teacher setting, and training step.

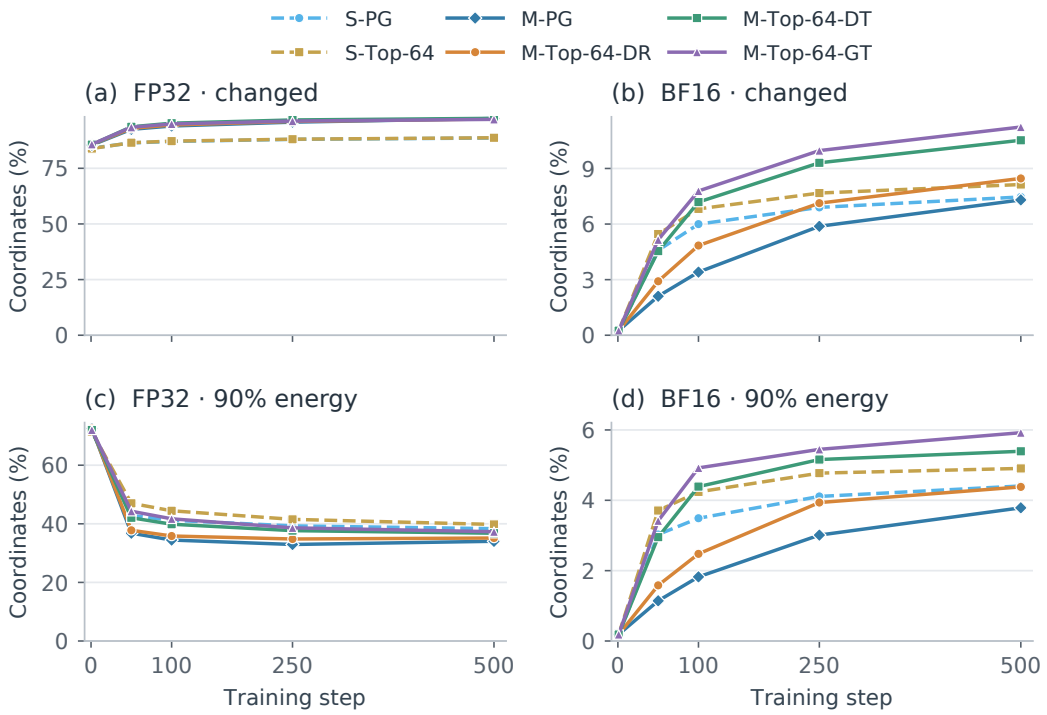

Figure 9: **Cumulative BF16 changes remain much sparser than FP32 changes.** PG and top-64 trajectories under domain-response averaging in single-teacher and MOPD training, with additional top-64 MOPD runs using domain-token/global-token averaging. Panels (a,b) show nonzero coordinate fractions; panels (c,d) show the fraction of coordinates containing 90% of squared displacement from initialization. Lines connect steps 1, 50, 100, 250, and 500; dashed lines denote single-teacher training. Figure labels use S/M for single-teacher/MOPD training and PG/Top-64 for the PG/top-64 intersection KL losses.

F.3 Additional optimizer and sampling controls

Figure 3(a) uses training logs from 500 steps per averaging rule. Boxes show interquartile ranges of the gradient fractions retained after clipping, whiskers show the 5th–95th percentiles, and labels give the fraction of training steps on which gradients are clipped. Panels (b,c) use the six batches and saved optimizer states described in Appendix B.1. Panel (c) first averages the four domains' probe-loss changes within each batch; error bars then show sample standard deviations computed from the three batches at each checkpoint.

The sampling-count experiment tests how averaging more token samples improves the approximation to the full-vocabulary KL gradient. Figure 8(a) uses three response batches from the MOPD student trained with the PG loss at step 100, with a 4,096-token response cap: 48 responses and 288 prefixes in total. Twelve fixed prefix subsets each receive two independent draws of 1, 16, or 64 actions per prefix. The top-64 intersection KL loss is evaluated once per subset. The mean PG gradient cosine to full KL rises from 0.54 to 0.87 and 0.97 as the sample count increases; top-64 exceeds 0.999. Panels (b,c) use the separate, shorter-response optimizer-history control in Appendix C.

