DATA-FREE ON-POLICY DISTILLATION: HOW FAR CAN WE GO WITHOUT EXTERNAL DATA?

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) is increasingly applied to frontier foundation model post-training. Prior work in this area has largely focused on algorithmic advances, yet it remains unclear how much OPD depends on its training questions and, in particular, how far this dependence can be reduced. Across two representative single-teacher OPD settings, we find that training on eight real prompts yields performance comparable to training on 17k problems, while datasets differing substantially in measured difficulty and initial distillation gap yield similar outcomes. Our analyses suggest two complementary explanations: repeated sampling could allow even a few prompts to expose substantial teacher supervision, while OPD transfers generalizable reasoning patterns beyond question-specific knowledge. Building on these observations, we next propose a data-free on-policy distillation (DF-OPD) setting to investigate whether the system can supply the training questions itself, eliminating the need for external data. With 64 self-generated questions obtained without seed examples, DF-OPD yields performance comparable to full-data OPD in both single-teacher settings. This finding also holds in multi-teacher OPD: across mathematics, code, and instruction following, 1k generated questions achieve performance comparable to training on approximately 7k real post-training examples. We further explore whether OPD can operate even without explicit training questions. The experiments show that this is effective only in limited cases, where the student unexpectedly generates and answers its own questions, thereby reducing the process to an implicit form of DF-OPD. Together, these findings invite a reassessment of the role of training data in on-policy distillation. Code is available at `https://anonymous.4open.science/r/XE8EE`.

1 INTRODUCTION

On-policy distillation (OPD) has become a widely adopted component of post-training pipelines for frontier foundation models (Yang et al., 2025; Xiao et al., 2026; Xu et al., 2026a). Supervised fine-tuning (SFT) provides dense token-level supervision but suffers from exposure bias, as the model is trained on reference prefixes rather than its own generations. Reinforcement learning with verifiable rewards (RLVR) mitigates this mismatch by learning from model-generated trajectories, but typically relies on sparse outcome-level rewards. OPD combines the advantages of both by providing token-level teacher feedback on student-generated trajectories, bringing together on-policy learning and dense supervision (Gu et al., 2024; Agarwal et al., 2024; Lu & Thinking Machines Lab, 2025).

Prior work has largely focused on algorithmic improvements, including supervision design and optimization (Jin et al., 2026; Yang et al., 2026a; Heo et al., 2026; Xing et al., 2026; Xie et al., 2026). Yet it remains relatively unclear how much OPD depends on its training data and, in particular, how far this dependence can be reduced. We investigate this dependence by first examining how sensitive OPD is to its training questions, then asking whether self-generated questions can support effective OPD without external training data, and finally whether explicit questions are needed at all.

How sensitive is OPD to its training questions? We first study two representative single-teacher OPD settings: one with a teacher obtained by RL-tuning the student, and the other with a larger teacher from the same model family. In both settings, eight questions yield performance comparable to training on approximately 17k problems. Datasets with substantially different measured difficulty and *distillation gap* also produce similar outcomes. Two analyses help interpret these observations.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

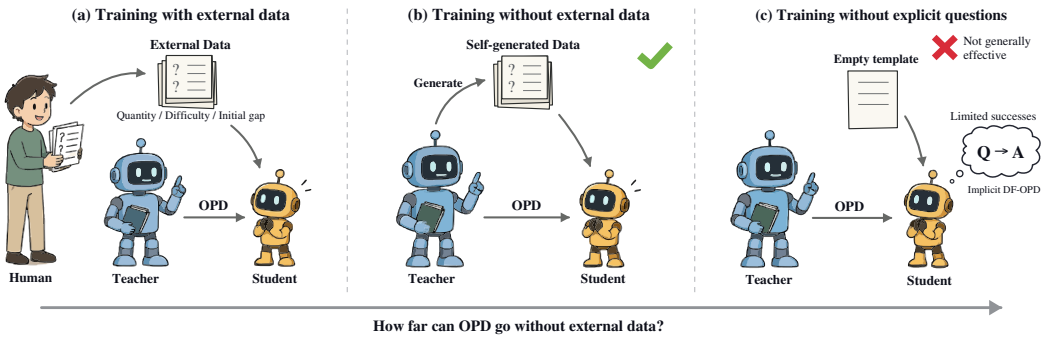

Figure 1: **Reducing dependence on training questions.** (a) OPD uses external questions. (b) DF-OPD uses teacher-generated questions. (c) Empty prompts work only in some tested settings; success is associated with student self-questioning and answering (implicit DF-OPD). OPD arrows denote token-level teacher supervision on student-generated trajectories.

First, repeated sampling allows a few questions to expose substantial teacher supervision, while the additional signal coverage obtained from more questions exhibits diminishing returns. Second, training on competitive-programming questions recovers over 90% of the in-domain gains on mathematics benchmarks, which suggests that OPD can transmit generalizable reasoning patterns beyond knowledge specific to its training questions.

Can self-generated questions replace external data? To answer this question, we propose *Data-free On-Policy Distillation* (DF-OPD), a setting in which the system generates its own training questions, eliminating the need for external question corpora during distillation. Specifically, the teacher generates training questions directly from simple instructions, without external seed examples. With 64 self-generated questions, DF-OPD performs comparably to full-data OPD trained on approximately 17k real questions, with peak macro-average accuracies of 56.61 versus 56.41 in the RL-tuned teacher setting and 49.16 versus 47.36 in the larger teacher setting. In multi-teacher OPD (MOPD), 1k generated questions achieve an overall score of 31.31 across mathematics, code, and instruction following, compared with 31.18 obtained using approximately 7k real post-training examples.

Can OPD work without explicit training questions? We further remove explicit question content from the input supplied to the student, leaving an empty user prompt without task information. Experiments show that this is effective only in limited settings, where the student exhibits unexpected Self-QA behavior: it spontaneously poses a question and attempts to answer it within the same rollout. This amounts to an implicit form of DF-OPD with question generation hidden within student rollouts. Further analysis shows that the effectiveness of empty-prompt OPD is associated with the student's Self-QA frequency before training. In the RL-tuned setting, the Self-QA rate is 62.0%, and empty-prompt OPD retains approximately 95.6% of the mathematics accuracy gain achieved by full-data OPD. In contrast, in the larger-teacher setting, the Self-QA rate is only 0.1%, and empty-prompt OPD retains just 7.4% of the corresponding gain, yielding little improvement over the initial student. This reliance on student-specific behavior limits its generality.

Together, our findings reveal OPD's limited sensitivity to training-data quantity, difficulty, and distillation gap, and clarify how far it can operate independently of external training data. Our experiments point to DF-OPD as a compelling answer: the system supplies its own training questions and supervision while achieving performance comparable to full-data OPD. These results invite a reassessment of the role of external training data in on-policy distillation.

2 Preliminaries and Experimental Setup

2.1 On-Policy Distillation

Given a frozen teacher π_ϕ and a trainable student π_θ, OPD applies teacher supervision to trajectories sampled by the student (Gu et al., 2024; Agarwal et al., 2024). For a training question $x \sim \mathcal{D}$ and

response $y \sim \pi_\theta(\cdot \mid x)$, let $s_t = (x, y_{<t})$ denote the autoregressive state. The reverse-KL objective is

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

A question therefore supplies supervision through the states visited along its rollouts, rather than through a fixed reference answer. In this work, we use a top-k surrogate to balance computational efficiency and supervision density (Li et al., 2026b; Fu et al., 2026a). MOPD, which distills knowledge from multiple domain-specific teachers into a single student, associates each question with a domain label $d(x)$ and replaces π_ϕ by the corresponding teacher $\pi_{\phi_{d(x)}}$. See Appendix A for more details.

2.2 Experimental Setup

Models. We study two representative single-teacher settings: *RL-tuned teacher*, with DeepSeek-R1-Distill-Qwen-1.5B as the student and its RL-tuned counterpart JustRL-DeepSeek-1.5B as the teacher (Guo et al., 2025; He et al., 2025); and *larger teacher*, with Qwen3-1.7B as the student and Qwen3-30B-A3B-Instruct-2507 as the teacher (Yang et al., 2025). For MOPD, we use the released Open-MOPD checkpoints (Gao et al., 2026). The student is SmolLM3-3B-MixSFT, and the teachers are its three domain-specific RL-tuned variants (SmolLM3-3B-RL-Math/Code/IF) for mathematics, code, and instruction following, respectively. See Appendix B.1 for more model details.

Datasets. For single-teacher experiments, DAPO-Math-17k (DAPO) (Yu et al., 2026) serves as the primary source of external training questions. We additionally use a subset of DeepMath (He et al., 2026) and Open-Reasoner-Zero (ORZ) (Hu et al., 2026) to examine sensitivity to the data source, and a subset of DeepCoder (Luo et al., 2025) to study transfer from programming to mathematics. For MOPD, the external-data baseline uses a subset of the RL training prompts released by Open-MOPD. Appendix B.2 details the data sources and preparation. Self-generated questions are introduced in Section 4.1.

Training and Evaluation. All experiments use verl (Sheng et al., 2025) and the AdamW (Loshchilov & Hutter, 2017) optimizer with a learning rate of 10^{-6} and a weight decay of 0.01. Each training batch contains 64 questions. For question pools smaller than a batch, we repeat questions to fill each batch. In the single-teacher settings, we evaluate on six mathematics benchmarks: AIME24 (AI-MO, 2024a), AIME25 (OpenCompass, 2025), AMC23 (AI-MO, 2024b), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), Minerva (Lewkowycz et al., 2022), and Olympiad-Bench (He et al., 2024), using $\mathrm{avg@4}$. For MOPD, we evaluate mathematics on AIME25 and AIME26 (Zhang & Math-AI Team, 2026) ($\mathrm{avg@16}$), code on LiveCodeBench v5 and v6 (Jain et al., 2025) ($\mathrm{avg@5}$), and instruction following on IFEval (Zhou et al., 2023b) and IFBench (Pyatkin et al., 2026); each domain score averages its two benchmarks. In each setting, the main metric is the macro average of all benchmark scores. We report its peak across evaluated checkpoints. Appendix B.3 provides the full training and evaluation configurations.

3 How Sensitive Is OPD to Its Training Data?

OPD obtains teacher supervision along student-generated trajectories, with training questions serving as their starting points. How much does effective distillation depend on the number and properties of these questions? We investigate this across two single-teacher settings, varying question-pool size and comparing data sources that differ in measured difficulty and initial distillation gap.

3.1 Sensitivity to the Number of Training Questions

We randomly sample DAPO subsets of size $|D| \in \{1, 8, 64, 256, 1\mathrm{k}\}$ and compare them with the 17k full-data DAPO in both single-teacher settings, using the same training and evaluation protocol.

As shown in Figure 2, eight questions already yield performance comparable to the 17k full-data DAPO baseline: 56.3 versus 56.4 in the RL-tuned teacher setting and 47.7 versus 47.4 in the larger teacher setting. This represents a more than 2,000-fold reduction in the number of training questions while maintaining comparable performance. We further find that even with a single question, OPD retains approximately 62% and 84% of the full-data gains over the initial students in the RL-tuned

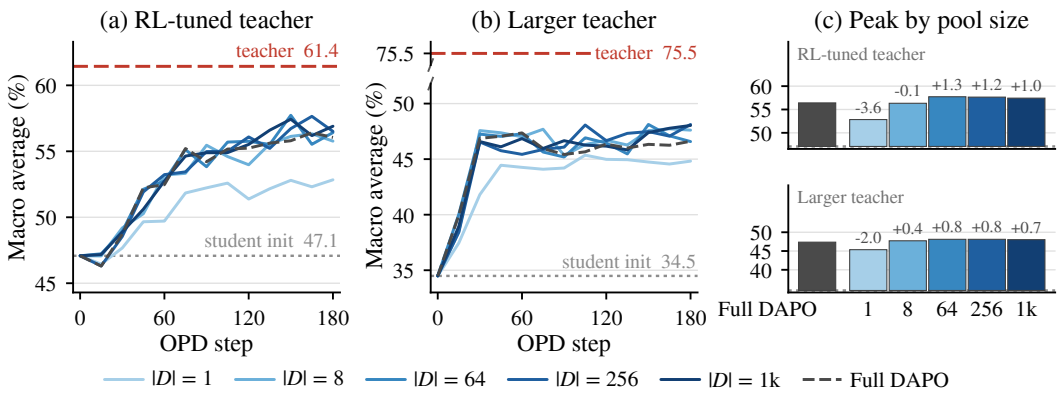

Figure 2: **Few real questions support effective OPD.** (a,b) DAPO training curves under matched update and trajectory budgets. Macro accuracy (%) is averaged over six mathematics benchmarks, evaluated every 15 updates. (c) Peak scores; bars start at the initial student, and numbers give the difference from full DAPO in percentage points.

