THE DESIGN SPACE OF ON-POLICY DISTILLATION: A UNIFIED FRAMEWORK AND EMPIRICAL ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) transfers reasoning capabilities from a stronger teacher to a student by training on student-generated responses. Yet it remains unclear which parts of OPD are essential for effective learning and which can be simplified. We introduce a unified training objective that organizes representative OPD methods along four design axes: rollout source, trajectory weighting, token-position weighting, and local supervision loss. Guided by this framework, we conduct a systematic empirical study of over 30 OPD configurations on mathematical reasoning tasks. We find that teacher feedback remains effective on student-generated trajectories. In our setting, training only on failed trajectories matches the accuracy of using all trajectories. At the token level, a random 100-token window remains competitive with full-token supervision, while SelecTKD, which emphasizes teacher–student agreement, achieves the highest accuracy among the tested token-weighting methods. Preserving teacher probability information remains important: adaptive vocabulary allocation retains more teacher probability mass at the same target average budget, whereas replacing probabilities with ranks or top-K membership lowers accuracy. Skew KL improves accuracy using a mixture of teacher and student probabilities, while hard clipping limits extreme learning signals. Finally, we combine these choices into a single OPD recipe and evaluate it on three larger teacher–student model pairs, improving accuracy by up to 1.29 points and reducing total training time by 15.3%. Together, our framework and findings clarify where OPD can be simplified and where richer teacher feedback remains important.

1 INTRODUCTION

On-policy distillation (OPD) transfers reasoning capabilities from a stronger teacher to a student by training on student-generated responses (Gu et al., 2024; Agarwal et al., 2024). The student determines which reasoning paths to take, while the teacher provides token-level guidance along those paths. This combination of student generation and teacher guidance makes OPD a practical approach to reasoning-model post-training (Yang et al., 2025; Lu, 2025). OPD now spans methods that change trajectory collection, select responses or token positions, and modify the teacher's learning signal (Ko et al., 2024; Xu et al., 2026; Li et al., 2026c; Fu et al., 2026). However, comparing complete recipes makes it difficult to isolate which design choices drive their gains. We therefore ask: *what do these methods change in a common student update, and which choices matter for learning?*

To answer this question, we introduce a unified training objective with four components: *rollout source*, *trajectory weighting*, *token-position weighting*, and *local supervision loss*. By expressing existing methods in this form, we can compare how they collect trajectories and use teacher feedback, and organize our experiments around the design choices they change.

We conduct a systematic empirical study of over 30 OPD configurations on mathematical reasoning tasks. Teacher feedback remains effective along student-generated reasoning paths: student-only rollouts achieve 32.82% average accuracy, compared with 21.80% for teacher-only generation and 31.20% for the strongest teacher-assisted variant. This feedback also remains useful on failed attempts, with failure-only training matching all-trajectory supervision in our setting. Within a response, a random 100-token window remains competitive with full-token supervision (33.62% vs. 32.82%), while SelecTKD (Huang et al., 2025) achieves the highest accuracy among the tested token-weighting

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.

methods. Preserving teacher probability information, however, remains important: replacing it with ranks or membership lowers accuracy. Adaptive vocabulary allocation retains more teacher probability mass and reaches 35.49% accuracy versus 33.20% for fixed allocation at the same target average budget. Skew KL (Ko et al., 2024) and hard clipping also improve accuracy, highlighting the importance of the local loss and update strength.

Finally, we combine these choices into a single OPD recipe and evaluate it on three larger teacher–student model pairs, improving accuracy by up to 1.29 points and reducing total training time by 15.3%. Together, these findings clarify where OPD can be simplified and where richer teacher feedback remains important.

Our contributions are:

- **A unified formulation of OPD.** We introduce a common training objective that organizes representative OPD methods along four design dimensions, making their shared structure and differences explicit.
- **A systematic empirical study of OPD design choices.** We evaluate over 30 configurations on mathematical reasoning tasks, examining how design choices affect accuracy and efficiency and identifying where simpler alternatives remain competitive.
- **Empirical guidance for efficient and simple OPD.** We combine choices across all four components and evaluate the resulting configuration on three larger teacher–student model pairs, improving accuracy by up to 1.29 points and reducing total training time by 15.3%.

2 Related Work

On-policy distillation and unified perspectives. Knowledge distillation trains students to match teacher predictions (Hinton et al., 2015) or imitate teacher-generated text (Kim & Rush, 2016). DAgger and ImitKD collect teacher feedback on states reached by the learner (Ross et al., 2011; Lin et al., 2020). MiniLLM and GKD similarly supervise prefixes in student-generated responses (Gu et al., 2024; Agarwal et al., 2024). DeepSeek-R1 trains smaller models on teacher-generated solutions (DeepSeek-AI, 2025), while KDRL and Nemotron-Cascade 2 combine distillation with reinforcement learning (Xu et al., 2025; Yang et al., 2026b). Existing frameworks distinguish training contexts and labels (Li et al., 2026a) or address student exploration and teacher reliability (Hou et al., 2026). We compare OPD methods through four design components.

Selective supervision. Selective supervision allocates feedback across examples, responses, and token positions. Methods select pre-training data offline (Gu et al., 2025), filter trajectories (Li et al., 2026c; Akhondzadeh et al., 2026), or select and reweight tokens (Xu et al., 2026; Wang et al., 2026b; Huang et al., 2025). Other approaches add response rankings (Kim & Baek, 2026), shorten rollouts (Zhou et al., 2026), or account for gradient-estimation reliability in sparse supervision (Sheng et al., 2026). We compare representative methods with uniform supervision and simple controls to assess how coverage and placement affect learning.

Teacher feedback and distillation objectives. Local supervision depends on the vocabulary entries, teacher information, and loss used. OPD analyses examine teacher–student compatibility and shared high-probability tokens (Li et al., 2026b), and motivate teacher top-K support matching (Fu et al., 2026). Feedback can also use standardized logits (Sun et al., 2024) or vocabulary rankings (Bassam et al., 2025). Objectives include f-divergence formulations (Wen et al., 2023), KL-constrained reinforcement learning (Yang et al., 2026a), and skew divergences (Ko et al., 2024; 2025), with additional entropy or credit regulation (Jin et al., 2026; Wang et al., 2026a; Zhao et al., 2026). We examine which teacher information and local losses most effectively support learning.

3 A Unified Formulation of On-Policy Distillation

3.1 Preliminaries

We consider knowledge distillation between a trainable autoregressive student π_θ and a fixed teacher π_T, defined over a shared vocabulary $\mathcal{V}$. Given a prompt x, a response $y = (y_1, \ldots, y_T)$ is generated sequentially, where T denotes the response length. At position t, the generation context is the prefix $s_t = (x, y_{<t})$.

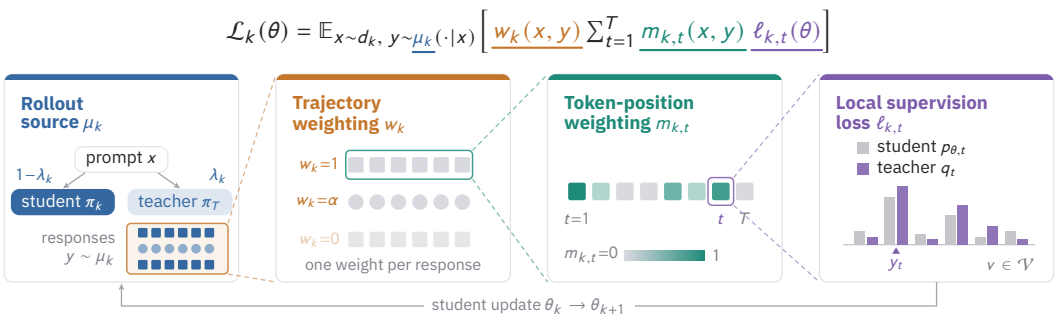

Figure 1: Our unified OPD objective has four components: rollout source, trajectory weighting, token-position weighting, and local supervision loss.

On-policy distillation (OPD) trains the student using teacher feedback on its own generated responses (Agarwal et al., 2024; Lu, 2025). At iteration k, let $\pi_k = \pi_{\theta_k}$ denote the current student policy. We sample a prompt $x \sim d_k$, where d_k is the prompt distribution, and collect a response $y \sim \pi_k(\cdot \mid x)$. At each resulting prefix s_t, the student and teacher define the next-token distributions

$$p_{\theta,t}(v) = \pi_\theta(v \mid s_t), \qquad q_t(v) = \pi_T(v \mid s_t), \qquad v \in \mathcal{V}.$$

Both distributions are therefore evaluated on the same student-generated context.

A common local objective for OPD minimizes the reverse KL divergence between the student and teacher:

$$\mathcal{L}_k^{\mathrm{OPD}}(\theta) = \mathop{\mathbb{E}}_{\substack{x\sim d_k \\ y\sim\pi_k(\cdot|x)}} \left[\sum_{t=1}^{T} D_{\mathrm{KL}}(p_{\theta,t} \,\|\, q_t) \right], \tag{1}$$

where $D_{\mathrm{KL}}(p\|q) = \sum_{v\in\mathcal{V}} p(v) \log \frac{p(v)}{q(v)}$. The collected responses and teacher predictions are treated as fixed data during each update. Gradients are taken only through the student distribution $p_{\theta,t}$.

3.2 The Design Space of On-Policy Distillation

Building on the local formulation in Eq. (1), we revisit how existing OPD methods collect responses and use teacher feedback. Their design choices concern the rollout distribution, the allocation of supervision across responses and token positions, and the construction of the local loss. We express these choices in a unified objective:

$$\mathcal{L}_k^{\mu,w,m,\ell}(\theta) = \mathop{\mathbb{E}}_{\substack{x\sim d_k \\ y\sim\mu_k(\cdot|x)}} \left[w_k(x,y) \sum_{t=1}^{T} m_{k,t}(x,y) \ell_{k,t}(\theta) \right]. \tag{2}$$

The rollout distribution and weights are held fixed during each update, following the optimization convention in Section 3.1. The objective reduces to Eq. (1) when $\mu_k = \pi_k$, $w_k = m_{k,t} = 1$, and $\ell_{k,t} = D_{\mathrm{KL}}(p_{\theta,t}\|q_t)$. This formulation organizes OPD methods along four design dimensions, each corresponding to a distinct choice in the training procedure.

Rollout source μ. The distribution $\mu_k(\cdot \mid x)$ specifies how responses are generated for a given prompt and hence which prefixes are available for supervision.

Trajectory weighting w. The nonnegative weight $w_k(x, y)$ scales the aggregate loss of a response, controlling its contribution relative to other collected trajectories.

Token-position weighting m. The nonnegative weight $m_{k,t}(x, y)$ controls the contribution of position t to the response loss, allocating supervision within each trajectory.

Local supervision loss ℓ. The loss $\ell_{k,t}(\theta)$ specifies how teacher feedback supervises the student at a given prefix. It determines what teacher information is retained and how that information enters the training objective.

As a concrete instantiation, sampled-token OPD uses the teacher–student log-ratio as a detached token credit (Lu, 2025; Fu et al., 2026):

$$a_{k,t} = \log \frac{q_t(y_t)}{\pi_k(y_t \mid s_t)}, \qquad \ell_{k,t}^{\text{samp}}(\theta) = -\,\text{sg}(a_{k,t}) \log p_{\theta,t}(y_t),$$

where sg denotes stop-gradient. The credit is computed using the rollout policy π_k and held fixed during optimization. Positive credit encourages the sampled token, whereas negative credit discourages it. Combined with $\mu_k = \pi_k$ and $w_k = m_{k,t} = 1$, this loss gives a sampled-token instantiation of Eq. (2).

Figure 1 illustrates how these four choices determine the student update, while Table 1 organizes representative methods within this design space. Appendices A and C provide detailed instantiations and supporting derivations. Our experiments use this decomposition to uncover how these choices shape learning and to identify principles for simpler, more efficient OPD.

Table 1: **Representative instantiations of the unified OPD formulation.** Entries are grouped by their primary design dimension in Eq. (2). Appendix A details the evaluated variants.

Design	Defining modification
Rollout source μ	
Response mixing (Agarwal et al., 2024) (Li et al., 2026a)	$\mu_k = (1-\lambda_k)\pi_k + \lambda_k \nu_T$: mix student rollouts with cached teacher responses ν_T, using a fixed or annealed teacher fraction λ_k.
AggreVaTe (Li et al., 2026a)	Let the teacher complete a student-generated prefix.
Trajectory weighting w	
FiRe (Li et al., 2026c)	$w_k = \mathbf{1}\{\bar{\ell}_T \geq \tau_k\}$: threshold the mean teacher log-probability $\bar{\ell}_T$, with entropy-based token weighting.
RG-OPD (Akhondzadeh et al., 2026)	$w_k = \mathbf{1}\{R\Delta_k > 0\}$, $\Delta_k = \sum_t a_{k,t}$: match the log-likelihood-ratio sign to the outcome R ($+1$ correct, -1 incorrect).
Token-position weighting m	
TIP (Xu et al., 2026)	$m_{k,t} = \mathbf{1}\{I_t \geq \tau_k\}$: select positions with high importance I_t, combining student uncertainty and teacher–student disagreement.
SelecTKD (Huang et al., 2025)	$m_{k,t} = 1$ when the student's top token is among the teacher's top five, and $m_{k,t} = 0.01$ otherwise.
Local supervision loss ℓ	
Distributional loss (Agarwal et al., 2024; Ko et al., 2024)	$\ell_{k,t} = \mathcal{D}(p_{\theta,t}, q_t)$: reverse/forward KL, Skew KL, or JSD.
Vocabulary support (Fu et al., 2026; Li et al., 2026b)	Restrict the loss to $S_t \subseteq \mathcal{V}$: teacher top-$K$, teacher–student overlap or union, or adaptive support.
Teacher information (Bassam et al., 2025) (Sun et al., 2024)	Retain teacher probabilities, token ranks, top-K membership, or standardized log-probabilities.
Credit transformation (Wang et al., 2026a) (Zhao et al., 2026)	$\ell_{k,t} = -\,\text{sg}(c_{k,t}) \log p_{\theta,t}(y_t)$: use clipped, soft-log, or probability-power credit $c_{k,t}$.

4 Experimental Setup

Models and training. We train Qwen3-1.7B-Base with Qwen3-4B-Instruct-2507 as the teacher (Yang et al., 2025) on 14,476 problems from Nemotron-Cascade-RL-Math (Wang et al., 2025). Vanilla uses student-generated rollouts ($\mu_k = \pi_k$), uniform trajectory and token weights ($w_k = m_{k,t} = 1$), and the sampled-token loss $\ell_{k,t}^{\text{samp}}$ defined in Section 3.2. We sample one rollout per prompt at temperature 1 and top-$p = 1$, with a maximum response length of 16,384 tokens. Training uses

Table 2: **Main results across the four OPD components. Bold** marks the best variants per column within each block. Δ is relative to Vanilla (pp).

	Benchmark accuracy (%) ↑					Overall	
Method	**AIME24**	**AIME25**	**AMC23**	**Minerva**	**MATH500**	**Average**	**Δ (pp)**
Vanilla	5.7	7.6	42.8	15.3	66.6	**32.82**	—
(a) Rollout source μ_k							§5.1
Fixed mixing	3.8	6.2	34.7	14.5	64.1	29.55	−3.27
Teacher-only	0.0	1.2	25.6	12.0	48.9	21.80	−11.02
DAgger	**5.4**	6.7	37.5	**15.7**	**65.5**	31.19	−1.63
AggreVaTe	5.0	**7.1**	**40.9**	13.6	64.2	**31.20**	**−1.62**
(b) Trajectory weighting w_k							§5.2
Correct-only	5.0	5.0	39.4	15.1	62.6	30.51	−2.31
Failure-only	**5.6**	**7.6**	43.2	**15.4**	66.1	**32.82**	**0.00**
FiRe	5.0	7.5	42.4	**15.4**	**67.2**	32.80	−0.02
RG-OPD	5.5	7.4	**43.3**	15.1	66.4	32.80	−0.02
(c) Token-position weighting $m_{k,t}$							§5.3
Prefix	3.8	6.7	**44.1**	12.7	61.5	30.87	−1.95
Suffix	0.4	2.1	23.4	11.5	49.5	21.44	−11.38
Random window	6.2	**9.2**	42.5	**17.2**	67.1	33.62	+0.80
Random scatter	**7.1**	7.5	39.1	15.9	65.8	32.02	−0.80
TIP	5.6	7.7	43.0	15.4	66.5	32.87	+0.05
SelecTKD	6.7	5.4	43.1	16.9	**70.7**	**34.19**	**+1.37**
TA-OPD	6.7	5.4	40.3	16.4	62.9	31.41	−1.41
(d) Local supervision loss $\ell_{k,t}$							§5.4
Vocabulary support and budget allocation							
Teacher top-K	1.1	2.0	28.2	12.7	49.6	23.03	−9.79
Intersection	7.1	7.1	43.1	15.4	68.2	33.46	+0.64
Union (FKL)	6.2	7.9	41.6	15.7	67.1	32.85	+0.03
Union (RKL)	4.2	2.9	31.2	11.6	46.9	23.33	−9.49
Student-tail	7.1	5.8	41.9	17.6	69.7	33.93	+1.11
Fixed budget	5.4	7.1	43.8	15.2	67.6	33.20	+0.38
Quantile budgeting	**7.9**	**10.0**	**45.0**	**18.1**	**69.9**	**35.49**	**+2.67**
Teacher information							
Standardized values	3.2	**6.0**	**38.0**	**14.1**	**63.8**	**30.13**	**−2.69**
Rank only	1.0	1.5	21.7	11.2	45.8	20.01	−12.81
Membership only	**3.8**	3.3	31.6	10.9	56.3	25.57	−7.25
Distillation loss							
Skew KL	**7.9**	8.3	43.1	**17.0**	**69.9**	**34.54**	**+1.72**
Forward KL	2.9	7.9	44.4	16.5	68.6	33.73	+0.91
JSD	7.0	8.2	43.0	15.4	69.3	33.83	+1.01
EOPD (adapted)	6.0	8.3	42.6	13.9	64.9	32.13	−0.69
Squared log-ratio	5.5	7.0	42.1	14.9	67.3	32.63	−0.19
OPD + stop-prob. distillation	6.2	**8.8**	**45.0**	16.3	69.0	34.45	+1.63
Update strength							
Hard clipping	**8.3**	8.3	41.9	15.9	**69.6**	**33.93**	**+1.11**
Positive-side clipping	3.0	4.0	34.6	13.5	57.6	27.29	−5.53
Soft-log compression	6.4	7.4	**43.9**	**16.0**	69.0	33.92	+1.10
Power credit	6.2	**9.6**	43.1	15.0	69.3	33.83	+1.01
Tanh transform	4.1	3.9	33.7	13.3	58.3	27.33	−5.49

AdamW (Loshchilov & Hutter, 2019) with a learning rate of 10^{-6} and batch size 128. Detailed settings are provided in Appendix B.