G Intersection Coverage and Domain-Level Breakdown

G.1 Probability mass retained by the candidate intersection

To assess how concentrated student predictions support the close gradient agreement in Qwen, we measure the probability retained by the actual student–teacher intersection. The MOPD students

trained with the PG or top-64 intersection KL loss, both with domain-response averaging, are evaluated at steps 100 and 500. At each checkpoint, four cached batches provide one response per domain and six sampled positions per response, giving 96 prefixes. Each response is scored by its domain teacher. We record student mass $p_\theta(I_{64} \mid h)$ on the intersection $I_{64} = S_{64} \cap T_{64}$ and its fraction of the student's top-64 mass, $r = p_\theta(I_{64} \mid h)/p_\theta(S_{64} \mid h)$.

Mean intersection mass exceeds 99.9% at each checkpoint, and the mean retained fraction r exceeds 99.97%. Therefore, little of the student's top-64 mass is excluded by disagreement with the assigned teacher's candidate set. Among all 384 prefixes, the minimum intersection mass is 96.55% and every intersection is nonempty. Top-64 therefore retains nearly all student probability throughout the sampled Qwen comparisons.

G.2 Coverage during top-16 training

The offline top-64 intersection KL loss audit is complemented by coverage measurements throughout 500 training steps with the top-16 intersection KL loss and domain-response averaging. In 8,000 logged slices of four responses, mean intersection mass is $p_\theta(I_{16}) = 99.62\%$, mean retained fraction is $r = p_\theta(I_{16})/p_\theta(S_{16}) = 99.88\%$, and mean intersection size is $|I_{16}| = 15.05$. Each slice averages valid tokens within responses and then responses equally. Candidate sets are fixed at rollout, and probabilities are measured during training.

The mean empty-intersection fraction is 0.0027%. The lowest slice-mean mass is 86.67% and the largest slice-mean empty-intersection fraction is 1.85%. These measurements show that high average coverage during training coexists with occasional slices of lower coverage.

G.3 Domain-level effects of loss averaging

An average probe-loss reduction can conceal an increase on an individual domain. We disaggregate Figure 3(c) using the same six batches and twelve update branches per batch (Appendix B.1), with one held-out response per domain. At step 250, saved-state Adam under domain-response, domain-token, and global-token averaging increases instruction-following KL by $0.362/0.293/0.272 \times 10^{-3}$, respectively; the corresponding four-domain changes are $0.041/-0.002/-0.027 \times 10^{-3}$. Domain-token and global-token averaging therefore improve the mean probe loss while moving instruction following away from its teacher (Figure 10).

H SmolLM3 Diagnostics and Capability Evaluation

H.1 Models and response coverage

SmolLM3 provides a complementary setting in which candidate truncation retains less student probability, allowing us to examine gradient approximation beyond the highly concentrated Qwen predictions. We use the SmolLM3-3B student, three RL teachers for mathematics, code, and instruction following, and exported students. The top-64 students use domain-response, domain-token, or global-token averaging with Adam at learning rate 2.5×10^{-7}; the PG student uses domain-response averaging with SGD at learning rate 0.0025.

Diagnostic prompts use Open-MOPD data. We shuffle unique raw messages within each domain and use disjoint coverage, normalization, and probe prompt splits. Generation preserves dataset messages, disables thinking in the chat template, and uses temperature 1, top-$p = 1$, a 2,048-token prompt cap, and a 4,096-token response cap. The initialization and the top-64/domain-response student each generate a natural response bank of 384 responses, with equal numbers from each domain. Sampling up to eight uniformly spaced positions per response gives 3,053 prefixes for the initialization and 3,058 for the trained student. Additional students score the trained student's 3,058 prefixes.

