EXTREMELY SPARSE SUPERVISION INCENTIVIZES REASONING ABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models demonstrate increasingly strong reasoning capabilities through effective post-training. Yet, prevailing post-training methods optimize over massive numbers of tokens, implicitly assuming that effective learning must be token-intensive. We revisit this assumption in the on-policy distillation setting, which naturally admits dense teacher supervision at every generated token. Using the Qwen3 family, **we discover a counter-intuitive phenomenon: reasoning can be effectively incentivized by an extremely small fraction of generated tokens—as few as one or two tokens per reasoning trajectory, corresponding to only 0.05% of all tokens**. Surprisingly, this sparse supervision in most cases matches or surpasses full-token training in improving reasoning ability, despite excluding the vast majority of generated tokens from the training objective. This phenomenon is consistently observed across nine teacher–student configurations spanning different model scales on mathematical reasoning tasks, and is further validated on coding reasoning, Llama models and Proximal Policy Optimization-based reinforcement learning with verifiable reward. Interestingly, such extremely sparse supervision may be closer to the natural learning process: rather than correcting every step word by word, one reflects on a few critical reasoning steps, updates one's prior understanding, and continues the trial-and-error, avoiding micro-level corrections while remaining remarkably effective. Overall, our results challenge the assumption that effective post-training must be token-intensive and point to a new direction for understanding and designing more efficient post-training algorithms.

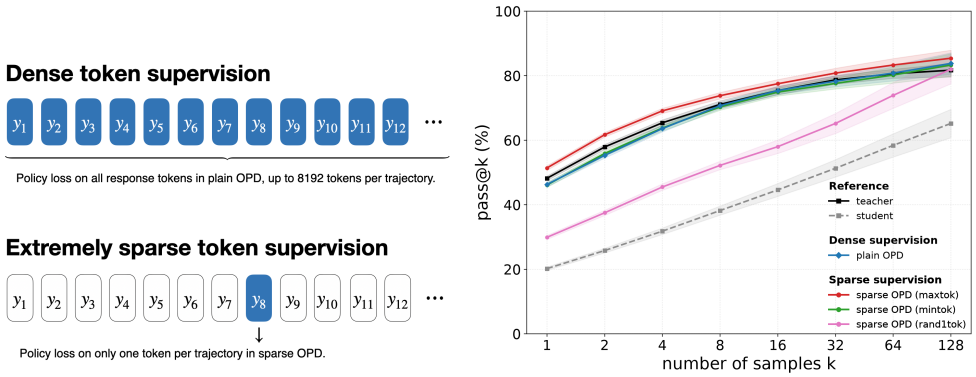


Figure 1: **Illustration of the mechanism and effectiveness of sparse OPD.** Left: sparse OPD supervises only one token per trajectory. Right: AIME 2025 results for Qwen3-8B (student) ← Qwen3-4B-Instruct-2507 (teacher), comparing the base student, teacher, plain OPD, and three sparse OPD variants: *maxtok*, *mintok*, and *rand1tok*. *maxtok* and *mintok* select the tokens with the most positive and negative supervision signal, respectively, while *rand1tok* randomly selects one token per trajectory. Shaded areas denote 95% confidence intervals. Remarkably, *maxtok* produces a student that outperforms both plain OPD and the teacher, *mintok* matches plain OPD, and even *rand1tok* substantially improves over the base student. **All three sparse OPD variants only use one token per trajectory.**

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.

1 INTRODUCTION

Post-training has become a critical stage for aligning large language models (LLMs) with human preferences (Ouyang et al., 2022; Bai et al., 2022) and eliciting their reasoning capabilities (Lightman et al., 2024; Shao et al., 2024; Guo et al., 2025; Team et al., 2026). Several seminal post-training paradigms have emerged, including reinforcement learning from human feedback (RLHF), reinforcement learning from verifiable rewards (RLVR), and on-policy distillation (OPD), etc. Similar to pre-training, which scales model capability through increasing compute and dataset size, collecting hundreds of trillions of tokens in total pretraining data (Kaplan et al., 2020; Hoffmann et al., 2022), post-training has also been traditionally regarded as a token-intensive process. For example, in the DAPO training procedure (Yu et al., 2026) on the Qwen2.5-32B Base model (Team, 2025), approximately 1,500 RL updates are performed, with each update collecting 8,192 rollouts of up to 16,384 tokens per trajectory. This corresponds to a total of hundreds of billions of generated tokens contributing gradient signals during RL optimization to improve the model's reasoning capability. OPD improves the training efficiency of RL-based post-training by leveraging dense token-level supervision. From an information-theoretic perspective, Lu & Thinking Machines Lab (2025) argues that reinforcement learning provides only $O(1)$ bits of learning signal per episode (Schulman & Thinking Machines Lab, 2025), while distillation provides $O(T)$ bits per episode, where T denotes the sequence length. Consequently, OPD can substantially improve training efficiency over RLVR approaches by extracting richer learning signals from each generated trajectory. Nevertheless, OPD still operates in a token-intensive regime, retaining dense supervision over hundreds of millions of generated tokens during training.

Recently, a computationally efficient variant of OPD, called sampled-token OPD, has attracted substantial attention due to its effectiveness in fine-tuning both small-scale models (Lu & Thinking Machines Lab, 2025) and large-scale frontier models (Yang et al., 2025; Team et al., 2026). It improves the student model by receiving token-level supervision from the teacher model, yielding a different learning paradigm from classic RLVR algorithms, such as Group Relative Policy Optimization (GRPO; Shao et al. (2024)) and Decoupled Clip and Dynamic sAmpling Policy Optimization (DAPO; Yu et al. (2026)). In particular, GRPO and DAPO fine-tune models based on external reward signals for domains with automatically verifiable outcomes, including mathematics (Zhang & Math-AI., 2024; Yu et al., 2026) and coding (Chen et al., 2021). Their optimization objective has a clear interpretation: policy updates reinforce reasoning trajectories that lead to correct solutions while suppressing those that do not, thereby directly improving the model's reasoning capability. Moreover, the advantage assigned to each token admits a natural credit assignment interpretation, quantifying its contribution to the final outcome. In stark contrast, sampled-token OPD assigns each token a supervision signal that reflects teacher preference, rather than quantifying its contribution to the correctness of the final answer. This token-level dense supervision has been considered one of the key advantages of OPD (Lu & Thinking Machines Lab, 2025; Li et al., 2026; Xiao et al., 2026), but also makes it difficult to precisely understand how OPD improves the student's reasoning capability. To better understand the mechanism of OPD, we systematically sparsify its token-level supervision and examine how OPD performs under increasingly sparse supervision. Our empirical results discover a counter-intuitive phenomenon: supervising only a few thousand tokens can induce substantial parameter changes and significant improvements in reasoning capability, challenging the conventional assumption that dense token-level supervision is necessary for reasoning improvement.

We summarize our main findings and contributions as follows:

- Using Qwen3 series models on math reasoning benchmarks as a testbed, across nine teacher-student combinations spanning large-scale teacher/small-scale student, same-scale teacher-student, and small-scale teacher/large-scale student settings, we consistently observe that supervising only one randomly selected token per trajectory consistently improves the reasoning capacity of the student model. Further, selectively supervising only one or two tokens with extreme teacher-student probability difference can match and in most cases outperform standard OPD with dense signal.
- The extremely sparse supervision also provides a useful lens for investigating and understanding OPD by isolating different components of the token-level learning signal and examine their individual contributions. First, we reveal a non-monotonic relationship between the amount of supervision and reasoning performance: as supervision becomes increasingly sparse, performance initially degrades but can recover and even surpass dense OPD under

extreme sparsity. Second, an expressive student, as in the same-scale teacher–student and large-scale student/small-scale teacher settings, benefits more from tokens with positive supervision signal, whereas a less expressive student in the large-scale teacher/small-scale student setting benefits more from tokens with negative supervision signal.

- Although standard OPD is formulated as minimizing the teacher–student reverse KL divergence, the sparse OPD variants that achieve the best reasoning performance do not necessarily reduce this divergence. In fact, we observe that they can even increase the teacher–student reverse KL divergence, suggesting that the reasoning improvement induced by OPD cannot be explained solely by making the student distribution closer to the teacher distribution. Finally, we further validate the extremely sparse supervision phenomenon beyond the Qwen3–OPD–Math setting through cross-task (coding reasoning), cross-family (Llama series), and cross-algorithm (PPO-based RLVR) experiments.

More broadly, our results suggest that post-training does not necessarily need to be token-intensive to substantially improve reasoning capability. A surprisingly small amount of token-level supervision can produce improvements comparable to, and sometimes larger than, dense supervision. This finding opens a new direction for understanding and designing post-training, motivating a shift toward more token- and memory-efficient training paradigms.

2 PRELIMINARY

In this section, we introduce the formulation of on-policy distillation and provide the necessary background and motivation for its use. Denote a LLM with parameters θ by π_θ. Given a prompt $\boldsymbol{x}$, the model generates a response sequence $\boldsymbol{y} = (y_1, \ldots, y_T)$ according to $\boldsymbol{y} \sim \pi_\theta(\cdot|\boldsymbol{x})$. Specifically, the sequence is generated autoregressively, where at each step t, conditioned on the prompt $\boldsymbol{x}$ and the prefix $\boldsymbol{y}_{1:t-1}$, the next token is sampled as $y_t \sim \pi_\theta(\cdot|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$. Denote $|\boldsymbol{y}|$ as the length of $\boldsymbol{y}$ and the vocabulary of the tokenizer as $\mathcal{V}$. For any two distributions p and q over $\mathcal{V}$, define the Kullback–Leibler (KL) divergence between p and q as $\mathrm{KL}(p||q) = \sum_{y\in\mathcal{V}} p(y)\log(p(y)/q(y))$.

2.1 SUPERVISED FINE-TUNING AS SAMPLED-TOKEN OFF-POLICY DISTILLATION

Knowledge distillation (KD, Hinton et al. (2015)) transfers knowledge from a teacher model π^{teacher} to a student model π_θ by minimizing the divergence between their output distributions. For autoregressive language models, KD minimizes the token-level full-vocabulary *forward* KL divergence $\mathcal{L}_{\mathrm{KD}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{y}\sim\pi^{\text{teacher}}(\cdot|\boldsymbol{x})}\big[\sum_{t=1}^{|\boldsymbol{y}|} D_{\mathrm{KL}}\big(\pi^{\text{teacher}}(\cdot|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})\|\pi_\theta(\cdot|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})\big)\big]$, where $\mathcal{D}$ is a dataset collected by the teacher model π^{teacher}, thus KD is also known as the off-policy distillation. Replacing the full-vocabulary KL divergence in $\mathcal{L}_{\mathrm{KD}}(\theta)$ by an unbiased estimator using teacher-sampled tokens, $\log \pi^{\text{teacher}}(y_t|\boldsymbol{x}, \boldsymbol{y}_{1:t-1}) - \log \pi_\theta(y_t|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$, and removing the first term that is independent of θ, we obtain the classic supervised fine-tuning (SFT, Wei et al. (2021)) objective: $\mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{y}\sim\pi^{\text{teacher}}(\cdot|\boldsymbol{x})}\big[\sum_{t=1}^{|\boldsymbol{y}|}\log \pi_\theta(y_t|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})\big]$. SFT can therefore be viewed as off-policy distillation with sampled tokens, which bypasses the computation burden of obtaining the full-vocabulary teacher distribution at each token.

2.2 SAMPLED-TOKEN ON-POLICY DISTILLATION

In forward KL distillation, the student model is optimized on prefixes sampled from the teacher distribution, and not on those induced by the student model. During inference, the student may generate prefixes that are never observed during training and thus fail to reason reliably. This is the classic compounding error issue in sequence generation tasks (Ross et al., 2011; Bengio et al., 2015) caused by the distribution mismatch between training and inference. On-policy distillation (OPD, Agarwal et al. (2024); Gu et al. (2024); Yang et al. (2025)) addresses this issue by sampling trajectories from the student model π_θ, allowing the teacher to provide supervision signals on prefixes that are actually visited by the student. Specifically, OPD minimizes the following *reverse* KL objective function: $\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\big[\sum_{t=1}^{|\boldsymbol{y}|} D_{\mathrm{KL}}\big(\pi_\theta(\cdot|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})\|\pi^{\text{teacher}}(\cdot|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})\big)\big]$, where the tokens are generated on-policy by the student model, i.e., $\boldsymbol{y} \sim \pi_\theta(\cdot|\boldsymbol{x})$. An efficient and prevailing variant of OPD proposed by Lu & Thinking Machines Lab (2025) replaces the full-vocabulary inverse KL divergence at each step t by an unbiased estimator, $\log \pi_\theta(y_t|\boldsymbol{x}, \boldsymbol{y}_{1:t-1}) -$

$\log \pi^{\text{teacher}}(y_t|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$, using the student-sampled token y_t. The objective function is

$$\mathcal{L}(\theta) = -\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}, \boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\left[\sum_{t=1}^{|\boldsymbol{y}|} r_t(\boldsymbol{x}, \boldsymbol{y}_{1:t-1})\right], \tag{2.1}$$

where $r_t(\boldsymbol{x}, \boldsymbol{y}_{1:t-1}) = \log \pi^{\text{teacher}}(y_t|\boldsymbol{x}, \boldsymbol{y}_{1:t-1}) - \log \pi_\theta(y_t|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$ is the teacher supervision at each token. *From a theoretical perspective, the optimal policy that optimizes the OPD objective (2.1) is exactly the teacher model $\pi^{teacher}$.* Thus the teacher supervision signal r_t guides the student to improve by imitating the teacher model: a positive (negative) supervision signal indicates the teacher assigns a higher (lower) probability to the sampled token y_t than the student, thereby encouraging the student to increase (decrease) the likelihood of generating that token. We refer to the sampled-token OPD simply as OPD in the rest of paper unless otherwise specified. OPD has recently been adopted in the post-training of frontier large language models (Xiao et al., 2026; Team et al., 2026), where it proves remarkably effective.

3 Extremely Sparse Supervision Incentivizes Reasoning Ability

Token-level dense supervision is widely regarded as a key advantage of OPD (Lu & Thinking Machines Lab, 2025; Li et al., 2026; Xiao et al., 2026). Recent studies have demonstrated the effectiveness of OPD and its variants on the Qwen3 family (Yang et al., 2025; Lu & Thinking Machines Lab, 2025; Li et al., 2026; Yang et al., 2026; Xu et al., 2026), making it a natural testbed for asking *whether dense supervision is indeed essential*. We systematically sparsify token-level supervision and uncover a surprising phenomenon: even extremely sparse supervision can effectively improve the reasoning capability of the student.

Randomly Supervising One Token per Trajectory. As a warm-up, consider an extreme case where only one randomly selected token from each trajectory is supervised and contributes to the gradient calculation and parameter update. Define the token-level advantage $A_t = \log \pi^{\text{teacher}}(y_t|\boldsymbol{x}, \boldsymbol{y}_{1:t-1}) - \log \pi_\theta(y_t|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$, and the PPO-style ϵ-clipped per-token loss $\ell_t(\theta) = -\min(w_t(\theta)A_t, \text{clip}(w_t(\theta), 1-\epsilon, 1+\epsilon)A_t)$, where $w_t(\theta) = \pi_\theta(y_t|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})/\pi_{\theta_{\text{old}}}(y_t|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$. The OPD objective is the token-mean loss over all response tokens, $\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}, \boldsymbol{y}\sim\pi_{\theta_{\text{old}}}(\cdot|\boldsymbol{x})}\left[\frac{1}{|\boldsymbol{y}|}\sum_{t=1}^{|\boldsymbol{y}|}\ell_t(\theta)\right]$. We insert a mask m_t and get

$$\mathcal{L}_{\text{SOPD}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}, \boldsymbol{y}\sim\pi_{\theta_{\text{old}}}(\cdot|\boldsymbol{x})}\left[\frac{1}{|\boldsymbol{y}|}\sum_{t=1}^{|\boldsymbol{y}|}\ell_t(\theta)\cdot m_t\right]. \tag{3.1}$$

We denote any algorithm minimizing (3.1) with sparse masks as the **sparse OPD**. Different types of masks induce various sparse OPD variants. As a warm-up, we randomly select *one* token t from each response and set $m_t = 1, m_{t'} = 0$ for any $t' \neq t$. We denote this sparse OPD variant as *rand1tok*.