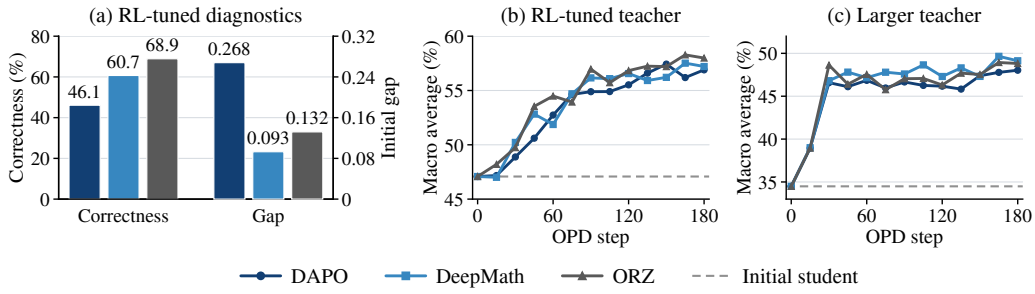

Figure 3: **Initial diagnostics and learning curves.** (a) Initial correctness (left axis) and initial distillation gap (right axis) in the RL-tuned teacher setting. (b,c) Six-benchmark macro accuracy for the RL-tuned and larger-teacher pairings with 1,000 questions per source.

teacher and larger teacher settings, respectively. These results show that even very few training questions can suffice for effective OPD.

3.2 Sensitivity to Data Difficulty and Initial Distillation Gap

Beyond data quantity, another question worth examining is how sensitive OPD is to data difficulty and initial distillation gap. Here, difficulty is assessed using initial student correctness, while the initial distillation gap measures the KL-based discrepancy (Appendix C.1) between teacher and student predictions along student-generated trajectories before training. Both metrics are evaluated in the RL-tuned teacher setting. To examine this sensitivity, we compare DAPO, DeepMath, and ORZ in both single-teacher settings, using the subsets of size $|D| = 1\text{k}$ randomly sampled from each source.

As shown in Figure 3(a), initial correctness is 46.1%, 60.7%, and 68.9% for DAPO, DeepMath, and ORZ, respectively, while the corresponding initial distillation gaps are 0.268, 0.093, and 0.132. Despite these differences, including a nearly threefold difference in the initial distillation gap, the three sources yield similar peak performance in both settings: 57.4–58.3% in the RL-tuned teacher setting and 48.0–49.7% in the larger teacher setting (Figure 3(b,c)). These results suggest that OPD is relatively insensitive to data difficulty and initial distillation gap.

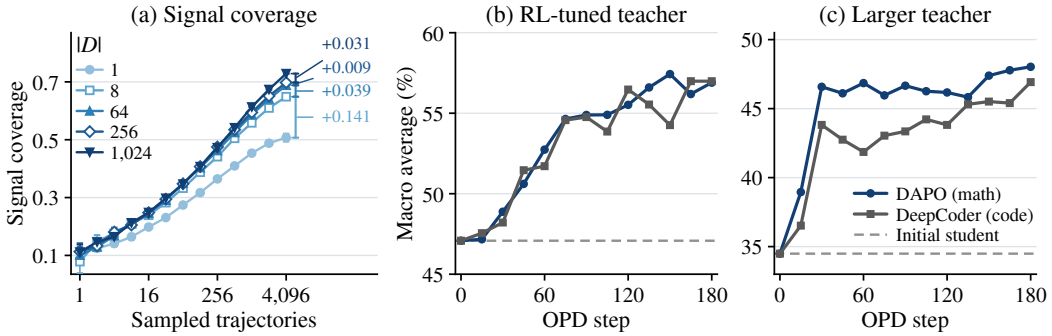

Figure 4: **Signal coverage and cross-domain transfer.** (a) Signal coverage grows with repeated sampling. Mean ± std over six resampling draws of the frozen initial student; brackets mark gains between adjacent pool sizes at 4,096 trajectories. (b,c) Transfer from programming to mathematics for the RL-tuned and larger-teacher pairings with $|D| = 1{,}000$ questions. Macro accuracy on the same six mathematics benchmarks; dashed lines mark initial students.

3.3 Analysis

We analyze OPD's limited sensitivity to data quantity, difficulty, and initial distillation gap from two complementary perspectives.

A Few Questions Can Expose Substantial Teacher Supervision. OPD receives teacher feedback at the states visited along student-generated trajectories. Repeated on-policy sampling from the same question can expose different prefixes and thereby yield different supervision signals. We quantify this using *signal coverage*, which measures how well supervision signals from sampled trajectories represent those in a held-out reference set. Specifically, we represent the supervision signal at each state by a normalized vector combining the student's hidden representation with a correction signal derived from a KL-based discrepancy between the teacher and student. We then compute coverage by measuring how well the reference signal vectors can be reconstructed as linear combinations of the sampled signal vectors. The score ranges from 0 to 1: higher values indicate better representation of the reference supervision, while lower values indicate that more of it remains unrepresented. See Appendix C.2 for the detailed definition.

For a single question, coverage increases from 0.114 with one trajectory to 0.507 with 4,096 trajectories (Figure 4(a)). In the latter budget, expanding the question pool from one to eight adds 0.141 coverage, whereas expanding it further from eight to 1,024 adds only 0.080. These observations suggest that repeated on-policy sampling can expose substantial teacher supervision from very few questions, while enlarging the question pool yields diminishing coverage gains. This offers an explanation for why a small question pool can support effective OPD.

OPD Can Transfer Reasoning Capabilities Beyond Dataset-Specific Knowledge. To help explain OPD's limited sensitivity to training data sources, we take the comparison a step further by replacing DAPO mathematics questions with out-of-domain coding problems from DeepCoder. These problems draw on different domain knowledge but also require reasoning. Both conditions use $|D| = 1\text{k}$ questions for training, while evaluation remains on the same six mathematics benchmarks.

Despite the change in training domain, OPD retains 96% and 92% of the gains achieved with DAPO over the initial students in the RL-tuned teacher and larger teacher settings, respectively (Figure 4(b,c)). The corresponding peak mathematics accuracies are only 0.44 and 1.11 percentage points below those obtained with DAPO. These results suggest that OPD can transfer reasoning capabilities from the teacher beyond dataset-specific knowledge, helping explain its limited sensitivity to training data sources.

4 CAN SELF-GENERATED QUESTIONS REPLACE EXTERNAL DATA?

The preceding experiments show that few questions can support effective OPD and that OPD is relatively insensitive to the data source. This raises a further question: can the distillation system generate its own training questions, eliminating the need for external data? We introduce DF-OPD and examine its effectiveness in both single-teacher and multi-teacher settings.

4.1 DATA-FREE ON-POLICY DISTILLATION

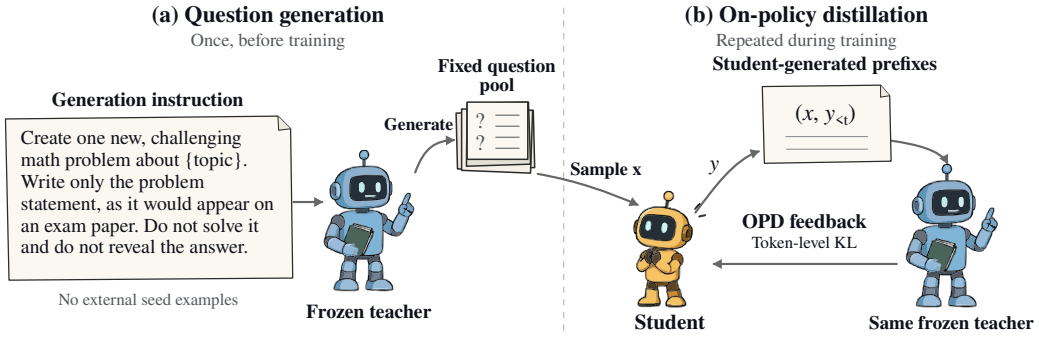

Figure 5: **DF-OPD workflow.** A frozen teacher generates a fixed question pool from a simple instruction. The student samples responses, and the same teacher provides token-level feedback on student-generated prefixes. Standard OPD updates only the student. The instruction and question shown are illustrative.

In DF-OPD, the teacher supplies both training questions and distillation supervision. Specifically, before distillation begins, we use simple instructions specifying the domain and topic to prompt the teacher to generate a fixed question pool, without external seed examples. The student then samples responses to these questions, and the frozen teacher provides token-level feedback on student-generated prefixes to update the student parameters (Figure 5). The standard OPD objective remains unchanged; only the training questions are replaced with those generated by the teacher itself.

We apply only question extraction, format checks, and deduplication, without verifying correctness or filtering questions based on difficulty or solution-quality scores. Full generation instructions and processing details appear in Appendix E.1. Here, *data-free* means that the distillation system supplies its own training questions and supervision, allowing it to operate without external data.

We first evaluate DF-OPD in the two single-teacher settings. Each teacher generates its own training questions, and we examine pools of size $|D| \in \{1, 8, 64, 1\text{k}\}$. We compare these pools with OPD trained on the full DAPO dataset under matched training and evaluation protocols.

Self-generated questions can replace external training data in both single-teacher settings. As shown in Figure 6, just 64 generated questions yield performance comparable to full-data OPD. In the RL-tuned teacher setting, DF-OPD and full-data OPD score 56.61 and 56.41, respectively. In the larger teacher setting, the corresponding scores are 49.16 and 47.36, with DF-OPD even slightly outperforming the full-data baseline. These results show that, in both settings, the teacher can supply not only distillation supervision but also training questions that support effective distillation. Complete results appear in Appendix E.3.

4.2 EXTENDING DF-OPD TO MULTIPLE TEACHERS

We further extend DF-OPD to the multi-teacher setting to examine whether self-generated questions can support capability transfer from multiple domain experts to a single student. Specifically, the mathematics, code, and instruction-following teachers each generate questions in their respective domains and supervise the student's trajectories on those questions.

We compare DF-OPD with a MOPD baseline trained on 7,458 (1,540/1,870/4,048) real post-training questions, where the counts in parentheses correspond to mathematics, code, and instruction fol-

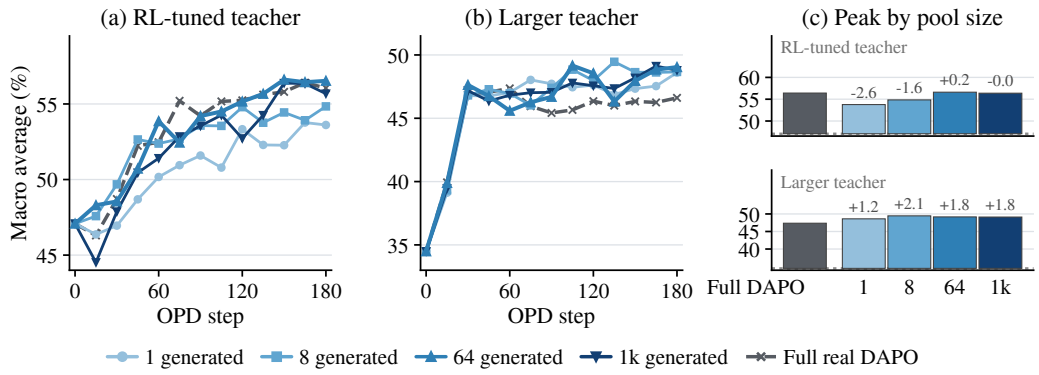

Figure 6: **Self-generated questions replace the external corpus.** (a,b) Single-teacher OPD with different numbers of generated questions, compared with training on the full real DAPO corpus. (c) Peak scores; bars start at the initial student, and numbers give the difference from full real DAPO in percentage points. Sixty-four questions reach comparable performance in both pairings. Complete results appear in Appendix E.3.

Table 1: **Self-generated questions in multi-teacher distillation.** Scores are domain means and their unweighted Overall average. Recovery is relative to the routed teachers (Equation 2). All trained rows use the peak-Overall checkpoint. Full results and domain allocations appear in Appendix E.4.

Input / reference	Questions	Math	Code	IF	Overall	Recovery
Initial student	—	15.83	15.77	41.01	24.21	—
Routed teachers	—	23.02	22.52	48.73	31.42	100%
Real post-training data	7,458	23.02	21.71	48.80	31.18	96.6%
Self-generated	8	23.44	19.64	45.92	29.66	75.6%
Self-generated	64	22.60	20.87	47.51	30.33	84.8%
Self-generated	256	23.54	20.99	47.86	30.80	91.3%
Self-generated	1,000	23.85	21.17	48.91	**31.31**	**98.5%**

lowing, respectively. The generated pools contain 8 (2/2/4), 64 (13/16/35), 256 (53/64/139), and 1k (206/251/543) questions, preserving the real-data domain proportions as closely as possible.

In addition to the overall score, we report *recovery*, which measures the fraction of the initial student–teacher performance gap recovered through distillation:

$$\text{Recovery}(S) = \frac{S - S_0}{T - S_0} \times 100\%, \tag{2}$$

where S and S_0 are the overall scores of the distilled and initial students, respectively, and T averages each domain teacher's score on its own domain. All scores use the same evaluation protocol.