Evaluation. We evaluate on AIME24, AIME25, AMC23, Minerva (Lewkowycz et al., 2022), and MATH500 (Hendrycks et al., 2021; Lightman et al., 2024), using temperature 0.7, nucleus sampling (Holtzman et al., 2020) with top-$p = 0.95$, and a maximum response length of 32,768 tokens. Our main evaluation setting averages results over three training seeds and uses avg@32 on AIME24, AIME25, and AMC23 and avg@3 on Minerva and MATH500. The overall score is the equally weighted average of pooled AIME, AMC23, Minerva, and MATH500 accuracies.

5 What Matters in OPD?

In this section, we use our unified framework to examine where training trajectories should come from (Section 5.1), which trajectories should contribute to learning (Section 5.2), how to weight supervision across tokens (Section 5.3), and which teacher information and loss function to use at each supervised position (Section 5.4). Table 2 summarizes the benchmark results across these four components.

5.1 Rollout source: how should we generate training trajectories?

Should training trajectories come from the student, the teacher, or both? Teacher-generated trajectories may offer better demonstrations and prefixes on which the teacher can provide more confident feedback. Student-generated trajectories cover the states the student actually visits, including its mistakes, helping reduce the mismatch between training and deployment. We vary the rollout source μ_k using student-only and teacher-only generation, along with three teacher-assisted controls. Across responses, we use a fixed teacher fraction following GKD (Agarwal et al., 2024) or a decreasing fraction in our DAgger control (Ross et al., 2011; Li et al., 2026a). Within a response, our AggreVaTe control lets the teacher complete a student-generated prefix (Ross & Bagnell, 2014; Li et al., 2026a). These choices test whether teacher generation helps and whether its timing and placement matter.

Table 2 reports the benchmark accuracies of the rollout-source methods. Student-only Vanilla achieves the highest average accuracy of 32.82, while teacher-only sampling reaches 21.80. Fixed mixing improves on teacher-only sampling to 29.55. DAgger and AggreVaTe come closest to Vanilla, reaching 31.19 and 31.20, respectively. In these configurations, teacher-assisted methods substantially outperform teacher-only training, with DAgger and AggreVaTe achieving accuracy close to Vanilla.

As shown in Figure 2(b,c), teacher-assisted methods produce shorter responses later in training and retain higher policy entropy. In particular, higher entropy alone does not identify a more effective rollout source. This pattern is consistent with the role of μ_k in Eq. (2): student-generated rollouts place teacher supervision directly on contexts encountered by the evolving student. Teacher feedback can thus support learning without requiring the teacher to generate the training trajectories.

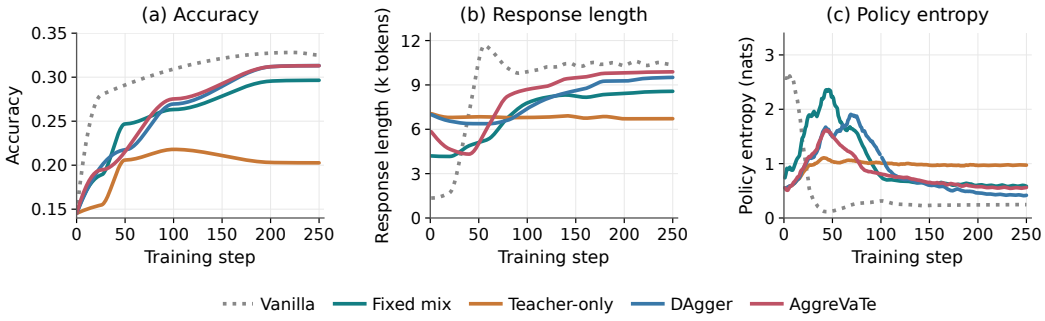

Figure 2: Training dynamics of methods with different rollout sources.

Finding 1

Teacher-assisted methods substantially outperform teacher-only training but remain comparable to student-only rollouts. We therefore use student-only rollouts as the default for OPD.

5.2 Trajectory selection: which trajectories should we train on?

Student-generated trajectories vary in quality, making it natural to concentrate training on those that appear most useful. Trajectory selection implements this choice through the weight w_k in Eq. (2), which controls each trajectory's contribution to the update. We compare uniform weighting (Vanilla, $w_k = 1$) with correct-only and failure-only training to test whether final-answer correctness

identifies useful supervision. We then consider teacher feedback beyond the final outcome through two adapted methods. FiRe (Li et al., 2026c) filters trajectories by mean teacher log-probability and additionally applies entropy-based token weights, while RG-OPD (Akhondzadeh et al., 2026) combines correctness with the teacher–student likelihood ratio. Together, these controls test whether outcome-based or teacher-guided selection improves upon retaining every trajectory.

Table 2(b) shows that failure-only matches Vanilla at 32.82, while correct-only reaches 30.51. Failure-only retains 92.3% of trajectories, whereas correct-only retains just 2.8%. Selecting only correct trajectories thus discards most opportunities for local teacher guidance and coincides with occasional gradient spikes (Appendix B.2). An incorrect final answer does not necessarily make the local losses $\ell_{k,t}$ along a trajectory uninformative. Failure-only preserves these learning opportunities while potentially reducing distillation compute.

Teacher-guided selection also retains comparable accuracy with fewer supervised trajectories: FiRe and RG-OPD both reach 32.80, while FiRe reaches 33.26 at its best evaluated checkpoint. These results show that selection can reduce supervision coverage while preserving useful teacher guidance.

Finding 2

Trajectory selection can achieve accuracy comparable to uniform trajectory weighting while using fewer responses for distillation. We therefore use failure-only trajectory selection as the default in our setting.

5.3 Token weighting: how should we allocate supervision across tokens?

At the token level, $m_{k,t}$ controls how much supervision each position receives. Hard selection sets unselected weights to zero; soft weighting adjusts their relative contributions. Vanilla uses $m_{k,t} = 1$ at every position. We first adapt teacher-guided methods: TIP (Xu et al., 2026) and TA-OPD (Wang et al., 2026b) apply hard selection based on token importance and teachability, respectively. SelecTKD (Huang et al., 2025) instead uses soft weighting, downweighting positions where the student's preferred token falls outside the teacher's top-ranked set. We then consider an ESR-inspired prefix baseline (Zhou et al., 2026), which supervises the first 100 tokens while retaining full generation. Starting from this baseline, we design three variants with the same 100-token budget: a fixed suffix, a random contiguous window, and randomly scattered positions. These positional controls compare early versus late supervision, fixed versus random placement, and contiguous versus scattered coverage.

As shown in Figure 3, the token-weighting methods exhibit different accuracy, response-length, and policy-entropy dynamics. Table 2(c) shows that a random window reaches 33.62, ahead of scattered positions at 32.02, a fixed prefix at 30.87, and a suffix at 21.44. Suffix supervision leaves earlier reasoning without direct feedback; prefix supervision misses later reasoning and stopping decisions. Random windows cover different stages across training, showing that a small supervision budget remains useful when its placement varies. SelecTKD reaches 34.19, the highest reported accuracy among the evaluated token-weighting methods. Its soft weighting adjusts emphasis without removing feedback entirely. Supervision-coverage results and short-generation controls appear in Appendix B.3.

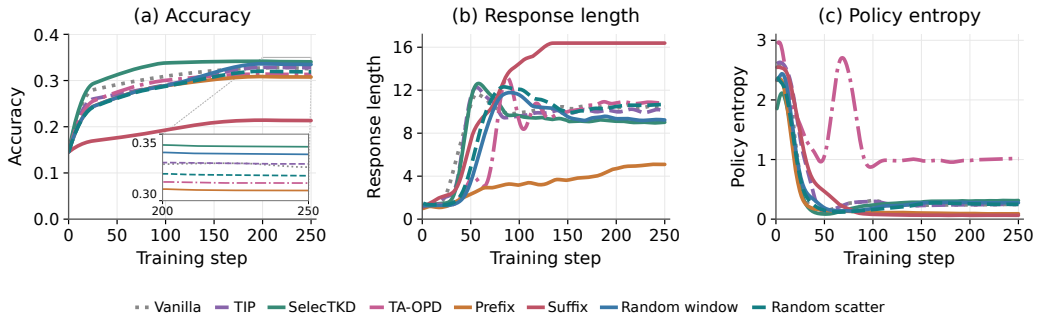

Figure 3: Training dynamics of token-weighting methods.

Finding 3

Token selection and reweighting can retain accuracy close to uniform token supervision, but where supervision is applied matters. We use SelecTKD as the default token-weighting method.

5.4 Local supervision loss: how should we use teacher feedback?

The local loss $\ell_{k,t}(\theta)$ determines how teacher feedback supervises each selected position. We examine three choices in its construction: which vocabulary entries enter the loss, what teacher information to retain, and how to convert that information into a loss and control update strength.

Allocating the vocabulary budget. Token weighting allocates supervision across sequence positions; vocabulary budgeting allocates vocabulary entries within each supervised position. Vanilla computes credit from the sampled token. We examine whether constructing the signal over a selected vocabulary support improves learning, then ask which candidates to retain and how many to allocate at each position. For support selection, we adapt teacher top-K support from local support matching (LSM) (Fu et al., 2026) and teacher–student intersection support (Li et al., 2026b), then add union support as a complementary control. For budget allocation, student-tail support sets the support size using cumulative student probability mass, while quantile budgeting allocates a shared vocabulary budget across positions. A fixed-budget baseline tests whether the same number of candidates should be retained at every position.

The vocabulary-support comparison in Table 2(d) shows that intersection outperforms teacher top-K, while union is sensitive to the KL direction. Adaptive budgets perform better: student-tail reaches 33.93, and quantile budgeting leads at 35.49, compared with 33.20 for fixed allocation and 32.82 for Vanilla. Figure 4 compares accuracy, retained teacher mass, and support size. Quantile budgeting uses a shared probability threshold, allocating more entries where more tokens are favored by either model. At the same target average budget, it retains 97.7% of teacher probability mass versus 54.7% for fixed allocation. Results for additional vocabulary-support methods appear in Appendix B.4.

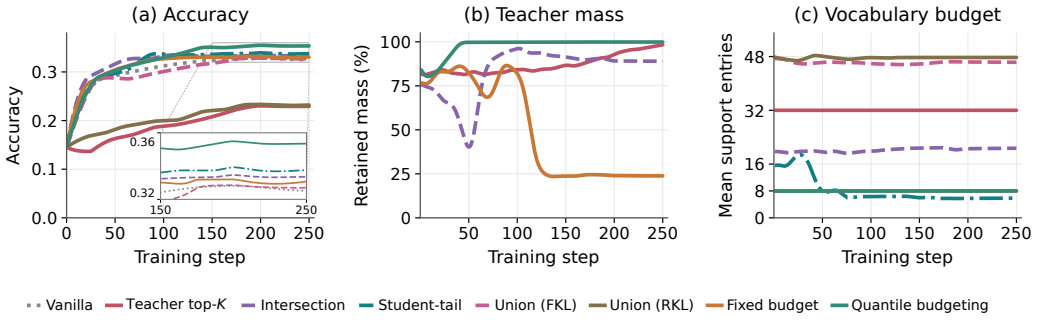

Figure 4: Training dynamics of vocabulary-support methods.

Retaining teacher information. To examine whether teacher feedback can be simplified within the selected vocabulary support, we compare z-scored log-probabilities, the teacher's ranking of vocabulary tokens, and its top-K token set as alternatives to the original probability values. The teacher-information comparison in Table 2(d) shows that all three simplifications reduce accuracy: standardized values reach 30.13, rank-only 20.01, and membership-only 25.57, versus Vanilla's 32.82. Rank and membership supervision omit the probability ratios that determine update strength. These results favor retaining numerical teacher probabilities within the selected support.

Shaping the learning signal. With teacher probabilities retained, we examine the form of $\ell_{k,t}$ and the strength of the credit used in its sampled-token variants. For the loss form, we compare Vanilla's sampled reverse-KL loss with forward KL, Jensen–Shannon divergence (Agarwal et al., 2024), and Skew KL (Ko et al., 2024). We also test an EOPD-inspired gated forward-KL term (Jin et al., 2026) and a squared log-ratio loss. Stop-probability distillation adds an auxiliary loss to OPD that matches the student's stopping probability to the teacher's at each position. To control update strength, we compare Vanilla's unmodified log-ratio with hard clipping and soft-log compression (Wang et al.,

2026a), power credit (Zhao et al., 2026), and a tanh transform. Positive-side clipping tests whether both tails of the credit should be bounded.

The loss comparison in Table 2(d) shows that Skew KL reaches 34.54 and stop-probability distillation reaches 34.45. By mixing teacher and student probabilities, Skew KL softens the penalty on tokens favored only by the student; it changes the local loss rather than the rollout distribution. The adapted EOPD and squared log-ratio losses remain below Vanilla in the reported comparisons. Among the credit transformations, hard clipping bounds extreme log-ratios while preserving moderate signals and reaches 33.93, close to soft-log compression and power credit. Additional ablations are given in Appendix B.4.

Finding 4

Preserving teacher probability information matters for accuracy, while adaptive vocabulary allocation and moderated local updates can improve it. We use quantile budgeting and Skew KL as defaults, with hard clipping to limit extreme learning signals.

6 Scaling the Combined Configuration

We next test whether the choices motivated by Section 5 remain effective when combined on larger teacher–student pairs, measuring both accuracy and full-run training time. The combined configuration uses student-generated rollouts, failure-only trajectory selection, SelecTKD token weighting, and quantile vocabulary budgeting with Skew KL and hard clipping. We compare it with Vanilla on three Qwen3 teacher–student pairs: 8B $\rightarrow$ 4B, 32B $\rightarrow$ 4B, and 32B $\rightarrow$ 8B, using Qwen3-Base students. We train each configuration for 250 steps. Evaluation uses avg@32 on AIME24, AIME25, and AMC23 and avg@3 on Minerva and MATH500. The reported accuracy is the equally weighted average over pooled AIME, AMC23, Minerva, and MATH500.

Table 3: Average benchmark accuracy and full-run training time across larger teacher–student pairs.

	Accuracy (%)			Training time (h)		
Teacher $\rightarrow$ student	**Vanilla**	**Ours**	Δ	**Vanilla**	**Ours**	**Speedup**
8B $\rightarrow$ 4B	42.40	42.84	+0.44	12.38	11.07	1.12$\times$
32B $\rightarrow$ 4B	40.81	40.99	+0.18	9.33	7.44	1.25$\times$
32B $\rightarrow$ 8B	43.18	44.47	+1.29	12.39	10.37	1.19$\times$

Table 3 compares accuracy and training efficiency across the three pairs. The reported accuracy gains are 0.44, 0.18, and 1.29 percentage points for the three pairs, respectively, with 1.12–$1.25\times$ full-run training speedups. Aggregated across the three teacher–student pairs, total training time decreases from 34.10 to 28.88 hours, corresponding to a 15.3% reduction. Most of the time savings occur during rollout collection, with smaller savings in student updates.

7 Conclusion

We introduced a unified framework that organizes OPD methods along four design dimensions and evaluated over 30 configurations on mathematical reasoning tasks. Our results show that teacher feedback remains effective on student-generated responses, including failed attempts, and that selective supervision across responses and tokens can retain competitive accuracy. Teacher probability values remain important: replacing them with ranks or top-K membership reduces accuracy. Adaptive vocabulary allocation and adjustments to the local loss and update strength also improve accuracy in our experiments. Guided by these findings, we combine choices across all four dimensions into a single OPD recipe. Across three larger teacher–student model pairs, this configuration improves accuracy by up to 1.29 points and reduces total training time by 15.3%. These results provide practical guidance for making OPD simpler and more efficient.

AI USE STATEMENT

The authors developed the research ideas and system design and led the manuscript writing. Generative AI tools assisted with implementation and manuscript polishing. The authors are responsible for the final content.

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 (ICLR)*, 2024. URL https://arxiv.org/abs/2306.13649.

Mohammad Sadegh Akhondzadeh, Vijay Lingam, Atula Tejaswi, Chanakya Ekbote, Sujay Sanghavi, and Aleksandar Bojchevski. Reward-gated on-policy distillation. *arXiv preprint arXiv:2607.04037*, 2026. URL https://arxiv.org/abs/2607.04037.

Ejafa Bassam, Dawei Zhu, and Kaigui Bian. PLD: A choice-theoretic list-wise knowledge distillation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 136090–136112, 2025. doi: 10.52202/085713-4542. URL https://arxiv.org/abs/2506.12542.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

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. URL https://arxiv.org/abs/2603.25562.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, pp. 32694–32717, 2024. URL https://arxiv.org/abs/2306.08543v4.

Yuxian Gu, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. MiniPLM: Knowledge distillation for pre-training language models. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=tJHDw8XfeC.

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 *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://arxiv.org/abs/2103.03874.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, Chengquan Zhang, Zhuotao Tian, Han Hu, Yi Yang, Fei Wu, and Hehe Fan. Uni-OPD: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*, 2026. URL https://arxiv.org/abs/2605.03677.