Models and Datasets. For the teacher models, we first train the Qwen3-4B-Base model with GRPO for one and five epochs, obtaining **Qwen3-4B-GRPO-1ep** and **Qwen3-4B-GRPO-5ep**, respectively. We additionally include **Qwen3-4B-Instruct-2507** and **Qwen3-30B-A3B-Instruct-2507** as two off-the-shelf teacher models. For the student models, we consider **Qwen3-1.7B-Base**, **Qwen3-4B-Base**, **Qwen3-1.7B**, **Qwen3-4B**, and **Qwen3-8B**. Unless otherwise specified, both teacher and student models operate in *no-think* mode. Using these models, we construct nine teacher–student families (Table 1) spanning three representative regimes: *large-scale teacher/small-scale student*, *same-scale teacher and student*, and *small-scale teacher/large-scale student*. These regimes cover varying degrees of teacher–student capability/representation gaps. For both GRPO and OPD training, we use DAPO-Math-17K (Yu et al., 2026) dataset. Our setup follows a standard math reasoning RLVR and OPD setting. We provide the detailed training configuration in Appendix F, Table 27.

Evaluation Metrics and Benchmarks. We are interested in measuring the **reasoning ability boundary** of the student models fine-tuned by the OPD and sparse OPD. Following the evaluation protocol in the literature (Chen et al., 2026; Zhu et al., 2026), we adopt the unbiased low-variance estimator of pass@k proposed by Chen et al. (2021), which measures whether the model can produce a correct answer within k independent attempts. Specifically, for each prompt $\boldsymbol{x}_i$ in an evaluation dataset $\mathcal{D}$, we generate n responses $\{\boldsymbol{y}_j\}_{j=1}^n$ and count the number of correct responses as c_i. Then the unbiased estimator of pass@k on dataset $\mathcal{D}$ is defined as $\text{pass@}k = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|} 1 - \binom{n-c_i}{k}/\binom{n}{k}$.

Table 1: The nine Qwen3 OPD families: student ← teacher configurations.

Family	Student	Teacher	Regime	Results
1	Qwen3-1.7B-Base	Qwen3-4B-GRPO-1ep	large teacher/small student	Table 6, Figure 4
2	Qwen3-1.7B-Base	Qwen3-4B-GRPO-5ep	large teacher/small student	Table 7, Figure 5
3	Qwen3-4B-Base	Qwen3-4B-GRPO-1ep	same-scale teacher student	Table 8, Figure 6
4	Qwen3-4B-Base	Qwen3-4B-GRPO-5ep	same-scale teacher student	Table 9, Figure 7
5	Qwen3-1.7B	Qwen3-4B-Instruct-2507	large teacher/small student	Table 10, Figure 8
6	Qwen3-8B	Qwen3-30B-A3B-Instruct-2507	large teacher/small student	Table 11, Figure 9
7	Qwen3-4B	Qwen3-30B-A3B-Instruct-2507	large teacher/small student	Table 12, Figure 10
8	Qwen3-1.7B	Qwen3-30B-A3B-Instruct-2507	large teacher/small student	Table 13, Figure 11
9	Qwen3-8B	Qwen3-4B-Instruct-2507	small teacher/large student	Table 14, Figure 12

We calculate a spectrum of pass@k with $k \in \{1, 2, 4, 8, \cdots, 128\}$ using $n = 256$. For each pass@k estimator, we also provide the 95% confidence interval (CI) to account for the randomness induced by finite (n=256) sampling. In particular, $\mathrm{Var}(\text{pass@}k) = \frac{1}{|\mathcal{D}|^2}\sum_{i=1}^{|\mathcal{D}|}\mathrm{Var}(g_k(c_i, n))$, where $g_k(c_i, n) = 1 - \binom{n-c_i}{k}/\binom{n}{k}$. Denote the probability of generating a correct response for $\boldsymbol{x}_i$ as p_i, we have $c_i \sim \mathrm{Binomial}(n, p_i)$. Replacing the unknown p_i by $\hat{p}_i = c_i/n$, we have $\widehat{\mathrm{Var}}(g_k(c_i, n)) = \sum_{j=0}^{n} g_k^2(j, n) \cdot \mathrm{Binomial}(j; n, \hat{p}_i) - \left(\sum_{j=0}^{n} g_k(j, n) \cdot \mathrm{Binomial}(j; n, \hat{p}_i)\right)^2$, where $\mathrm{Binomial}(j; n, \hat{p}_i) = \binom{n}{j}\hat{p}_i^j(1 - \hat{p}_i)^{n-j}$. Inserting the variance estimator to $\mathrm{Var}(\text{pass@}k)$, we have $\hat{\sigma} = \sqrt{\frac{1}{|\mathcal{D}|^2}\sum_{i=1}^{|\mathcal{D}|}\widehat{\mathrm{Var}}(g_k(c_i, n))}$ and the CI is calculated as $\text{pass@}k \pm 1.96 \cdot \hat{\sigma}$. We report pass@$k$ on AIME 24 (Zhang & Math-AI., 2024) and AIME 25 (Zhang & Math-AI., 2025).

We also adopt avg@k to measure the **efficiency in sampling correct responses**. We set $k = 8$: for each prompt $\boldsymbol{x}_i$, the model generates eight responses, among which c_i are correct. Then $\text{avg@8} = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|} c_i/8$. We report avg@8 on three math benchmarks, AIME 24, AIME 25 and HMMT-Feb 25 (Balunovic et al., 2026), as well as the mean avg@8 over the three benchmarks.

Experiment Results: *rand1tok* Effectively Enables Reasoning Improvement. We compare the performance of student model fine-tuned by *rand1tok* with (1) the base student model; (2) the student model fine-tuned with plain OPD, as well as (3) the teacher model. We select Family 8 (Figure 2 and Table 2), Family 4 and 9 (Figure 3, Table 4 and Table 3) as representative cases of the large-scale teacher/small-scale student, same-scale teacher–student and small-scale teacher/large-scale student settings, respectively. The complete results across all nine families are provided in Appendix D. Recall that *rand1tok* discards almost all of the token-level supervision used by plain OPD: a trajectory containing thousands of tokens provides supervision through only a single randomly-selected token. One might expect OPD to completely fail to transfer the teacher's capabilities under such an extreme reduction in supervision. However, the pass@k results reveal the opposite: ***rand1tok* consistently improves the reasoning capability of base students across all nine teacher–student families**. Moreover, in Families 1–3, *rand1tok* surpasses plain OPD in the large-k regime of pass@$k = 128$, suggesting that extreme sparsification can sometimes lead to broader reasoning exploration. The avg@8 results further demonstrate that ***rand1tok* consistently improves sampling efficiency of base students**. The consistent improvement across both pass@k and avg@8 metrics, as well as nine diverse configurations, indicates this phenomenon is not an artifact of evaluation variance.

4 Extreme Sparse Supervision Can Outperform Dense OPD

The surprising effectiveness of *rand1tok* motivates us to further investigate the extremely sparse supervision in OPD. Random token selection provides an approximately unbiased estimator (up to scale) of the dense OPD objective, suggesting that *rand1tok* still optimizes the same underlying teacher–student alignment objective, albeit with substantially higher variance. However, the success of random sparse supervision raises several deeper questions: *Is dense token-level supervision necessary for improving reasoning ability? Does the reasoning ability improvement of the student model arise purely from better mimicking the teacher model? How different token-level signals play distinct roles in shaping the student's reasoning capability?* In this section, we investigate sparse

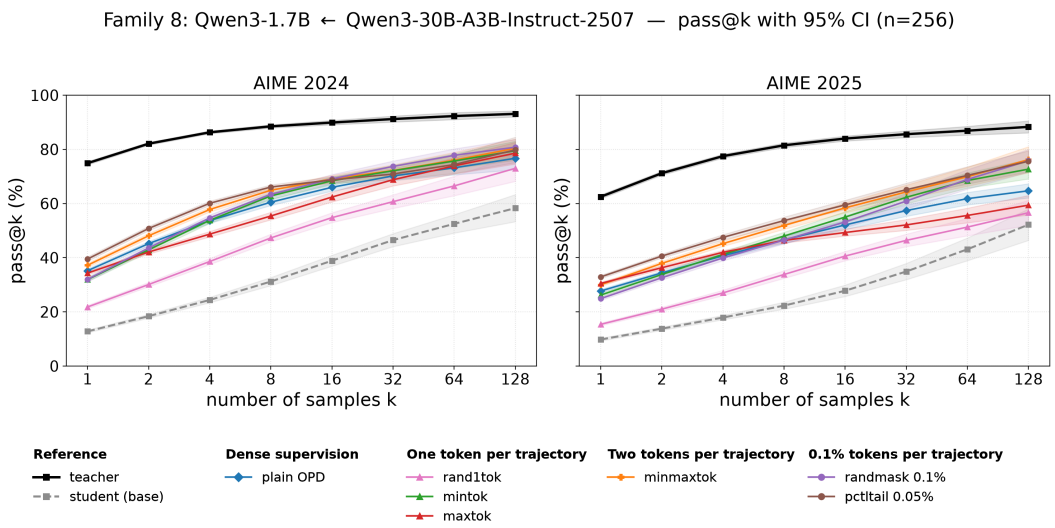


Figure 2: Family 8 pass@k curves across AIME 24 and 25. Shaded area is 95% confidence interval.

OPD variants that selectively supervise tokens with extreme teacher–student probability differences, and provide empirical answers to the above questions.

4.1 Tokens with Extreme Supervision Signal

We focus on the tokens with the most positive and most negative supervision signal, as well as their combination within each trajectory, and investigate whether selecting tokens with extreme supervision signal provides more effective sparse supervision than random token selection. Specifically, if the token mask in (2.1) is defined as $m_t = \mathbb{1}\left\{t = \arg\max_i r_i\right\}$ then only the token with the maximum supervision signal contributes to the gradient update. We refer to this sparse OPD variant as *maxtok*. Such tokens typically receive large positive supervision signal (e.g., $r_t > 3$), indicating that the teacher assigns substantially higher probability to the sampled token than the student. Empirically, these tokens are usually associated with high student entropy.

Conversely, if the token mask in (2.1) is defined as $m_t = \mathbb{1}\left\{t = \arg\min_i r_i\right\}$, then only the token with the minimum supervision signal contributes to the gradient update. We refer to this variant as *mintok*. Signal on these tokens is typically extremely negative (e.g., $r_t < -16$), indicating that the teacher assigns exponentially lower probability to the sampled token than the student. Unlike *maxtok*, the student entropy on these tokens can be either high or low. Next we combine the two selection strategies by retaining both the maximum and minimum tokens in each trajectory: $m_t = \mathbb{1}\left\{t = \arg\max_i r_i \ \vee \ t = \arg\min_i r_i\right\}$. We refer to this sparse OPD variant as *minmaxtok*.

Finally, for ablations we slightly increase the supervision budget from one token per trajectory to 0.1% of the generated tokens, corresponding to approximately 3–4 supervised tokens per trajectory. Specifically, the token mask m_t in (2.1) is independently sampled from a Bernoulli distribution with success probability $p = 0.001$, i.e., $P(m_t = 1) = 0.001$ and $P(m_t = 0) = 0.999$. We refer to this sparse OPD variant as *randmask 0.1%*. Moreover, let $q_{0.05\%}$ and $q_{99.95\%}$ denote the 0.05th and 99.95th percentiles of the supervision signal within a trajectory. We define $m_t = \mathbb{1}\left\{r_t \le q_{0.05\%} \ \vee \ r_t \ge q_{99.95\%}\right\}$ and refer to this sparse OPD variant as *pctltail 0.05%*.

4.2 Experiment Results

We introduce an additional evaluation metric in the following analysis. Standard OPD aims to minimize the reverse KL divergence between student and teacher. We are interested in how different variants of sparse OPD change the reverse KL. Given a model π_θ, we let π_θ generate one response for each prompt $\boldsymbol{x}_i$ in a dataset $\mathcal{D}$ and calculate the full-vocabulary reverse

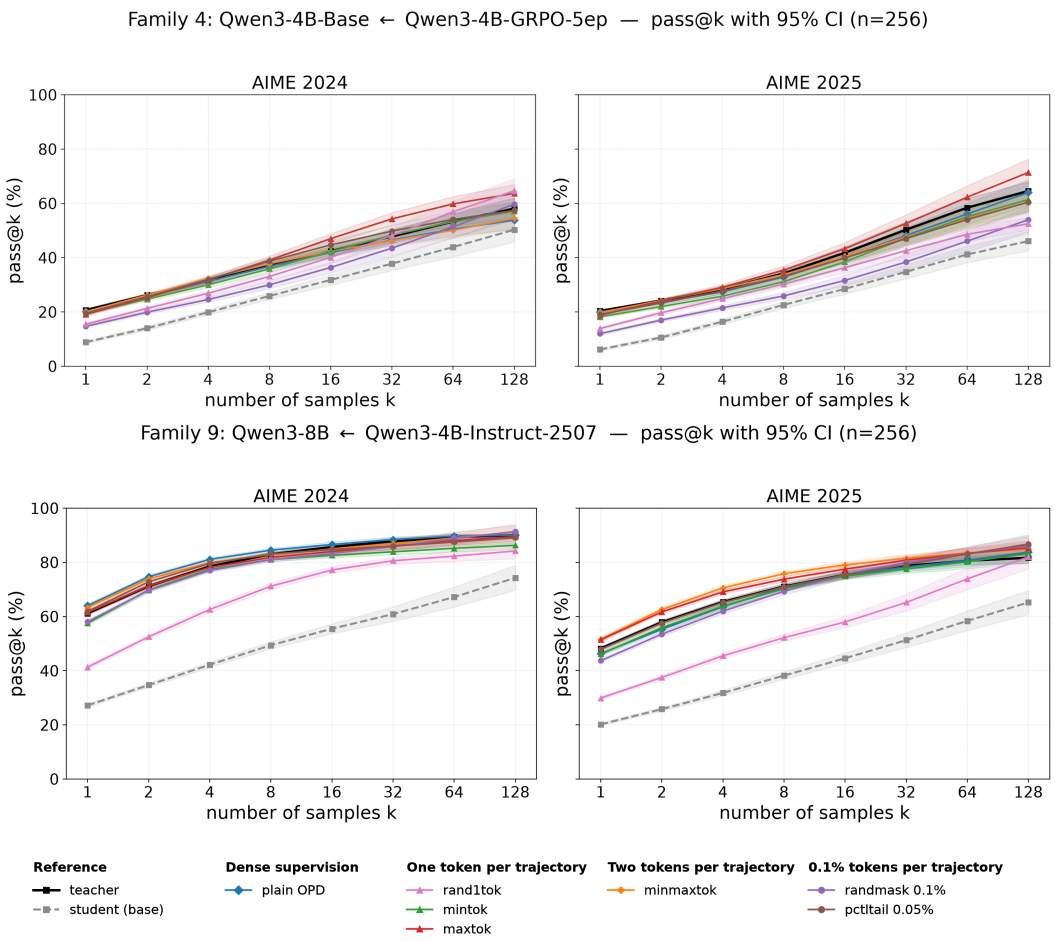


Figure 3: Family 4 and Family 9 pass@k curves across AIME 24 and 25. Shaded area is the 95% confidence interval.