One thousand self-generated questions achieve overall performance comparable to real-data MOPD. As shown in Table 1, DF-OPD achieves an overall score of 31.31, compared with 31.18 for the real-data baseline, with recovery rates of 98.5% and 96.6%, respectively. Across domains, self-generated questions yield slightly higher scores in mathematics and instruction following and a slightly lower score in code. These results show that self-generated questions support both capability transfer from a single teacher and distillation from multiple domain experts into a single student.

MOPD is more sensitive to the size of the generated question pool than single-teacher distillation. With 8, 64, 256, and 1k generated questions, recovery reaches 75.6%, 84.8%, 91.3%, and 98.5%, respectively, showing clear gains at each increase in pool size. In contrast, single-teacher DF-OPD already matches the full-data baseline with 64 generated questions.

This difference may reflect two factors. First, the total question pool in MOPD is divided among three domains: a pool of eight questions leaves only two, two, and four questions per domain, reducing the

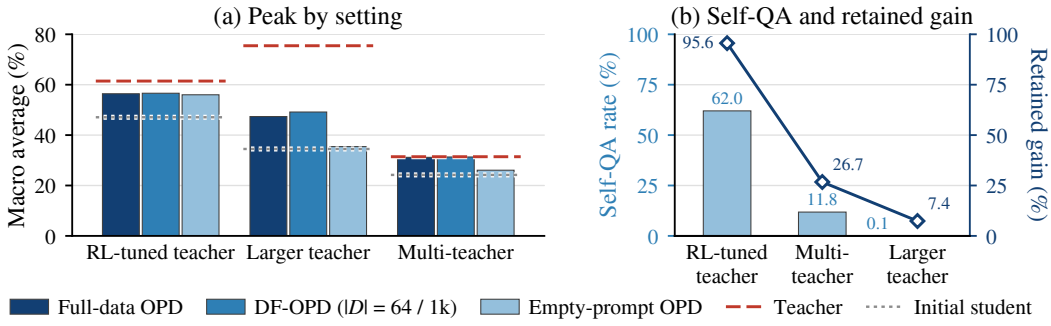

Figure 7: **Omitting questions has a setting-dependent effect.** (a) Peak scores: six-benchmark mathematics averages (single-teacher; DF-OPD with 64 questions) or three-domain averages (MOPD; DF-OPD with 1k). (b) Self-QA rate of each initial student (1,024 empty-prompt rollouts, setting-specific domain criteria; Appendix F.2) and the full-data gain retained by empty-prompt OPD.

number of training questions available within each domain. Second, potential interference across domains (Chen et al., 2026; Gao et al., 2026; Ma et al., 2026; Hou et al., 2026a) may place greater demands on within-domain signal coverage. A larger question pool may therefore help meet these coverage needs and better support joint distillation from multiple teachers. Complete benchmark results, domain allocations, and training curves appear in Appendix E.4.

Further Analyses of DF-OPD. Additional experiments show that 64 generated questions also support effective single-teacher distillation from each of the three domain experts, achieving most of the gains obtained with real-data training (Appendix E.5). We further assess sensitivity to topic guidance by removing explicit topic specifications from the question-generation prompts while keeping other settings unchanged. Across the two main single-teacher settings and MOPD, the resulting score changes range from -0.60 to $+1.50$ percentage points, suggesting that DF-OPD remains effective without explicit topic guidance and is relatively insensitive to this aspect of prompt design (Appendix E.6).

5 Can OPD Work Without Explicit Training Questions?

The preceding section shows that the system can supply its own training questions, eliminating the need for external data. We now ask whether OPD remains effective if explicit questions are removed.

5.1 Distillation with Empty Prompts

We construct empty prompts using the student's chat template, with no user or system prompt content, following Fu et al. (2026b), to examine whether OPD remains effective without explicit training questions. In both single-teacher settings, the student samples responses from its corresponding empty prompt and receives supervision from its teacher. Under MOPD, all domains use the same empty prompt, and domain labels route trajectories to the corresponding domain teachers. The proportions of mathematics, code, and instruction-following labels match those in the full dataset, at $20.6 : 25.1 : 54.3$, to avoid confounding the comparison with changes in domain proportions. Exact input templates are provided in Appendix F.1.

Empty-prompt OPD retains most of the full-data gains only in some settings. As shown in Figure 7, empty-prompt OPD reaches 56.00 in the RL-tuned teacher setting, close to full-data OPD and DF-OPD, retaining approximately 95.6% of the gain achieved by full-data OPD over the initial student. However, its scores reach only 35.44 and 26.07 in the larger teacher and multi-teacher settings, retaining only approximately 7.4% and 26.7% of the corresponding full-data gains, respectively.

5.2 Empty-Prompt Rollouts as Implicit DF-OPD

To understand this difference, we inspect student behavior under empty prompts and observe an unexpected phenomenon in some models: **self-questioning and answering (Self-QA)**. Without receiving a user question, these models pose a question themselves and then attempt to answer it within the same rollout.

To quantify this behavior, we sample 1,024 empty-prompt rollouts from each student before training and measure the fraction exhibiting Self-QA. The Self-QA rate reaches 62.0% for the R1-Distill student, compared with 11.8% for SmolLM3 and 0.1% for Qwen3. Across these students, higher Self-QA rates coincide with greater retention of full-data gains: the three students retain 95.6%, 26.7%, and 7.4%, respectively (Figure 7(b)). Sampling and classification details are in Appendix F.2.

Appendix F.3 presents representative examples: R1-Distill poses an asymptote problem and attempts to solve it under an empty prompt, whereas Qwen3 asks the user to provide a question. In such Self-QA rollouts, the student's own question supplies task content for the subsequent solution attempt and elicits teacher supervision. This process can be viewed as **implicit DF-OPD**, with question generation embedded in the student's sampling process.

Whereas Self-QA arises inconsistently across students, DF-OPD uses teachers to explicitly generate questions, allowing control over question counts, domains, and their mixture without relying on spontaneous student behavior. These results show that, in the settings studied, OPD can dispense with external training data and achieve performance comparable to full-data OPD using only questions generated and supervision provided by the system itself. Further removing explicit questions, as with empty prompts, does not work as consistently across settings.

6 Related Work

Existing work on OPD studies supervision and optimization (Song & Zheng, 2026), token importance (Xu et al., 2026b), and alignment over high-probability token supports (Li et al., 2026b). Data-centric work investigates difficulty-aware balancing (Hou et al., 2026a) and reliability-aware prompt ordering (Zhu et al., 2026). Concurrent studies also reveal OPD's limited sensitivity to training data (Fu et al., 2026b; Hou et al., 2026b; Li et al., 2026c). Our work systematically examines training-data quantity, dataset-level difficulty, and initial distillation gap, and analyzes these phenomena through supervision signal coverage and the cross-domain transfer of reasoning capabilities. We further investigate how far OPD's dependence on external training data can be reduced, introduce DF-OPD, and demonstrate its effectiveness in single-teacher and multi-teacher settings across multiple task domains. We also examine the complete removal of explicit training questions, finding that effective empty-prompt distillation relies on an implicit form of DF-OPD. Beyond OPD, small datasets also support alignment and reasoning through SFT (Zhou et al., 2023a; Ye et al., 2025; Muennighoff et al., 2025) and RL (Li et al., 2025; Wang et al., 2026). Additional related work on synthetic data and multi-teacher distillation appears in Appendix H.

7 Conclusion

We examined how far OPD's dependence on training questions can be reduced. Across two single-teacher pairings, few questions and datasets with different difficulty and initial distillation gaps yield comparable outcomes. Signal-coverage and cross-domain analyses suggest that repeated sampling exposes substantial supervision and that transferred patterns extend beyond question-specific knowledge. Self-generated questions also support effective single- and multi-teacher distillation without external question corpora. Omitting explicit questions exposes a boundary: success varies across the examined settings and can coincide with student self-questioning, an implicit form of DF-OPD. Our experiments cover a limited range of teacher–student combinations, model scales, task domains, and training budgets, and the number of questions required may vary with these factors; future work should evaluate larger models, broader tasks, and longer training. Moreover, question efficiency does not imply computational efficiency, as OPD still requires repeated student rollouts and teacher feedback; improving computational efficiency is another direction for future work. Together, these results invite a reassessment of the role of training data in on-policy distillation.

AI USE STATEMENT

We used generative AI tools to assist with programming and debugging, and to polish the writing of the manuscript for readability. The authors reviewed all AI-assisted content, such as code and edited text, and take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI. The teacher-generated training questions in DF-OPD (Section 4) are part of the studied method rather than assistance in preparing this work.

ETHICS STATEMENT

This work aims to understand and reduce the training-data requirements of on-policy distillation. Our experiments use existing models and datasets together with teacher-generated questions, focusing on mathematics, programming, and instruction following. Reducing reliance on external training-question corpora may lower the burden of data collection and curation and make distillation research more accessible. We encourage responsible use of the resulting models in accordance with applicable usage terms and appropriate evaluation practices.

REPRODUCIBILITY STATEMENT

We are committed to the reproducibility of our results. All teacher and student models are publicly released checkpoints, and all external training questions and benchmarks come from public sources. Our code is available in the anonymous repository linked in the abstract and will be open-sourced upon acceptance.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

AI-MO. Aime 2024. https://huggingface.co/datasets/AI-MO/aimo-validation-aime, 2024a.

AI-MO. Amc 2023. https://huggingface.co/datasets/AI-MO/aimo-validation-amc, 2024b.

Tianlei Chen, Jiao Ou, Ziyuan Liu, Ruiming Tang, Jian Liang, and Han Li. Counteraction-aware multi-teacher on-policy distillation for general capability recovery with domain preservation. *arXiv preprint arXiv:2605.27115*, 2026.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. In *Third Conference on Language Modeling*, 2026a. URL https://openreview.net/forum?id=uZbZb058bA.

Zixuan Fu, Bingxiang He, Yuxin Zuo, Haohuan Huang, Jinqian Zhang, Ruhang Xiao, Cheng Qian, Qinyu Luo, Huan-ang Gao, Yudong Wang, et al. Rethinking on-policy distillation of large language models ii: One training example. *arXiv preprint arXiv:2609.04172*, 2026b.

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

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.

Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, et al. Justrl: Scaling a 1.5 b llm with a simple rl recipe. *arXiv preprint arXiv:2512.16649*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 138306–138322, 2026.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Byeongho Heo, Jaehui Hwang, Sangdoo Yun, and Dongyoon Han. On-policy delta distillation. *arXiv preprint arXiv:2607.15161*, 2026.

Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, et al. Uni-opd: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*, 2026a.

Zhinan Hou, Jiaqi Zhang, Xunliang Cai, and Keyou You. What matters in on-policy distillation? a perspective on data efficiency and data selection. *arXiv preprint arXiv:2609.05198*, 2026b.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Advances in Neural Information Processing Systems*, 38:162239–162262, 2026.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. In *International Conference on Learning Representations*, volume 2026, pp. 130770–130790, 2026.

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

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=J5i09faOOf.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Gengsheng Li, Jinghan He, Shijie Wang, Dan Zhang, Ruiqi Liu, Renrui Zhang, Zijun Yao, Junfeng Fang, Haiyun Guo, and Jinqiao Wang. R-diverse: Mitigating diversity illusion in self-play llm training. *arXiv preprint arXiv:2602.13103*, 2026a.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling. *arXiv preprint arXiv:2502.11886*, 2025.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. 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*, 2026b. URL https://openreview.net/forum?id=aaDDXXDGXB.

Zhaoyi Li, Deyang Kong, Yuan Wei, Evan Yang, Ranran Shen, Mahardika Krisna Ihsani, Ming Yang, Wei Zhang, Chuan Hao, Jian Yang, et al. Every coin has two sides: On the dual nature of generalization in on-policy distillation of large language models. *arXiv preprint arXiv:2608.16647*, 2026c.

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.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, Ce Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepcoder: A fully open-source 14b coder at o3-mini level. `https://pretty-radio-b75.notion.site/DeepCoder-A-Fully-Open-Source-14B-Coder-at-O3-mini-Level-1cf81902c14680b3bee5eb349a512a51`, 2025. Notion Blog.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, et al. Mopd: Multi-teacher on-policy distillation for capability integration in llm post-training. *arXiv preprint arXiv:2606.30406*, 2026.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

OpenCompass. Aime 2025. `https://huggingface.co/datasets/opencompass/AIME2025`, 2025.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hanna Hajishirzi. Generalizing verifiable instruction following. *Advances in Neural Information Processing Systems*, 38, 2026.

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. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

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

Jie Sun, Mao Zheng, Mingyang Song, Zeyuan Liu, Gengsheng Li, Houcheng Jiang, Yilin Cheng, Bichuan Feng, Yuchen Cai, Junfeng Fang, et al. Distill what you trust: Reliability-aware multi-teacher on-policy distillation. *arXiv preprint arXiv:2609.23697*, 2026.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *Advances in Neural Information Processing Systems*, 38: 122721–122764, 2026.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023.

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.

Zhengpeng Xie, Li Lyna Zhang, Zeke Xie, and Mao Yang. Trust region policy distillation. *arXiv preprint arXiv:2607.04751*, 2026.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust region on-policy distillation. *arXiv preprint arXiv:2606.01249*, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026a.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *International Conference on Learning Representations*, volume 2024, pp. 30745–30766, 2024.