Haiduo Huang, Jiangcheng Song, Yadong Zhang, and Pengju Ren. SelecTKD: Selective token-weighted knowledge distillation for LLMs. *arXiv preprint arXiv:2510.24021*, 2025. URL https://arxiv.org/abs/2510.24021.

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. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2603.07079.

Minsang Kim and Seung Jun Baek. Explain in your own words: Improving reasoning via token-selective dual knowledge distillation. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=zph7e5JaXc.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2402.03898.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31044–31062, 2025. URL https://proceedings.mlr.press/v267/ko25a.html.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay 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 *Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857, 2022. doi: 10.52202/068431-0278. URL https://arxiv.org/abs/2206.14858.

Changhao Li, Rushi Qiang, Jiawei Huang, Chenxiao Gao, Chao Zhang, Niao He, and Bo Dai. Revisiting DAgger in the era of LLM-agents. *arXiv preprint arXiv:2605.12913*, 2026a. URL https://arxiv.org/abs/2605.12913.

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. *arXiv preprint arXiv:2604.13016*, 2026b. URL https://arxiv.org/abs/2604.13016.

Yuying Li, Leqi Zheng, Yongzi Yu, Wenrui Zhou, Xuchang Zhong, Xing Hu, Jing Jin, Hangjie Yuan, and Tao Feng. Filter, then reweight: Rethinking optimization granularity in on-policy distillation. *arXiv preprint arXiv:2606.02684*, 2026c. URL https://arxiv.org/abs/2606.02684.

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 *International Conference on Learning Representations*, pp. 39578–39601, 2024. URL https://arxiv.org/abs/2305.20050.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6121–6133, 2020. doi: 10.18653/v1/2020.emnlp-main.494. URL https://aclanthology.org/2020.emnlp-main.494/.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1711.05101.

Kevin Lu. On-policy distillation. Thinking Machines Lab: Connectionism, 2025. URL https://thinkingmachines.ai/blog/on-policy-distillation/.

Stephane Ross and J. Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014. URL https://arxiv.org/abs/1406.5979.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Huanxin Sheng, Zhiling Ye, Haonan Wang, Jian Wang, Jinjie Gu, and Jian Kang. 1% of tokens can be enough: On gradient estimation in on-policy distillation. *arXiv preprint arXiv:2609.24432*, 2026. URL https://arxiv.org/abs/2609.24432.

Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15731–15740, 2024. URL https://arxiv.org/abs/2403.01427.

Boxin Wang, Chankyu Lee, Nayeon Lee, Sheng-Chieh Lin, Wenliang Dai, Yang Chen, Yangyi Chen, Zhuolin Yang, Zihan Liu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nemotron-Cascade: Scaling cascaded reinforcement learning for general-purpose reasoning models. *arXiv preprint arXiv:2512.13607*, 2025. URL https://arxiv.org/abs/2512.13607.

Rui Wang, Hongru Wang, Yi Chen, Boyang Xue, Tianqing Fang, Wenhao Yu, and Kam-Fai Wong. Demystifying on-policy distillation: Roles, pathologies, and regulations. *arXiv preprint arXiv:2607.13399*, 2026a. URL https://arxiv.org/abs/2607.13399.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation. *arXiv preprint arXiv:2605.26844*, 2026b. URL https://arxiv.org/abs/2605.26844.

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, 2023. doi: 10.18653/v1/2023.acl-long.605. URL https://aclanthology.org/2023.acl-long.605/.

Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. KDRL: Post-training reasoning LLMs via unified knowledge distillation and reinforcement learning. *arXiv preprint arXiv:2506.02208*, 2025. URL https://arxiv.org/abs/2506.02208.

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. URL https://arxiv.org/abs/2604.14084.

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. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

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*, 2026a. URL https://arxiv.org/abs/2602.12125.

Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, Sheng-Chieh Lin, Chankyu Lee, Yangyi Chen, Dongfu Jiang, Jiafan He, Renjie Pi, Grace Lam, Nayeon Lee, Alexander Bukharin, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nemotron-Cascade 2: Post-training LLMs with cascade RL and multi-domain on-policy distillation. *arXiv preprint arXiv:2603.19220*, 2026b. URL https://arxiv.org/abs/2603.19220.

Anhao Zhao, Junlong Tong, Yingqi Fan, Ping Nie, Wenjie Li, and Xiaoyu Shen. PowerOPD: Stabilizing on-policy distillation with bounded power transformation. *arXiv preprint arXiv:2606.17199*, 2026. URL https://arxiv.org/abs/2606.17199.

Ziheng Zhou, Jiaqi Li, Huacong Tang, Ying Nian Wu, and Demetri Terzopoulos. Less is more: Early stopping rollout for on-policy distillation. *arXiv preprint arXiv:2605.27028*, 2026. URL https://arxiv.org/abs/2605.27028.

A Derivation and Instantiations of the Unified OPD Objective

This appendix explains how the local formulation in Section 3 gives rise to the four-component objective, then specifies each method through its rollout source, trajectory weight, position weight, and local loss. Each instantiation is followed by the objective obtained by substitution. Table 4 provides an index of the methods and controls covered here. Appendix C establishes the sampled/direct relation and the effect of support normalization.

A.1 From local distillation to the unified objective

Consider one training iteration. The student collects responses, receives teacher feedback at their prefixes, and updates its parameters using those observations. A response contains several supervised positions, and a batch contains several responses. The objective therefore needs to specify both the loss at each position and how those losses are combined.

At iteration k, let $\pi_k = \pi_{\theta_k}$ be the collecting student. At prefix $s_t = (x, y_{<t})$, write $p_{\theta,t}(v) = \pi_\theta(v \mid s_t)$ and $q_t(v) = \pi_T(v \mid s_t)$. We abbreviate $p_t = p_{\theta_k,t}$ when computing statistics before the update. The prompt distribution d_k is shared across the comparisons.

Start with one position. Teacher feedback at s_t defines a local loss $\ell_{k,t}(\theta)$. In Eq. (1), this is $D_{\mathrm{KL}}(p_{\theta,t} \| q_t)$. Other choices can compare selected vocabulary entries, use coarser teacher information, or supervise the sampled token. The loss specifies what is learned from the feedback available at this prefix.

Combine positions within a response. Given a collected response y, assign position t a nonnegative weight $m_{k,t}(x, y)$. Its weighted sum of local losses is

$$A_k(x, y; \theta) = \sum_{t=1}^{T} m_{k,t}(x, y)\ell_{k,t}(\theta). \tag{3}$$

Setting every weight to one gives equal weight to all positions. A zero weight omits that position's loss, while an intermediate weight reduces its contribution. This choice allocates supervision within a response without changing how the response was generated.

Combine responses from the rollout source. Let $\mu_k(\cdot \mid x)$ specify how a response is generated and let $w_k(x, y) \geq 0$ scale its entire loss. Averaging over prompts and collected responses gives

$$\begin{aligned} \mathcal{L}_k^{\mu,w,m,\ell}(\theta) &= \mathbb{E}_{x\sim d_k,\, y\sim\mu_k(\cdot|x)}\left[w_k(x, y)A_k(x, y; \theta)\right] \\ &= \mathbb{E}_{x\sim d_k,\, y\sim\mu_k(\cdot|x)}\left[w_k(x, y)\sum_{t=1}^{T} m_{k,t}(x, y)\ell_{k,t}(\theta)\right]. \end{aligned} \tag{4}$$

The two choices act at different stages: μ_k determines which responses and prefixes are collected, while w_k determines how much each collected response contributes. For example, teacher-assisted generation changes μ_k, whereas discarding a response after checking its answer changes w_k. Together with the position weights and local loss, these operations yield the four components in Eq. (2).

Optimize on the collected data. During each update, the responses, teacher predictions, sampling laws, and weights are fixed. A weight computed from the student is evaluated using π_k. With sg denoting stop-gradient, this convention is explicit in

$$\mathcal{L}_k^{\mu,w,m,\ell}(\theta) = \mathbb{E}_{x\sim d_k,\, y\sim\mu_k}\left[\mathrm{sg}(w_k(x, y))\sum_{t=1}^{T}\mathrm{sg}(m_{k,t}(x, y))\ell_{k,t}(\theta)\right]. \tag{5}$$

Only student-dependent terms inside $\ell_{k,t}$ are differentiated. Any randomness in response or position selection is included in the expectation. The preliminary objective is recovered with $\mu_k = \pi_k$, $w_k = m_{k,t} = 1$, and the reverse-KL local loss.

Reading the method instantiations. Below, each displayed tuple lists $(\mu_k, w_k, m_{k,t}, \ell_{k,t})$ in this order, followed by the objective that the method minimizes. For compactness, $\mathbb{E}_{\rho_k}$ denotes $\mathbb{E}_{x\sim d_k,\, y\sim\rho_k(\cdot|x)}$, and $\sum_t$ runs over the collected response. We retain the notation $\ell_{k,t}^{\mathrm{samp}}$ and

$a_{k,t} = \log[q_t(y_t)/p_t(y_t)]$ for the sampled loss and detached credit defined in Section 3.2. Arguments (x, y) on weights are omitted when clear.

We assign a factor shared by all positions of a response to w, a factor scaling a position's entire loss to m, and changes to the local comparison to ℓ. This operational convention makes composite methods explicit even when factors could be rearranged algebraically. The instantiations describe our evaluated adaptations and controls. Finite-batch normalization is specified in Appendix B.1.

A.2 Rollout source: constructing the response distribution

These methods change how the response is collected while retaining uniform weights and sampled log-ratio supervision. Let $\nu_T(\cdot \mid x)$ be the empirical distribution of cached teacher responses. A response-level mixture first chooses its source, giving

$$\mu_k^\lambda(y \mid x) = (1 - \lambda_k)\pi_k(y \mid x) + \lambda_k \nu_T(y \mid x). \tag{6}$$

The local loss is evaluated on the resulting prefixes, regardless of which source generated them.

Fixed response mixing. Following the rollout-distribution choice in GKD (Agarwal et al., 2024), the fixed-mixture control sets $\lambda_k = 0.5$. Its instantiation is

$$\begin{aligned}(\mu_k, w_k, m_{k,t}, \ell_{k,t}) &= (\mu_k^{0.5}, 1, 1, \ell_{k,t}^{\text{samp}}),\\ \mathcal{L}_k^{\text{fixed}}(\theta) &= \mathbb{E}_{\mu_k^{0.5}}\left[\sum_t \ell_{k,t}^{\text{samp}}(\theta)\right].\end{aligned} \tag{7}$$

Teacher-only responses. Setting $\lambda_k = 1$ selects every response from the teacher cache:

$$\begin{aligned}(\mu_k, w_k, m_{k,t}, \ell_{k,t}) &= (\nu_T, 1, 1, \ell_{k,t}^{\text{samp}}),\\ \mathcal{L}_k^{\text{teacher}}(\theta) &= \mathbb{E}_{\nu_T}\left[\sum_t \ell_{k,t}^{\text{samp}}(\theta)\right].\end{aligned} \tag{8}$$

Here the teacher changes the collected prefixes, while the local objective remains sampled distillation. Cached responses need not have the same law as fresh samples from π_T.

DAgger-style response mixing. The DAgger-inspired control (Ross et al., 2011; Li et al., 2026a) gradually transfers generation from teacher to student with $\lambda_k = \max(1 - k/250, 0)$. Thus

$$\begin{aligned}(\mu_k, w_k, m_{k,t}, \ell_{k,t}) &= (\mu_k^{\lambda_k}, 1, 1, \ell_{k,t}^{\text{samp}}),\\ \mathcal{L}_k^{\text{DAgger}}(\theta) &= \mathbb{E}_{\mu_k^{\lambda_k}}\left[\sum_t \ell_{k,t}^{\text{samp}}(\theta)\right].\end{aligned} \tag{9}$$

The schedule changes μ_k over training. This response-level adaptation retains the OPD loss and does not introduce the original DAgger dataset-aggregation and teacher-label objective.

AggreVaTe-style handoff. This control changes the source within a response (Ross & Bagnell, 2014; Li et al., 2026a). Sample κ uniformly from $\{0, \ldots, T_{\max}(k)\}$, where $T_{\max}(k)$ increases linearly from zero to 16,384 over 250 steps. The student generates up to κ tokens. If it has not terminated, the teacher completes that prefix. Writing μ_k^{handoff} for this rollout law gives

$$\begin{aligned}(\mu_k, w_k, m_{k,t}, \ell_{k,t}) &= (\mu_k^{\text{handoff}}, 1, 1, \ell_{k,t}^{\text{samp}}),\\ \mathcal{L}_k^{\text{handoff}}(\theta) &= \mathbb{E}_{\mu_k^{\text{handoff}}}\left[\sum_t \ell_{k,t}^{\text{samp}}(\theta)\right].\end{aligned} \tag{10}$$

The teacher now continues prefixes actually reached by the student. The adaptation changes the rollout law without adding the original AggreVaTe cost-to-go objective.

Short generation. The early-stopping control (Zhou et al., 2026) samples at most $K = 100$ student tokens. Let $\pi_k^{\leq K}$ denote this truncated rollout law. Then

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k^{\leq K}, 1, 1, \ell_{k,t}^{\text{samp}}),$$
$$\mathcal{L}_k^{\text{short}}(\theta) = \mathbb{E}_{\pi_k^{\leq K}}\left[\sum_{t=1}^{T} \ell_{k,t}^{\text{samp}}(\theta)\right], \qquad T \leq K. \tag{11}$$

Reaching the cap truncates the response without adding a sampled termination token. Appendix B.3 gives the conditions under which this produces the same expected update as generating a full response and supervising only its prefix.

SFT followed by OPD. The archived two-stage control first fits teacher responses with $\ell_{k,t}^{\text{CE}}(\theta) = -\log p_{\theta,t}(y_t)$, then switches to student rollouts and sampled distillation. Its two stages are

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\nu_T, 1, 1, \ell_{k,t}^{\text{CE}}),$$
$$\mathcal{L}_k^{\text{SFT}}(\theta) = \mathbb{E}_{\nu_T}\left[-\sum_t \log p_{\theta,t}(y_t)\right]. \tag{12}$$
$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{\text{samp}}),$$
$$\mathcal{L}_k^{\text{post-SFT}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \ell_{k,t}^{\text{samp}}(\theta)\right]. \tag{13}$$

The SFT parameters initialize the OPD stage. This procedure changes both the starting student and its training objective, so it is treated separately from the rollout substitutions above.

A.3 Trajectory weighting: selecting and scaling responses

These controls retain student rollouts and change the contribution of a completed response. Let $R \in \{-1, +1\}$ indicate verifier failure or success. Define the detached response statistics

$$\bar{\ell}_T(x, y) = \frac{1}{T}\sum_t \log q_t(y_t), \qquad \Delta_k(x, y) = \sum_t \log \frac{q_t(y_t)}{p_t(y_t)} = \sum_t a_{k,t}. \tag{14}$$

A trajectory weight scales every local loss in that response by the same amount.

Correct-only supervision. The verifier accepts only successful responses:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, \mathbf{1}\{R = +1\}, 1, \ell_{k,t}^{\text{samp}}),$$
$$\mathcal{L}_k^{\text{correct}}(\theta) = \mathbb{E}_{\pi_k}\left[\mathbf{1}\{R = +1\}\sum_t \ell_{k,t}^{\text{samp}}(\theta)\right]. \tag{15}$$

Failure-only supervision. The complementary gate retains unsuccessful responses:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, \mathbf{1}\{R = -1\}, 1, \ell_{k,t}^{\text{samp}}),$$
$$\mathcal{L}_k^{\text{failure}}(\theta) = \mathbb{E}_{\pi_k}\left[\mathbf{1}\{R = -1\}\sum_t \ell_{k,t}^{\text{samp}}(\theta)\right]. \tag{16}$$

Within each retained failure, the teacher still determines the signed credit at every token. The verifier selects responses without changing their local distillation loss.

Reward-gated OPD. RG-OPD (Akhondzadeh et al., 2026) keeps responses whose verifier outcome agrees with the sign of the teacher–student log-likelihood ratio:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, \mathbf{1}\{R\Delta_k > 0\}, 1, \ell_{k,t}^{\text{samp}}),$$
$$\mathcal{L}_k^{\text{RG}}(\theta) = \mathbb{E}_{\pi_k}\left[\mathbf{1}\{R\Delta_k > 0\}\sum_t \ell_{k,t}^{\text{samp}}(\theta)\right]. \tag{17}$$

A correct response passes when the teacher assigns it higher likelihood than the student. An incorrect response passes with the reverse preference. Thus the gate can retain both outcome types.

FiRe. FiRe (Li et al., 2026c) filters responses by teacher likelihood and reweights positions in those retained. Define $F_k(y) = \mathbf{1}\{\bar{\ell}_T \geq \tau_k\}$, where τ_k is the rolling 20th percentile of response scores. Its token weights are

$$\begin{aligned} b_{k,t} &= (1 + c^T_{k,t})(1 + c^S_{k,t}), \\ c^T_{k,t} &= 1 - \frac{H^T_t}{\max H^T}, \qquad c^S_{k,t} = \frac{H^S_t}{\max H^S}. \end{aligned} \tag{18}$$

Here H^T_t and H^S_t are teacher and student entropy. Teacher entropy uses the available top-K distribution, and the maxima follow the run's normalization scope. Lower teacher entropy and higher student entropy receive larger weights. Substitution gives

$$\begin{aligned} (\mu_k, w_k, m_{k,t}, \ell_{k,t}) &= (\pi_k, F_k, b_{k,t}, \ell^{\text{samp}}_{k,t}), \\ \mathcal{L}^{\text{FiRe}}_k(\theta) &= \mathbb{E}_{\pi_k}\left[F_k(y) \sum_t b_{k,t} \ell^{\text{samp}}_{k,t}(\theta) \right]. \end{aligned} \tag{19}$$

FiRe therefore changes both w and m. The sampled local loss remains the same.