KL as $\mathrm{revKL}(\pi_\theta, \pi^{\text{teacher}}) = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\sum_{t=1}^{|\boldsymbol{y}_i|}\mathrm{KL}(\pi_\theta(\cdot|\boldsymbol{x}_i, \boldsymbol{y}_{i,1:t-1})||\pi^{\text{teacher}}(\cdot|\boldsymbol{x}_i, \boldsymbol{y}_{i,1:t-1}))$. We set $\mathcal{D} = \{\text{AIME 24}, \text{AIME 25}, \text{HMMT-Feb 25}\}$.

Supervising One or Two Tokens per Trajectory Is Sufficient to Match or Outperform Plain OPD. Selected results are shown in Figures 2 and 3 and Tables 2 to 4, and complete results are postponed to Appendix D. The key takeaway is, across all nine teacher–student families, we can always identify a sparse OPD variant that matches (two out of nine families) or outperforms (seven out of nine families) plain OPD in terms of both reasoning capability boundary and sampling efficiency. In particular, *minmaxtok* and *pctltail 0.05%* are consistently among the strongest variants, and the best-performing sparse OPD variant is typically one of the two. Sparse OPD variants sometimes produce students that outperform their teachers, an effect that is particularly pronounced in Family 9, where the student has substantially greater representation capacity than the teacher. Moreover, *maxtok* performs particularly well in the same-scale teacher–student setting (Families 3 and 4) and the small-scale teacher/large-scale student setting (Family 9), achieving strong improvements in both pass@k and avg@8. In contrast, *mintok* performs consistently well across all nine families. Although it falls slightly short of plain OPD in a few cases, it remains highly competitive despite supervising only one token per trajectory. We postpone a detailed discussion on the effect of extreme-signal tokens in terms of student-teacher distribution gap and learning dynamics to Appendix B.

Sparse Supervision Does Not Simply Improve Teacher Imitation. According to the reverse KL divergence reported in Tables 2 to 4, sparse OPD variants do not necessarily reduce the teacher–student reverse KL divergence. On the contrary, many of the best-performing sparse OPD variants exhibit a

Table 2: Family 8: Qwen3-1.7B ← Qwen3-30B-A3B-Instruct-2507. avg@8 scores and mean averages over AIME24/AIME25/HMMT. *keep_frac (%)*: fraction of response tokens supervised among all generated tokens; *revKL*: reverse KL divergence between the student and teacher; *freeze (%)*: fraction of student network unchanged parameters after training.

	avg@8 (%)						
Method	AIME24	AIME25	HMMT	Mean	keep_frac (%)	revKL	freeze (%)
Teacher	72.5	62.1	42.5	59.0	–	0.000	–
Student	12.9	8.3	5.0	8.7	–	0.368	–
plain OPD	37.1	27.9	17.5	27.5	100	0.190	80.2
one token supervision per trajectory							
rand1tok	21.7	15.8	9.2	15.6	0.0379	0.299	95.0
mintok	32.9	30.0	17.5	26.8	0.0254	0.260	91.4
maxtok	35.8	32.1	19.2	29.0	0.0176	0.750	89.5
two tokens supervision per trajectory							
minmaxtok	40.4	29.6	16.7	28.9	0.0409	0.374	91.3
0.1% tokens supervision per trajectory							
randmask 0.1%	37.1	23.3	16.2	25.6	0.0988	0.267	92.6
pctltail 0.05%	38.8	31.2	20.4	**30.1**	0.1380	0.404	89.6

Table 3: Family 9: Qwen3-8B ← Qwen3-4B-Instruct-2507. avg@8 scores and mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
Method	AIME24	AIME25	HMMT	Mean	keep_frac (%)	revKL	freeze (%)
Teacher	62.1	48.3	30.0	46.8	–	0.000	–
Student	26.2	20.8	10.8	19.3	–	0.268	–
plain OPD	66.7	44.6	31.2	47.5	100	0.184	78.2
one token supervision per trajectory							
rand1tok	42.1	30.0	17.5	29.9	0.0392	0.253	95.5
mintok	55.4	47.1	25.8	42.8	0.0259	0.213	87.3
maxtok	63.8	50.8	29.6	48.1	0.0167	1.116	90.7
two tokens supervision per trajectory							
minmaxtok	62.9	55.0	30.0	**49.3**	0.0472	0.319	88.1
0.1% tokens supervision per trajectory							
randmask 0.1%	59.6	45.4	28.3	44.4	0.0999	0.279	94.1
pctltail 0.05%	62.1	47.5	28.8	46.1	0.120	0.254	86.1

larger reverse KL divergence. This is particularly striking for Families 4 and 9, where *maxtok* substantially increases the reverse KL divergence while producing students that significantly outperform plain OPD and even their teachers. Across the nine teacher–student families, the strongest reasoning performance is generally not achieved by the student with the smallest reverse KL divergence to the teacher. These observations suggest that the reasoning improvement induced by OPD cannot be fully explained by simply making the student distribution closer to the teacher distribution.

Memory Saving and Response Length. A direct benefit of sparse supervision is memory saving: since only a small number of tokens contribute to the gradient calculation, we can discard the full-vocabulary logit information for the majority of masked tokens. Thus, for the sparse OPD variants the logit memory is effectively erased. We defer a detailed discussion to Appendix E. Finally, we compare the average response length across all checkpoints. We observe no clear pattern that sparse OPD variants systematically increase or decrease response length relative to plain OPD, the response length of all variants remains within a reasonable range.

Table 4: Family 4: Qwen3-4B-Base ← Qwen3-4B-GRPO-5ep. avg@8 scores and mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
Method	AIME24	AIME25	HMMT	Mean	keep_frac (%)	revKL	freeze (%)
Teacher	20.8	17.9	8.8	15.8	–	0.000	–
Student	9.6	7.9	0.8	6.1	–	0.441	–
plain OPD	23.3	17.5	7.5	16.1	100	0.005	75.0
one token supervision per trajectory							
rand1tok	12.1	13.8	6.7	10.8	0.0601	0.078	90.2
mintok	19.2	17.5	6.2	14.3	0.0369	0.018	81.9
maxtok	22.9	18.8	7.5	**16.4**	0.0364	0.428	89.5
two tokens supervision per trajectory							
minmaxtok	21.7	19.6	5.0	15.4	0.0622	0.015	81.8
0.1% tokens supervision per trajectory							
randmask 0.1%	16.2	12.5	5.4	11.4	0.1000	0.119	88.6
pctltail 0.05%	19.2	18.3	5.4	14.3	0.1770	0.012	80.9

5 Discussion and Limitations

Qwen3 family provides an ideal testbed for studying post-training, where plain OPD is robust and effective across all nine teacher–student families, and the sparse OPD variants built upon it exhibit similarly stable behavior. To assess whether the phenomenon extends beyond Qwen3, math reasoning and OPD, we conduct **cross-task, cross-model-family and cross-training-paradigm** experiment in Appendix C. In particular, we use Llama-3.1-8B-Instruct as the student and Llama-3.3-70B-Instruct as the teacher. We again identify extremely sparse OPD variants that achieve performance comparable to dense OPD. We further test the phenomenon of extremely sparse supervision on coding reasoning OPD and the PPO-RLVR setting. Experiment results show that this phenomenon broadly holds across different model families, reasoning tasks, and post-training algorithms. Nevertheless, our systematic investigation remains limited in model-family and task diversity. When extending our experiments to other model families, including Gemma 3 (Gemma Team et al., 2025), Gemma 4 (Gemma Team et al., 2026) and Mistral 3 (Liu et al., 2026), we found that directly training with plain sampled-token OPD does not improve the student models. One possible reason is we did not first perform SFT on the student models before applying OPD. Since our study builds upon successful OPD training, we therefore do not include these model families in the present study. Evaluating extremely sparse supervision across a broader range of models, post-training algorithms, and domains remains an interesting direction for future work.

Despite the limitation, the improvement in reasoning capability is substantial and remarkably consistent across the settings considered in this work. We believe this phenomenon provides a useful lens for understanding the role of supervision density in post-training and motivates new algorithm development. An intriguing direction is whether the small number of informative tokens can be identified without access to a white-box teacher. If so, assigning targeted positive or negative supervision signal to these tokens could potentially lead to new post-training methods that require neither a white-box teacher nor an outcome-level reward. A concurrent work (Ding & Zhang, 2026) provides evidence in this direction, showing that much of the benefit of OPD can be recovered by identifying low-probability student tokens and applying a fixed negative advantage, without relying on the teacher's token-level supervision. Extremely sparse supervision may also provide a way to reduce the reliance on shared tokenizer between teacher and student, potentially enabling cross-family distillation. Beyond on-policy training, it is unknown whether similar sparsification can be applied to off-policy training, where large amounts of training data are already available, and under suitable conditions sparsification may enable effective updates to focus on a small subset of informative tokens while largely excluding irrelevant or noisy supervision.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the writing and implementing methods. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, propose or refine hypotheses, designing or providing feedback on research methodology or experiments, assisting with translation, cleaning and reformatting dataset, supporting qualitative and thematic data analysis, interpret results. We have reviewed all AI-assisted work. For example we read everything written and checked the code written by AI. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024. 3, 13

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. 2, 12

Mislav Balunovic, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *Advances in Neural Information Processing Systems*, 38, 2026. 5

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015. 3

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. 2, 4, 15

Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Advances in Neural Information Processing Systems*, 38:57654–57689, 2026. 4

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025. 15

Yi Ding and Ruqi Zhang. Does on-policy distillation really distill? from noisy teacher to self-improvement. *arXiv preprint arXiv:2608.31046*, 2026. 9

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. 9

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026. 9

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 15

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024. 3, 14

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 2, 12

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 2

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025. 15

Eric Jang. A beginner's guide to variational methods: Mean-field approximation. https://blog.evjang.com/2016/08/variational-bayes.html, August 2016. 14

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 2

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*, 2026. 2, 4

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*, volume 2024, pp. 39578–39601, 2024. 2

Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026. 9

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation. 2, 3, 4, 13

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. *Advances in Neural Information Processing Systems*, 38:132119–132138, 2026. 13

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 2, 12

Stéphane 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*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011. 3

John Schulman and Thinking Machines Lab. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/. 2

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 15, 16

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 2, 12, 15

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024. 16

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026. 2, 4, 13

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025. 2

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 38:115452–115486, 2026. 14

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021. 3, 12

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. 15

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026. 2, 4

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. 4, 27

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 2, 3, 4, 13

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026. 4

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026. 2, 4, 12

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024. 2, 5

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025. 5

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. *Advances in Neural Information Processing Systems*, 38:126546–126573, 2026. 4

A Related Work

Post-training has become a central stage for improving the capabilities and behavior of LLMs, playing an important role in both instruction following and reasoning. Supervised fine-tuning (SFT, Wei et al. (2021)) adapts pretrained models using curated demonstrations, while reinforcement learning from human feedback (RLHF) further optimizes model behavior according to human preferences (Ouyang et al., 2022; Bai et al., 2022). Reinforcement learning from verifiable rewards (RLVR) has emerged as an effective approach for eliciting reasoning capabilities on tasks with automatically verifiable outcomes, such as mathematics and coding (Shao et al., 2024; Guo et al., 2025; Yu et al., 2026). In RLVR, models generate responses on-policy and receive rewards based on the correctness

Table 5: A summary of all sparse OPD variants and the number of tokens supervised per trajectory.

Variant	Number of tokens supervised per trajectory
plain OPD	all tokens (up to 8192 tokens per trajectory)
randmask 0.1%	0.1% of tokens (uniformly at random)
pctltail 0.05%	0.1% of tokens (bottom and top 0.05% of tokens by supervision signal)
rand1tok	one token (uniformly at random)
minmaxtok	two tokens (tokens with the highest and lowest supervision signal)
mintok	one token (token with the highest supervision signal)
maxtok	one token (token with the lowest supervision signal)

of the resulting solutions, with policy optimization reinforcing trajectories that lead to desirable outcomes while suppressing unsuccessful ones. More recently, on-policy distillation (OPD) provides a complementary paradigm for post-training. Rather than relying on a scalar outcome reward, OPD generates trajectories from the student model and obtains token-level supervision from a stronger teacher model (Agarwal et al., 2024; Lu & Thinking Machines Lab, 2025). This dense token-level supervision is one of the defining advantages of OPD and has contributed to its recent adoption for reasoning-oriented post-training (Yang et al., 2025; Team et al., 2026). Our work is motivated by this line of studies but is fundamentally orthogonal to them. Rather than identifying which tokens are most informative, we discover a counter-intuitive phenomenon that has not been previously reported in literature: extremely sparse supervision, down to one token per trajectory, can effectively incentivize reasoning ability.

B Ablation Studies and Additional Analysis

Given the effectiveness of *mintok* and *maxtok*, we further sweep the amount of extremely negative and positive tokens used for supervision. Specifically, we set the token mask m_t in (3.1) to $m_t = \mathbb{1}\{A_t < \tau\}$ or $m_t = \mathbb{1}\{A_t > \tau\}$, where τ controls the fraction of the extreme tokens selected for supervision. We denote the resulting sparse OPD variants as *at* $< \tau$ and *at* $> \tau$, respectively. Experiment results are postponed to Appendix D (Figures 13 to 17 and Tables 15 to 23).

A Non-Monotonic Relationship Between Sparsity and Performance. Across our experiments, we frequently observe a non-monotonic relationship between the amount of supervised tokens and the reasoning performance. For example, Tables 19 to 21 show that as the number of supervised tokens decreases from dense supervision, performance initially degrades, but then recovers and can even surpass dense OPD when the supervision becomes extremely sparse. However, when the supervision becomes too sparse, the learning signal eventually becomes insufficient to produce meaningful improvement. This suggests that reasoning performance can peak at an intermediate level of extreme sparsity, rather than varying monotonically with the amount of token-level supervision.

Sparse Subnetwork Update Mukherjee et al. (2026) discover that RLVR training updates only a small subnetwork of LLM comprising just 5%-30% of the parameters. We are interested in whether the standard OPD admits the same phenomenon and whether sparse OPD variants admit sparser subnetwork updates. As a quantification, we report two sparsity metrics: the fraction of activated tokens during the training (keep_frac (%)) and the fraction of parameters updated in the neural network (freeze (%)). Specifically, given a base student model π_{θ^0} and an OPD fine-tuned student model π_θ, the former is calculated as keep_frac% $= \frac{\#\text{activated tokens}}{\#\text{all tokens generated}}$ and the latter is calculated as freeze(%) $= \sum_{j=1}^{N} \mathbb{1}\{|\theta_j^0 - \theta_j| < 1e-5\}/N$, where N is the dimension of θ. The results in Tables 2 to 4 show that standard OPD also updates only a small fraction of the student's parameters, consistent with the sparse-update phenomenon observed in RLVR. Furthermore, sparser token supervision generally results in sparser parameter updates. In general, we can conclude that across all nine teacher–student families, *there exists at least one sparse OPD variant that supervises only* 0.01%–0.1% *of generated tokens and updates only* $\sim$ 10% *of the model parameters, yet matches or even outperforms plain OPD.*

What Are Those Tokens? We have demonstrated that tokens receiving extreme positive or negative supervision signal are particularly effective at incentivizing reasoning ability. It is therefore intriguing to ask what these tokens actually represent. Extremely sparse supervision offers a unique advantage for studying this question: with only two activated tokens per trajectory, we can directly inspect them by eye and characterize the types of decisions that receive these unusually strong learning signals. To this end, we audit the tokens activated by *minmaxtok* among the first 20 steps in the sparse OPD setting with a Qwen3-1.7B student and a Qwen3-4B-Instruct-2507 teacher. We summarize the results in Table 24 and Table 25. We find that tokens with extremely positive supervision signal tend to have clear semantic meaning, whereas tokens with extremely negative supervision signal are predominantly correct-but-teacher-dispreferred tokens, with only a small fraction corresponding to genuine mathematical errors.