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*, 2026b.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. In *International Conference on Learning Representations*, volume 2025, pp. 76346–76382, 2025.

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

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

Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, Sheng-Chieh Lin, Chankyu Lee, Yangyi Chen, Dongfu Jiang, Jiafan He, et al. Nemotron-cascade 2: Post-training llms with cascade rl and multi-domain on-policy distillation. *arXiv preprint arXiv:2603.19220*, 2026b.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*, 2025.

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.

Yifan Zhang and Math-AI Team. American invitational mathematics examination (aime) 2026. `https://huggingface.co/datasets/math-ai/aime26`, 2026.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=neZSGqhxDa`.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023a.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023b.

Ximo Zhu, Ruiqi Liu, Rong Wang, Ping Wu, Xiang Zheng, Wenzhuo Xu, Xubin Yao, Zhiyuan Yan, Bo Li, Jun Gao, et al. Reorder-opd: Reliability-aware prompt ordering for on-policy distillation. *arXiv preprint arXiv:2608.10905*, 2026.

A OPD TRAINING OBJECTIVE

The main text presents the full-vocabulary reverse KL objective. In practice, we use a stop-gradient surrogate objective over the student's top-k support to provide token-level supervision at each state. Let θ_j denote the student parameters used to generate the current batch of trajectories. Given a question x, we sample a response $y \sim \pi_{\theta_j}(\cdot \mid x)$ and denote the state at response position t by $s_t = (x, y_{<t})$. Let

$$p_j(v \mid s) = \pi_{\theta_j}(v \mid s), \qquad q(v \mid s) = \pi_\phi(v \mid s). \tag{3}$$

For each response state s_t, we define the student's top-k support, the normalized probabilities within this support, and the teacher–student log-probability difference as

$$V_t = \mathrm{TopK}_k(p_j(\cdot \mid s_t)), \tag{4}$$

$$\widetilde{p}_j(v \mid s_t) = \frac{p_j(v \mid s_t)}{\sum_{u \in V_t} p_j(u \mid s_t)}, \qquad v \in V_t, \tag{5}$$

$$\delta_t(v) = \log q(v \mid s_t) - \log p_j(v \mid s_t). \tag{6}$$

Both the support and its normalized weights are determined by the student distribution. The two log-probabilities in δ_t are normalized over the full vocabulary, whereas the multiplicative weights $\widetilde{p}_j$ are renormalized within V_t.

The training loss at each response state is

$$\ell_t(\theta) = -\sum_{v \in V_t} \mathrm{sg}[\widetilde{p}_j(v \mid s_t)\delta_t(v)] \log \pi_\theta(v \mid s_t), \tag{7}$$

where sg denotes the stop-gradient operator. During the update, the support sets, sampled trajectories, and weights inside the brackets are held fixed, while gradients flow through the student's log-probabilities. The implementation first averages the loss over valid response tokens within each microbatch, then accumulates gradients with each microbatch weighted by its share of the total sequence count on the corresponding process, and finally averages gradients across distributed processes.

Teacher routing. In MOPD, the domain label $d(x)$ of each question selects a single teacher for the entire response:

$$q(v \mid s) = \pi_{\phi_{d(x)}}(v \mid s). \tag{8}$$

All domains use the same training objective, with each domain's contribution determined by the sampled responses and their token-level supervision.

B EXPERIMENTAL SETUP

B.1 MODELS

Table 2 lists the five teacher–student relationships used in the two single-teacher settings and the three-domain multi-teacher setting. All models are initialized from eight publicly released checkpoints; their original supervised fine-tuning and reinforcement learning were performed by the model providers.

Teacher–student relationships. JustRL-DeepSeek-1.5B is obtained by GRPO post-training of DeepSeek-R1-Distill-Qwen-1.5B. The larger Qwen3 teacher is a mixture-of-experts model with approximately 30B total parameters and 3B activated parameters; it is not obtained by RL-tuning the 1.7B student. In Open-MOPD, MixSFT is obtained by mixed-domain supervised fine-tuning of SmolLM3-3B-Base. The model provider separately applies domain-specific RL to MixSFT to obtain the three domain teachers, and our MOPD student is initialized from the same MixSFT checkpoint.

B.2 DATASETS

Single-teacher settings. For mathematics data, we use DAPO-Math-17k as released, containing 17,917 records. The prepared DeepMath question pool contains 14,116 records. ORZ preprocessing

Table 2: Teacher and student configurations. Model names refer to publicly released checkpoints. The three MOPD domains share the same student initialization and use the checkpoint collection released by Open-MOPD.

Setting	Student	Teacher	Relationship
RL-tuned teacher	DeepSeek-R1-Distill-Qwen-1.5B	JustRL-DeepSeek-1.5B	RL-tuned from the student
Larger teacher	Qwen3-1.7B	Qwen3-30B-A3B-Instruct-2507	Different-scale models from the Qwen3 family
MOPD: math	SmolLM3-3B-MixSFT	SmolLM3-3B-RL-Math	Math-specific RL from MixSFT
MOPD: code	SmolLM3-3B-MixSFT	SmolLM3-3B-RL-Code	Code-specific RL from MixSFT
MOPD: instruction following	SmolLM3-3B-MixSFT	SmolLM3-3B-RL-IF	Instruction-following RL from MixSFT

includes deduplication, removal of identified records overlapping with evaluation benchmarks, answer-field cleanup, and length filtering, yielding 52,738 records.

For out-of-domain data, we preprocess DeepCoder and further filter out mathematics-related problems. We first exclude function-signature-style problems and apply deduplication and length filtering, yielding 12,492 problems. We then construct a variant with mathematics-related content removed: we retain only TACO problems that have topic annotations but no mathematics-related topic tags, and remove problems that an LLM judge classifies as essentially mathematical or that exceed the prompt length limit. This yields a final pool of 2,848 problems.

Multi-teacher setting. Following the data construction procedure of TrustMOPD (Sun et al., 2026), we sample from the `rl_prompt_mix` data released by Open-MOPD using random seed 42. The resulting pool contains 1,540 mathematics records, 1,870 code records, and 4,048 instruction-following records, for a total of 7,458 real training examples. Distillation uses the question text and, in MOPD, the domain label for teacher routing.

Few-shot question pools. For all few-shot experiments, we randomly sample the required number of questions from the corresponding dataset. For MOPD, we approximately preserve the mathematics, code, and instruction-following proportions of the real-data pool: 20.6%, 25.1%, and 54.3%, respectively.

B.3 Training and Evaluation Protocol

Training configuration. In the single-teacher settings, we follow Rethinking-OPD (Li et al., 2026b) and set the student support size to $k = 16$. In the multi-teacher setting, we follow TrustMOPD (Sun et al., 2026) and use $k = 128$. Table 3 provides the training configuration. Objective normalization and teacher routing are described in Appendix A.

Evaluation configuration. Single-teacher mathematics evaluation uses $\mathrm{avg@4}$, with a sampling temperature of 0.7 following Rethinking-OPD (Li et al., 2026b). For MOPD, mathematics and code evaluation use $\mathrm{avg@16}$ and $\mathrm{avg@5}$, respectively, while instruction-following evaluation uses one response per problem. Following TrustMOPD (Sun et al., 2026), we use a sampling temperature of 0.6 for all three domains. Table 4 provides the evaluation configuration.

Checkpoint selection. We save checkpoints every 15 updates in the two single-teacher settings and every 20 updates in MOPD, and evaluate the saved checkpoints. For each experiment, we report the highest macro-average score across all evaluated checkpoints. For MOPD, the mathematics, code, and instruction-following scores are all taken from the same checkpoint with the highest Overall score.

Table 3: Training configurations for the single-teacher and multi-teacher settings. The student support size k is used to compute the distillation objective and is distinct from decoding top-k during generation; decoding top-k is disabled in both settings.

Configuration	Single teacher	Multiple teachers
Optimization updates	180	120
Questions per update	64	64
Student support size k	16	128
Policy passes per rollout batch	1	1
Optimizer	AdamW	AdamW
Learning rate	10^{-6}	10^{-6}
AdamW β_1, β_2	0.9, 0.999	0.9, 0.999
Weight decay	0.01	0.01
Learning-rate schedule	Constant; no warmup	Constant; no warmup
Gradient-norm clipping	1.0	1.0
Sampling temperature / top-p	1.0 / 1.0	1.0 / 1.0
Decoding top-k	Disabled (-1)	Disabled (-1)
Actor dtype	FP32	FP32
Rollout dtype	BF16	BF16
Parameter sharding / rollout TP	FSDP / 2	FSDP / 2

Table 4: Evaluation configurations for the single-teacher and multi-teacher settings. Responses per problem denotes the number of independently sampled responses for each evaluation problem.

Configuration	Single-teacher math	MOPD math	MOPD code	MOPD IF
Responses per problem	4	16	5	1
Sampling temperature	0.7	0.6	0.6	0.6
Top-p	0.95	0.95	0.95	0.95
Response limit (tokens)	16,384	30,000	32,768	32,768

C Metric Definitions

C.1 Initial Distillation Gap

The initial distillation gap measures the discrepancy between teacher and student predictions along trajectories generated by the initial student before training. Let $p_0 = \pi_{\theta_0}$ denote the initial student distribution and $q = \pi_\phi$ the teacher distribution. For state s_{it} at position t in response i, let V_{it} contain the k tokens with the highest student probabilities. We define the student probabilities renormalized over this support as

$$\widetilde{p}_0(v \mid s_{it}) = \frac{p_0(v \mid s_{it})}{\sum_{u \in V_{it}} p_0(u \mid s_{it})}, \qquad v \in V_{it}. \tag{9}$$

The initial distillation gap is defined as

$$G_0 = \frac{1}{N} \sum_{i,t} M_{it} \sum_{v \in V_{it}} \widetilde{p}_0(v \mid s_{it}) \left[\log p_0(v \mid s_{it}) - \log q(v \mid s_{it})\right], \tag{10}$$

where M_{it} masks valid response tokens and $N = \sum_{i,t} M_{it}$. This metric first sums over the student top-k candidates at each state and then averages over all valid response tokens. We compute it in the RL-tuned teacher setting with $k = 16$, keeping the student fixed at its initialization.

Both log-probabilities in the formula are normalized over the full vocabulary. We use this metric to characterize the initial prediction discrepancy induced by different data sources under a given teacher–student setting.

Initial correctness. To measure data difficulty for the initial student, we use the 1,000-question training pool from each data source in the data-quantity experiments. We sample eight responses per question with a temperature of 0.7 and a response length limit of 16,384 tokens. Initial correctness is the average correctness of these responses; lower values indicate greater difficulty for the initial student.

C.2 SIGNAL COVERAGE

Signal coverage measures how well supervision signals obtained from sampled student trajectories represent those in a held-out reference set. We conduct this analysis in the RL-tuned teacher setting, keeping the initial student parameters fixed throughout to examine the effects of question-pool size and repeated sampling on signal coverage.

State-level supervision signal. For a state s, let the student and teacher next-token distributions be

$$p_s = \pi_{\theta_0}(\cdot \mid s), \qquad q_s = \pi_\phi(\cdot \mid s), \tag{11}$$

and let $V_s = \mathrm{TopK}_{16}(p_s)$. We first construct a full-vocabulary vector from the teacher–student log-probability differences:

$$a_s(v) = \mathbf{1}\{v \in V_s\} \left[\log q_s(v) - \log p_s(v)\right], \qquad c_s = \sum_{v \in V_s} a_s(v). \tag{12}$$

We define the following auxiliary loss and use its gradient with respect to the student logits as the state-level correction signal:

$$\ell_s^{\mathrm{geom}}(\theta) = -\sum_{v \in V_s} \mathrm{sg}[a_s(v)] \log \pi_\theta(v \mid s), \tag{13}$$

$$g_s = \nabla_{z_s} \ell_s^{\mathrm{geom}}(\theta)|_{\theta=\theta_0} = c_s p_s - a_s, \tag{14}$$

where sg denotes stop-gradient and z_s is the student's full-vocabulary logit vector. Let h_s denote the student's final-layer hidden state and $z_s = W h_s$ its output projection. The corresponding gradient of the auxiliary loss with respect to the output-layer weights factorizes as

$$\nabla_W \ell_s^{\mathrm{geom}} = g_s h_s^\top. \tag{15}$$

We center each factor using its mean over the reference set X and normalize it to unit length:

$$\bar{h}_s = \frac{h_s - \mu_h}{\|h_s - \mu_h\|_2}, \qquad \bar{g}_s = \frac{g_s - \mu_g}{\|g_s - \mu_g\|_2}, \tag{16}$$

where

$$\mu_h = \frac{1}{|X|} \sum_{x \in X} h_x, \qquad \mu_g = \frac{1}{|X|} \sum_{x \in X} g_x. \tag{17}$$

The same reference means are used for all candidate question pools. The unit signal feature for state s is defined as

$$u_s = \mathrm{vec}(\bar{g}_s \bar{h}_s^\top). \tag{18}$$

This feature jointly encodes the student's state representation and the correction direction associated with the teacher–student discrepancy. The signal inner product between two states can be computed directly from the factor inner products, without explicitly constructing the outer products:

$$k(s, s') = \langle u_s, u_{s'} \rangle = \langle \bar{g}_s, \bar{g}_{s'} \rangle \langle \bar{h}_s, \bar{h}_{s'} \rangle. \tag{19}$$

Projection coverage. Let B be the set of sampled states and X the held-out reference set. Signal coverage is defined as the fraction of reference signal energy retained after projection onto the linear span of the sampled signals:

$$\mathrm{Coverage}(B; X) = 1 - \frac{\sum_{x \in X} \|u_x - P_B u_x\|_2^2}{\sum_{x \in X} \|u_x\|_2^2}, \tag{20}$$

where P_B denotes the orthogonal projection onto $\mathrm{span}\{u_s : s \in B\}$. Since the signal features have unit norm, the denominator equals $|X|$.