Sequence-mean aggregation. Giving each response its average token loss corresponds, in the unnormalized objective, to a length weight $w_k = 1/T$:

$$\begin{aligned} (\mu_k, w_k, m_{k,t}, \ell_{k,t}) &= (\pi_k, 1/T, 1, \ell^{\text{samp}}_{k,t}), \\ \mathcal{L}^{\text{seq}}_k(\theta) &= \mathbb{E}_{\pi_k}\left[\frac{1}{T} \sum_t \ell^{\text{samp}}_{k,t}(\theta) \right]. \end{aligned} \tag{20}$$

For B responses, its empirical form is

$$\mathcal{L}_{\text{seq}}(\theta) = \frac{1}{B} \sum_{i=1}^{B} \frac{1}{T_i} \sum_{t=1}^{T_i} \ell^{\text{samp}}_{i,t}(\theta). \tag{21}$$

Each response contributes one mean loss regardless of its length. This averaging rule must be retained when implementing the tuple. Appendix B.1 contrasts it with pooled-token normalization.

A.4 Token-position weighting: allocating supervision within a response

These methods generate full student responses and change the contribution of individual positions. Let H_t be student entropy, Ω_t the available teacher support, and

$$H_t = -\sum_v p_t(v) \log p_t(v), \qquad d_t = \sum_{v \in \Omega_t} q_t(v) \log \frac{q_t(v)}{p_t(v)}. \tag{22}$$

The restricted sum d_t is a disagreement statistic, not necessarily a normalized KL. Hats below indicate the configured normalization over the packed micro-batch, and Q_a denotes an empirical a-quantile.

TIP. TIP (Xu et al., 2026) combines student uncertainty and disagreement using $I_t = \hat{H}_t + \hat{d}_t - \hat{H}_t \hat{d}_t$. The soft-OR score is high when either input is high. Define $M^{\text{TIP}}_t = \mathbf{1}\{I_t \geq Q_{.5}(I)\}$. Its instantiation is

$$\begin{aligned} (\mu_k, w_k, m_{k,t}, \ell_{k,t}) &= (\pi_k, 1, M^{\text{TIP}}_t, \ell^{\text{samp}}_{k,t}), \\ \mathcal{L}^{\text{TIP}}_k(\theta) &= \mathbb{E}_{\pi_k}\left[\sum_t M^{\text{TIP}}_t \ell^{\text{samp}}_{k,t}(\theta) \right]. \end{aligned} \tag{23}$$

The importance score selects where supervision is applied. The local loss still uses the sampled log-ratio credit.

TA-OPD. Teachability-aware selection (Wang et al., 2026b) also checks whether teacher probability lies on candidates the student considers plausible. Let $c_t = q_t(\Omega_t \cap \mathrm{Top}_{16}(p_t))$, $J_t = \hat{d}_t \hat{c}_t$, and $M_t^{\mathrm{TA}} = \mathbf{1}\{J_t \geq Q_{.95}(J)\}$. Then

$$\begin{aligned}(\mu_k, w_k, m_{k,t}, \ell_{k,t}) &= (\pi_k, 1, M_t^{\mathrm{TA}}, \ell_{k,t}^{\mathrm{samp}}),\\ \mathcal{L}_k^{\mathrm{TA}}(\theta) &= \mathbb{E}_{\pi_k}\left[\sum_t M_t^{\mathrm{TA}} \ell_{k,t}^{\mathrm{samp}}(\theta)\right].\end{aligned} \tag{24}$$

Our adaptation targets 5% retention. A large disagreement is selected only when the overlap score is also large. Both TIP and TA-OPD compute their quantiles over the available packed micro-batch, which can contain only one long response.

SelecTKD. SelecTKD (Huang et al., 2025) uses $A_t = \mathbf{1}\{\arg\max_v p_t(v) \in \mathrm{Top}_5(q_t)\}$ to test whether the student agrees with the teacher's top-ranked set. Its soft weighting gives

$$\begin{aligned}(\mu_k, w_k, m_{k,t}, \ell_{k,t}) &= (\pi_k, 1, 0.01 + 0.99A_t, \ell_{k,t}^{\mathrm{samp}}),\\ \mathcal{L}_k^{\mathrm{SelecT}}(\theta) &= \mathbb{E}_{\pi_k}\left[\sum_t (0.01 + 0.99A_t)\ell_{k,t}^{\mathrm{samp}}(\theta)\right].\end{aligned} \tag{25}$$

Agreed positions receive unit weight, and the remaining positions retain weight 0.01. The agreement test sets the weight without relabeling the sampled token.

Positional controls. These controls test supervision placement with a common budget $K = 100$. Let $K_T = \min(K, T)$. For a random window, draw U uniformly from $\{1, \ldots, T - K_T + 1\}$. For scattered selection, draw $\mathcal{U}_{K_T}$ uniformly over subsets of $\{1, \ldots, T\}$ of size K_T. The following four cases retain the same sampled loss and full rollout.

Prefix supervision. The ESR-inspired prefix control (Zhou et al., 2026) supervises the first K_T positions:

$$\begin{aligned}(\mu_k, w_k, m_{k,t}, \ell_{k,t}) &= (\pi_k, 1, \mathbf{1}\{t \leq K_T\}, \ell_{k,t}^{\mathrm{samp}}),\\ \mathcal{L}_k^{\mathrm{prefix}}(\theta) &= \mathbb{E}_{\pi_k}\left[\sum_{t=1}^{K_T} \ell_{k,t}^{\mathrm{samp}}(\theta)\right].\end{aligned} \tag{26}$$

Unlike short generation, this control collects the full response and changes only m.

Suffix supervision. Selecting the last K_T positions gives

$$\begin{aligned}(\mu_k, w_k, m_{k,t}, \ell_{k,t}) &= (\pi_k, 1, \mathbf{1}\{t > T - K_T\}, \ell_{k,t}^{\mathrm{samp}}),\\ \mathcal{L}_k^{\mathrm{suffix}}(\theta) &= \mathbb{E}_{\pi_k}\left[\sum_{t=T-K_T+1}^{T} \ell_{k,t}^{\mathrm{samp}}(\theta)\right].\end{aligned} \tag{27}$$

Random window. A uniformly placed contiguous window gives

$$\begin{aligned}(\mu_k, w_k, m_{k,t}, \ell_{k,t}) &= (\pi_k, 1, \mathbf{1}\{U \leq t < U + K_T\}, \ell_{k,t}^{\mathrm{samp}}),\\ \mathcal{L}_k^{\mathrm{window}}(\theta) &= \mathbb{E}_{\pi_k, U}\left[\sum_{t=U}^{U+K_T-1} \ell_{k,t}^{\mathrm{samp}}(\theta)\right].\end{aligned} \tag{28}$$

Random scatter. Selecting K_T positions without a contiguity constraint gives

$$\begin{aligned}(\mu_k, w_k, m_{k,t}, \ell_{k,t}) &= (\pi_k, 1, \mathbf{1}\{t \in \mathcal{U}_{K_T}\}, \ell_{k,t}^{\mathrm{samp}}),\\ \mathcal{L}_k^{\mathrm{scatter}}(\theta) &= \mathbb{E}_{\pi_k, \mathcal{U}_{K_T}}\left[\sum_{t \in \mathcal{U}_{K_T}} \ell_{k,t}^{\mathrm{samp}}(\theta)\right].\end{aligned} \tag{29}$$

The expectations for the two randomized controls include their position-selection randomness. All four controls supervise the entire available response when $T < K$.

A.5 Local supervision loss: constructing the teacher–student comparison

These methods change the local loss or the teacher information used to construct it. We organize them by discrepancy, vocabulary support, retained information, and sampled-credit transformation. Unless an additional gate is stated, student rollouts and unit weights are retained.

A.5.1 Distributional losses and auxiliary supervision

We first give the full-vocabulary objectives. A restricted-support implementation also specifies the construction in Appendix A.5.2.

Direct reverse KL. The local loss weights teacher–student discrepancies by student probability:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, D_{\mathrm{KL}}(p_{\theta,t} \| q_t)),$$
$$\mathcal{L}_k^{\mathrm{RKL}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \sum_v p_{\theta,t}(v) \log \frac{p_{\theta,t}(v)}{q_t(v)}\right]. \tag{30}$$

Forward KL. Reversing the arguments weights the comparison by teacher probability:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, D_{\mathrm{KL}}(q_t \| p_{\theta,t})),$$
$$\mathcal{L}_k^{\mathrm{FKL}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \sum_v q_t(v) \log \frac{q_t(v)}{p_{\theta,t}(v)}\right]. \tag{31}$$

Both are direct distributional objectives (Agarwal et al., 2024; Fu et al., 2026). Appendix C.1 proves the conditional expected-gradient relation between direct reverse KL and sampled OPD.

Skew KL. Let $M_{\alpha,t} = \alpha p_{\theta,t} + (1-\alpha) q_t$, with $0 \le \alpha < 1$. Skew KL compares the student with this mixture (Ko et al., 2024):

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, D_{\mathrm{KL}}(p_{\theta,t} \| M_{\alpha,t})),$$
$$\mathcal{L}_k^{\mathrm{SKL}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t D_{\mathrm{KL}}(p_{\theta,t} \| M_{\alpha,t})\right]. \tag{32}$$

The student appears in both arguments and is differentiated through both. At $\alpha = 0$ the loss recovers reverse KL. The mixture moderates discrepancies where the teacher probability is small.

Jensen–Shannon divergence. For $M_{\beta,t} = \beta p_{\theta,t} + (1-\beta) q_t$, define $\ell_{k,t}^{\mathrm{JS}} = \beta D_{\mathrm{KL}}(p_{\theta,t} \| M_{\beta,t}) + (1-\beta) D_{\mathrm{KL}}(q_t \| M_{\beta,t})$, where $0 < \beta < 1$. Then

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{\mathrm{JS}}),$$
$$\mathcal{L}_k^{\mathrm{JS}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \{\beta D_{\mathrm{KL}}(p_{\theta,t} \| M_{\beta,t}) + (1-\beta) D_{\mathrm{KL}}(q_t \| M_{\beta,t})\}\right]. \tag{33}$$

Gradients pass through $M_{\beta,t}$ in both KL terms. The symmetric choice is $\beta = 0.5$, with $\beta = 0.1$ and 0.9 as additional controls. Varying β changes both the balance of the two comparisons and the scale of the unnormalized loss.

Sampled OPD with gated forward KL. The entropy-aware control (Jin et al., 2026) augments the sampled loss using a detached coefficient $e_{k,t} \ge 0$, which includes the entropy gate and its scale:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{\mathrm{samp}} + e_{k,t} \ell_{k,t}^{\mathrm{FKL}}),$$
$$\mathcal{L}_k^{\mathrm{EOPD}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \left\{\ell_{k,t}^{\mathrm{samp}}(\theta) + e_{k,t} D_{\mathrm{KL}}(q_t \| p_{\theta,t})\right\}\right]. \tag{34}$$

Our adaptation estimates teacher entropy using the top-32 probabilities. The gate scales only the auxiliary term, so it belongs inside ℓ. A position weight $m = e$ would scale both terms.

Squared log-ratio loss. The k2 control differentiates the squared discrepancy at the sampled token:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \frac{1}{2}[\log q_t(y_t) - \log p_{\theta,t}(y_t)]^2),$$
$$\mathcal{L}_k^{\mathrm{k2}}(\theta) = \mathbb{E}_{\pi_k}\left[\frac{1}{2}\sum_t \left(\log q_t(y_t) - \log p_{\theta,t}(y_t)\right)^2\right]. \tag{35}$$

Its negative gradient is $[\log q_t(y_t) - \log p_{\theta,t}(y_t)]\nabla_\theta \log p_{\theta,t}(y_t)$. At θ_k this matches the sampled reference in Eq. (72), but the log-ratio changes with θ during optimization.

Stop-probability distillation. Let $p_{\theta,t}^{\mathrm{stop}}$ and q_t^{stop} be the student and teacher stopping probabilities. Define the binary cross-entropy

$$\ell_{k,t}^{\mathrm{aux}}(\theta) = -q_t^{\mathrm{stop}} \log p_{\theta,t}^{\mathrm{stop}} - (1 - q_t^{\mathrm{stop}}) \log(1 - p_{\theta,t}^{\mathrm{stop}}). \tag{36}$$

Adding it with unit weight gives

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{\mathrm{samp}} + \ell_{k,t}^{\mathrm{aux}}),$$
$$\mathcal{L}_k^{\mathrm{stop}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \left\{\ell_{k,t}^{\mathrm{samp}}(\theta) + \ell_{k,t}^{\mathrm{aux}}(\theta)\right\}\right]. \tag{37}$$

The auxiliary term supervises stopping at every prefix, including positions where the sampled token does not end the response.

A.5.2 Vocabulary support and its local loss

A support $S_t \subseteq \mathcal{V}$ determines which candidates enter the local comparison (Fu et al., 2026; Li et al., 2026b). It changes ℓ while retaining position t. Setting $m_{k,t} = 0$ instead removes that position's entire loss. Support membership is computed before the update and held fixed.

Loss definitions used below. For a nonempty support with positive model mass, conditional matching uses

$$\tilde{p}_{\theta,t}^{S_t}(v) = \frac{p_{\theta,t}(v)}{p_{\theta,t}(S_t)}, \qquad \tilde{q}_t^{S_t}(v) = \frac{q_t(v)}{q_t(S_t)}, \qquad v \in S_t. \tag{38}$$

We write $\ell_{k,t}^{S,D}(\theta) = D(\tilde{p}_{\theta,t}^{S_t}, \tilde{q}_t^{S_t})$ for the chosen conditional comparison. Here $D_{\mathrm{R}}(p,q) = D_{\mathrm{KL}}(p\|q)$ and $D_{\mathrm{F}}(p,q) = D_{\mathrm{KL}}(q\|p)$. The student normalizer is differentiated. Superscripts on the conditional distributions are omitted when the support is clear. A support choice and a loss choice are both required to specify a method. The tuples below keep D explicit for support families and specialize it when comparing KL directions.

For sampled support supervision, define

$$\ell_{k,t}^{\mathrm{samp},S}(\theta) = -\,\mathrm{sg}(c_{k,t}^{S_t}) \log p_{\theta,t}(y_t). \tag{39}$$

The credit $c_{k,t}^{S_t}$ is constructed from the available support information. For example, using a detached support divergence D_t as a sampled penalty gives $c_{k,t}^{S_t} = -D_t$, rather than differentiating D_t directly. Thus a sampled and a direct rule need not share an update even with the same support.

A tail-bucket construction additionally retains the complement mass. With $\bar{p}_{\theta,t} = 1 - p_{\theta,t}(S_t)$ and $\bar{q}_t = 1 - q_t(S_t)$, define

$$\ell_{k,t}^{\mathrm{bucket},S}(\theta) = \sum_{v \in S_t} p_{\theta,t}(v) \log \frac{p_{\theta,t}(v)}{q_t(v)} + \bar{p}_{\theta,t} \log \frac{\bar{p}_{\theta,t}}{\bar{q}_t}. \tag{40}$$

Conditional matching compares relative probabilities inside S_t, whereas the bucket loss also constrains mass outside it. In particular, conditional KL on a singleton is zero, while sampled credit for that token can be nonzero. Appendix C.2 explains how normalization changes the objective and its update.

Teacher top-K: direct comparison. Let $S_t^q(K) = \mathrm{Top}_K(q_t)$ and set $S_t = S_t^q(K)$, with $K = 32$ in the top-32 controls. Direct conditional matching gives

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{S^q, D}),$$
$$\mathcal{L}_k^{\text{topK-direct}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t D(\tilde{p}_{\theta,t}^{S^q}, \tilde{q}_t^{S^q})\right]. \tag{41}$$

The direct reverse-KL control takes $D = D_{\mathrm{R}}$.

Teacher top-K: sampled comparison. Using the same candidate set to construct a detached sampled credit gives

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{\text{samp},S^q}),$$
$$\mathcal{L}_k^{\text{topK-sampled}}(\theta) = \mathbb{E}_{\pi_k}\left[-\sum_t \mathrm{sg}(c_{k,t}^{S_t^q}) \log p_{\theta,t}(y_t)\right]. \tag{42}$$

Teacher top-K: tail bucket. Retaining the top-K candidates and their aggregated complement gives

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{\text{bucket},S^q}),$$
$$\mathcal{L}_k^{\text{topK-bucket}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \ell_{k,t}^{\text{bucket},S^q}(\theta)\right]. \tag{43}$$

The local bucket loss is defined in Eq. (40). The tuple identifies the bucket objective, separately from any sampled surrogate used to optimize it.

Teacher–student intersection. With $S_t^p(K) = \mathrm{Top}_K(p_t)$, let $S_t^\cap = S_t^p(K) \cap S_t^q(K)$. It restricts the comparison to candidates favored by both models:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{S^\cap, D}),$$
$$\mathcal{L}_k^{\text{intersection}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t D(\tilde{p}_{\theta,t}^{S^\cap}, \tilde{q}_t^{S^\cap})\right]. \tag{44}$$

The conditional construction assumes a nonempty intersection. Empty supports require a specified skip or fallback rule.

Teacher–student union. Let $S_t^\cup = S_t^p(K) \cup S_t^q(K)$ include candidates favored by either model. The two KL directions give distinct instances:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{S^\cup, D_{\mathrm{R}}}),$$
$$\mathcal{L}_k^{\text{union-RKL}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t D_{\mathrm{KL}}(\tilde{p}_{\theta,t}^{S^\cup} \| \tilde{q}_t^{S^\cup})\right]. \tag{45}$$

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{S^\cup, D_{\mathrm{F}}}),$$
$$\mathcal{L}_k^{\text{union-FKL}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t D_{\mathrm{KL}}(\tilde{q}_t^{S^\cup} \| \tilde{p}_{\theta,t}^{S^\cup})\right]. \tag{46}$$

Vocabulary-budget definitions. For the available support Ω_t, score candidates by $r_t(v) = \max(p_t(v), q_t(v))$. A pooled quantile threshold defines

$$S_t^{\text{quant}} = \{v \in \Omega_t : r_t(v) \geq \tau_{\mathcal{B}}\} \cup \{\arg\max_v q_t(v)\}. \tag{47}$$

The threshold targets eight retained entries per position on average over the packed micro-batch, with the teacher's top token included. The per-sequence rule uses a separate threshold τ_y for each response and is denoted S_t^{qseq}. Fixed-eight uses a fixed count at each position, denoted S_t^8. These set definitions induce the following objectives.