Why and How Extreme Token Supervision Works. One motivation for investigating tokens with extreme supervision signal is the mode-seeking property of reverse KL minimization (Jang, 2016; Gu et al., 2024), particularly when the student distribution is less expressive than the teacher, as in the large-scale teacher/small-scale student setting. By penalizing probability mass assigned to regions where the teacher assigns low probability, reverse KL encourages the student to concentrate its probability mass on the teacher's high-probability modes rather than spreading it across multiple suboptimal alternatives. As illustrated in Figure 19, the optimal student distribution q can assign moderately higher density than the target teacher distribution p around the right mode. Thus upon converging, there should be few generated tokens with extremely negative or positive supervision signal, and instead all generated tokens should receive moderate positive or negative supervision signal. This suggests that during training, tokens with extremely negative or positive supervision signal provide strong directional learning signals, while tokens with moderate positive or negative supervision signal could largely be noise and should not contribute to the gradient update or be involved in shaping the student distribution. There is, however, an important caveat for extremely positive supervision signal. When the student's representation capacity is limited, strongly reinforcing a single teacher-preferred mode may cause the student's probability mass to shift excessively toward different modes across training, potentially leading to unstable or conflicting updates. This may explain why *maxtok* does not perform well in the large-scale teacher/small-scale student regime, including Qwen3-1.7B-Base (Figures 4 and 5), Qwen3-1.7B (Figures 2 and 8) and Qwen3-4B (Figure 10), whereas *mintok* exhibits more stable performances.

In contrast, in the same-scale teacher–student and small-scale teacher/large-scale student settings, the student has substantially greater representation capacity and can better approximate the target distribution p, as illustrated on the right of Figure 19. In this regime, tokens with extreme supervision signal still provide particularly strong directional signals for optimization. Because the student has sufficient capacity to accommodate these updates, reinforcing such extreme signals need not force it to choose among incompatible modes. This interpretation is consistent with our results: in Families 3 (Figure 6), 4 and 9 (Figure 3), *maxtok* is consistently among the strongest sparse OPD variants and can even produce students that outperform their teachers. Moreover, we observe distinct effects on the learning dynamics of actor entropy (Figure 18). *mintok* tends to decrease actor entropy, consistent with the fact that supervision on tokens with extremely negative supervision reward suppresses probability mass on behaviors that the teacher strongly disfavors. In contrast, *maxtok* tends to increase actor entropy. Notably, tokens with extremely positive supervision signal are often associated with high student entropy. Such tokens correspond to the "forking" tokens (Wang et al., 2026), which are positions where the student model is uncertain about the current token choice, and the choice of this token strongly determines the subsequent generation path. Reinforcing these tokens therefore encourages the student to allocate probability to behaviors that it currently under-explores.

Hypothesis. The motivation for supervising one or two tokens with extreme supervision signal comes from the natural learning process: a learner first attempts to solve a problem, then reflects on a few consequential reasoning steps, incorporates the resulting feedback into its existing knowledge, and improves through subsequent trial and error. In contrast, standard dense-token supervision on every token may be highly redundant or even harmful. In an on-policy trajectory, later tokens are conditioned on the particular prefix produced by the current student. Some of these prefixes may result from an early mistake and may never occur again after the model parameters are updated. Consequently, applying dense supervision to every subsequent token may correct behaviors conditioned on states that are unlikely to be encountered after the parameter update, making such supervision largely ineffective

or even redundant. In contrast, targeted interventions, such as correcting one particularly important error or reinforcing one useful behavior per response, may be sufficient to shift the student's future behavior towards generating correct answer, especially given student model's substantial pretrained knowledge and existing reasoning capabilities. Moreover, since the teacher model is not perfect, sparse supervision may also reduce the risk of unnecessarily transferring the teacher's mistakes or limitations. Interestingly, such extremely sparse supervision may be closer to the natural learning process: rather than correcting every step word by word, one reflects on a few critical reasoning steps, updates one's prior understanding, and continues the trial-and-error, avoiding micro-level corrections while remaining remarkably effective.

Remark. We have empirically established an intriguing phenomenon: an extremely small amount of token-level supervision can match or even surpass dense supervision in plain OPD, despite discarding the vast majority of the available token-level training signal. The fact this phenomenon emerges under multiple sparse supervision strategies, across diverse teacher–student configurations, calls into question whether the effectiveness of OPD is fundamentally driven by its dense token-level supervision. We emphasize our goal is neither to argue the learning signal in OPD necessarily originates from any particular type of token nor to advocate for a specific token-selection strategy. We also do not claim supervising one single token per trajectory represents the limit of achievable sparsity. For example, our preliminary experiments suggest supervising only the minimum-signal token in *incorrect* trajectories can achieve comparable performances. The sparse OPD still requires the same number of on-policy rollouts as the plain OPD, which dominates the overall computational cost and training time. We leave a systematic investigation of sparser or more targeted supervision schemes and their potential for reducing overall training cost to future work.

C Cross-Task, Cross-Family and Cross-Algorithm Validation

In this section, we investigate whether the phenomenon of extremely sparse supervision for incentivizing reasoning ability generalizes to other tasks, model families and post-training algorithms. For cross-task validation, we consider code reasoning (Chen et al., 2021) as an additional task, which has been widely studied in the literature. For cross-family validation, we choose the Llama 3 series (Grattafiori et al., 2024), an established and widely used model family for LLM research. Its earlier release also provides a natural safeguard against potential data contamination in our evaluation benchmarks. For cross-algorithm validation, we choose PPO (Schulman et al., 2017), a widely adopted post-training algorithm for incentivizing reasoning. PPO is particularly suitable for our study because, unlike GRPO (Shao et al., 2024) and REINFORCE (Williams, 1992), it provides token-level supervision through the advantage function, even when the underlying reward is sparse and outcome-based. In contrast, GRPO and REINFORCE assign the same trajectory-level reward signal to all tokens within the same response. We therefore focus on PPO to examine whether the sparse supervision phenomenon extends to a post-training algorithm with token-level learning signals beyond OPD. Our preliminary experiments with GRPO and REINFORCE found that they do not improve the model under the extremely sparse supervision regime, and we thus do not include them in this work.

Sparse OPD on Coding Reasoning. To evaluate whether the effectiveness of sparse OPD extends beyond mathematical reasoning, we conduct experiments on coding reasoning. We use the Eurus-RL-Code dataset (Cui et al., 2025) for training, randomly sampling 12K prompts from the full dataset. For evaluation, we use LiveCodeBench v6 (Jain et al., 2025) and the Eurus-RL-Code validation set. The former evaluates out-of-distribution generalization, while the latter evaluates in-distribution performance. From each evaluation dataset, we randomly sample 50 prompts, as evaluation with $n = 256$ generations per prompt is particularly time-consuming.

We consider two settings of Strong-to-Weak distillation: (1) Qwen3-30B-A3B-Instruct-2507 as the teacher and Qwen3-4B as the student, representing the large-scale teacher/small-scale student setting; and (2) Qwen3-4B-Instruct-2507 as the teacher and Qwen3-8B as the student, representing the small-scale teacher/large-scale student setting. All models operate in no-think mode throughout training and evaluation, and other experimental settings follow those in Section 3. Experiment results on pass@k are shown in Figures 20 and 21. The observations from coding reasoning distillation closely resemble those from mathematical reasoning distillation. Specifically, *rand1tok* significantly improves the base student's reasoning ability. Among the sparse OPD variants, *minmaxtok* and

pctltail perform best, both of them outperform plain OPD in the large k regime. In the large-scale teacher/small-scale student setting, *mintok* is also quite effective and consistently outperforms *maxtok* on both benchmarks, while in the small-scale teacher/large-scale student setting, *maxtok* is more effective than *mintok*.

Sparse OPD with Llama Models. As for the cross-family validation, we choose Llama-3.1-8B-Instruct as the student and Llama-3.3-70B-Instruct as the teacher for Strong-to-Weak distillation in math reasoning tasks. All other experimental settings follow those in Section 3. Experiment results are presented in Figure 22. In particular, *mintok* and *pctltail 0.05%* generally match the performance of plain OPD, whereas the other sparse OPD variants fail to improve the base student. Although the reasoning improvement from plain OPD is less pronounced than that observed across the nine Qwen3 families, it is still statistically significant. More importantly, we again identify extremely sparse OPD variants that achieve performance comparable to dense OPD, suggesting that the extremely-sparse-supervision phenomenon extends beyond the Qwen3 model family. Taken together, these results lead us to conjecture that extremely sparse supervision incentivizing reasoning ability is closely associated with successful dense OPD: when the conditions for successful OPD are satisfied, the phenomenon emerges as well. In the next section, however, we show that this phenomenon is not exclusive to OPD: under PPO, sparse supervision can yield meaningful reasoning improvements even when dense supervision causes training to collapse.

Sparse PPO for RLVR. Sparse PPO uses the same objective function (3.1) as sparse OPD, only the token-level advantage A_t is replaced with generalized advantage estimation (Schulman et al., 2017). Thus, all sparse variants, *rand1tok, mintok, maxtok, minmaxtok, rand1tok 0.1%* and *pctltail 0.05%*, can be implemented with sparse PPO. Experiment configuration for PPO is provided in Appendix F, Table 28. Other experiment setup follows that in Section 3. Specifically, when we tune the PPO algorithm with 0.8.0 verl (Sheng et al., 2024), we find PPO keeps failing but sparse variants work. We then switch to 0.9.0 verl with the same configuration, this time PPO works well. Experiment results are provided in Figure 23: 0.8.0 version verl based PPO fails to improve the base model, while several 0.8.0 version verl based sparse PPO variants, including *rand1tok, mintok, maxtok, pctltail 0.05%* significantly improves the base model. While 0.9.0 dev version verl based plain PPO successfully improves the base model. 0.9.0 dev version verl based sparse PPO variants, including *rand1tok, mintok, maxtok, pctltail 0.05%*, also significantly improves the base model, and on AIME 24 they match or outperform plain PPO when k is large; on AIME 25 they fall short of the plain PPO. But a closer examination shows that 0.8.0 version verl based *mintok, maxtok, pctltail 0.05%* outperforms 0.9.0 version verl based plain PPO when k is large. But overall sparse PPO variants do not achieve the same level of sampling efficiency as 0.9.0 verl based plain PPO.

The weaker performances of sparse PPO compared with sparse OPD may stem from the different token-level learning signals used by the two methods. In PPO, token-level advantages are estimated from sparse outcome rewards through temporal-difference-based credit assignment, so they can be noisy and affected by the bias and variance of the advantage estimator. In contrast, OPD obtains its token-level signal directly from the teacher's token preference, without requiring token-level credit to be inferred from the outcome reward. As a result, keeping only a small number of tokens may be more effective in OPD, where the selected tokens still carry a direct teacher-derived signal. We hypothesize that this difference partly explains why sparse supervision is more effective in OPD than in PPO, and leave a more systematic investigation to future work.

D Full Experiment Results

In this section, we provide complete experiment results across the nine families of Strong-to-Weak distillation. Main results on pass@k are provided in Figures 4 to 12, main results on avg@8 are provided in Tables 6 to 14. Ablation results of threshold sweeping on pass@k are provided in Figures 13 to 17, ablation results of threshold sweeping on avg@8 are provided in Tables 15 to 23.