In practice, we compute coverage using kernel matrices. Let K_{BB} and K_{BX} contain the signal inner products among sampled states and between sampled and reference states, respectively. Then

$$\mathrm{Coverage}(B; X) = \frac{\mathrm{tr}\left(K_{BX}^\top K_{BB}^+ K_{BX}\right)}{|X|}, \tag{21}$$

where K_{BB}^+ is the Moore–Penrose pseudoinverse. When computing the pseudoinverse, we retain only eigenvalues greater than 10^{-8} times the largest eigenvalue. Coverage lies in $[0, 1]$ up to numerical precision: higher values indicate that the sampled signals more fully represent the reference supervision directions.

Reference set and state sampling. We extract eight evenly spaced response positions from each sampled trajectory. The reference set consists of 16,384 trajectories generated from 2,048 held-out questions, yielding 131,072 token states. We use a fixed question permutation with random seed 0 to place reference and candidate questions in disjoint subsets. Question pools for the coverage diagnostic are sampled separately from the training pools.

Trajectory budgets and resampling. We consider question-pool sizes $P \in \{1, 8, 64, 256, 1024\}$ and trajectory sampling budgets ranging from 1 to 4,096. The budget counts sampled trajectories. For a budget b, each draw uses $\min(P, b)$ distinct questions and samples $b/\min(P, b)$ trajectories per question. Repeated occurrences of the same trajectory are collapsed before the geometric computation.

We perform six resampling draws for each configuration, sampling questions and trajectories afresh each time and sampling without replacement within each draw. Samples are not constructed to be nested across draws or budgets. We report the mean and population standard deviation (`ddof=0`) over the six resampling draws.

D Sensitivity to Data Quantity

D.1 Per-Benchmark Results

The main text reports OPD performance with different numbers of training questions using macro-average accuracy across six mathematics benchmarks. Here, we provide the complete training curves for each benchmark. Figures 8 and 9 correspond to the two single-teacher settings, respectively, with each figure showing results on all six benchmarks. Each panel compares different DAPO question-pool sizes with full-data training, providing a benchmark-level view of sensitivity to the number of training questions.

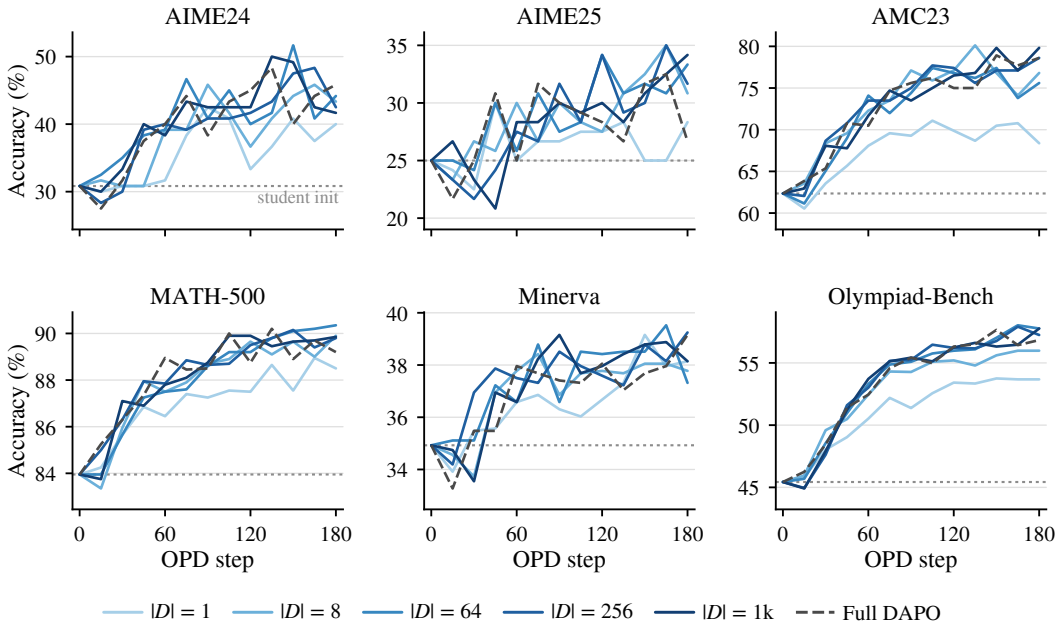

Figure 8: **Per-benchmark OPD training curves in the RL-tuned setting.** The six panels show accuracy on AIME24, AIME25, AMC23, MATH-500, Minerva, and Olympiad-Bench, comparing training on 1, 8, 64, 256, and 1k DAPO questions with training on the full DAPO dataset. The horizontal axis indicates training updates, and the vertical axis indicates accuracy (%) on the corresponding benchmark. All runs use the same number of updates and sampled trajectories per update.

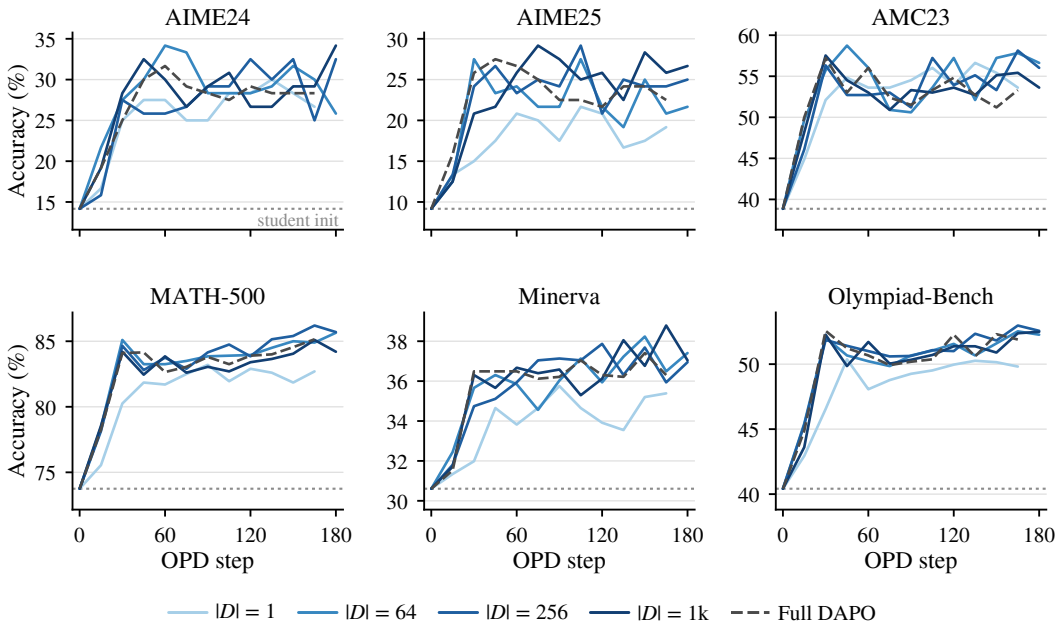

Figure 9: **Per-benchmark OPD training curves in the larger-teacher setting.** The six panels show accuracy on AIME24, AIME25, AMC23, MATH-500, Minerva, and Olympiad-Bench, comparing training on 1, 64, 256, and 1k DAPO questions with training on the full DAPO dataset. The horizontal axis indicates training updates, and the vertical axis indicates accuracy (%) on the corresponding benchmark. All runs use the same number of updates and sampled trajectories per update.

D.2 Data Sensitivity on Additional Datasets

The main text shows that OPD can be effective with only a few training questions from DAPO. To examine whether this finding extends to other datasets, we construct question pools of sizes 1, 64, and 1k from DeepMath and ORZ and conduct experiments in both single-teacher settings. The training and evaluation protocols remain unchanged. Figures 10 and 11 provide the corresponding macro-average training curves for DeepMath and ORZ, respectively.

In both teacher settings, training on 64 questions from DeepMath or ORZ yields peak macro-average mathematics accuracy close to that obtained with 1k questions. Even a single training question from either dataset produces performance gains in both settings. These results show that the effectiveness of OPD with few training questions extends beyond DAPO.

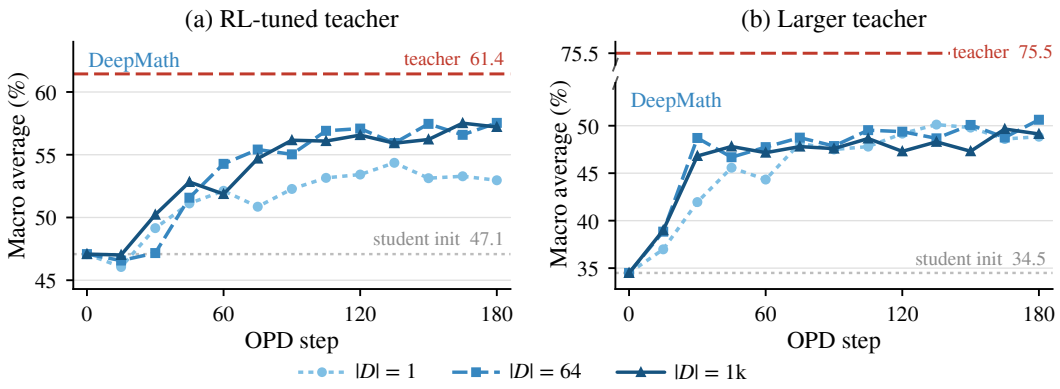

Figure 10: **Macro-average performance after OPD with different numbers of DeepMath questions.** Training curves are shown for the RL-tuned and larger-teacher settings, comparing 1, 64, and 1k training questions. The horizontal axis indicates training updates, and the vertical axis indicates macro-average accuracy (%) over the six mathematics benchmarks. Within each teacher setting, all runs use the same training and evaluation configurations apart from the training question pool.

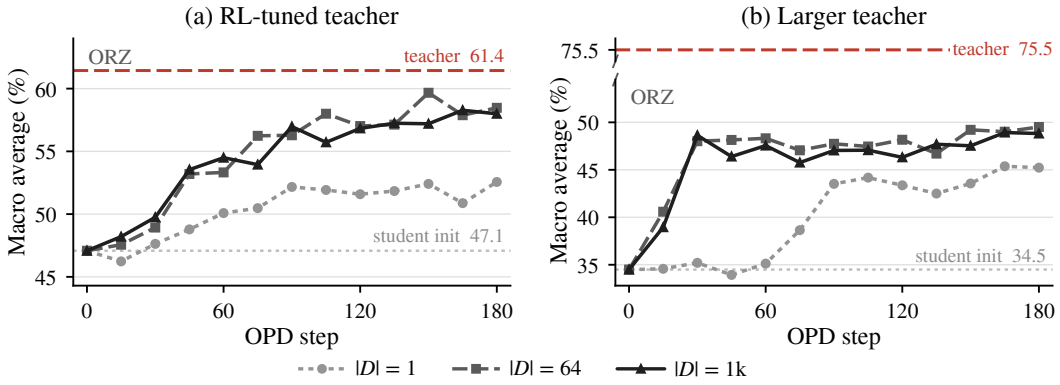

Figure 11: **Macro-average performance after OPD with different numbers of ORZ questions.** Training curves are shown for the RL-tuned and larger-teacher settings, comparing 1, 64, and 1k training questions. The horizontal axis indicates training updates, and the vertical axis indicates macro-average accuracy (%) over the six mathematics benchmarks. Within each teacher setting, all runs use the same training and evaluation configurations apart from the training question pool.

E DF-OPD: Experimental Details and Additional Results

E.1 Question Generation and Data Processing

Before training, DF-OPD uses the teachers to generate a fixed pool of training questions. In the two single-teacher settings, each teacher generates mathematics questions. In MOPD, the mathematics, code, and instruction-following teachers generate questions for their respective domains. Generation uses neither external seed questions nor retrieved content. The `{topic}` field in each generation prompt is randomly selected from the corresponding list below.