Quantile budgeting. A shared threshold allocates a variable number of entries to each position:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{S^{\mathrm{quant}},D}),$$
$$\mathcal{L}_k^{\mathrm{quantile}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \ell_{k,t}^{S^{\mathrm{quant}},D}(\theta)\right]. \tag{48}$$

Per-sequence quantile budgeting. Choosing the threshold independently within each response gives

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{S^{\mathrm{qseq}},D}),$$
$$\mathcal{L}_k^{\mathrm{qseq}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \ell_{k,t}^{S^{\mathrm{qseq}},D}(\theta)\right]. \tag{49}$$

This changes support selection. It does not apply the sequence-mean aggregation in Eq. (21).

Fixed-eight support. Keeping eight candidates per position gives

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{S^{8},D}),$$
$$\mathcal{L}_k^{\mathrm{fixed8}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \ell_{k,t}^{S^{8},D}(\theta)\right]. \tag{50}$$

The three rules change the vocabulary support inside ℓ while retaining every sequence position.

Student-tail support. Order the available teacher candidates by q_t, and let $S_{t,j}^q$ contain the first j entries. Choose

$$k_t^\pi = \min\{j : p_t(S_{t,j}^q) \geq 1 - \varepsilon\}, \qquad S_t^{\mathrm{tail}} = S_{t,k_t^\pi}^q, \qquad \varepsilon = 0.05. \tag{51}$$

This retains enough teacher-ranked candidates to cover the target student mass. Its mapping is

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{S^{\mathrm{tail}},D}),$$
$$\mathcal{L}_k^{\text{student-tail}}(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \ell_{k,t}^{S^{\mathrm{tail}},D}(\theta)\right]. \tag{52}$$

When the available candidates cannot reach the target, the diagnostic records that failure. The 95% mass is therefore a target, rather than a guarantee at every position.

Gates and token/event representation. Write $G_{k,t}^{\mathrm{rep}}$, $G_{k,t}^{\mathrm{HQ}}$, and $G_{k,t}^{\mathrm{state}}$ for the detached repetition, HQ, and combined state gates. Let $\ell_{k,t}^{\mathrm{tail},j}$ denote the student-tail local loss with the token/event representation used by control j. The HQ/state and exact-carrier controls score individual token coordinates, whereas the student-tail reference aggregates terminal-event mass. Distinguishing them avoids attributing a composite implementation change solely to m.

Student-tail with repetition gate. The repetition control suppresses selected local losses through

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, G^{\mathrm{rep}}_{k,t}, \ell^{\mathrm{tail,rep}}_{k,t}),$$
$$\mathcal{L}^{\text{tail-rep}}_k(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t G^{\mathrm{rep}}_{k,t}\ell^{\mathrm{tail,rep}}_{k,t}(\theta)\right]. \tag{53}$$

Student-tail with HQ gate. The HQ control uses its corresponding detached gate and local representation:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, G^{\mathrm{HQ}}_{k,t}, \ell^{\mathrm{tail,HQ}}_{k,t}),$$
$$\mathcal{L}^{\text{tail-HQ}}_k(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t G^{\mathrm{HQ}}_{k,t}\ell^{\mathrm{tail,HQ}}_{k,t}(\theta)\right]. \tag{54}$$

Student-tail with state gate. The combined state-gate control gives

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, G^{\mathrm{state}}_{k,t}, \ell^{\mathrm{tail,state}}_{k,t}),$$
$$\mathcal{L}^{\text{tail-state}}_k(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t G^{\mathrm{state}}_{k,t}\ell^{\mathrm{tail,state}}_{k,t}(\theta)\right]. \tag{55}$$

Student-tail with exact carrier. The exact-carrier control changes how probability is assigned to the scored token or terminal event. Writing its local loss as $\ell^{\mathrm{tail,carrier}}_{k,t}$ gives

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell^{\mathrm{tail,carrier}}_{k,t}),$$
$$\mathcal{L}^{\text{tail-carrier}}_k(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \ell^{\mathrm{tail,carrier}}_{k,t}(\theta)\right]. \tag{56}$$

A.5.3 Teacher information: probabilities, ranks, and membership

Let $\mathcal{I}_t$ denote the retained teacher information, and let $\mathcal{D}_{\mathcal{I}}$ denote its comparison loss:

$$\ell^{\mathcal{I}}_{k,t}(\theta) = \mathcal{D}_{\mathcal{I}}(p_{\theta,t}; \mathcal{I}_t). \tag{57}$$

Ranks retain $\mathcal{I}_t = \mathrm{rank}(q_t)$ over the comparison set, following list-wise distillation (Bassam et al., 2025). Membership retains $\mathcal{I}_t(v) = \mathbf{1}\{v \in S^q_t(K)\}$. Their comparison losses are denoted ℓ^{rank} and ℓ^{member}. A probability-value anchor contributes ℓ^{anc}, including its configured weight. These names distinguish the information and comparison used by each control. Ranks and membership are not probability distributions that can be substituted directly into a KL expression.

Rank-only supervision. Using only the ordering comparison gives

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell^{\text{rank}}_{k,t}),$$
$$\mathcal{L}^{\text{rank}}_k(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \ell^{\text{rank}}_{k,t}(\theta)\right]. \tag{58}$$

Rank supervision with an anchor. Restoring a numerical probability target adds the anchor term:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell^{\text{rank}}_{k,t} + \ell^{\text{anc}}_{k,t}),$$
$$\mathcal{L}^{\text{rank+anc}}_k(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \{\ell^{\text{rank}}_{k,t}(\theta) + \ell^{\text{anc}}_{k,t}(\theta)\}\right]. \tag{59}$$

Membership-only supervision. A set-membership comparison specifies accepted candidates without their ordering or relative probabilities:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell^{\text{member}}_{k,t}),$$
$$\mathcal{L}^{\text{member}}_k(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \ell^{\text{member}}_{k,t}(\theta)\right]. \tag{60}$$

Membership supervision with an anchor. The anchored variant gives

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell^{\text{member}}_{k,t} + \ell^{\text{anc}}_{k,t}),$$
$$\mathcal{L}^{\text{member+anc}}_k(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \{\ell^{\text{member}}_{k,t}(\theta) + \ell^{\text{anc}}_{k,t}(\theta)\}\right]. \tag{61}$$

Removing an anchor changes ℓ while retaining the same source and selection rules. Membership-only supervision differs from conditional support matching, which uses teacher probabilities within the selected set.

Standardized teacher values. This control supplies $\mathcal{I}^z_t = z(\log q_t)$, removing a common location and scale over the comparison set (Sun et al., 2024). Let $\ell^z_{k,t} = \mathcal{D}_z(p_{\theta,t}; \mathcal{I}^z_t)$. Its instantiation is

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell^z_{k,t}),$$
$$\mathcal{L}^{\text{zvalue}}_k(\theta) = \mathbb{E}_{\pi_k}\left[\sum_t \mathcal{D}_z(p_{\theta,t}; \mathcal{I}^z_t)\right]. \tag{62}$$

The loss still uses numerical teacher information, but its absolute probability magnitudes have been removed. The comparison and standardization scope are part of ℓ.

A.5.4 Sampled-credit transformations

For any detached credit $c_{k,t}$, define

$$\ell^c_{k,t}(\theta) = -\operatorname{sg}(c_{k,t}) \log p_{\theta,t}(y_t). \tag{63}$$

The following controls (Wang et al., 2026a; Zhao et al., 2026) replace the sampled log-ratio credit while keeping the response source and both weighting levels fixed. For each transform j, write $\ell^j_{k,t}$ for Eq. (63) with $c_{k,t} = c^j_{k,t}$.

Soft-log credit. Let $c_{k,t}^{\mathrm{soft}} = \mathrm{sgn}(a_{k,t}) \log(1 + |a_{k,t}|)$. This preserves the sign and compresses large magnitudes:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{\mathrm{soft}}),$$
$$\mathcal{L}_k^{\mathrm{soft}}(\theta) = \mathbb{E}_{\pi_k}\left[-\sum_t \mathrm{sg}(c_{k,t}^{\mathrm{soft}}) \log p_{\theta,t}(y_t)\right]. \tag{64}$$

Hard clipping. Let $c_{k,t}^{\mathrm{clip}} = \mathrm{clip}(a_{k,t}, -b, b)$. The main control uses $b = 10$, with $b = 2.303$ as the tighter ablation:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{\mathrm{clip}}),$$
$$\mathcal{L}_k^{\mathrm{clip}}(\theta) = \mathbb{E}_{\pi_k}\left[-\sum_t \mathrm{sg}(c_{k,t}^{\mathrm{clip}}) \log p_{\theta,t}(y_t)\right]. \tag{65}$$

Positive-side clipping. With $c_{k,t}^{\mathrm{pos}} = \min(a_{k,t}, b_+)$, only positive credit is capped:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{\mathrm{pos}}),$$
$$\mathcal{L}_k^{\mathrm{posclip}}(\theta) = \mathbb{E}_{\pi_k}\left[-\sum_t \mathrm{sg}(c_{k,t}^{\mathrm{pos}}) \log p_{\theta,t}(y_t)\right]. \tag{66}$$

Power credit. The PowerOPD-inspired transform uses $c_{k,t}^{\mathrm{power}} = q_t(y_t)^\alpha - p_t(y_t)^\alpha$, $\alpha > 0$ (Zhao et al., 2026):

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{\mathrm{power}}),$$
$$\mathcal{L}_k^{\mathrm{power}}(\theta) = \mathbb{E}_{\pi_k}\left[-\sum_t \mathrm{sg}(c_{k,t}^{\mathrm{power}}) \log p_{\theta,t}(y_t)\right]. \tag{67}$$

It is bounded for probabilities in $[0, 1]$ and preserves the sign of their difference. It depends on the two absolute probabilities, rather than only their log-ratio.

Tanh credit. Let $c_{k,t}^{\tanh} = b \tanh(a_{k,t}/s)$, where b sets the output magnitude and s the input scale. This smoothly saturates both signs:

$$(\mu_k, w_k, m_{k,t}, \ell_{k,t}) = (\pi_k, 1, 1, \ell_{k,t}^{\tanh}),$$
$$\mathcal{L}_k^{\tanh}(\theta) = \mathbb{E}_{\pi_k}\left[-\sum_t \mathrm{sg}(c_{k,t}^{\tanh}) \log p_{\theta,t}(y_t)\right]. \tag{68}$$

For all five cases, the tuple changes only ℓ. Each transform scales the sampled-token gradient by its detached credit, as shown in Eq. (72).

Table 4: **Index of OPD designs and their formal instantiations.** Methods are grouped by design family and linked to their component assignments and objectives. Parameter variants share an instantiation. Composite and staged controls are included.

Design family	Methods and controls	Instantiation
Reference	Sampled OPD (Vanilla)	Section 3.2
Rollout source		
Response source	Fixed mixing, teacher-only, DAgger-style mixing, AggreVaTe-style handoff	(7)–(10)
Rollout horizon	Short generation	(11)
Staged training	SFT followed by OPD (archived control)	(12)–(13)
Trajectory weighting		
Response selection	Correct-only, failure-only, RG-OPD	(15)–(17)
Composite weighting	FiRe (trajectory and position weights)	(19)
Length weighting	Sequence-mean aggregation	(20)
Token-position weighting		
Teacher-guided	TIP, TA-OPD, SelecTKD	(23)–(25)
Positional	Prefix, suffix, random window, random scatter	(26)–(29)
Local supervision loss		
Discrepancy	Direct reverse KL, forward KL, Skew KL, JSD (including mixture-weight variants)	(30)–(33)
Local objectives	Entropy-gated FKL, squared log-ratio, stop-probability distillation	(34)–(37)
Support construction	Teacher top-K (direct, sampled, tail bucket), intersection, union (RKL/FKL)	(41)–(46)
Vocabulary budget	Quantile, per-sequence quantile, fixed-eight, student-tail	(48)–(52)
Support controls	Student-tail with repetition gate, HQ gate, state gate, or exact carrier	(53)–(56)
Teacher information	Rank and membership (with/without anchors), standardized values	(58)–(62)
Sampled credit	Soft-log, hard clipping (including tighter clipping), positive-side clipping, power, tanh	(64)–(68)

B Experimental Details and Additional Results

B.1 Training and evaluation configuration

Table 5 lists the shared training parameters.

Table 5: Training configuration.

Parameter	Value
Student	Qwen3-1.7B-Base
Teacher	Qwen3-4B-Instruct-2507
Training dataset	Nemotron-Cascade-RL-Math
Training examples	14,476
Maximum prompt length	1,024 tokens
Maximum response length	16,384 tokens
Sampling temperature / top-p	1.0 / 1.0
Training batch / mini-batch size	128 / 128
Optimizer	AdamW
Learning rate / schedule	10^{-6} / constant
Weight decay	0.01
Adam betas	$(0.9, 0.999)$
Gradient-norm clipping	1.0
PPO ratio clipping	0.2 for the sampled-gradient branch

Loss aggregation. For a batch of responses, write $w_i = w_k(x_i, y_i)$ and $m_{it} = m_{k,t}(x_i, y_i)$. The empirical loss is $\widehat{\mathcal{L}}(\theta) = \sum_{i,t} \alpha_{it} \ell_{it}(\theta)$. Our default token mean and the sequence-mean control use

$$\alpha_{it}^{\text{token}} = \frac{w_i m_{it}}{\sum_{j,u} w_j m_{ju}}, \tag{69}$$

$$\alpha_{it}^{\text{sequence}} = \frac{w_i}{\sum_j w_j} \frac{m_{it}}{\sum_u m_{iu}}. \tag{70}$$

Token mean pools the selected token mass across responses. Sequence mean first averages within each response, recovering Eq. (21) under uniform weights. These expressions require positive denominators. The default assigns zero loss to an empty selection. For sequence means, responses with no selected positions require an explicit exclusion or zero-loss convention. All weights and batch denominators are held fixed during each update.

Sampled-loss implementation. For detached credit $c_{k,t}$, the sampled branch uses the PPO-clipped surrogate with $\rho_{\theta,t} = p_{\theta,t}(y_t)/p_t(y_t)$:

$$\ell_{k,t}^{\text{PPO}}(\theta) = -\min\{\rho_{\theta,t} c_{k,t}, \text{clip}(\rho_{\theta,t}, 1-\epsilon, 1+\epsilon) c_{k,t}\}, \qquad \epsilon = 0.2. \tag{71}$$

At θ_k, the ratio is one and its gradient is $\nabla_\theta \log p_{\theta,t}(y_t)|_{\theta_k}$, recovering Eq. (72). Away from the collecting policy, the ratio and clipping can change the update. The credit-clipping controls bound $c_{k,t}$ before this surrogate is formed.

B.2 Additional trajectory-selection results

We further measure trajectory retention and gradient norms to examine supervision coverage and training stability. Table 6 summarizes accepted-trajectory counts and gradient norms.

Failure-only retains 93.27% of trajectories on average, compared with 2.10% for Correct-only. Correct-only retains 3.64 trajectories per batch on average and accepts none in 47 of 750 batches. Its gradient norm exceeds 10 in eight batches, each with only one to four accepted trajectories. Vanilla and Failure-only exhibit more stable gradient norms.

Table 7 reports benchmark accuracy under sequence-mean aggregation. This control retains every trajectory but averages token losses within each trajectory before averaging over trajectories. It changes the contribution of trajectory length without filtering by correctness or teacher information; the two aggregation rules are defined in Appendix B.1.

Sequence-mean aggregation reaches 31.95, compared with Vanilla's 32.82.

Table 6: Trajectory retention and gradient norms.

	Trajectory selection		Gradient norm			
Method	**Kept/batch**	**Empty**	**Median**	**99th pct.**	**Max**	**> 10**
Vanilla	128.00	0	0.81	6.33	8.04	0
Failure only	118.14	0	0.80	6.35	7.24	0
Correct only	3.64	47	0.66	10.04	39.56	8

Table 7: Math benchmark accuracy (%) with sequence-mean aggregation.

Method	**AIME24**	**AIME25**	**AMC23**	**Minerva**	**MATH500**	**Average**
Vanilla	5.7	7.6	42.8	15.3	66.6	32.82
Sequence-mean aggregation	5.4	6.2	39.4	16.5	66.1	31.95

B.3 Additional token-weighting results

Figure 5 compares accuracy and supervision coverage for the eight token-weighting methods in Table 2. Figure 5(b) reports selected-token fractions, or the full-weight fraction for SelecTKD.

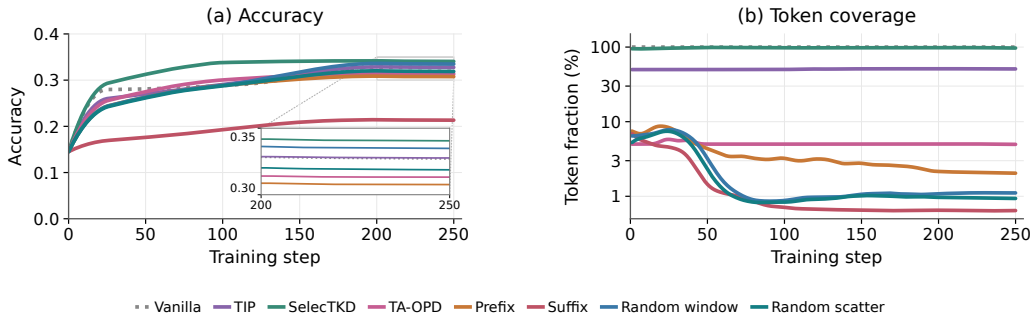

Figure 5: Token-weighting dynamics.