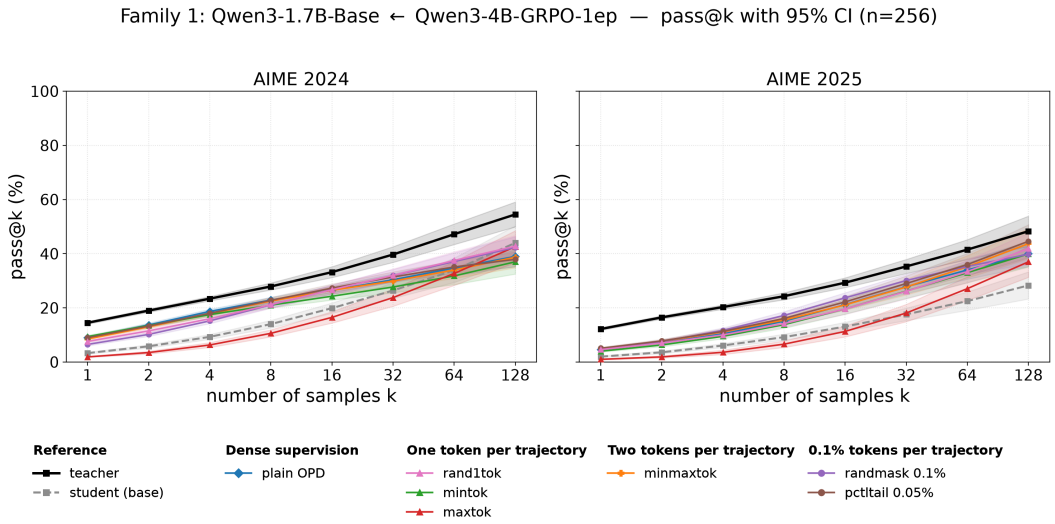


Figure 4: Family 1 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

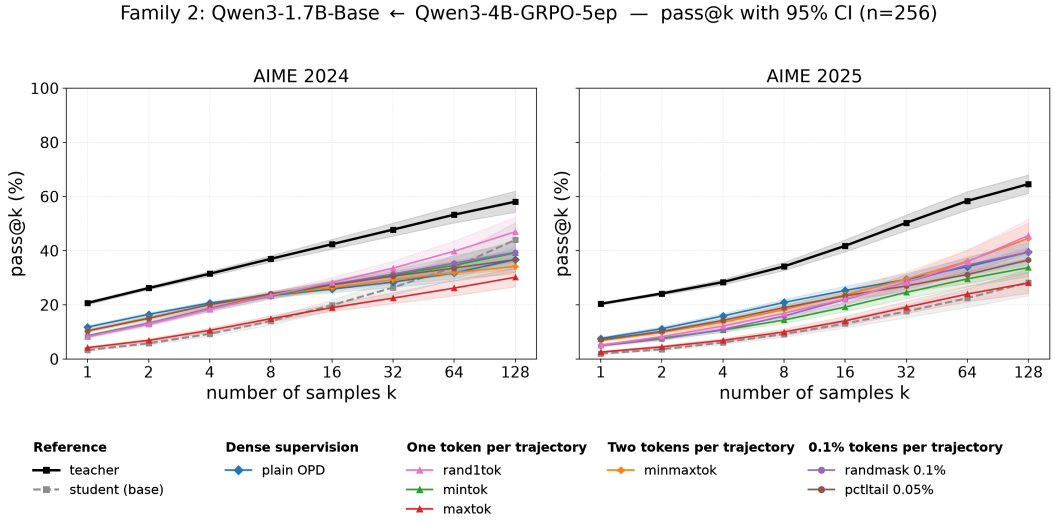


Figure 5: Family 2 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

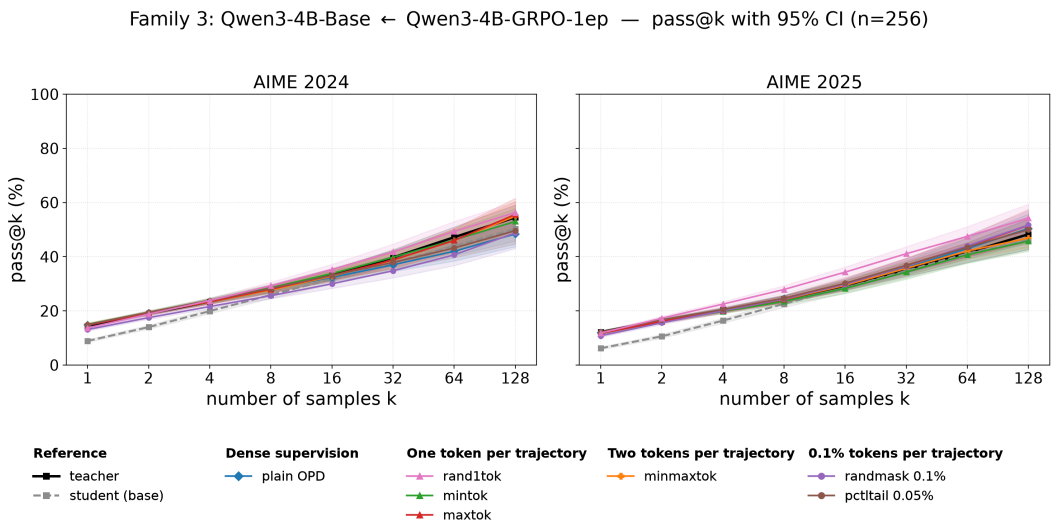


Figure 6: Family 3 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

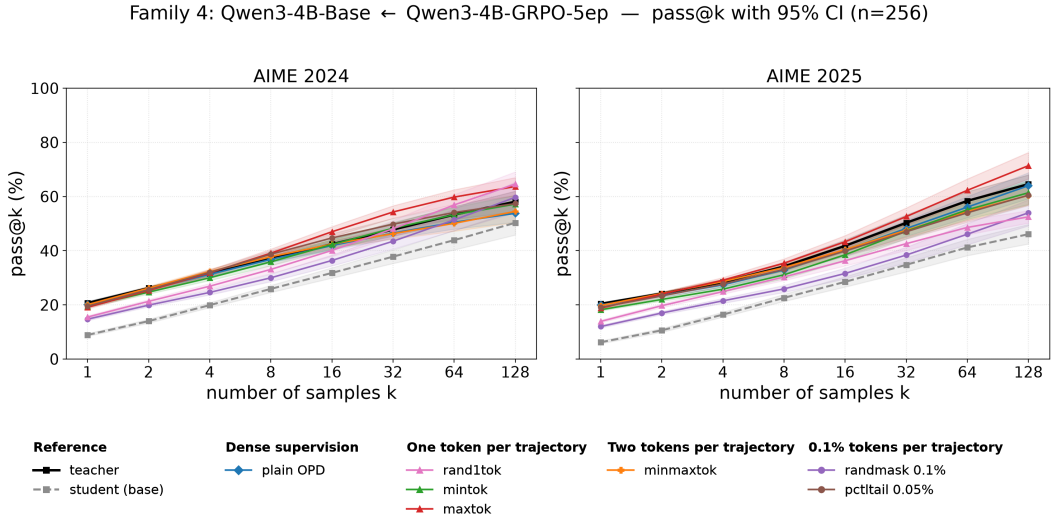


Figure 7: Family 4 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

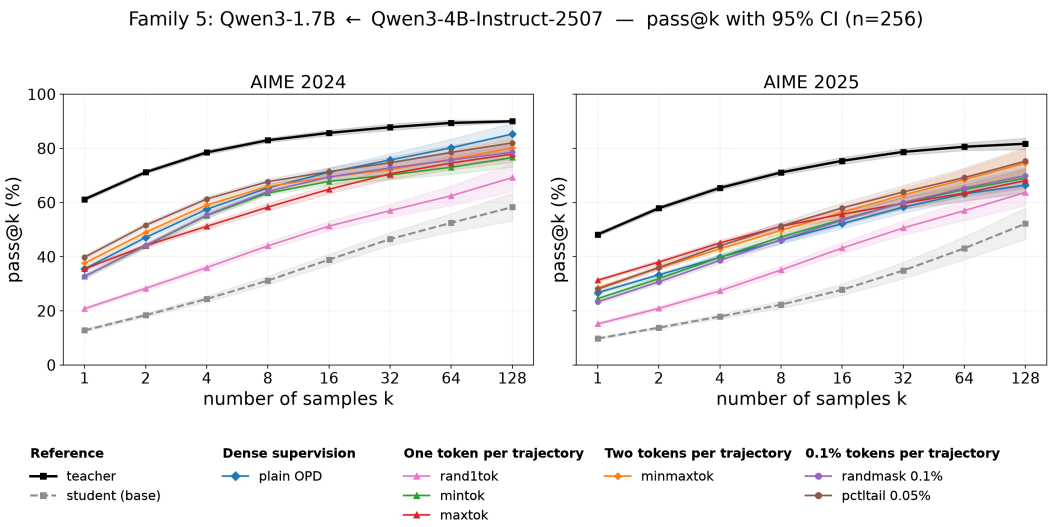


Figure 8: Family 5 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

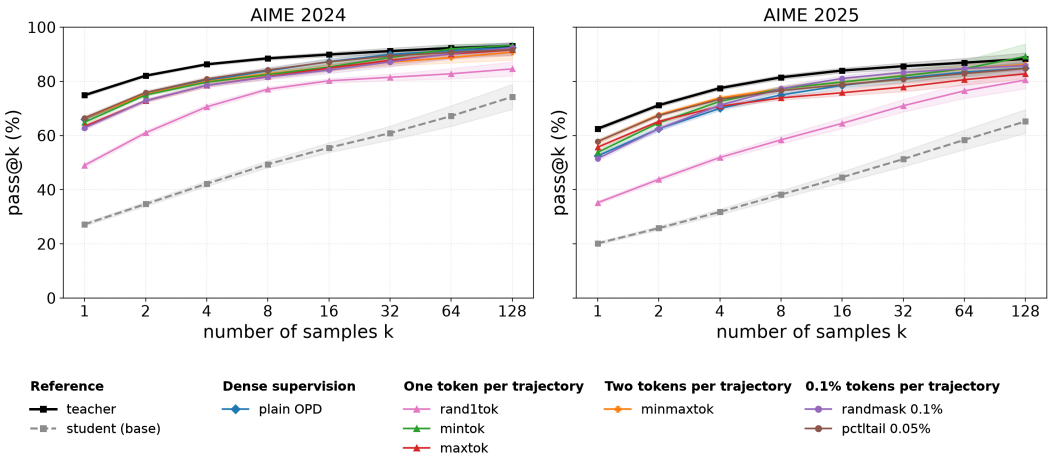


Figure 9: Family 6 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

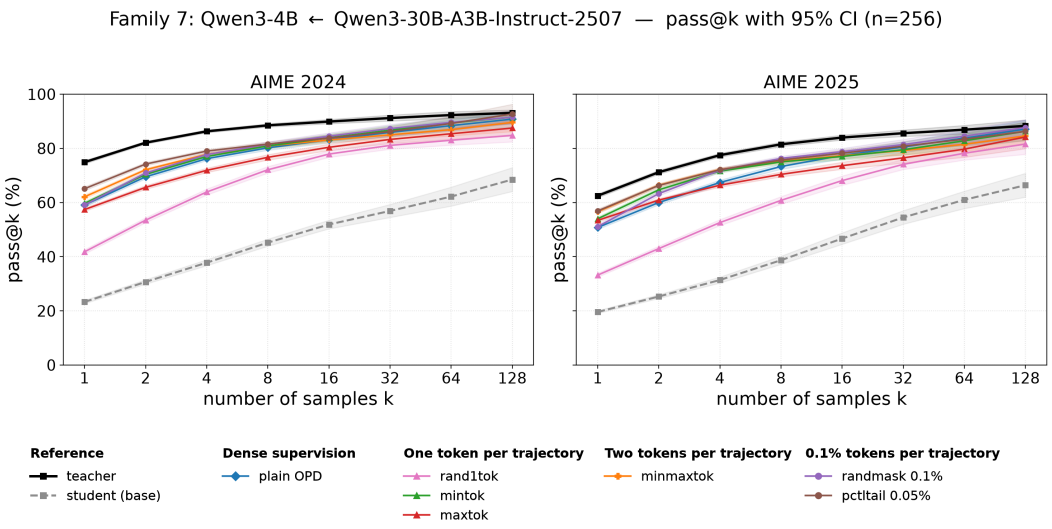


Figure 10: Family 7 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

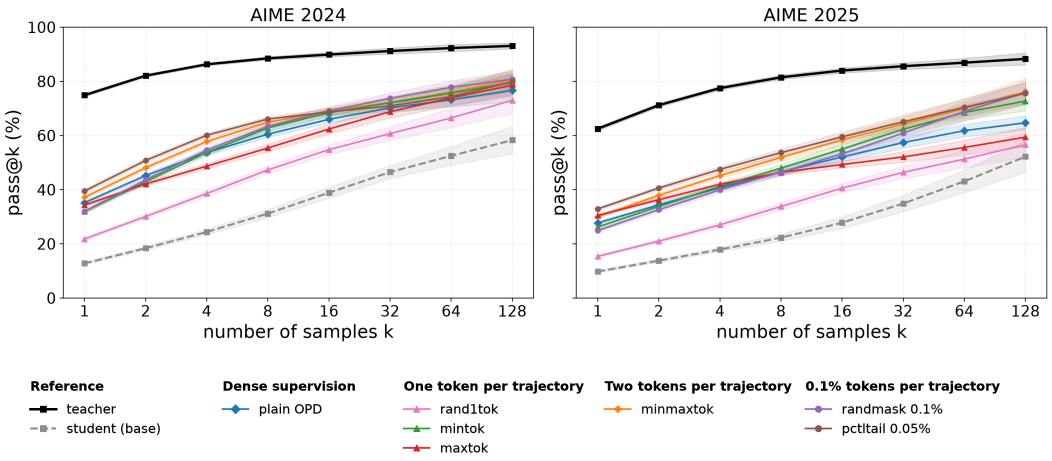


Figure 11: Family 8 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

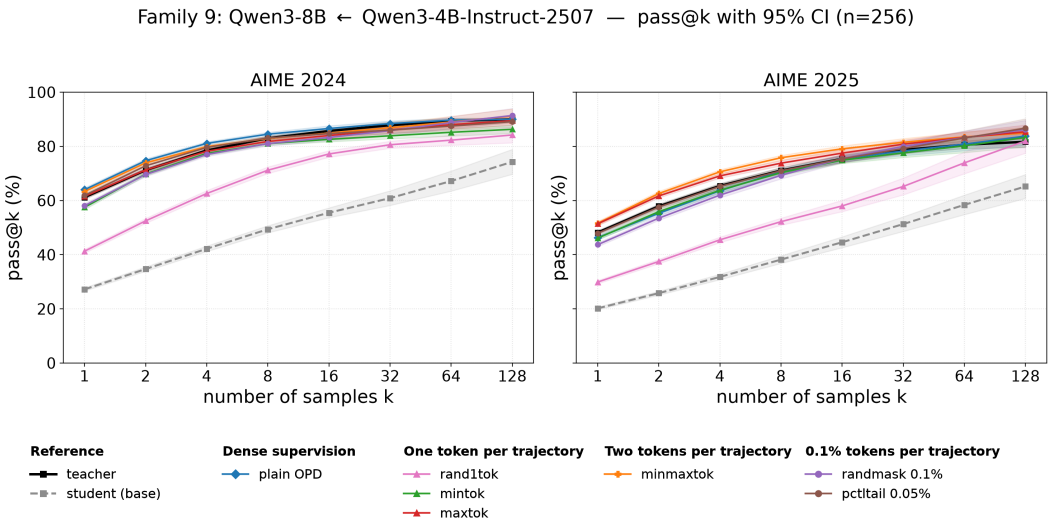


Figure 12: Family 9 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

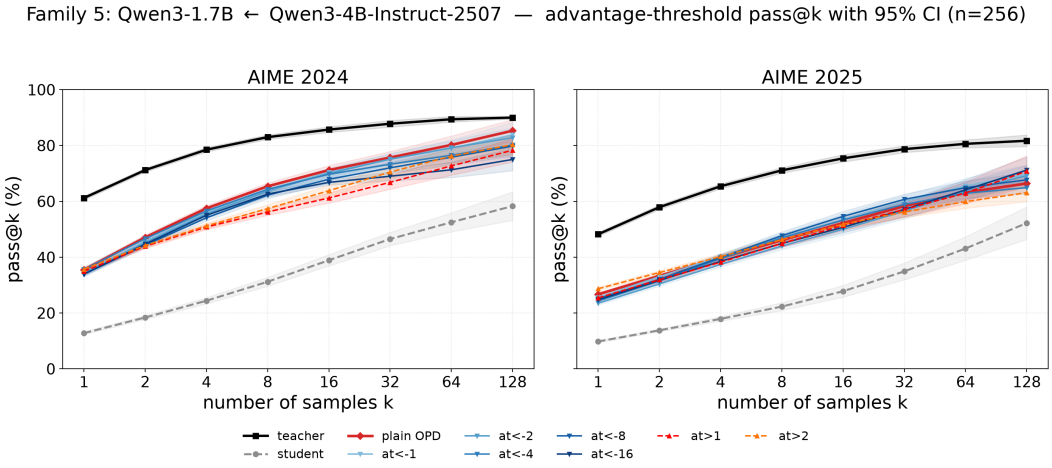


Figure 13: Family 5 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

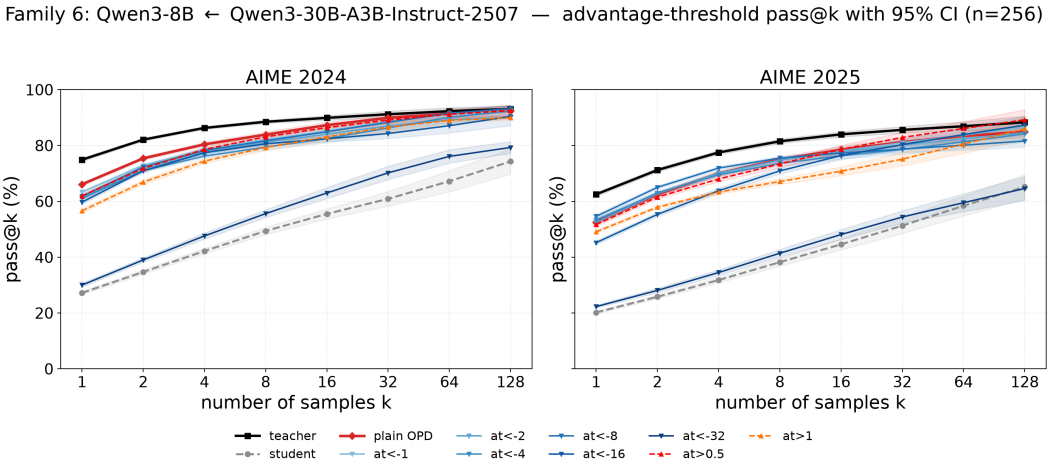


Figure 14: Family 6 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

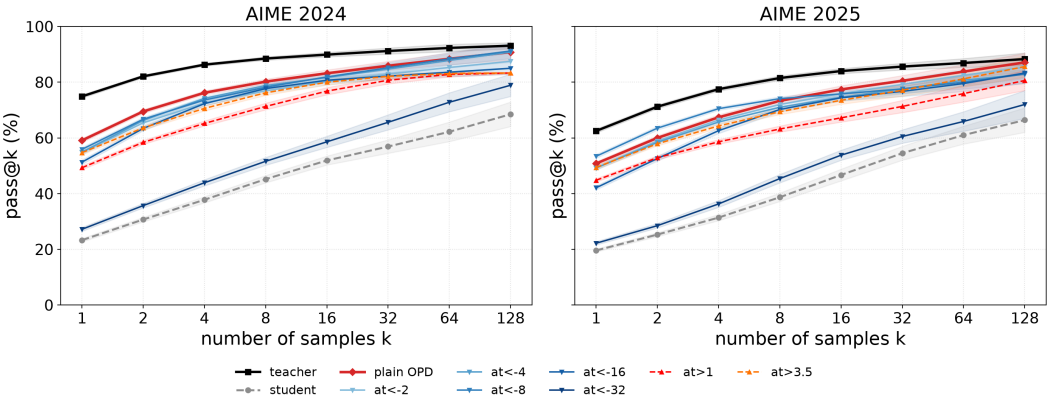


Figure 15: Family 7 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

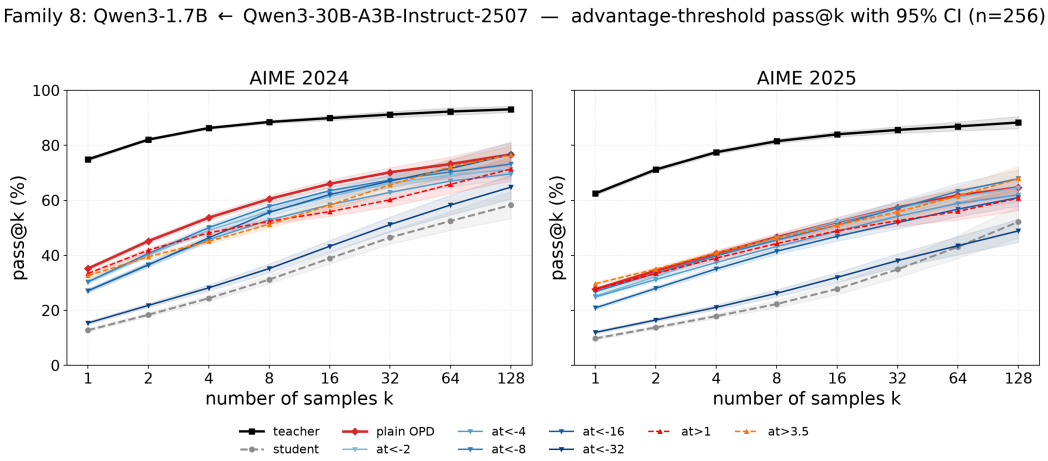


Figure 16: Family 8 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

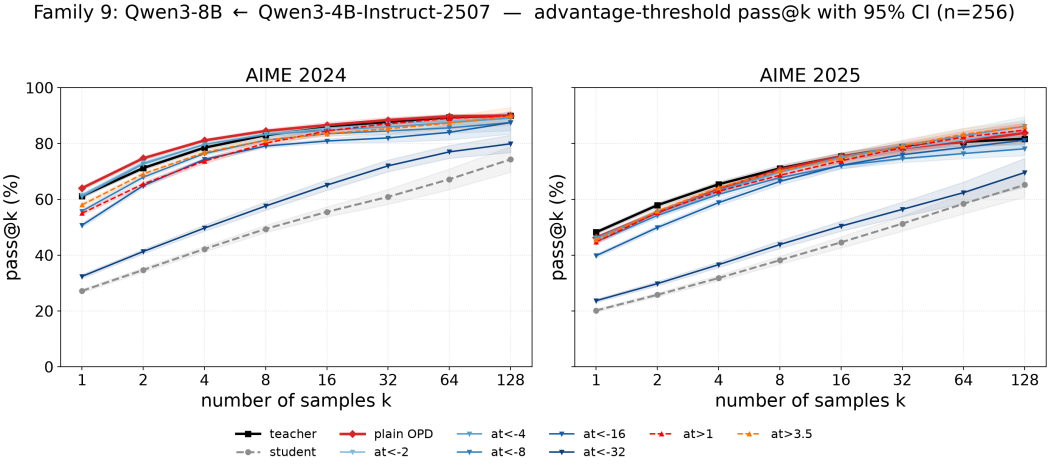


Figure 17: Family 9 pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

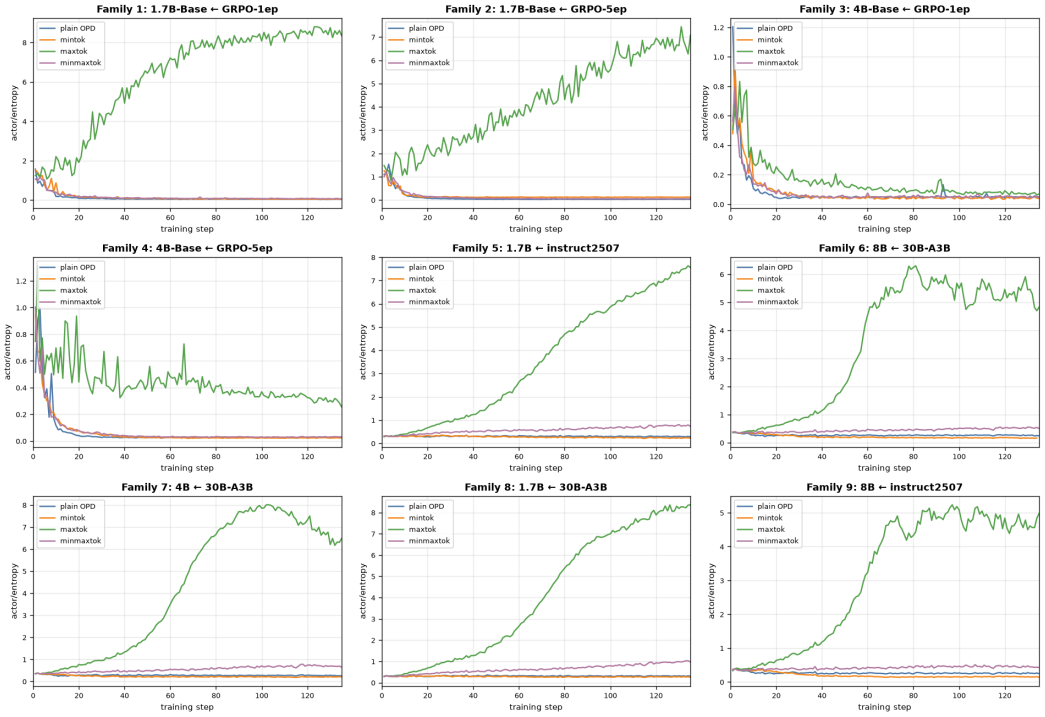


Figure 18: Learning dynamics of actor entropy across nine families.

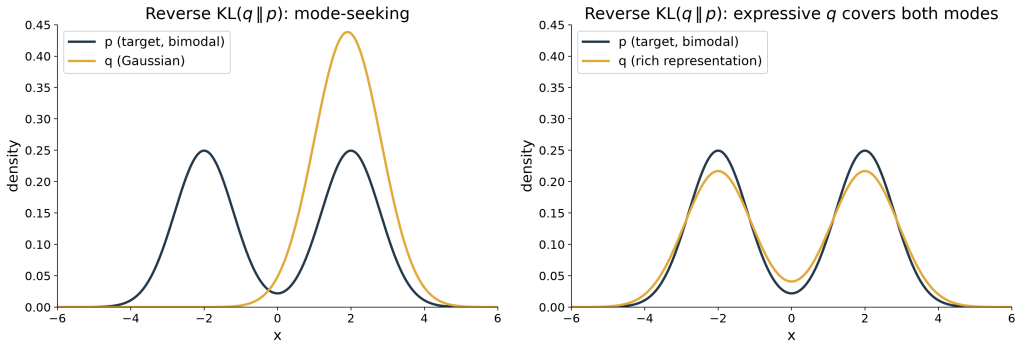


Figure 19: A toy-example illustration of minimizing reverse KL divergence between two distributions p and q. Left: the mode-seeking behavior of minimizing reverse KL divergence between a uni-modal Gaussian distribution q and a bimodal target distribution p. Right: if q is expressive enough, then minimizing reverse KL divergence recovers the target distribution p.

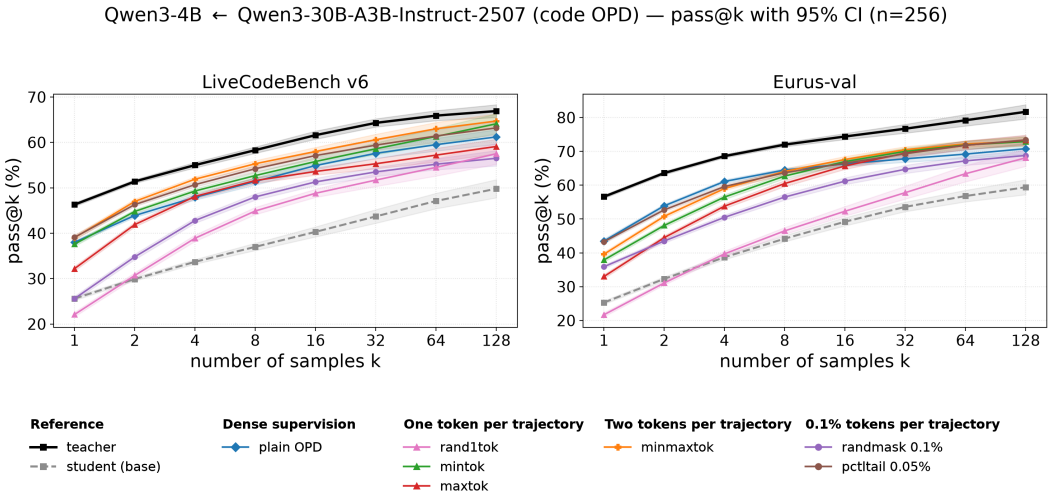


Figure 20: Cross-task validation with coding reasoning. Strong-to-Weak distillation setting with Qwen3-4B as student and Qwen3-30B-A3B-Instruct-2507 as teacher. Figures are pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across a subset of LiveCodeBench v6 and Eurus-RL-code dataset. Shaded area is the 95% confidence interval.

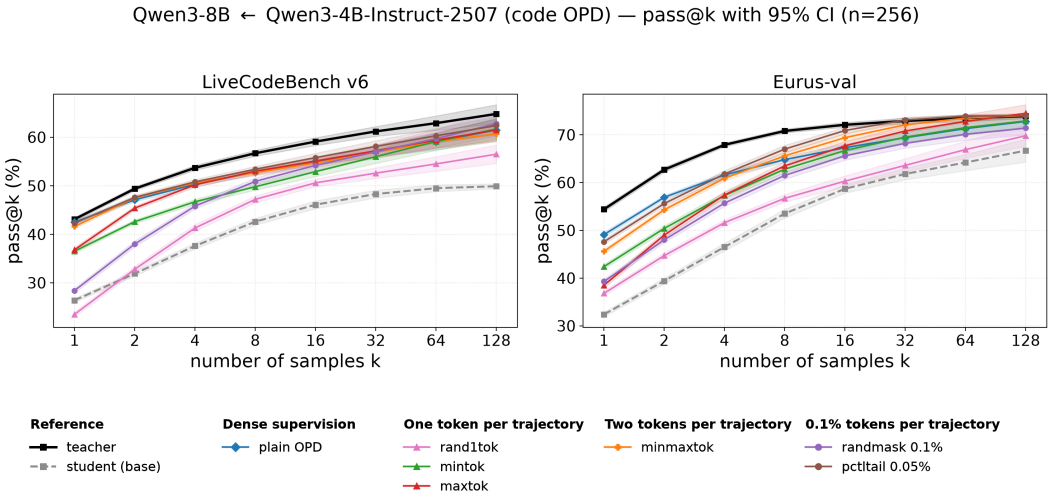


Figure 21: Cross-task validation with coding reasoning. Strong-to-Weak distillation setting with Qwen3-8B as student and Qwen3-4B-Instruct-2507 as teacher. Figures are pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across a subset of LiveCodeBench v6 and Eurus-RL-code dataset. Shaded area is the 95% confidence interval.

Table 6: Family 1: Qwen3-1.7B-Base ← Qwen3-4B-GRPO-1ep. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
Method	AIME24	AIME25	HMMT	Mean	keep_frac (%)	revKL	freeze (%)
Teacher	13.3	12.9	3.3	9.8	–	0.000	–
Student	1.7	0.8	0.4	1.0	–	0.156	–
plain OPD	8.3	5.0	0.4	4.6	100	0.030	82.2
one token supervision per trajectory							
rand1tok	7.1	5.4	0.8	4.4	0.0757	0.051	89.5
mintok	7.9	3.3	0.4	3.9	0.0521	0.033	86.0
maxtok	1.7	2.9	0.4	1.7	0.0427	0.414	87.6
two tokens supervision per trajectory							
minmaxtok	9.2	4.6	0.0	4.6	0.1150	0.054	86.0