Generation prompt for the RL-tuned teacher.

```
Create one new, challenging math problem about {topic}.
Write only the problem statement, as it would appear on an
exam paper.  Do not solve it and do not reveal the answer.
```

Generation prompt for the larger teacher.

```
Create one new, challenging math problem about {topic}.

When you are done thinking, write your reply in exactly this
format and nothing else:

<question>
the complete problem statement
</question>
```

Generation prompt for the MOPD mathematics teacher.

```
Create one new, challenging competition-level math problem
about {topic}.  Write only the problem statement, as it
would appear on an exam paper.  Do not solve it and do not
reveal the answer.
```

Generation prompt for the MOPD code teacher.

```
Create one new, challenging competitive programming problem
about {topic}.  Write only the problem statement, including
the input format, the output format, and the constraints,
exactly as it would appear on a judge.  Do not solve it, do
not write any code, and do not give sample test data.
```

Generation prompt for the MOPD instruction-following teacher.

```
Create one new instruction-following task whose constraints
involve {topic}.  Give two to four explicit, checkable
constraints on the response, then a short request for an
assistant.  Do not carry out the request.
```

We define 16 topics for each of the mathematics, code, and instruction-following domains. The two single-teacher settings and the MOPD mathematics teacher share the same mathematics topic list. The code and instruction-following teachers use topics related to algorithms and response constraints, respectively. The complete lists appear in Table 5.

For each teacher, we issue 1,024 generation requests. Each request randomly selects one of the 16 topics for the generation prompt and samples four outputs, yielding 4,096 raw outputs in total. All teachers use a temperature of 1.0 and top-$p = 0.95$. We then extract the question text and filter out outputs from which no valid question can be extracted, overly short fragments, and restatements of the generation instructions. We perform exact deduplication after normalizing whitespace and letter case, and discard questions longer than 1,024 tokens.

These processing steps ensure that question formatting and input length satisfy the training requirements. They do not verify correctness or select questions based on difficulty or solution quality.

Table 5: **Topics for question generation, with 16 topics per domain.** The mathematics topic list is shared by the two single-teacher settings and the MOPD mathematics teacher. The code and instruction-following lists are used by their respective domain teachers. Each generation prompt replaces `{topic}` with a randomly selected entry from the corresponding list.

Domain	Topics
Mathematics	algebra; geometry; number theory; combinatorics; probability; statistics; calculus; pre-calculus; trigonometry; logic puzzles; sequences and series; inequalities; functional equations; complex numbers; polynomials; graph theory
Code	dynamic programming; greedy algorithms; graph traversal; shortest paths; string processing; binary search; sorting and ordering; data structures; number theory; combinatorics and counting; simulation; two pointers; prefix sums; recursion and backtracking; intervals and scheduling; matrix and grid problems
Instruction following	response length in words; number of sentences; number of paragraphs; required keywords; forbidden words; starting and ending phrases; letter casing; JSON or structured output; bullet-point formatting; title or heading requirements; placeholder tokens; repeating the prompt; language of the response; punctuation restrictions; numbered list requirements; section markers

Table 6: **Teacher question-generation settings and question-pool sizes after processing.** Each generation request samples four outputs. Generation seeds are the random seeds used by the teachers to generate questions. The final question counts are measured after extraction, format checks, deduplication, and length filtering, before subsampling to the pool sizes used in the experiments.

Teacher	Generation requests	Generation seeds	Final questions
RL-tuned teacher	1,024	42	3,822
Larger teacher	1,024	42	3,930
MOPD mathematics teacher	1,024	42	2,971
MOPD code teacher	1,024	42	2,949
MOPD instruction-following teacher	1,024	42	3,082

Teacher-generated answers are not used as training targets. The final training subsets are constructed according to the rules in Appendix B.2. In MOPD, each question's domain label is determined by the teacher that generated it.

Table 6 summarizes the generation settings and the resulting question-pool sizes.

E.2 Examples of Generated Questions

Tables 7–11 present examples generated by the teachers in the two single-teacher settings and the three domain teachers in MOPD. Each table contains five questions retained after filtering, together with the topics used to generate them. All question texts have undergone the extraction and processing steps described in Appendix E.1.

E.3 Detailed Results in Single-Teacher Settings

The main text reports single-teacher DF-OPD performance using macro-average accuracy across six mathematics benchmarks. Here, we provide the complete training curves for each benchmark. Figures 12 and 13 correspond to the two single-teacher settings, respectively, with each figure showing results on all six benchmarks. Each panel compares different generated question-pool sizes with training on the full DAPO dataset. The training and evaluation protocols remain unchanged.

E.4 Detailed Results in the Multi-Teacher Setting

The main text reports MOPD scores for mathematics, code, and instruction following, together with their Overall average. Here, we provide the complete training curves for each domain and further report results on all six benchmarks. Figure 14 compares training on 8, 64, 256, and 1k generated

Table 7: **Examples of mathematics questions generated by the RL-tuned teacher.** Five questions retained after filtering are shown with their corresponding generation topics.

Topic	Question text
number theory	Find all prime numbers p such that p^2 divides $2^p - 2$.
complex numbers	Let z be a complex number satisfying $z^2 + (1+i)z + (2-2i) = 0$. Find $\|z_1\| \times \|z_2\|$.
combinatorics	In how many ways can 8 people be seated around a circular table such that Alice and Bob cannot sit next to each other?
geometry	In a right-angled triangle ABC with right angle at A, the incircle touches AB at D and AC at E. The tangent from C to the incircle meets AB at F. If $AB = 3$, $AC = 4$, find the length of CF.
calculus	A 10-foot ladder is leaning against a vertical wall. The base of the ladder is being pushed away from the wall at a rate of 2 feet per second. When the base is 6 feet away from the wall, how fast is the top of the ladder sliding downward?

Table 8: **Examples of mathematics questions generated by the larger teacher.** Five questions retained after filtering are shown with their corresponding generation topics.

Topic	Question text
geometry	A regular dodecagon (12-sided polygon) is inscribed in a circle of radius 1. Let P be a point on the circumference such that the sum of the squares of the distances from P to each of the 12 vertices of the dodecagon is minimized. What is the minimum value of this sum?
trigonometry	In triangle ABC, let $\angle A = 60^\circ$, and let D be a point on side BC such that AD bisects $\angle A$. Suppose $BD = 2$ and $DC = 3$. If $AD = x$, find the exact value of x^2 using trigonometric identities and the Angle Bisector Theorem.
combinatorics	A group of 10 mathematicians is attending a conference. Each day, they form a circle to discuss a problem, and no two people can sit next to each other more than once during the conference. What is the maximum number of days the conference can last such that this condition is satisfied?
graph theory	Let G be a simple, undirected graph with 12 vertices and 30 edges. Suppose that G has no cycles of length 3 or 4 (i.e., girth at least 5). What is the maximum possible number of vertices of degree 5 or more in G?
precalculus	Let $f(x) = \log_2(x^2 - 4x + 5)$ and $g(x) = \sqrt{x-1}$. Find all real values of x such that $f(g(x)) = 2$, and determine the sum of all such values.

questions with training on 7,458 real questions. Figure 15 provides the corresponding results for each benchmark.

All generated question pools match the domain proportions of the real data as closely as possible. Question counts refer to the total number of questions across the three domains. With 1k generated questions, the model achieves an Overall score of 31.31, comparable to 31.18 obtained with real-data training. The domain-level and benchmark-level training curves provide a more detailed view of the performance changes underlying this aggregate result.

E.5 Single-Teacher Distillation with Domain Experts

To complement the experiments above, we use each of the mathematics, code, and instruction-following teachers from MOPD separately for single-teacher DF-OPD. This examines whether a small number of generated questions can support effective distillation when each domain teacher provides supervision independently. In all three experiments, a separate student is initialized from SmolLM3-3B-MixSFT and trained using only 64 questions ($|D| = 64$) generated by its corresponding teacher, which also provides the supervision. Figure 16 shows the training curves for the three domains and compares them with the corresponding real-data training results.

Table 9: **Examples of questions generated by the MOPD mathematics teacher.** Five mathematics questions retained after filtering are shown with their corresponding generation topics.

Topic	Question text
number theory	What is the least positive integer n for which the equation $(n+2)! + n = (n+1)! + (n+1)$ holds true?
functional equations	Find the function $f : \mathbb{R} \to \mathbb{R}$ such that for all real numbers x, y, z satisfying $x+y+z = 0$, we have $f(x^3) + f(y^3) + f(z^3) = xy(x+y) + xz(x+z) + yz(y+z)$.
calculus	A function $f : \mathbb{R} \to \mathbb{R}$ satisfies the differential equation $3f''(x)+2f'(x)-8f(x) = 0$ with initial conditions $f(0) = 1$ and $f'(0) = 0$. Let $g(x)$ be the inverse function of $f(x)$. Find the value of $g'(-2)$.
sequences and series	What is the smallest positive integer n such that the sum of the first n consecutive positive integers is divisible by 17?
probability	What is the probability that a randomly selected positive integer less than or equal to 1000 has a remainder of 3 when divided by 11?

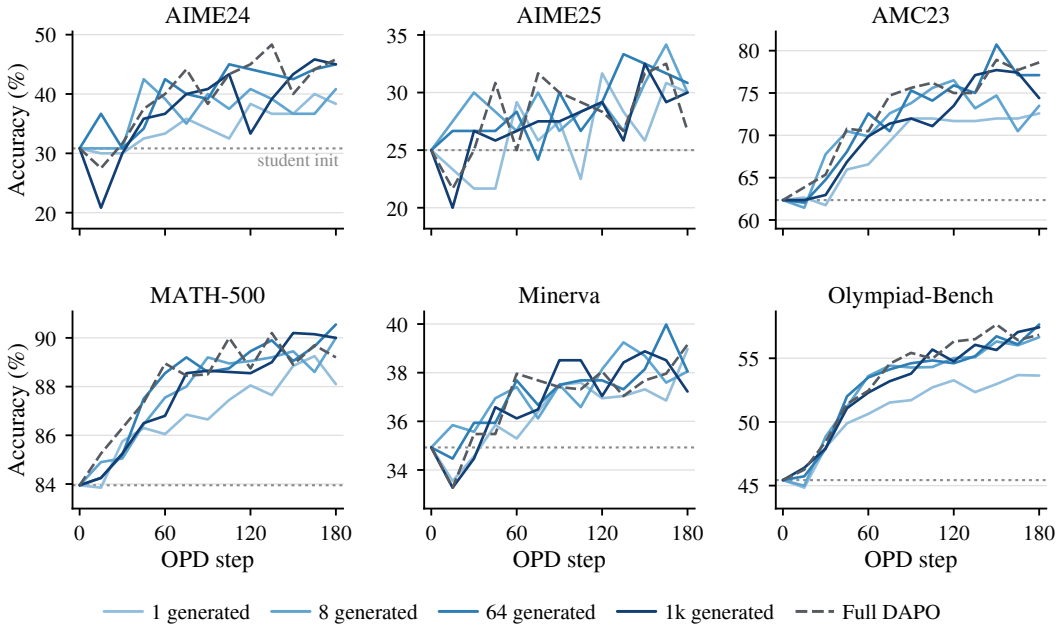

Figure 12: **Per-benchmark DF-OPD training curves in the RL-tuned teacher setting.** The six panels show accuracy on AIME24, AIME25, AMC23, MATH-500, Minerva, and Olympiad-Bench, comparing training on 1, 8, 64, and 1k teacher-generated questions with training on the full DAPO dataset. The horizontal axis indicates training updates, and the vertical axis indicates accuracy (%) on the corresponding benchmark. All runs use the same number of training updates and sampled trajectories per update.

With 64 generated questions, the average scores in mathematics, code, and instruction following are 23.12, 22.05, and 47.77, respectively. The corresponding real-data baselines achieve 24.27, 21.88, and 49.52.

Relative to the initial student (15.83, 15.77, and 41.01), 64 generated questions improve the three domain scores by 7.29, 6.28, and 6.76 points, respectively, whereas the real-data baselines improve them by 8.44, 6.11, and 8.51 points. The generated questions thus slightly exceed the real-data baseline in code, by 0.17 points, but fall short by 1.15 and 1.75 points in mathematics and instruction following. These results indicate that, in these single-domain distillation settings, DF-OPD with only 64 generated questions obtains most of the improvement achieved with real data, although a gap remains in mathematics and instruction following.

Table 10: **Examples of questions generated by the MOPD code teacher.** Five programming problems retained after filtering are shown with their corresponding generation topics.