The mean full-weight fraction for SelecTKD is 97.66%, while the selected fractions for TIP and TA-OPD average 50.62% and 5.08%, respectively. SelecTKD assigns weight 0.01 outside the full-weight set, so its nonzero-weight coverage remains 100%.

Short generation under prefix supervision. Table 8 compares the generation cost of Prefix, Short generation, and Vanilla. Prefix and Short generation both supervise the first 100 tokens; Short generation also stops sampling after that prefix.

Table 8: Generation cost under prefix supervision.

Method	**Tokens/response**	**Gen. (s)**	**Iter. (s)**
Vanilla	12,646.2	614.67	763.44
Prefix (full generation)	3,575.5	135.95	181.59
Short generation (100 tokens)	99.1	2.66	10.76

Mean response lengths are 99.1 tokens for Short generation, 3,575.5 for Prefix, and 12,646.2 for Vanilla. The mean iteration-time ratios are 16.88 for Prefix versus Short generation and 70.95 for Vanilla versus Short generation.

Relation to prefix supervision. Under the same autoregressive policy and termination convention, the distribution of the first $\min(K, T)$ tokens is unchanged by subsequently generating a suffix. Marginalizing out that suffix therefore preserves the expected prefix update when its weights, local losses, and normalization depend only on the retained prefix and are identical in both procedures.

Full-response rewards, length-dependent weights, or different batch denominators can violate this condition.

B.4 Additional supervision-signal results

Dynamic allocation retains more teacher probability mass for the same average number of vocabulary entries per position. Both quantile-budgeting variants retain about 97.7%, compared with 54.7% for the fixed-budget baseline.

Additional support ablations. Table 9 compares local losses on teacher top-K support, gate and carrier variants of student-tail support, and the scope of quantile budget allocation. Reference methods are repeated for comparison. The teacher top-K reference in Table 2 uses the sampled rule; direct and tail-bucket variants retain the same top-32 set but change its local loss. Per-sequence quantile budgeting restricts budget sharing to positions within one trajectory, whereas the default shares the budget across the packed microbatch. The fixed-budget baseline and both quantile variants target eight entries per position on average.

Table 9: Math benchmark accuracy (%) for additional vocabulary-support ablations.

Method	AIME24	AIME25	AMC23	Minerva	MATH500	Average
Vanilla	5.7	7.6	42.8	15.3	66.6	32.82
Teacher top-K: local loss						
Teacher top-K	1.1	2.0	28.2	12.7	49.6	23.03
Teacher top-K (direct)	4.0	3.6	34.8	12.8	58.5	27.48
Teacher top-K + tail bucket	1.6	2.0	24.5	8.5	44.0	19.71
Student-tail: gates and carrier						
Student-tail	7.1	5.8	41.9	17.6	69.7	33.93
Student-tail + repetition gate	6.2	9.7	42.7	19.1	71.4	35.29
Student-tail + HQ gate	7.5	6.2	42.2	15.6	70.5	33.77
Student-tail + state gate	9.2	5.4	43.4	17.8	70.9	34.86
Student-tail + exact carrier	7.5	8.3	44.1	18.4	70.9	35.31
Quantile budgeting: allocation scope						
Fixed budget	5.4	7.1	43.8	15.2	67.6	33.20
Per-sequence quantile budgeting	4.2	9.2	42.5	18.1	69.3	34.16
Quantile budgeting	7.9	10.0	45.0	18.1	69.9	35.49

The direct teacher top-K variant reaches an average accuracy of 27.48, compared with 23.03 for the sampled variant in Table 9. Sharing a support label does not make two implementations estimates of the same update: Eq. (73) requires its sampling and full-support conditions, whereas restricted supports and renormalization can alter the signal. Tail gates, exact-carrier controls, and union losses test interacting choices; they do not establish that increasing support or adding a gate is beneficial by default. Mean support size alone also does not measure teacher-forward cost.

Probability-anchor ablations. Table 10 reports benchmark accuracy for rank and membership rules with and without a probability anchor. The anchored variants retain a numerical teacher target alongside the coarser signal.

Table 10: Math benchmark accuracy (%) with and without probability anchors.

Method	AIME24	AIME25	AMC23	Minerva	MATH500	Average
Vanilla	5.7	7.6	42.8	15.3	66.6	32.82
Rank only	1.0	1.5	21.7	11.2	45.8	20.01
Rank + probability anchor	3.4	6.7	35.4	16.5	61.9	29.69
Membership only	3.8	3.3	31.6	10.9	56.3	25.57
Membership + probability anchor	3.7	5.0	33.8	12.8	59.1	27.51

Rank-only and Membership-only supervision reach average accuracies of 20.01 and 25.57, while their anchored variants reach 29.69 and 27.51. Both anchored variants remain below Vanilla's 32.82. The anchor supplies probability magnitudes that ordering or set membership omits. These results support

retaining numerical teacher feedback, while the remaining accuracy gap favors using probabilities directly in the distillation loss.

Clipping-threshold ablation. Table 11 reports benchmark accuracy after reducing the hard-clipping threshold from $b = 10$ to $b = 2.303$.

Table 11: Math benchmark accuracy (%) under different clipping thresholds.

Method	AIME24	AIME25	AMC23	Minerva	MATH500	Average
Vanilla	5.7	7.6	42.8	15.3	66.6	32.82
Hard clipping ($b = 10$)	8.3	8.3	41.9	15.9	69.6	33.93
Hard clipping ($b = 2.303$)	2.6	2.6	32.3	11.9	54.0	25.19

The reported average accuracies are 33.93 for $b = 10$ and 25.19 for $b = 2.303$. Reducing b maps a wider range of log-ratios to the same positive or negative cap, discarding differences in their magnitudes. Clipping should limit extreme corrections while preserving moderate ones; we retain $b = 10$ as the less restrictive default.

Jensen–Shannon divergence ablations. Table 12 reports benchmark accuracy under alternative JSD mixture weights.

Table 12: Math benchmark accuracy (%) under JSD mixture-weight ablations.

Method	AIME24	AIME25	AMC23	Minerva	MATH500	Average
Vanilla	5.7	7.6	42.8	15.3	66.6	32.82
JSD ($\beta = 0.1$)	1.3	1.6	21.5	8.8	38.6	17.58
JSD ($\beta = 0.5$)	7.0	8.2	43.0	15.4	69.3	33.83
JSD ($\beta = 0.9$)	2.2	2.8	28.3	9.8	48.2	22.19

The reported average accuracies are 17.58 for $\beta = 0.1$, 33.83 for $\beta = 0.5$, and 22.19 for $\beta = 0.9$. Changing β alters both the mixture distribution and the gradient scale: the unnormalized JSD loss vanishes as β approaches either endpoint. The mixture weight therefore controls update strength as well as the balance between teacher and student probabilities. A practical starting point is the symmetric mixture, $\beta = 0.5$, with the mixture weight and loss scale tuned jointly.

B.5 Training dynamics

Figures 6–16 present accuracy, response length, policy entropy, and gradient norm for all evaluated methods, each compared with Vanilla.

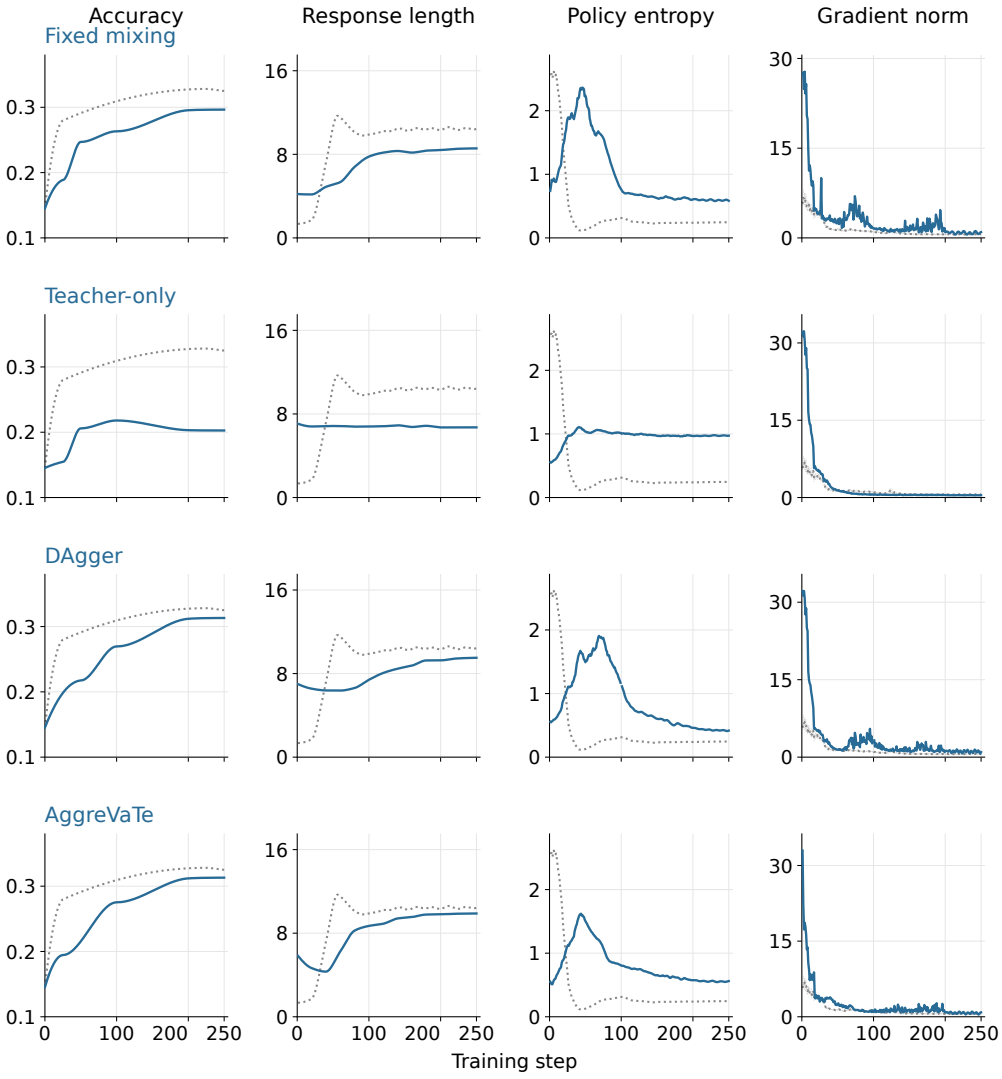

Figure 6: Training dynamics against Vanilla for rollout source, with response length in thousands of tokens, policy entropy in nats, and gradient norm in dimensionless units.

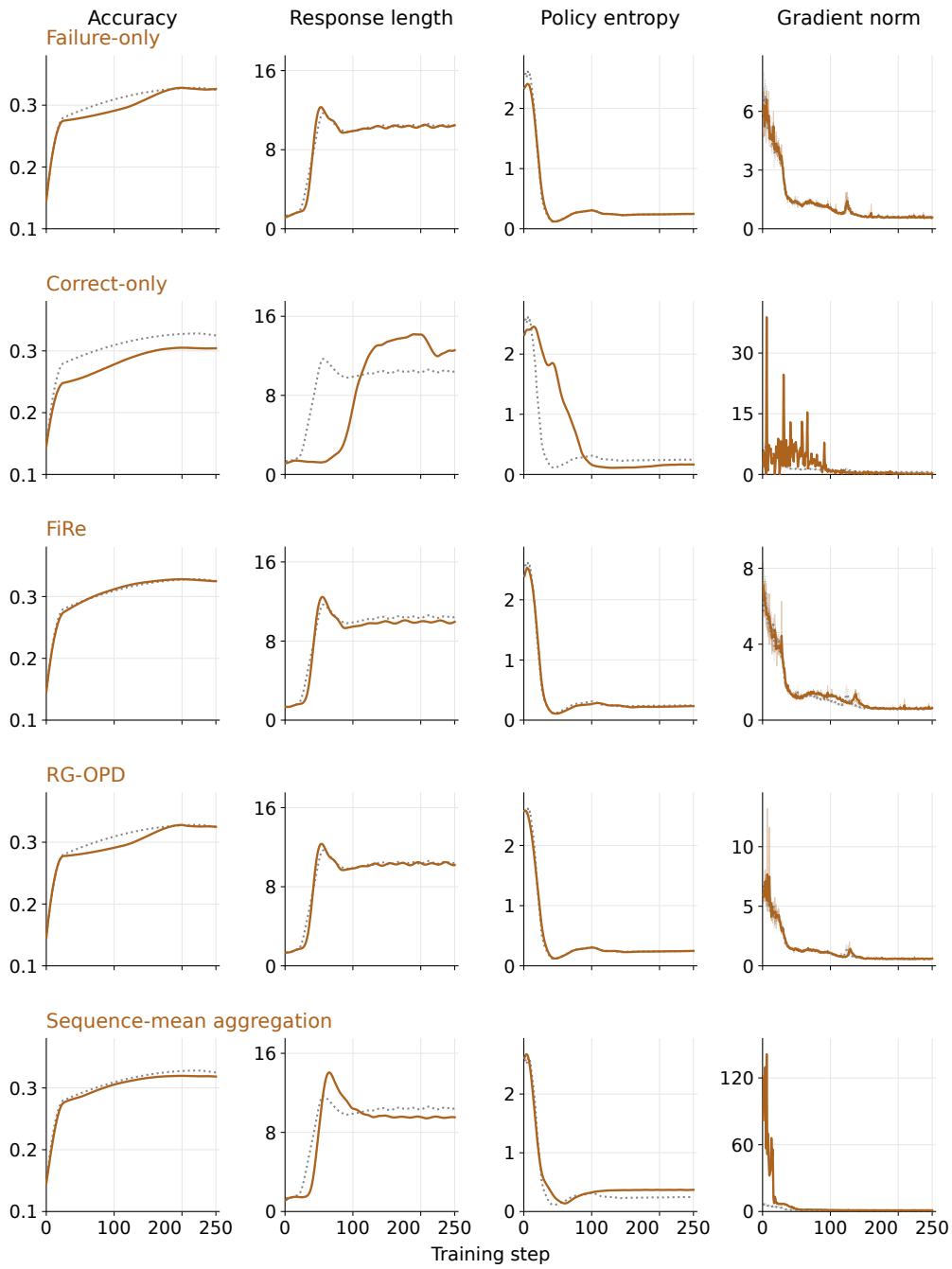

Figure 7: Training dynamics against Vanilla for trajectory selection, with response length in thousands of tokens, policy entropy in nats, and gradient norm in dimensionless units.

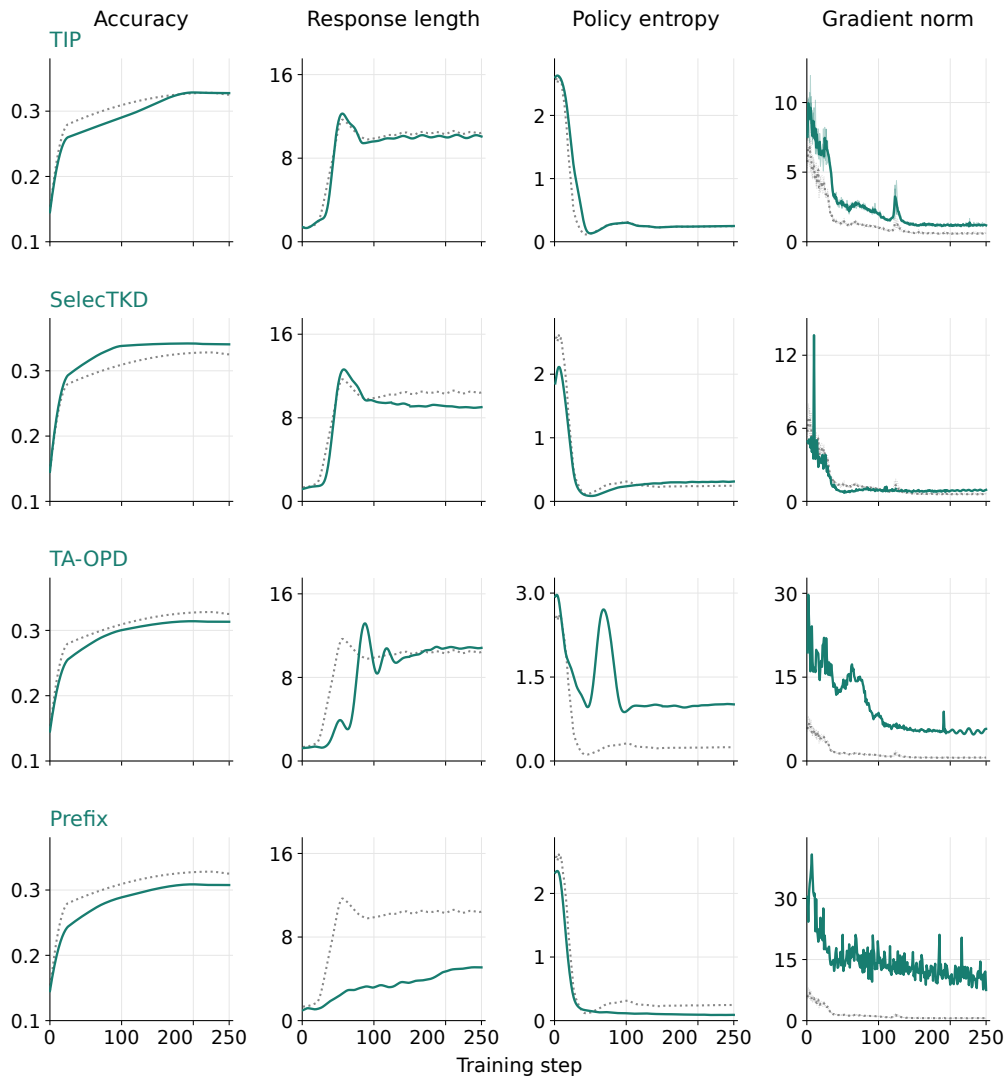

Figure 8: Training dynamics against Vanilla for token weighting (part 1), with response length in thousands of tokens, policy entropy in nats, and gradient norm in dimensionless units.