0.1% tokens supervision per trajectory							
randmask 0.1%	8.3	4.6	0.8	4.6	0.0986	0.060	88.1
pctltail 0.05%	10.8	7.1	0.0	**6.0**	0.267	0.044	85.7

Table 7: Family 2: Qwen3-1.7B-Base ← Qwen3-4B-GRPO-5ep. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
Method	AIME24	AIME25	HMMT	Mean	keep_frac (%)	revKL	freeze (%)
Teacher	20.8	17.9	8.8	15.8	–	0.000	–
Student	1.7	0.8	0.4	1.0	–	0.361	–
plain OPD	10.8	9.6	2.1	**7.5**	100	0.138	78.7
one token supervision per trajectory							
rand1tok	8.3	5.8	0.8	5.0	0.0611	0.156	89.1
mintok	7.9	3.3	0.8	4.0	0.0289	0.083	84.2
maxtok	4.6	3.8	0.4	2.9	0.0512	0.280	86.2
two tokens supervision per trajectory							
minmaxtok	10.0	6.7	0.8	5.8	0.0645	0.123	84.3
0.1% tokens supervision per trajectory							
randmask 0.1%	9.6	4.2	1.2	5.0	0.0995	0.120	88.2
pctltail 0.05%	7.9	7.5	2.1	5.8	0.1840	0.162	83.3

E Additional Results and Discussion

E.1 Memory Saving

During the backpropagation at each step, the student's gradient path holds two kinds of activation: the transformer backbone activations across all layers, and the output logits $[B, T, V]$ produced by the language model (LM) head, where B is the batch size, T is the response length, V is the vocabulary size. Because the vocabulary size V vastly exceeds the hidden width H, for example in Qwen3 models $V = 151{,}936$ versus $H \sim 2.5\text{–}4 \times 10^3$, the logit tensor, together with the softmax/KL

Table 8: Family 3: Qwen3-4B-Base ← Qwen3-4B-GRPO-1ep. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
Method	AIME24	AIME25	HMMT	Mean	keep_frac (%)	revKL	freeze (%)
Teacher	13.3	12.9	3.3	9.8	–	0.000	–
Student	9.6	7.9	0.8	6.1	–	0.082	–
plain OPD	15.8	14.2	4.2	11.4	100	0.001	84.0
one token supervision per trajectory							
rand1tok	12.9	10.4	5.0	9.4	0.0806	0.004	93.4
mintok	14.2	13.3	2.9	10.1	0.0673	0.001	88.5
maxtok	15.4	15.8	5.8	12.4	0.0731	0.001	90.3
two tokens supervision per trajectory							
minmaxtok	16.7	16.2	5.0	**12.6**	0.1370	0.000	88.4
0.1% tokens supervision per trajectory							
randmask 0.1%	15.4	12.9	2.5	10.3	0.1020	0.006	91.0
pctltail 0.05%	17.1	13.8	3.8	11.5	0.2770	0.001	88.1

Table 9: Family 4: Qwen3-4B-Base ← Qwen3-4B-GRPO-5ep. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
Method	AIME24	AIME25	HMMT	Mean	keep_frac (%)	revKL	freeze (%)
Teacher	20.8	17.9	8.8	15.8	–	0.000	–
Student	9.6	7.9	0.8	6.1	–	0.441	–
plain OPD	23.3	17.5	7.5	16.1	100	0.005	75.0
one token supervision per trajectory							
rand1tok	12.1	13.8	6.7	10.8	0.0601	0.078	90.2
mintok	19.2	17.5	6.2	14.3	0.0369	0.018	81.9
maxtok	22.9	18.8	7.5	**16.4**	0.0364	0.428	89.5
two tokens supervision per trajectory							
minmaxtok	21.7	19.6	5.0	15.4	0.0622	0.015	81.8
0.1% tokens supervision per trajectory							
randmask 0.1%	16.2	12.5	5.4	11.4	0.1000	0.119	88.6
pctltail 0.05%	19.2	18.3	5.4	14.3	0.1770	0.012	80.9

intermediates derived from it, is the single dominant term in the footprint of the backward pass. It is precisely this term that forces small microbatches and caps the trainable context length and model size. Sparse OPD updates the loss on only $O(B)$ positions (one or two tokens per trajectory). Under a gather-then-project realization, the LM head is applied only at those positions, so the retained logit activation shrinks from $[B, T, V]$ to $[B, V]$. Thus, for the sparse OPD variants the logit memory is effectively erased. For a detailed discussion on the implementation, which involves several engineering tricks, we refer readers to Xu et al. (2026).