Coverage statistics weight domains, responses within domains, and audited positions within responses equally. Mean-mass intervals resample responses independently within each domain 2,000 times using seeds 1042/1043/1044. The initialization and domain-response student use their own natural response banks; the remaining students use the latter bank. On the domain-response student's natural bank, the minimum top-64 intersection mass is 1.7×10^{-8}. The domain-token and

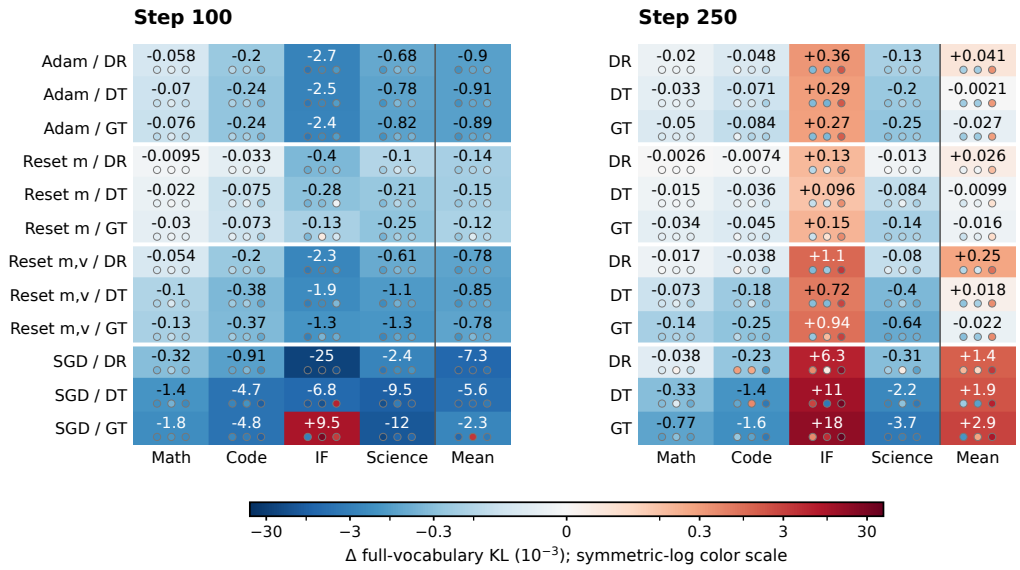

Figure 10: **Mean probe-loss reductions can coexist with domain-level increases.** Full-vocabulary KL after minus before BF16 writeback, in units of 10^{-3}. Cells show three-batch means; dots show batches 1042–1044 from left to right. The shared symmetric-log color scale is linear within ± 0.1. SGD is matched to each rule's saved-state Adam step norm.

global-token students each have one empty intersection when scoring the common bank. These low-coverage cases motivate the targeted gradient comparisons below. Figure labels use Initial and DR500 for the initialization and the domain-response student, respectively.

H.2 Gradient comparisons on typical and difficult prefixes

Within each domain, we select representative prefixes, prefixes above the domain's 75th entropy percentile, and prefixes whose top-64 intersection mass is below 0.99. The groups can overlap and use up to 32 distinct prompts per domain. The low-coverage code group has 21 eligible prompts, giving the low-coverage group 85 distinct prefixes from all domains combined. Each group is divided into three batches. Smaller candidate budgets, $k = 4, 8$, are selected using coverage alone.

The BF16 coverage audit sorts a top-128 list and takes its first k entries. Gradient diagnostics recompute top-k in FP32 at the exported BF16 parameter values. Student forward and backward passes use FP32; cached teacher targets use BF16 forward passes followed by FP32 log-softmax. One of the 85 frozen low-coverage prefixes crosses the 0.99 threshold during the FP32 diagnostic and remains in its original group. In FP32, the mean top-64 intersection masses are 99.05%, 95.68%, and 93.12% for the representative, high-entropy, and low-coverage groups, respectively.

Full-vocabulary KL, PG, and top-k intersection KL losses follow Section 2; empty intersections give zero loss. We retain zero- and small-norm cases and report relative gradient error alongside cosine to capture changes in both direction and magnitude. For intersection gradient g_{I_k} and full-vocabulary gradient g_{full}, relative error is $\|g_{I_k} - g_{\text{full}}\|/\|g_{\text{full}}\|$.

H.3 Supplementary capability observations

PG training with domain-response averaging and SGD raises MATH-500 accuracy from 50.60% at initialization to 53.80%. On AIME 2025, the initialization, the top-64/domain-token Adam student, and the PG/domain-response SGD student each solve four of 30 questions with eight samples per question. Only one solved question is common to all three models.