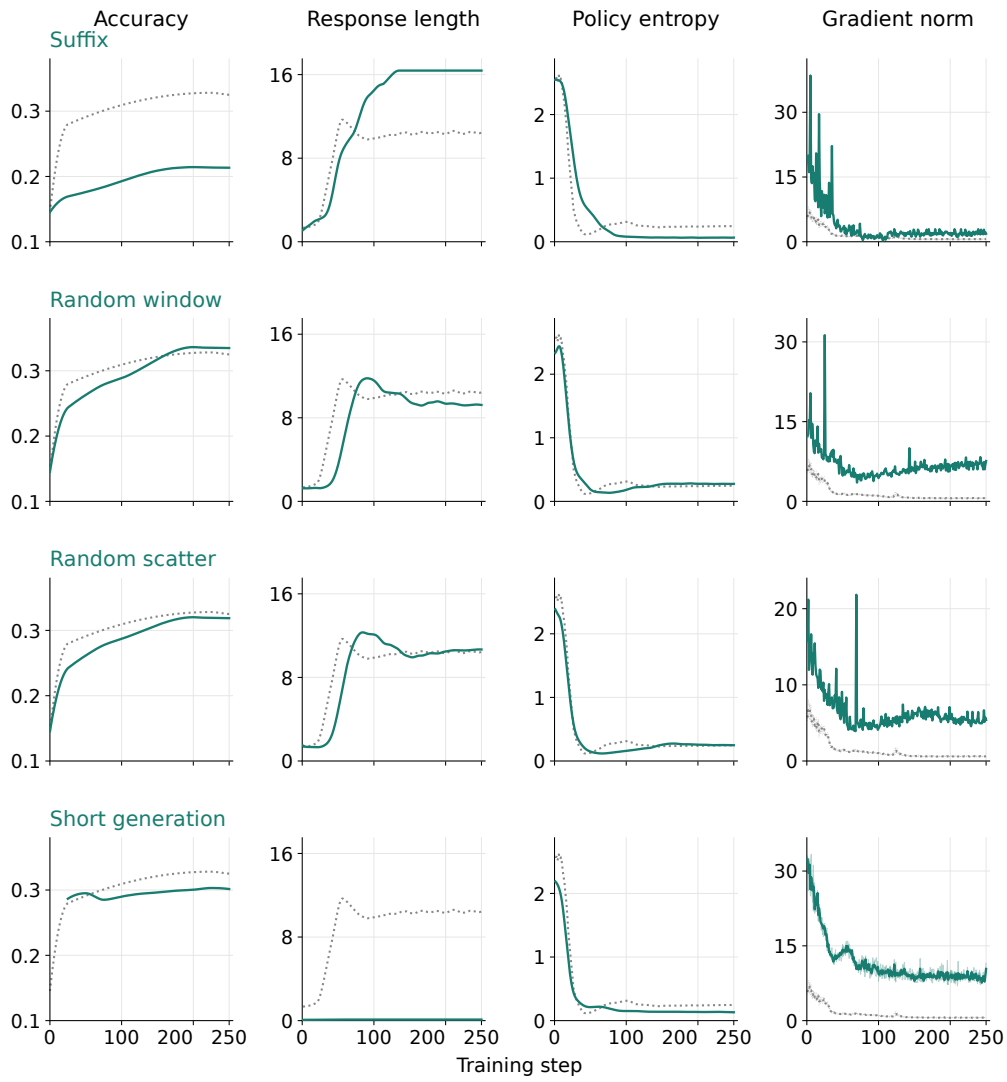

Figure 9: Training dynamics against Vanilla for token weighting (part 2), with response length in thousands of tokens, policy entropy in nats, and gradient norm in dimensionless units.

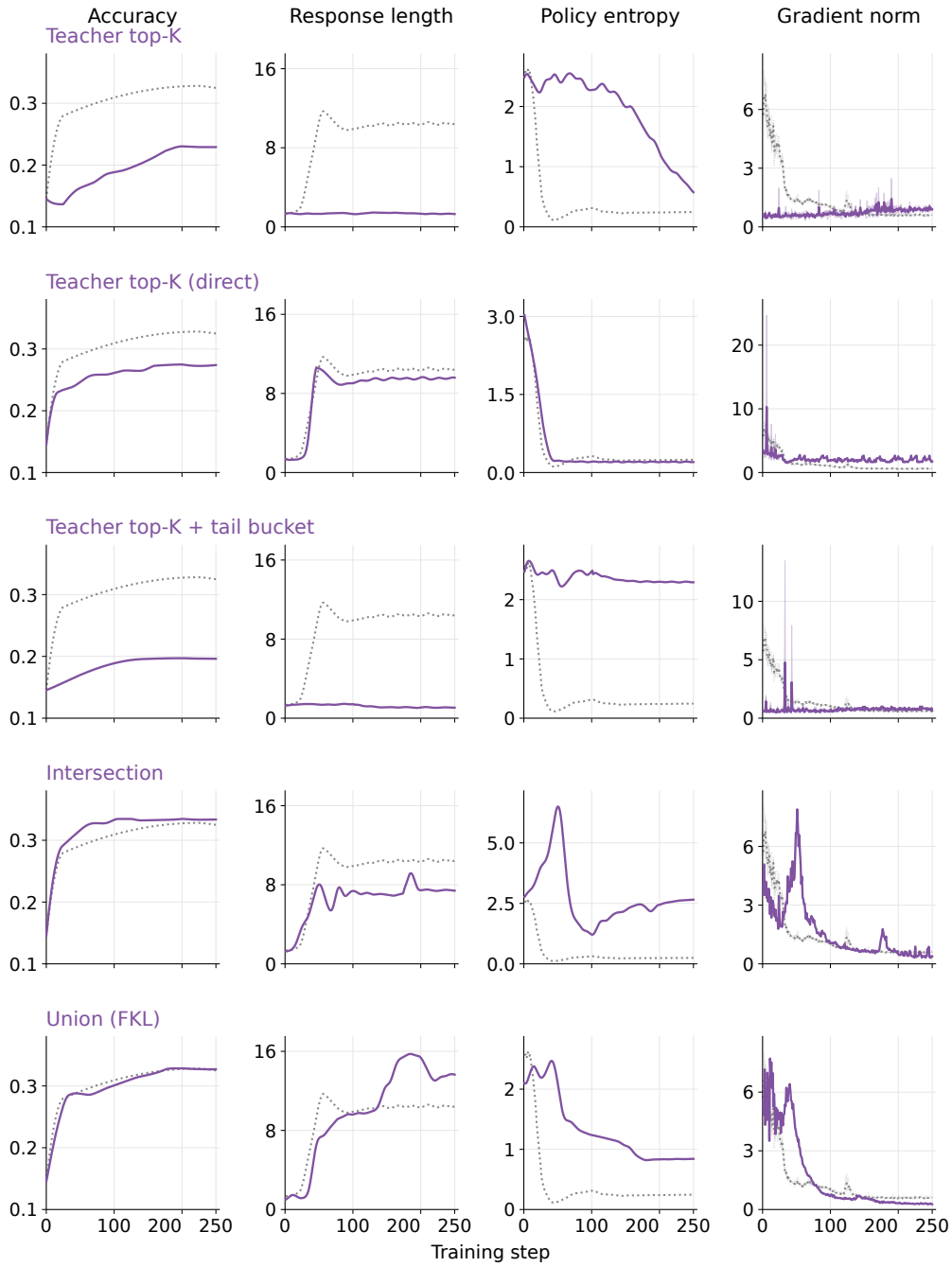

Figure 10: Training dynamics against Vanilla for vocabulary support (part 1), with response length in thousands of tokens, policy entropy in nats, and gradient norm in dimensionless units.

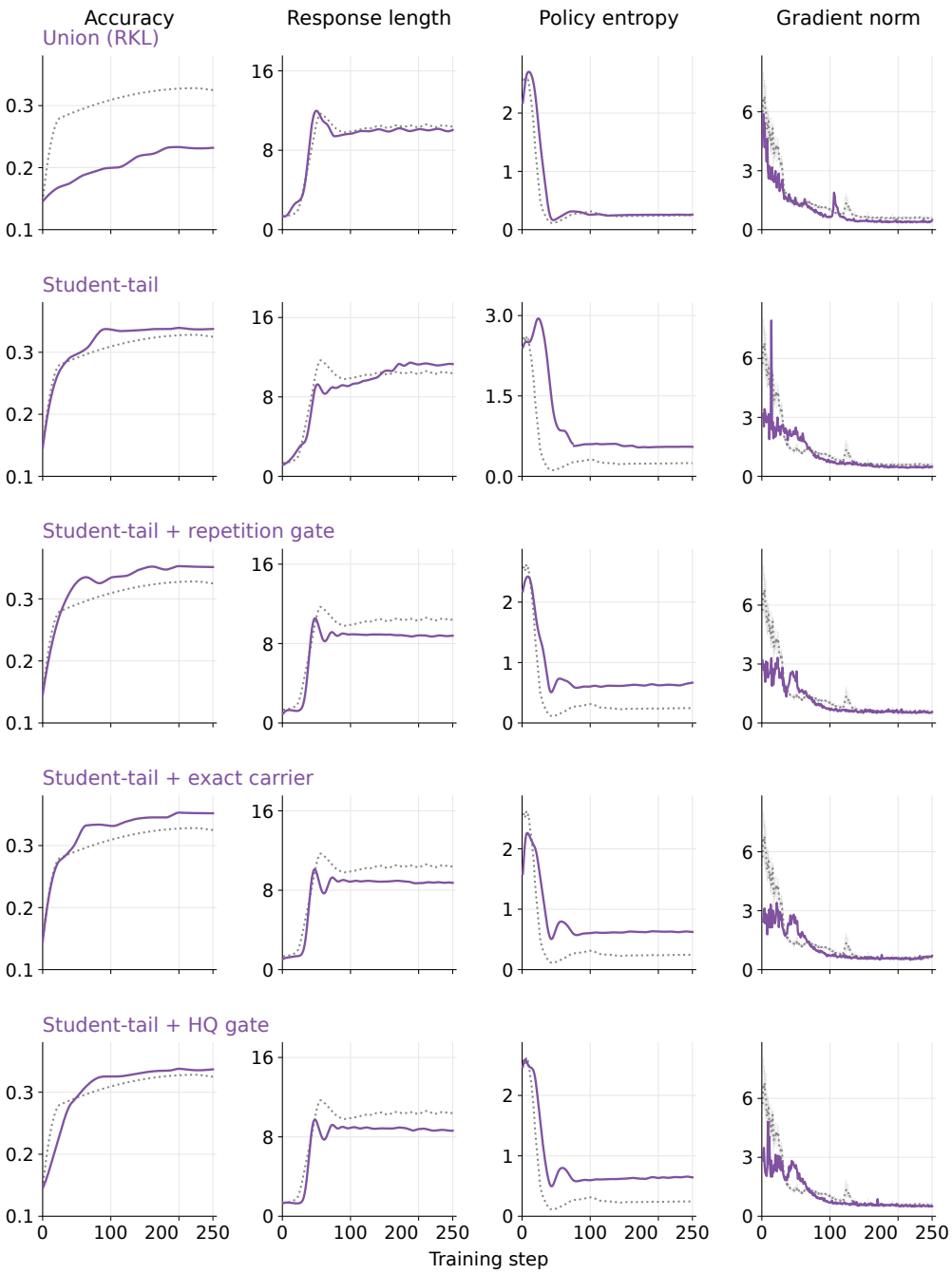

Figure 11: Training dynamics against Vanilla for vocabulary support (part 2), with response length in thousands of tokens, policy entropy in nats, and gradient norm in dimensionless units.

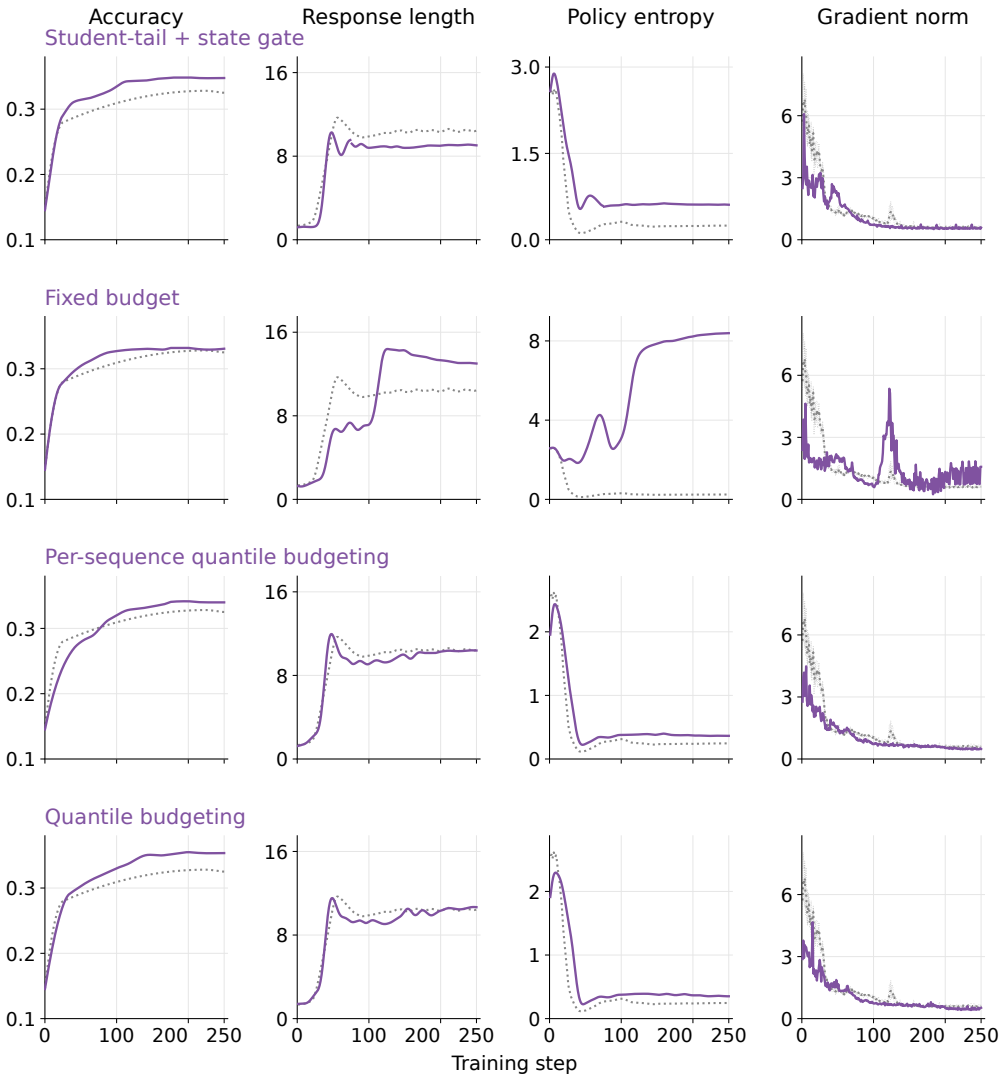

Figure 12: Training dynamics against Vanilla for vocabulary support (part 3), with response length in thousands of tokens, policy entropy in nats, and gradient norm in dimensionless units.

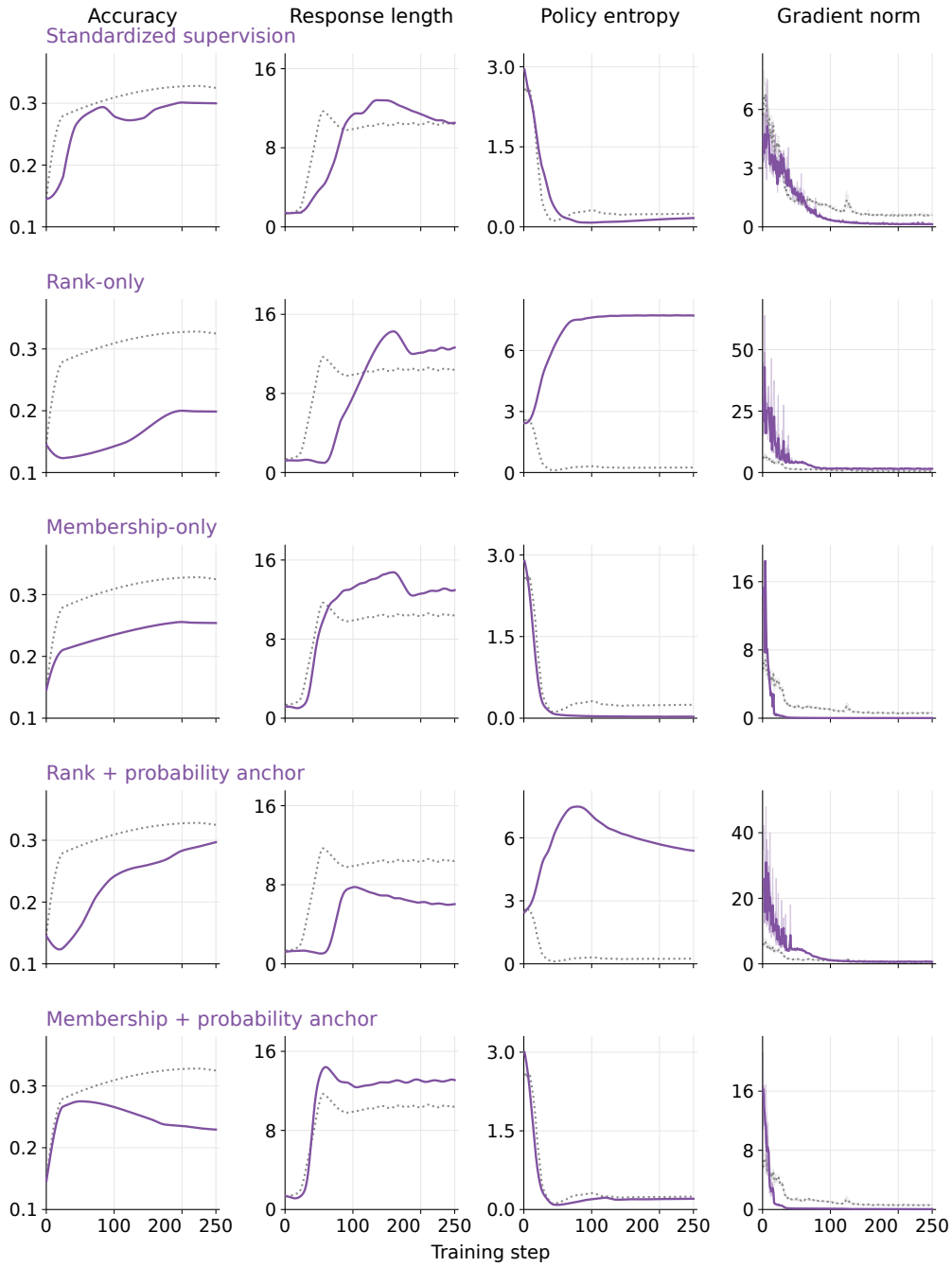

Figure 13: Training dynamics against Vanilla for teacher information, with response length in thousands of tokens, policy entropy in nats, and gradient norm in dimensionless units.