Table 10: Family 5: Qwen3-1.7B ← Qwen3-4B-Instruct-2507. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
Method	AIME24	AIME25	HMMT	Mean	keep_frac (%)	revKL	freeze (%)
Teacher	62.1	48.3	30.0	46.8	–	0.000	–
Student	12.9	8.3	5.0	8.7	–	0.371	–
plain OPD	36.7	25.8	15.8	26.1	100	0.318	80.1
one token supervision per trajectory							
rand1tok	19.6	16.7	9.2	15.1	0.0431	0.317	94.5
mintok	31.7	22.9	18.8	24.4	0.0240	0.366	90.8
maxtok	37.1	28.7	20.0	**28.6**	0.0169	1.227	89.8
two tokens supervision per trajectory							
minmaxtok	38.3	28.7	17.9	28.3	0.0414	0.481	90.8
0.1% tokens supervision per trajectory							
randmask 0.1%	35.4	21.2	15.0	23.9	0.0994	0.397	92.3
pctltail 0.05%	37.9	31.7	16.2	**28.6**	0.138	0.449	88.8

Table 11: Family 6: Qwen3-8B ← Qwen3-30B-A3B-Instruct-2507. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
Method	AIME24	AIME25	HMMT	Mean	keep_frac (%)	revKL	freeze (%)
Teacher	72.5	62.1	42.5	59.0	–	0.000	–
Student	26.2	20.8	10.8	19.3	–	0.268	–
plain OPD	65.4	54.2	29.2	49.6	100	0.158	78.8
one token supervision per trajectory							
rand1tok	50.4	37.9	22.9	37.1	0.0361	0.214	96.0
mintok	63.3	52.9	32.9	49.7	0.0253	0.192	88.2
maxtok	61.7	55.0	32.1	49.6	0.0220	0.721	89.9
two tokens supervision per trajectory							
minmaxtok	63.3	55.4	30.8	49.9	0.0391	0.263	88.9
0.1% tokens supervision per trajectory							
randmask 0.1%	62.5	53.3	32.1	49.3	0.100	0.214	91.3
pctltail 0.05%	67.1	55.8	34.6	**52.5**	0.148	0.236	86.9

E.2 Response Length

In this section, we compare the response length of all checkpoints obtained from the plain OPD and sparse OPD variants across all nine families. The response length statistics in Table 26 is calculated from response trajectories on AIME 2024, AIME 2025 and HMMT 2025 obtained during evaluation.

Table 12: Family 7: Qwen3-4B ← Qwen3-30B-A3B-Instruct-2507. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
Method	AIME24	AIME25	HMMT	Mean	keep_frac (%)	revKL	freeze (%)
Teacher	72.5	62.1	42.5	59.0	–	0.000	–
Student	24.6	20.0	12.9	19.2	–	0.303	–
plain OPD	60.4	47.9	34.6	47.6	100	0.169	74.5
one token supervision per trajectory							
rand1tok	41.7	35.8	21.2	32.9	0.0362	0.220	93.2
mintok	56.7	55.4	32.5	48.2	0.0243	0.203	86.1
maxtok	56.2	50.8	27.5	44.9	0.0178	0.724	87.8
two tokens supervision per trajectory							
minmaxtok	63.3	57.5	33.3	51.4	0.0429	0.306	86.2
0.1% tokens supervision per trajectory							
randmask 0.1%	58.3	51.7	35.0	48.3	0.1000	0.223	91.4
pctltail 0.05%	64.6	54.6	37.1	**52.1**	0.1470	0.271	84.0

Table 13: Family 8: Qwen3-1.7B ← Qwen3-30B-A3B-Instruct-2507. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
Method	AIME24	AIME25	HMMT	Mean	keep_frac (%)	revKL	freeze (%)
Teacher	72.5	62.1	42.5	59.0	–	0.000	–
Student	12.9	8.3	5.0	8.7	–	0.368	–
plain OPD	37.1	27.9	17.5	27.5	100	0.190	80.2
one token supervision per trajectory							
rand1tok	21.7	15.8	9.2	15.6	0.0379	0.299	95.0
mintok	32.9	30.0	17.5	26.8	0.0254	0.260	91.4
maxtok	35.8	32.1	19.2	29.0	0.0176	0.750	89.5
two tokens supervision per trajectory							
minmaxtok	40.4	29.6	16.7	28.9	0.0409	0.374	91.3
0.1% tokens supervision per trajectory							
randmask 0.1%	37.1	23.3	16.2	25.6	0.0988	0.267	92.6
pctltail 0.05%	38.8	31.2	20.4	**30.1**	0.1380	0.404	89.6

F Details on Experiment Configuration

In this section, we provide details on experiment configuration to reproduce all experiment results. Hyperparameters used to train the GRPO teachers, all OPD and sparse OPD variants are provided in Table 27. Hyperparameters used to train the PPO and sparse PPO variants are provided in Table 28. Prompts used for math and coding are as follows:

Prompt for math reasoning

Table 14: Family 9: Qwen3-8B ← Qwen3-4B-Instruct-2507. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
Method	AIME24	AIME25	HMMT	Mean	keep_frac (%)	revKL	freeze (%)
Teacher	62.1	48.3	30.0	46.8	–	0.000	–
Student	26.2	20.8	10.8	19.3	–	0.268	–
plain OPD	66.7	44.6	31.2	47.5	100	0.184	78.2
one token supervision per trajectory							
rand1tok	42.1	30.0	17.5	29.9	0.0392	0.253	95.5
mintok	55.4	47.1	25.8	42.8	0.0259	0.213	87.3
maxtok	63.8	50.8	29.6	48.1	0.0167	1.116	90.7
two tokens supervision per trajectory							
minmaxtok	62.9	55.0	30.0	**49.3**	0.0472	0.319	88.1
0.1% tokens supervision per trajectory							
randmask 0.1%	59.6	45.4	28.3	44.4	0.0999	0.279	94.1
pctltail 0.05%	62.1	47.5	28.8	46.1	0.120	0.254	86.1

Table 15: Family 1: Qwen3-1.7B-Base ← Qwen3-4B-GRPO-1ep. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
	AIME24	AIME25	HMMT	Mean	keep%	revKL	Spars.%
Teacher	13.3	12.9	3.3	9.8	–	0.000	–
Student	1.7	0.8	0.4	1.0	–	0.156	–
plain OPD	8.3	5.0	0.4	4.6	100	0.030	82.2
at<-1	7.9	5.0	0.4	4.4	3.7900	0.024	82.2
at<-2	9.2	3.8	0.0	4.3	2.3700	0.021	82.2
at<-8	7.1	5.8	1.2	**4.7**	0.1960	0.031	85.2
at<-16	6.7	6.2	0.0	4.3	0.0079	0.033	87.8
at<-32	2.5	2.5	0.0	1.7	0.0000	0.085	89.8
at>0.5	3.8	0.4	0.8	1.7	6.4700	0.379	74.4
at>1	0.8	0.0	0.4	0.4	1.9700	0.430	72.9
at>3.5	2.9	1.7	0.4	1.7	0.0045	0.189	90.4

```
MATH_INSTRUCTION = (
    "Please reason step by step, and put your final answer
    within \\boxed{}."
)
```

Prompt for code reasoning

```
CODE_INSTRUCTION = (
    "Write Python code to solve the problem. Present the code in\n"
    "```python\n"
    "Your code\n"
    "```\n"
```

Table 16: Family 2: Qwen3-1.7B-Base ← Qwen3-4B-GRPO-5ep. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
	AIME24	AIME25	HMMT	Mean	keep%	revKL	Spars.%
Teacher	20.8	17.9	8.8	15.8	–	0.000	–
Student	1.7	0.8	0.4	1.0	–	0.361	–
plain OPD	10.8	9.6	2.1	**7.5**	100	0.138	78.7
at<-1	9.6	7.1	2.1	6.2	4.6200	0.106	79.1
at<-2	9.6	5.4	2.1	5.7	3.8100	0.053	79.1
at<-8	10.8	7.5	1.2	6.5	1.4700	0.079	80.0
at<-16	11.2	5.8	2.1	6.4	0.3540	0.086	82.1
at<-32	8.8	3.8	0.4	4.3	0.0028	0.097	87.4
at>0.5	6.7	3.3	1.2	3.8	5.4600	0.629	78.2
at>1	2.5	0.8	0.4	1.2	2.1000	0.735	75.3
at>3.5	2.1	0.8	0.0	1.0	0.0073	0.317	89.2

Table 17: Family 3: Qwen3-4B-Base ← Qwen3-4B-GRPO-1ep. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
	AIME24	AIME25	HMMT	Mean	keep%	revKL	Spars.%
Teacher	13.3	12.9	3.3	9.8	–	0.000	–
Student	9.6	7.9	0.8	6.1	–	0.082	–
plain OPD	15.8	14.2	4.2	11.4	100	0.001	84.0
at<-1	19.6	16.2	5.0	**13.6**	0.1560	0.001	83.9
at<-2	15.8	16.2	4.2	12.1	0.1400	0.002	83.8
at<-4	18.3	12.9	3.8	11.7	0.0415	0.002	84.9
at<-8	13.8	12.5	2.9	9.7	0.0041	0.008	88.0
at<-16	10.8	7.5	1.2	6.5	0.0002	0.030	92.1
at<-32	10.0	5.8	1.2	5.7	0.0000	0.019	100.0
at>0.5	15.8	12.9	1.2	10.0	0.1660	0.003	87.9
at>1	15.8	12.5	3.3	10.6	0.0196	0.024	89.1
at>3.5	13.8	10.0	2.9	8.9	0.0001	0.030	92.7

```
    "at the end.\n"
    "You need to think first then write the Python code."
)
```

All OPD experiments are conducted on 8*A100 GPUs and 8*H100 GPUs, PPO experiments are conducted on 8*H100 GPUs, and evaluations are conducted on 8*H200 GPUs. Each experiment takes 5-10 hours on corresponding GPUs, the total amount of GPU hours for this project is about 10,000.

Table 18: Family 4: Qwen3-4B-Base ← Qwen3-4B-GRPO-5ep. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
	AIME24	AIME25	HMMT	Mean	keep%	revKL	Spars.%
Teacher	20.8	17.9	8.8	15.8	–	0.000	–
Student	9.6	7.9	0.8	6.1	–	0.441	–
plain OPD	23.3	17.5	7.5	16.1	100	0.005	75.0
at<-1	19.6	18.8	5.0	14.4	0.6920	0.007	75.2
at<-2	20.4	18.3	6.7	15.1	0.6830	0.006	78.1
at<-4	19.6	17.1	5.8	14.2	0.3780	0.008	78.7
at<-8	17.1	17.9	5.0	13.3	0.1840	0.015	80.0
at<-16	16.7	12.9	5.0	11.5	0.0339	0.024	82.2
at<-32	17.1	13.3	4.6	11.7	0.0007	0.067	88.5
at>0.5	24.2	21.2	7.9	**17.8**	1.1500	0.040	81.0
at>1	19.6	18.3	10.4	16.1	0.3510	0.193	85.5
at>3.5	18.3	13.8	5.4	12.5	0.0009	0.435	90.9

Table 19: Family 5: Qwen3-1.7B ← Qwen3-4B-Instruct-2507. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
	AIME24	AIME25	HMMT	Mean	keep%	revKL	Spars.%
Teacher	62.1	48.3	30.0	46.8	–	0.000	–
Student	12.9	8.3	5.0	8.7	–	0.371	–
plain OPD	36.7	25.8	15.8	26.1	100	0.318	80.1
at<-1	32.1	22.9	17.5	24.2	9.2400	0.291	81.5
at<-2	36.2	24.2	16.7	25.7	5.0200	0.300	81.7
at<-4	36.2	22.9	14.2	24.4	1.8500	0.275	83.2
at<-8	33.8	25.0	14.2	24.3	0.3700	0.278	86.3
at<-16	32.9	25.4	15.4	24.6	0.0302	0.352	90.6
at<-32	14.6	12.5	6.7	11.2	0.0002	0.379	93.6
at>0.5	35.4	32.5	15.8	**27.9**	7.5000	0.686	79.6
at>1	30.8	20.4	13.3	21.5	2.6600	1.011	80.2
at>2	37.5	28.7	17.1	27.8	0.3180	1.357	83.6
at>3.5	38.3	26.2	19.2	**27.9**	0.0136	1.231	89.6

Table 20: Family 6: Qwen3-8B ← Qwen3-30B-A3B-Instruct-2507. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
	AIME24	AIME25	HMMT	Mean	keep%	revKL	Spars.%
Teacher	72.5	62.1	42.5	59.0	–	0.000	–
Student	26.2	20.8	10.8	19.3	–	0.268	–
plain OPD	65.4	54.2	29.2	**49.6**	100	0.158	78.8
at<-1	64.6	53.8	30.0	49.4	6.0500	0.141	80.2
at<-2	65.8	51.2	30.0	49.0	2.8800	0.130	80.8
at<-4	59.2	50.8	28.3	46.1	0.8790	0.128	82.5
at<-8	63.7	52.9	30.0	48.9	0.1490	0.147	86.2
at<-16	57.9	43.8	29.2	43.6	0.0104	0.206	89.6
at<-32	28.7	21.2	15.0	21.7	0.0000	0.218	96.1
at>0.5	59.6	49.6	29.6	46.2	5.9900	0.415	80.5
at>1	53.8	50.8	26.7	43.8	1.8900	0.660	80.3
at>3.5	55.4	54.2	32.1	47.2	0.0104	0.820	88.0

Table 21: Family 7: Qwen3-4B ← Qwen3-30B-A3B-Instruct-2507. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
	AIME24	AIME25	HMMT	Mean	keep%	revKL	Spars.%
Teacher	72.5	62.1	42.5	59.0	–	0.000	–
Student	24.6	20.0	12.9	19.2	–	0.303	–
plain OPD	60.4	47.9	34.6	47.6	100	0.169	74.5
at<-2	56.7	47.1	31.7	45.1	3.1200	0.129	76.8
at<-4	57.1	53.3	29.6	46.7	0.9820	0.133	79.0
at<-8	58.3	53.3	32.5	**48.1**	0.1660	0.174	83.2
at<-16	47.9	42.1	29.2	39.7	0.0116	0.221	87.3
at<-32	25.8	22.5	15.8	21.4	0.0000	0.247	91.2
at>1	51.2	44.6	22.1	39.3	1.9900	0.701	76.2
at>3.5	54.2	52.1	30.0	45.4	0.0111	0.691	86.3

Table 22: Family 8: Qwen3-1.7B ← Qwen3-30B-A3B-Instruct-2507. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
	AIME24	AIME25	HMMT	Mean	keep%	revKL	Spars.%
Teacher	72.5	62.1	42.5	59.0	–	0.000	–
Student	12.9	8.3	5.0	8.7	–	0.368	–
plain OPD	37.1	27.9	17.5	**27.5**	100	0.190	80.2
at<-2	29.2	27.9	12.9	23.3	4.7700	0.116	82.1
at<-4	26.2	24.2	11.7	20.7	1.6800	0.093	83.6
at<-8	31.2	25.4	14.6	23.8	0.3290	0.157	87.4
at<-16	25.4	21.7	12.1	19.7	0.0336	0.296	91.3
at<-32	15.4	11.7	6.2	11.1	0.0002	0.354	92.3
at>1	32.9	27.5	15.0	25.1	2.7700	0.685	79.4
at>3.5	32.5	30.0	17.9	26.8	0.0143	0.789	89.1