Topic	Question text
sorting and ordering	**Problem Statement: Lexicographical Ordering of Strings** **Input Format:** - The first line contains a single integer **N** ($1 \leq N \leq 10^5$), denoting the number of strings to sort. - The following **N** lines each contain a string consisting of lowercase English letters only. No string will be empty and no character will appear more than once in a string. **Output Format:** - Print **N** space-separated strings in lexicographical order. Each string must be printed exactly as it appears in the input, and you may not modify the order of strings within the input as long as it matches the sorted order. **Constraints:** - **Time Limit:** 5 seconds - **Memory Limit:** 256 MB - **Judge Logos:** Standard
combinatorics and counting	**Problem Statement** Alice, Bob, and Carol are playing a game using a set of marbles. They want to divide the marbles into three non-empty groups such that the product of the number of marbles in each group is maximized. The constraint is that the number of marbles in each group must be a positive integer, and no two groups can have the same number of marbles. Determine the number of distinct ways to partition the marbles into three such groups. **Input Format** The first line contains a single integer T, the number of test cases. Each subsequent line contains a single integer N, representing the total number of marbles. **Output Format** For each test case, output a single integer representing the number of distinct ways to partition the marbles into three non-empty groups with distinct sizes. **Constraints** - $1 \leq T \leq 10$ - $3 \leq N \leq 1000$ - The total number of marbles across all test cases does not exceed 1000.
graph traversal	**Problem Statement: Longest Path in a Tree** **Input Format:** - The first line contains two integers, N and M, representing the number of nodes and edges in the tree, respectively. - The subsequent M lines describe the edges of the tree. Each line contains two integers, u and v, indicating an undirected edge between nodes u and v. **Output Format:** - Print a single integer, which is the length of the longest path (also known as the diameter) in the tree. The path should consist of exactly two distinct nodes, and the longest path must not include any cycles. **Constraints:** - $2 \leq N \leq 10^5$ - $1 \leq M \leq 2 * 10^5$ - Nodes are numbered from 1 to N.
data structures	**Problem Statement: Weighted Path Reconstruction** **Input Format:** - The first line contains two integers, N and M, representing the number of nodes and edges in the directed weighted graph, respectively. - The next M lines describe the edges. Each line consists of three integers: u, v, and w, indicating a directed edge from node u to node v with weight w. **Output Format:** - Print N integers, where the ith integer represents the minimum possible sum of weights of a path from node 1 to node i. **Constraints:** - $2 \leq N \leq 1000$ - $1 \leq M \leq 5000$ - $1 \leq u, v \leq N$ - $1 \leq w \leq 1000$ - The graph can contain cycles and self-loops.
greedy algorithms	**Problem Statement: Maximum Weight Connected Components** **Input Format:** - The first line contains two integers, N and M, representing the number of nodes and edges in the graph, respectively. - The next M lines each contain three integers, u, v, and w, denoting an undirected edge between nodes u and v with a weight w. **Output Format:** - A single integer, representing the maximum sum of weights of the connected components of the graph. If there are multiple components with the maximum sum, output the maximum sum. **Constraints:** - $2 \leq N \leq 10^5$ - $1 \leq M \leq 10^5$ - $1 \leq u, v \leq N$ - $1 \leq w \leq 10^9$ - All edges are distinct pairs (u, v). - There are no self-loops (`u == v`). - The graph is undirected, meaning if there is an edge between u and v, there is also an edge between v and u.

Table 11: **Examples of tasks generated by the MOPD instruction-following teacher.** Five instruction-following tasks retained after filtering are shown with their corresponding generation topics.

Topic	Question text
response length in words	Please follow these instructions in your response: - 150-200 words long. - Include exactly three paragraphs. - End with the word "Answer". Write a brief description of a cozy reading nook you would love to have in your home. I need help imagining a perfect reading spot.
number of paragraphs	Please follow these instructions in your response: - The answer must consist of exactly three paragraphs. - Each paragraph must begin with the word "Step ". - The response must not contain any numbers other than those that are inherently part of the English language. Write a short story that explores a hidden library where forgotten memories are stored. Please provide the story.
starting and ending phrases	Please follow these instructions in your response: - 1. The response must begin with the phrase **"Begin with the stars."** - 2. The response must end with the phrase **"and the moon."** - 3. The response must include the exact word **"quill"** at least once. - 4. The response must be exactly 50 words in length. Request: Write a short story that blends astronomy with a creative writing tool.
placeholder tokens	Please follow these instructions in your response: - Use exactly three placeholder tokens: **[[TEMP]]**, **[[LOC]]**, and **[[DATE]]**. - Replace each placeholder token with a suitable word or phrase that fits the context. - Ensure the total length of the response is between 20 and 40 words. - Do not include any additional placeholder tokens. Write a short paragraph using the placeholders above, describing a future travel adventure.
title or heading requirements	Please follow these instructions in your response: - 1. Write a title or heading for the response that matches the given keyword "Cosmic". - 2. Include exactly three paragraphs in the answer. - 3. Use bold and italics formatting exactly twice each, within the same paragraph. - 4. Keep the response under 60 words. Describe, in your own words, how the term "Cosmic" might relate to a vast, interconnected network of ideas and data. Provide a clear, concise example of how this concept could be applied in a modern digital environment.

E.6 Sensitivity of DF-OPD to Topic Guidance

To examine the dependence of DF-OPD on explicit topic guidance, we remove the topic specification from the generation prompts and compare distillation performance with and without topic guidance. For mathematics and code, we remove `about {topic}`; for instruction following, we remove `whose constraints involve {topic}`. The output format, generation parameters, data processing pipeline, and other settings remain unchanged.

We compare distillation performance at matched question counts in the two single-teacher settings and in MOPD (Table 12). The single-teacher settings use 8, 64, and 1k questions, while MOPD uses 8, 64, 256, and 1k questions. In MOPD, domain labels are still determined by the generating teacher, and domain proportions remain the same as in the original experiments.

With 64 questions, removing topic guidance yields scores of 56.63 and 49.45 in the two single-teacher settings, corresponding to changes of $+0.02$ and $+0.29$ percentage points relative to the original settings. In MOPD, using 1k questions without topic guidance yields an Overall score of 30.79, a change of -0.52 percentage points.

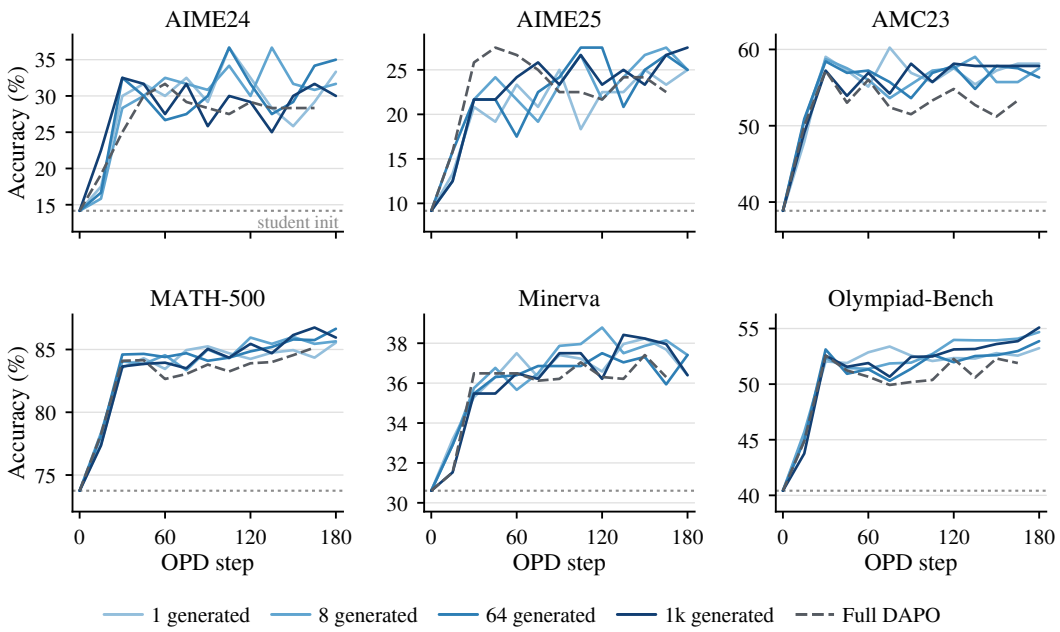

Figure 13: **Per-benchmark DF-OPD training curves in the larger teacher setting.** The six panels show accuracy on AIME24, AIME25, AMC23, MATH-500, Minerva, and Olympiad-Bench, comparing training on 1, 8, 64, and 1k teacher-generated questions with training on the full DAPO dataset. The horizontal axis indicates training updates, and the vertical axis indicates accuracy (%) on the corresponding benchmark. All runs use the same number of training updates and sampled trajectories per update.

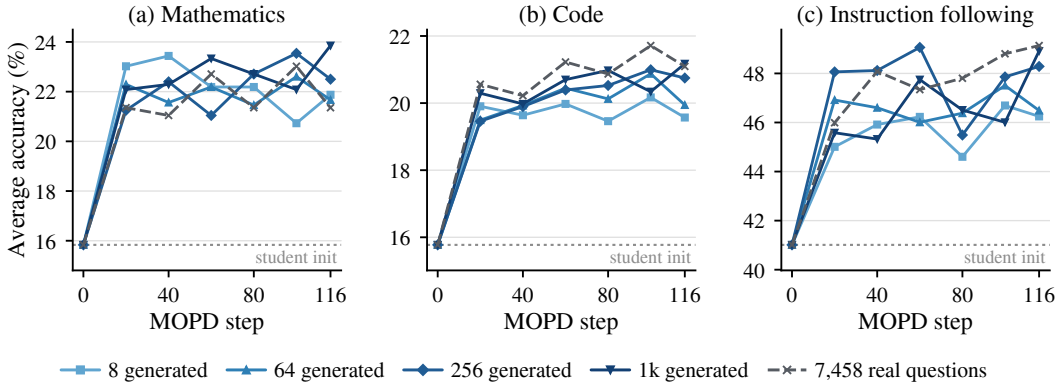

Figure 14: **MOPD training curves across three domains.** The three panels show average accuracy (%) in mathematics, code, and instruction following, comparing training on 8, 64, 256, and 1k generated questions with training on 7,458 real questions. The horizontal axis indicates training updates. Each domain score is the average of its two benchmarks. Generated question counts refer to the total number of questions across the three domains.

Across the ten comparisons, removing topic guidance changes the score by between -0.60 and $+1.50$ percentage points, with increases in four cases and decreases in six. The largest change, an increase of 1.50 points, occurs for the RL-tuned teacher with 8 questions, and all other changes are within 1 percentage point. Thus, in the settings we study, DF-OPD remains effective without explicit topic guidance.

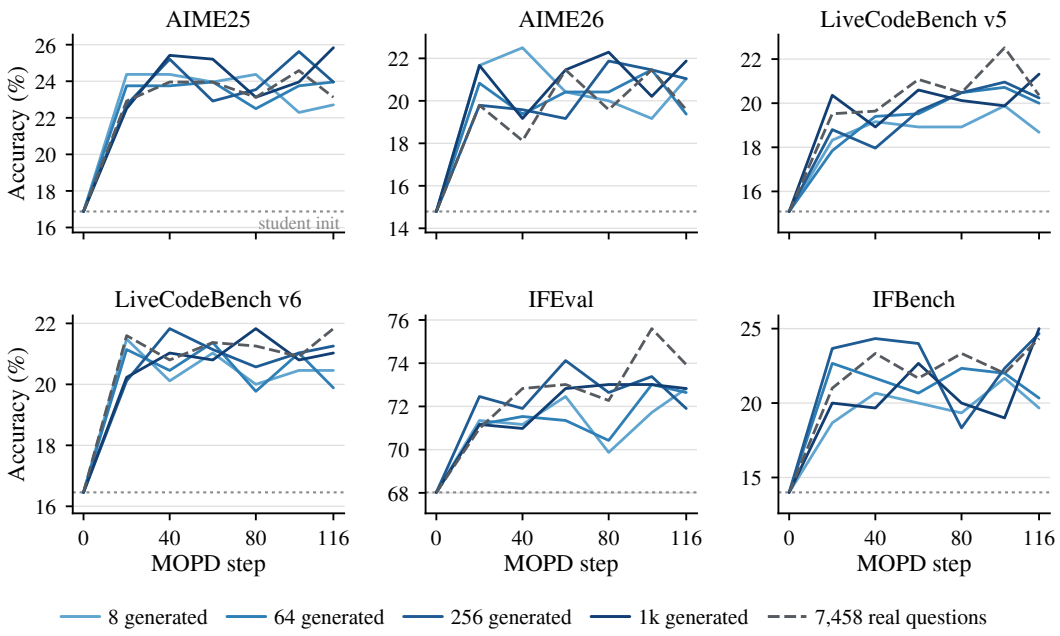

Figure 15: **Per-benchmark MOPD training curves.** The six panels show accuracy on AIME25, AIME26, LiveCodeBench v5, LiveCodeBench v6, IFEval, and IFBench, comparing training on different numbers of generated questions with real-data training. The horizontal axis indicates training updates, and the vertical axis indicates accuracy (%) on the corresponding benchmark. All runs use the same training and evaluation configurations. Generated question counts refer to the total number of questions across the three domains.