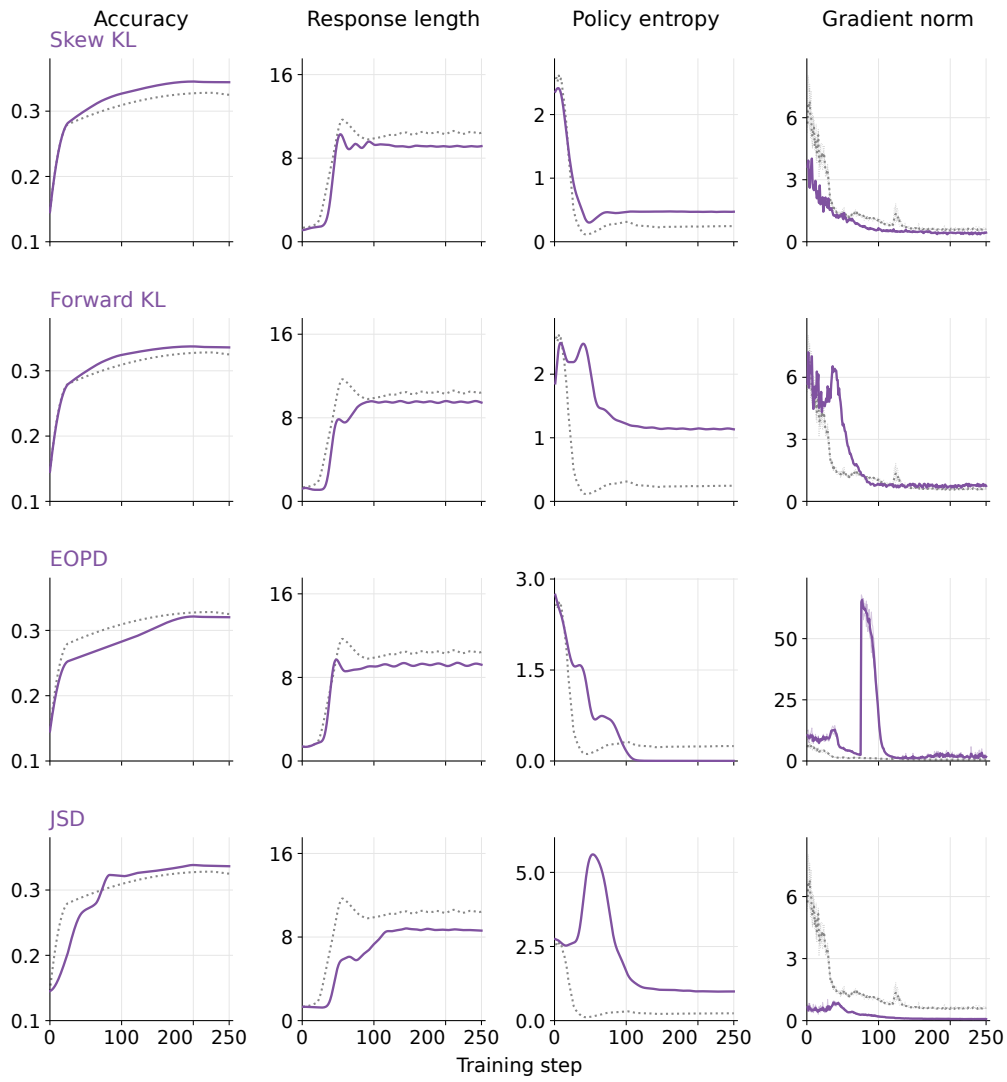

Figure 14: Training dynamics against Vanilla for distillation losses (part 1), with response length in thousands of tokens, policy entropy in nats, and gradient norm in dimensionless units.

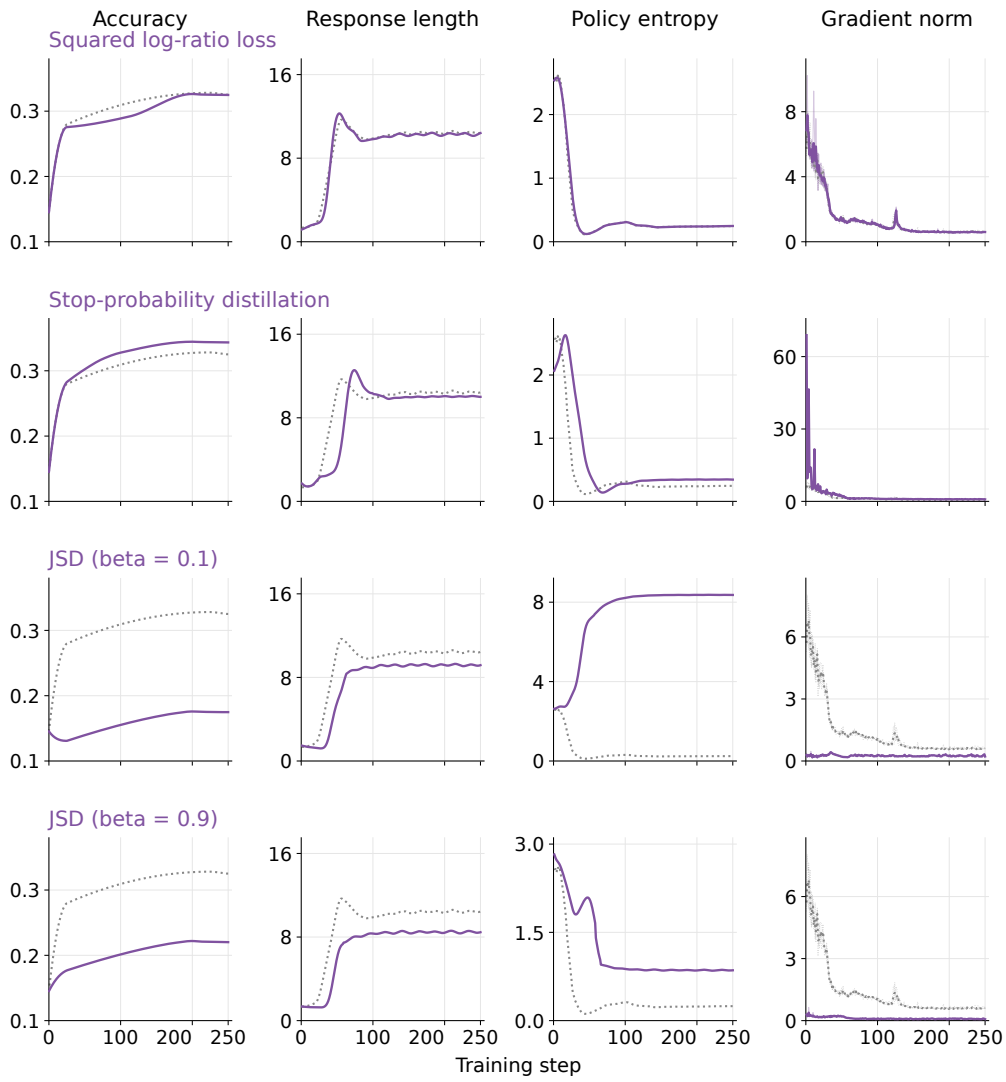

Figure 15: Training dynamics against Vanilla for distillation losses (part 2), with response length in thousands of tokens, policy entropy in nats, and gradient norm in dimensionless units.

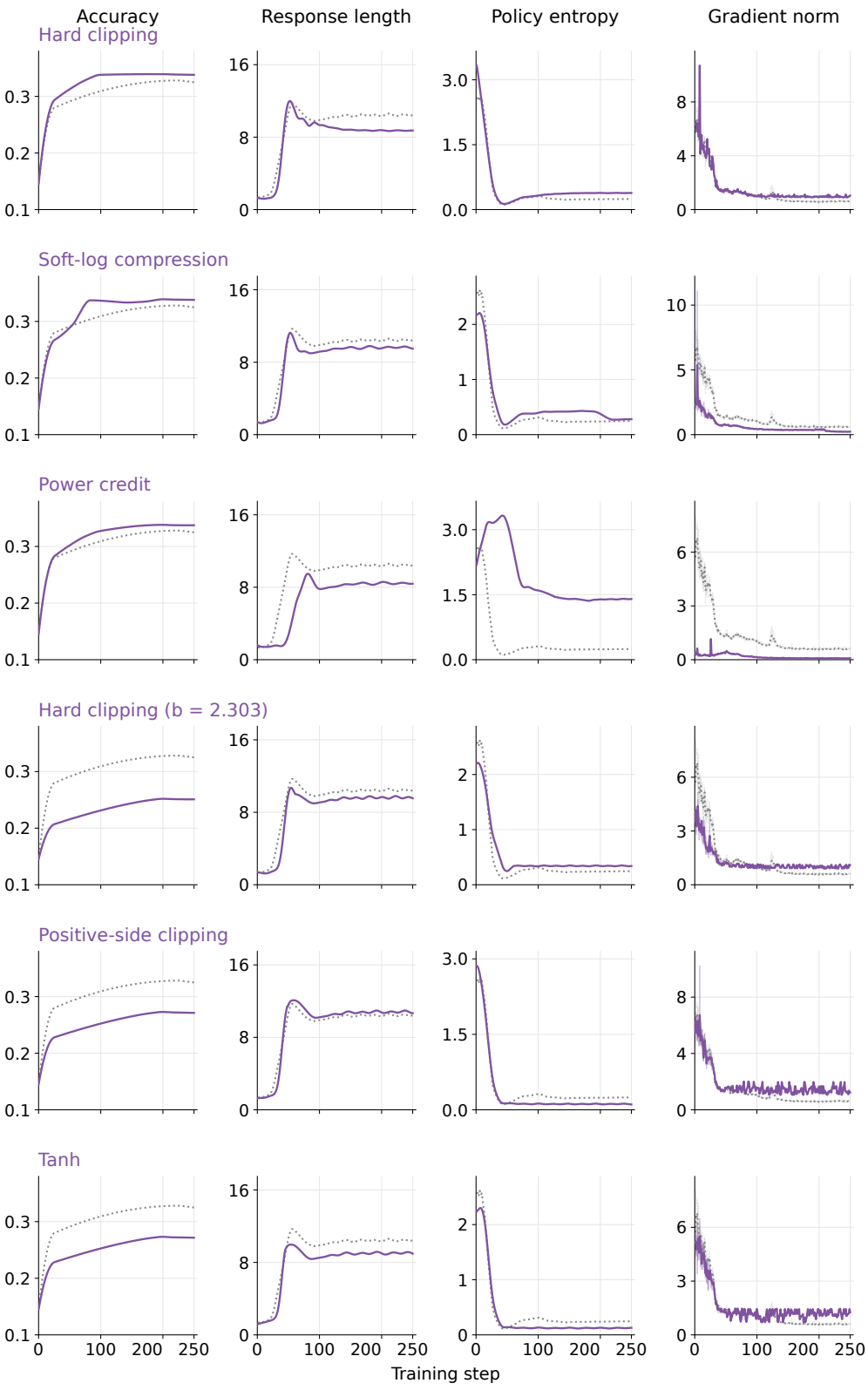

Figure 16: Training dynamics against Vanilla for update strength, with response length in thousands of tokens, policy entropy in nats, and gradient norm in dimensionless units.

C Supporting Derivations for Local Supervision

This appendix connects the sampled-token objective in Section 3.2 to the reverse-KL formulation in Section 3.1, then explains how vocabulary-support normalization changes the local comparison. These two results clarify the loss and support choices instantiated in Appendix A.5. Throughout, collected prefixes, teacher predictions, and selection decisions are fixed during differentiation.

C.1 Sampled supervision and reverse KL

Fix a prefix s_t and write $p_t = p_{\theta_k,t}$. Assume positive student and teacher probabilities on the finite vocabulary. Let $f_t(v) = \nabla_\theta \log p_{\theta,t}(v)|_{\theta_k}$ be the score vector. For a detached credit $c_{k,t}$, differentiating the sampled loss in Eq. (63) gives the local descent direction

$$-\nabla_\theta \ell^c_{k,t}(\theta)\big|_{\theta_k} = c_{k,t} f_t(y_t). \tag{72}$$

Thus the credit determines the sign and magnitude of the sampled-token update. The log-ratio credit uses $c_{k,t} = a_{k,t} = \log[q_t(y_t)/p_t(y_t)]$, while the transformations in Appendix A.5.4 change this scale.

To relate the log-ratio update to direct reverse KL, note that normalization implies $\sum_v p_t(v) f_t(v) = \nabla_\theta \sum_v p_{\theta,t}(v)|_{\theta_k} = 0$. Differentiating the KL and applying this identity yields

$$\begin{aligned} -\nabla_\theta D_{\mathrm{KL}}(p_{\theta,t}\|q_t)\big|_{\theta_k} &= -\sum_v p_t(v)\left[\log\frac{p_t(v)}{q_t(v)} + 1\right] f_t(v) \\ &= \sum_v p_t(v) \log\frac{q_t(v)}{p_t(v)} f_t(v) \\ &= \mathbb{E}_{y_t\sim p_t}[a_{k,t} f_t(y_t) \mid s_t]\,. \end{aligned} \tag{73}$$

The sampled update therefore has the same conditional expectation as the direct reverse-KL descent direction at θ_k. The equality concerns gradients at the collecting policy. It does not identify the loss values or subsequent updates, and sampling introduces variance.

The sampling law and credit are essential to the identity. A weight determined by the prefix can multiply both sides, but a gate depending on the sampled token or completed response generally cannot be moved outside the conditional expectation. Teacher-generated tokens, nonlinear credit transformations, and changes to the comparison support therefore require a separate calculation. In particular, the identity does not equate every sampled and direct method that uses the same candidate set.

C.2 Normalization on restricted vocabulary supports

The support comparisons in Section 5.4 vary both the available candidates and how their probabilities enter the loss. Consider a fixed, nonempty support S_t with positive probability mass under both models. Conditional matching uses the normalized distributions in Eq. (38). Since $\log \tilde{p}_{\theta,t}(v) = \log p_{\theta,t}(v) - \log p_{\theta,t}(S_t)$, its score is

$$\tilde{f}_t(v) = \nabla_\theta \log \tilde{p}_{\theta,t}(v)\big|_{\theta_k} = f_t(v) - \sum_{u \in S_t} \tilde{p}_t(u) f_t(u), \qquad v \in S_t. \tag{74}$$

The second term differentiates the student normalizer. Applying the reverse-KL calculation above to these conditional distributions, or directly differentiating conditional cross-entropy, gives

$$\begin{aligned} -\nabla_\theta D_{\mathrm{KL}}(\tilde{p}_{\theta,t} \| \tilde{q}_t)\big|_{\theta_k} &= \sum_{v \in S_t} \tilde{p}_t(v) \log \frac{\tilde{q}_t(v)}{\tilde{p}_t(v)} \tilde{f}_t(v), \\ -\nabla_\theta D_{\mathrm{KL}}(\tilde{q}_t \| \tilde{p}_{\theta,t})\big|_{\theta_k} &= \sum_{v \in S_t} \tilde{q}_t(v) \tilde{f}_t(v). \end{aligned} \tag{75}$$

These losses compare relative probabilities within S_t. They are unchanged if the total probability assigned to that set changes while its conditional distribution stays fixed. For a singleton support, both conditional distributions equal one, so the loss and its gradient vanish. Simply dropping terms outside S_t from a full-vocabulary KL does not perform this conditional normalization.

The tail-bucket objective retains additional information about the total mass of the selected set. Write $P_\theta = p_{\theta,t}(S_t)$ and $Q = q_t(S_t)$, with $0 < P_\theta, Q < 1$. Expanding $p_{\theta,t}(v) = P_\theta \tilde{p}_{\theta,t}(v)$ and $q_t(v) = Q\tilde{q}_t(v)$ in Eq. (40) gives

$$\ell_{k,t}^{\mathrm{bucket},S_t}(\theta) = D_{\mathrm{KL}}(\mathrm{Bern}(P_\theta) \| \mathrm{Bern}(Q)) + P_\theta D_{\mathrm{KL}}(\tilde{p}_{\theta,t} \| \tilde{q}_t), \tag{76}$$

where $\mathrm{Bern}(P)$ assigns probabilities P and $1 - P$ to membership in the support and its complement. The first term supervises that mass allocation, while the second compares candidates within the support. Both P_θ and $\tilde{p}_{\theta,t}$ are differentiated.

A sampled loss constructed from the same support instead follows Eq. (72) with its chosen credit. It need not vanish on a singleton and need not equal the derivative of either conditional matching or the tail-bucket objective. These distinctions explain why specifying a vocabulary support alone is insufficient to identify the supervision used in the experiments.