Table 23: Family 9: Qwen3-8B ← Qwen3-4B-Instruct-2507. avg@8 scores of the base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 2024, AIME 2025 and HMMT Feb 2025. Mean averages over AIME24/AIME25/HMMT. keep_frac (%): fraction of response tokens supervised among all generated tokens; revKL: reverse KL divergence between the student and teacher; freeze (%): fraction of student network unchanged parameters after training.

	avg@8 (%)						
	AIME24	AIME25	HMMT	Mean	keep%	revKL	Spars.%
Teacher	62.1	48.3	30.0	46.8	–	0.000	–
Student	26.2	20.8	10.8	19.3	–	0.268	–
plain OPD	66.7	44.6	31.2	**47.5**	100	0.184	78.2
at<-2	60.4	49.6	28.3	46.1	2.7400	0.191	80.2
at<-4	63.7	45.8	26.7	45.4	0.8540	0.214	81.8
at<-8	55.8	45.0	28.3	43.0	0.1480	0.200	85.3
at<-16	52.9	40.4	25.0	39.4	0.0100	0.170	89.1
at<-32	36.2	24.2	13.3	24.6	0.0000	0.280	92.1
at>1	51.2	43.8	23.8	39.6	1.5640	0.836	81.7
at>3.5	63.7	46.7	25.0	45.1	0.0070	0.956	91.7

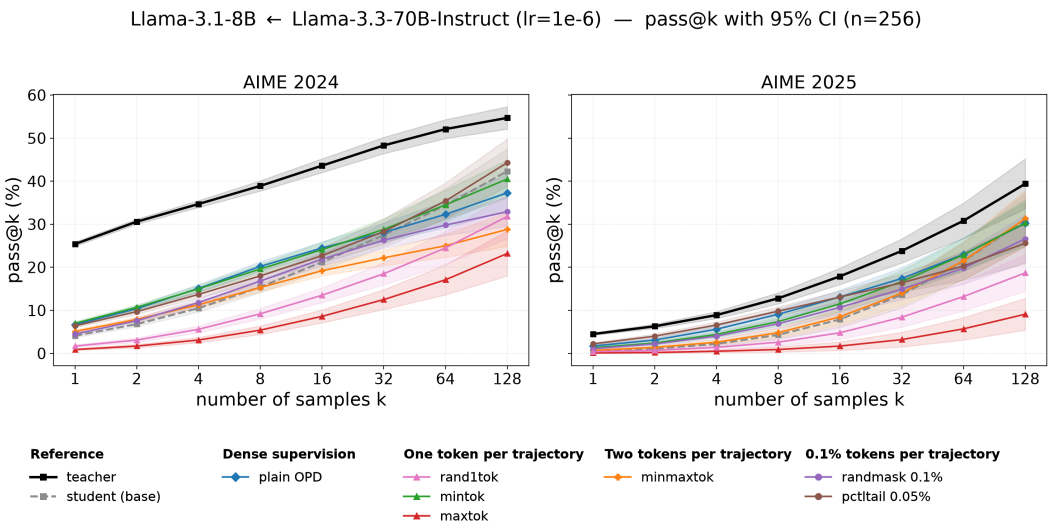


Figure 22: Cross-family validation with Llama models. Strong-to-Weak distillation setting with Llama-3.1-8B-Instruct as student and Llama-3.3-70B-Instruct as teacher. Figures are pass@k curves of base student model, teacher model, student model trained with plain OPD and student models trained with sparse OPD variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval.

Table 24: Categories of tokens with extremely positive supervision signal activated by *minmaxtok*. We audit the trajectories generated in the first 20 steps.

category	**description**	**examples**
`digit / value`	a numeral, or a single digit inside a larger number	`4`, `9`, the `5` in `1<5>0`, the `7` in `2<7>2`, a variable-as-value like `a`, `x`, `k`
`math-mode`	LaTeX delimiter or command opener — switches into a formula	`$`, `$$`, start of `\frac`, `\cdot`, `(`, `[`, `{`, `^`, `=`, `sum`, `quad`
`content word`	ordinary prose word (noun/verb/adjective) carrying lexical meaning	`determinant`, `function`, `symmetric`, `analyze`, `chooses`, `consider`, `rotates`, `circular`, `horizontal`, `valid`, `constraint`, `smaller`, `teams`, `axis`
`discourse hedge`	reasoning-flow / stance marker — steers, transitions, or backtracks	`But`, `Wait`, `Actually`, `So`, `Therefore`, `Thus`, `Since`, `Now`, `Let`, `Try`, `Note`, `Check`, `First`, `Alternatively`
`format separator`	whitespace / markdown / structural punctuation — organizes layout, no meaning	`space`, `\n`, `\n\n`, `--`, `###`, `####`, `**`, `:`, `\n\n`, `- (bullet)`, `,`, `>`
`CJK`	Chinese token (content or connective), when the trajectory reasons in Chinese	观察 (observe), 注意 (note), 回忆 (recall), 但 (but), 答案 (answer), 设 (let), 生成 (generate), 我们 (we)

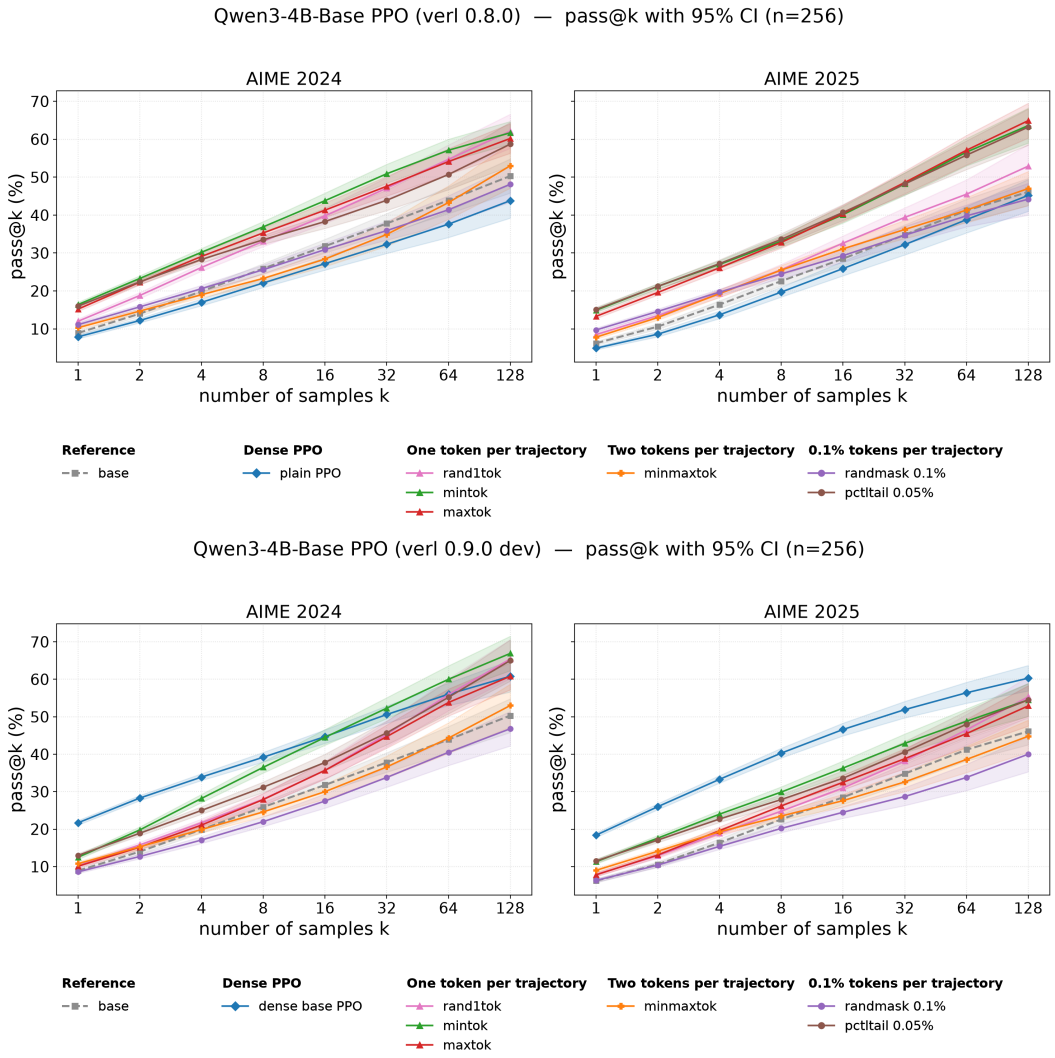


Figure 23: Pass@k curves of base model, model trained with plain PPO and models trained with sparse PPO variants across AIME 24 and AIME 25. Shaded area is the 95% confidence interval. Top: experiment results with verl 0.8.0. Bottom: experiment results with verl 0.9.0 dev.

Table 25: Categories of tokens with extremely negative supervision signal activated by *minmaxtok*. We audit the trajectories generated in the first 20 steps.

category	description	examples
`math / LaTeX fragment`	notation the student wrote (no error)	`\frac`, `=`, `^`, `$`, `{`, `}`, `\`, `_`, `(`, `left`, `right`, `sqrt`
`correct content word`	ordinary word right in context	`function`, `rectangle`, `perpendicular`, `distance`
`correct digit / value`	numeral that is arithmetically correct	the 7 in $\sigma(4) = 7$; the 2 in $13 \times 12 = 156$
`format separator`	whitespace / markdown / structural punctuation	`space`, `\n`, `\n\n`, `--`, `###`, `**`, `:`, `comma`
`stop <|im_end|>`	premature end-of-turn after `\boxed{...}`	`<|im_end|>` or `<|endoftext|>` right after the boxed answer
`genuine math error`	wrong committed digit/value (a real mistake)	$\mathrm{LCM}(15, 16) = 272$ ($\to 240$); $\mathrm{median}\{0, 120, 240\} = 240$ ($\to 120$); $\sin^2(\pi/2) = 0$ ($\to 1$)
`non-English token`	fluent, correct Chinese the student switched to	生成 (generate), 计算 (compute), 图 (graph)

Table 26: Average response length (number of tokens) per checkpoint, each cell presents *correct*/*all*. *correct*: mean length of correct responses; *all*: mean over all responses. F1 to F9 represent Family 1 to Family 9, respectively. '–' marks a method not included for that family.

Method	F1	F2	F3	F4	F5	F6	F7	F8	F9
Teacher	1121/2789	2949/6293	1121/2789	2949/6293	4451/9028	5110/6606	5110/6606	5110/6606	4451/9028
Student	1524/2874	1524/2874	1012/1619	1012/1619	1524/2874	1667/3607	1724/3619	1524/2874	1667/3607
plain OPD	1069/5740	3088/10193	1626/3988	2854/6486	5894/11138	7096/11760	7191/12395	7831/16643	5888/9059
rand1tok	950/4056	1623/7329	1644/3169	1722/5916	2457/5613	4684/8612	3891/7974	2744/6643	3146/6775
minmaxtok	1051/3691	2325/6864	1576/3923	2912/6189	7706/15692	9106/16248	10244/16328	8669/15479	7539/13320
mintok	995/5735	1947/9687	1825/4621	2749/6630	6070/13817	7494/14862	8215/15723	6519/13906	6143/12047
maxtok	1574/1749	3056/10694	1415/4028	3272/5182	12342/17283	15026/20484	15902/21346	15294/18449	12388/16326
randmask 0.1%	1044/5090	2948/8367	1282/2947	2500/3676	4878/12749	7454/13130	7538/14130	6619/14809	5963/11074
pctltail 0.05%	1545/4181	2312/6407	1445/4354	2752/6352	7888/15511	9543/16479	10774/18015	9441/18845	6321/10638
at<-1	1067/9122	2527/12958	1810/4092	3019/6920	5147/10665	7279/13425	–	–	–
at<-2	1083/9570	3349/12996	1308/4727	2949/7262	5107/10890	6734/14557	6882/15559	6786/23986	5473/8497
at<-4	–	–	1613/3830	2443/7059	5349/12578	7789/17518	7837/17162	8471/26584	5441/8913
at<-8	1119/7222	2039/11384	1255/2585	2042/4962	5674/14873	7908/17234	8435/17702	8698/24773	5368/10524
at<-16	931/5740	2381/8257	2039/11384	1860/6230	6132/14363	5764/11221	5548/10388	3893/8264	8486/15591
at<-32	1585/3804	2532/10186	1165/4968	2244/7020	1627/3201	2016/4158	2078/4676	1876/3546	2663/4939
at>0.5	1362/2194	4437/5452	1321/2295	3117/5545	9993/16503	9682/14322	–	–	–
at>1	1429/1938	3800/4844	1205/1943	3261/5696	10727/18594	15320/20065	13730/20286	14712/21807	10432/15027
at>2	–	–	–	–	14081/20855	–	–	–	–
at>3.5	1260/2656	1560/5097	1393/2706	3047/5313	12358/17680	15159/19158	16598/20923	13893/17616	10319/14457

Table 27: Training configuration for GRPO (teacher RL) and on-policy distillation (OPD).

	GRPO (teacher RL)	**OPD (distillation)**
Framework	verl 0.8.0	verl 0.8.0 (native OPD)
Objective	GRPO	Policy-gradient on-policy distillation (reverse KL)
Advantage estimator	GRPO	GRPO
Base / student model	Qwen3-4B-Base	Qwen3-{1.7B, 4B, 8B}-Base
Teacher	—	separate vLLM pool (Qwen3-4B / 30B-A3B)
Training data	DAPO-Math-17k	DAPO-Math-17k
Train batch size (prompts)	128	128
PPO mini-batch	64	64
Rollouts per prompt (n)	8	1
Max prompt length	1024	1024
Max response length	8192	8192
Learning rate	1×10^{-6}	1×10^{-6}
LR warmup ratio	0.0	0.0
Weight decay	0.0	0.0
Gradient clip	1.0	1.0
PPO clip ratio (ϵ)	0.2	0.2
Entropy coefficient	0.0	0.0
KL loss	on, coef 0.001, low_var_kl	off
KL in reward	off	off
Rollout temperature / top-p	1.0 / 1.0	1.0 / 1.0
Epochs	5 (and 1-epoch variant)	1 (4B/8B), 2 (1.7B)
Optimizer	AdamW (verl default)	AdamW (verl default)
Precision	bf16	bf16
Gradient checkpointing	yes	yes

Table 28: PPO training configuration.

Component	Configuration
Algorithm	
Advantage estimator	GAE ($\gamma = 1.0$, $\lambda = 1.0$)
KL regularization	Disabled
KL reward / loss coefficient	0
Actor	
Learning rate	1×10^{-5}
LR schedule / warmup	Constant / None
Weight decay	0.1
PPO epochs / mini-batch	1 / 64
Clip range	0.2 / 0.2
Dual clip	10.0
Entropy coefficient	0
Gradient clipping	1.0
Loss aggregation	Token-mean
Max tokens per GPU	24,576
Precision	BF16
FSDP size	8
Optimizer offload	Enabled
Parameter offload	Disabled
Critic	
Learning rate	1×10^{-5}
LR schedule / warmup	Constant / None
Weight decay	0.1
Value clip range	0.5
Gradient clipping	1.0
Critic warmup	50 steps
Max tokens per GPU	24,576
Precision	BF16
FSDP size	8
Optimizer / parameter offload	Enabled / Enabled
Rollout (vLLM)	
Trajectories per prompt	8
Temperature / top-p / top-k	1.0 / 1.0 / -1
Maximum model length	10,240
Tensor parallelism	1
GPU memory utilization	0.5
Chunked prefill	Enabled
Max batched tokens / sequences	49,152 / 128
Data and Training	
Dataset	DAPO-math (17,398 prompts)
Training batch size	128
Steps per epoch	135
Training epochs / total steps	1 / 135
Maximum prompt / response length	2,048 / 8,192
Reward function	`math_verify` (naive manager)
Hardware	8 GPUs, 1 node
RL Framework	
verl version	0.8.0 and 0.9.0-dev