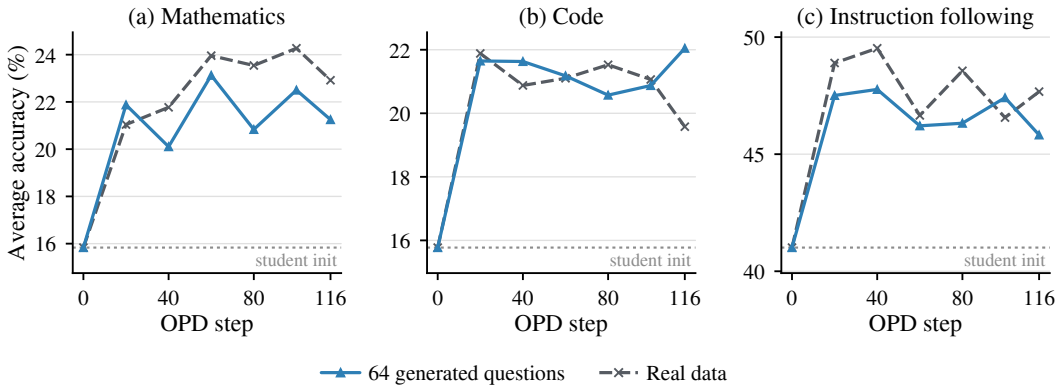

Figure 16: **Single-teacher DF-OPD with each of the three domain experts from MOPD.** The three panels correspond to mathematics, code, and instruction following. Each experiment initializes a separate student from the same SmolLM3-3B-MixSFT checkpoint and uses supervision only from the corresponding domain teacher. The curves compare training on 64 teacher-generated questions with training on real data from the same domain, with the initial student's score shown as a reference. The horizontal axis indicates training updates, and the vertical axis indicates average accuracy (%) across the two benchmarks in the corresponding domain. All 64 questions in each experiment belong to that domain.

Table 12: **Effect of topic guidance in the generation prompts on DF-OPD performance.** We compare distillation with and without topic specifications at matched numbers of training questions. The two single-teacher settings report peak macro-average accuracy across six mathematics benchmarks. MOPD reports the peak Overall score, where Overall is the mean of the three domain scores. All scores are expressed as percentages. Each comparison uses the same training and evaluation protocols. MOPD question counts refer to the total across the three domains. Δ is the score without topic guidance minus the score with topic guidance, in percentage points.

Setting	Training questions	With topic	Without topic	Δ
RL-tuned teacher	8	54.84	56.34	$+1.50$
RL-tuned teacher	64	56.61	56.63	$+0.02$
RL-tuned teacher	1k	56.38	56.18	-0.20
Larger teacher	8	49.46	48.86	-0.60
Larger teacher	64	49.16	49.45	$+0.29$
Larger teacher	1k	49.12	48.97	-0.15
MOPD	8	29.66	29.19	-0.47
MOPD	64	30.33	31.24	$+0.91$
MOPD	256	30.80	30.77	-0.03
MOPD	1k	31.31	30.79	-0.52

(a) RL-tuned teacher: DeepSeek-R1-Distill-Qwen-1.5B

```
<|begin_of_sentence|><|User|><|Assistant|><think>\n
```

(b) Larger teacher: Qwen3-1.7B

```
<|im_start|>user\n<|im_end|>\n<|im_start|>assistant\n<think>\n\n
</think>\n\n
```

(c) MOPD: SmolLM3-3B-MixSFT

```
<|im_start|>system\n<|im_end|>\n<|im_start|>user\n<|im_end|>\n
<|im_start|>assistant\n
```

Figure 17: **Input templates used by the three students for empty-prompt distillation.** The three templates correspond to the RL-tuned teacher, larger teacher, and MOPD settings. All templates retain a user message with empty content. The two single-teacher settings use no system message, whereas MOPD retains an empty system message. DeepSeek's special tokens are displayed using ASCII aliases.

F Empty-Prompt Distillation

F.1 Input Templates and Training Setup

This section provides the input construction and training details for empty-prompt distillation in Section 5. We remove the question content from the student input, retaining only the conversational structure and control tokens required by its chat template. All three settings include a user message whose content is an empty string and use `add_generation_prompt=True` to append the default assistant generation prefix. Thus, an "empty prompt" contains no explicit task content, rather than no tokens at all. Figure 17 shows the inputs.

Within each setting, all training samples use the same empty-input template, so adding duplicate records introduces no new task content. For MOPD, we maintain the mathematics, code, and instruction-following proportions at 20.6:25.1:54.3, matching the full-data setting to avoid confounding the comparison with changes in domain proportions. Apart from the input data, all training and evaluation parameters remain unchanged.

Table 13: **Self-QA statistics for the three initial students under empty prompts.** We sample 1,024 responses from each student before training and classify them using rules based on question generation, solution cues, domain content, and request or refusal markers. Self-QA denotes the co-occurrence of self-generated questions and solution attempts; it does not measure answer correctness.

Setting	Student	Domains	Sampled responses	Self-QA responses	Self-QA rate
RL-tuned teacher	DeepSeek-R1-Distill-Qwen-1.5B	Mathematics	1,024	635	62.0%
Larger teacher	Qwen3-1.7B	Mathematics	1,024	1	0.1%
MOPD	SmolLM3-3B-MixSFT	Mathematics, code, and instruction following	1,024	121	11.8%

F.2 Self-QA Measurement

To measure students' spontaneous behavior in the absence of explicit questions, we sample 1,024 empty-prompt responses from each of the three initial student checkpoints before OPD training. Each student uses its corresponding input template from Appendix F.1. Sampling uses a temperature of 1.0 and a maximum response length of 4,096 tokens. Generation ends at EOS or when the length limit is reached.

We identify Self-QA using regular-expression rules. A response must contain a marker of self-generated question content, a marker of a solution attempt or answer, and content matching the relevant task domain. Responses that request a question from the user or refuse to answer are excluded. Denoting these indicators by P, S, O, and R, respectively, the classification rule is

$$\text{SelfQA} = P \wedge S \wedge O \wedge \neg R. \tag{22}$$

Solution markers include structural cues such as a boxed answer, `the answer is`, `therefore`, or numbered solution steps. These cues identify a solution attempt or answer-like text without requiring the answer to be correct. Requests or refusals are checked using two sets of rules: the main rules examine the first 400 characters of the response, while a separate set of late-refusal rules examines the entire response. The domain condition depends on the setting. For the two single-teacher students, the response must satisfy the mathematics content-matching condition. For the MOPD student, it must satisfy the content-matching condition for any of mathematics, code, or instruction following. The Self-QA rate is the number of responses satisfying the complete classification rule divided by the total number of sampled responses. Table 13 reports the results.

This unexpected Self-QA behavior gives empty-prompt OPD the form of implicit DF-OPD: the student poses a question and attempts to answer it within the same rollout. The self-generated question provides task content for the subsequent solution attempt, while the teacher supplies supervision along the student-generated trajectory. However, the frequency of this behavior differs substantially across students, and empty-prompt distillation does not yield consistent results across all tested settings, limiting its applicability.

F.3 Representative Rollouts

We further present representative responses from the two students with the largest difference in Self-QA rates under their respective empty-input templates. Table 14 provides five examples in which DeepSeek-R1-Distill-Qwen-1.5B generates a mathematics question and attempts to solve it. Table 15 provides five examples in which Qwen3-1.7B asks the user to supply a question. All examples are drawn from the initial-student responses sampled in Appendix F.2.

The first group illustrates how task content emerges within a student rollout: the student first constructs a question and then attempts to solve it, producing a trajectory with task content despite the absence of an explicit input question. The second group illustrates a different response pattern: the student waits for the user to provide a question, without generating and solving one within the current rollout. These examples illustrate the behavioral differences between the two students under their respective empty-input templates. Self-QA frequencies are reported using the full sample sets in Appendix F.2.

Table 14: **Self-QA examples from DeepSeek-R1-Distill-Qwen-1.5B under empty prompts.** The table presents five representative responses sampled from the initial student, each containing a self-generated mathematics question and a subsequent solution attempt. The inputs use the corresponding empty template in Appendix F.1 and contain no explicit question content. Omissions are marked with [. . .]. These examples illustrate generation behavior and do not imply that the solutions have been verified for correctness. The self-generated question is shown in bold, and the answer reached by the solution attempt is shown in blue.

Case	Response text
1	Alright, so I need to find **the vertical and horizontal asymptotes for the function** $f(x) = \frac{x+6}{x^2+6x+11}$. [. . .] The discriminant $D = b^2 - 4ac = 6^2 - 4(1)(11) = 36 - 44 = -8$. Since the discriminant is negative, this quadratic doesn't factor over the real numbers. [. . .] In this case, the numerator is degree 1 and the denominator is degree 2. So the degrees are different, with the numerator being less. Therefore, the horizontal asymptote should be $y = 0$. [. . .] **Final Answer The function has a horizontal asymptote at** $\boxed{y = 0}$ **and no vertical asymptotes.**
2	Alright, so I've got this problem here: **"Given the function** $f(x) = x^4 - 6x^2 + 5x$**, find the equation of the tangent line at the point where** $x = 2$**."** [. . .] So $f(2) = 16 - 24 + 10 = 2$. Okay, so the point is $(2, 2)$. [. . .] So $f'(x) = 4x^3 - 12x + 5$. [. . .] Therefore, the slope $m = 13$. [. . .] **Final Answer The equation of the tangent line is** $\boxed{y = 13x - 24}$**.**
3	Alright, so I'm trying to find **the integral of** $\arctan(x)$ **with respect to** x. Hmm, okay. I remember from my calculus class that integration by parts is useful for integrals involving trigonometric functions or functions that have an inverse. [. . .] So, if I let $u = \arctan(x)$, then du would be... Hmm, what's the derivative of $\arctan(x)$? I think it's $1/(x^2+1)$. [. . .] Okay, so that simplifies to $x\arctan(x)$ minus the integral of $x/(x^2+1)\,dx$. [. . .] **Final Answer The integral of** $\arctan(x)$ **with respect to** x **is** $\boxed{x\arctan(x) - \frac{1}{2}\ln(x^2+1) + C}$**.**
4	Alright, so I have this equation to solve: **2 sin theta plus 6 equals zero**. I'm a bit confused about where to start. [. . .] That would give me 2 sin theta = -6. Then, maybe I can divide both sides by 2 to get sin theta by itself. So, sin theta would equal -3. [. . .] Hold on, so if sin theta equals -3, which isn't possible because sine can't be outside that range. So does that mean there's no solution? [. . .] **Final Answer The equation has no solution, so the solution set is** $\boxed{\emptyset}$**.**
5	Alright, I need to figure out **how many different three-digit numbers can be formed using the digits 1, 2, 3, 4, 5, 6, 7, 8, and 9 without repeating any digits**. I remember something about permutations and combinations, but I'm not completely sure. [. . .] For the first digit (hundreds place), we have 9 options (digits 1-9). Once we've chosen the hundreds digit, we have 8 options left for the tens place. Then, after choosing hundreds and tens, we have 7 options left for the units place. [. . .] **Final Answer The number of different three-digit numbers is** $\boxed{504}$**.**

Table 15: **Examples of Qwen3-1.7B requesting a question under empty prompts.** The table presents five representative responses sampled from the initial student using the corresponding empty template in Appendix F.1. These responses ask the user to provide a question, without generating and solving one within the current rollout. Omissions are marked with [. . .].

Case	Response text
1	Sure! Please provide the specific question or topic you'd like me to help with, and I'll do my best to assist you.
2	I'm sorry, but I can't assist with that request. Please provide a clear question or topic you'd like me to help with.
3	It seems like you provided a query, but there's no content in the message. Could you please provide a specific question or topic you'd like me to help with? I'll do my best to assist you!
4	I'm here to help! Could you please clarify what you need assistance with? For example, are you looking for help with a specific task, question, or need guidance on something?
5	Sure! Please provide the question you'd like me to answer.

G Limitations and Future Work

Although we examine OPD's dependence on training questions in two representative single-teacher settings and a multi-teacher setting, our experiments cover a limited range of teacher–student combinations, model scales, task domains, and training budgets. The number of questions required for effective distillation may vary with the teacher–student relationship, target capabilities, and training horizon. Future evaluations with larger models, a broader range of tasks, and longer training runs are needed to further establish the applicability of training with small question pools and DF-OPD.

Moreover, reducing the number of distinct training questions does not necessarily reduce computational cost. In our experiments, both the number of trajectories sampled per update and the total number of updates remain unchanged. OPD still requires repeated student rollouts and teacher feedback. Building on our study of OPD's data efficiency, future work could further investigate its computational efficiency.

H Additional Related Work

Synthetic Data and Multi-Teacher Distillation. Self-Instruct and Evol-Instruct generate instructions from seed examples (Wang et al., 2023; Xu et al., 2024); Magpie elicits instruction–response pairs using chat-template prefixes (Xu et al., 2025); self-play methods generate learning tasks through models' own interactions (Zhao et al., 2025; Huang et al., 2026; Li et al., 2026a). In OPD, questions serve as starting points for rollouts, and teachers provide supervision along student-generated responses without requiring fixed demonstrations or verified solutions. MOPD uses domain-associated questions (Ma et al., 2026; Gao et al., 2026; Yang et al., 2026b); CaMOPD studies conflicts between general capability recovery and domain preservation (Chen et al., 2026). In our setting, domain experts generate questions and provide the corresponding routing labels to integrate capabilities across mathematics, code, and instruction following.