BEYOND PROMPT COUNT: HOW DATA SHAPES TRANSFER IN ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) trains students using teacher feedback on their own sampled responses, yet how prompt choice shapes transfer across teacher–student pairs remains poorly understood. We systematically study prompt quantity, source, and selection across RL- and SFT-continuation pairs and cross-model settings. We find that OPD can be highly prompt-efficient: a few prompts can approach large-pool performance, with four DAPO prompts matching the observed mathematics score of 3,840 DeepMath prompts. However, prompt utility is relational rather than intrinsic: changing only the teacher can reverse the relative effectiveness of mathematics and code prompts. To characterize these transfer differences, we analyze parameter and functional changes across prompt supports and model pairs. Functional alignment with the teacher varies across supports and target tasks; in continuation pairs, teacher-aligned prediction changes can coexist with weak parameter alignment. Continued OPD on effective supports can restore performance after unfavorable transfer. Finally, targeted selection does not consistently outperform uniform random sampling, and filtering out a source that performs poorly alone yields no consistent gain across three paired support draws. Overall, our results distinguish prompt efficiency from prompt interchangeability and show that effective data choice depends on the teacher–student pair and target capability, with random sampling providing a competitive baseline in the studied settings. Our code is available at https://anonymous.4open.science/r/dissecting-opd-13D4.

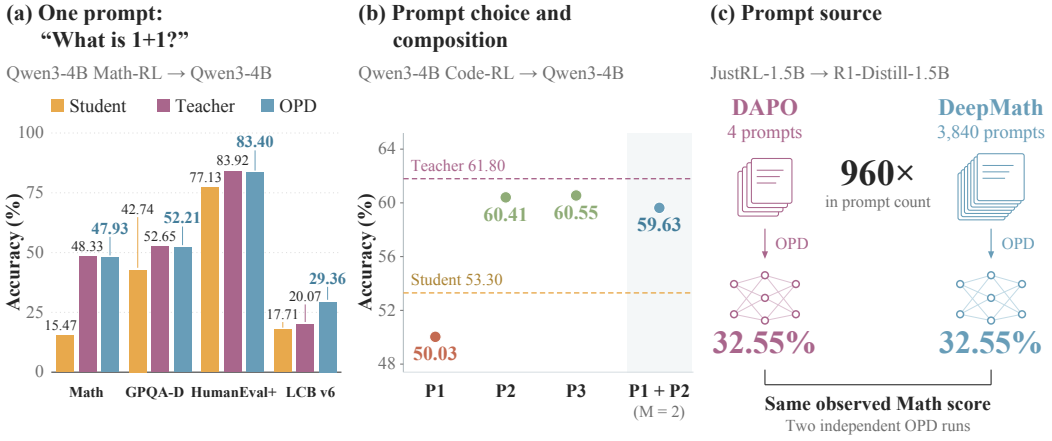


Figure 1: **Prompt efficiency does not imply prompt interchangeability.** A single prompt can induce substantial transfer, but outcomes vary with prompt identity and composition. Moreover, a few prompts from one source can match the observed mathematics score of thousands from another. These examples motivate our study of prompt count, source, teacher–student pairing, and selection. In (b), P_1–P_3 denote the individual code prompts described in Appendix C.4, while $P_1 + P_2$ denotes joint OPD using the first two prompts.

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

1 INTRODUCTION

On-policy distillation (OPD) is increasingly used in large language model (LLM) post-training to transfer capabilities to smaller students (Yang et al., 2025; Zheng et al., 2025), consolidate domain-specialized teachers (Xiao et al., 2026; Li et al., 2026a; DeepSeek-AI, 2026), and recover capabilities acquired in earlier training stages (Zeng et al., 2026).

In OPD, the student samples its own responses, and the teacher provides token-level feedback at the prefixes visited by the student (Lin et al., 2020; Agarwal et al., 2024; Lu & Thinking Machines Lab, 2025). A fixed prompt can elicit different trajectories as the student evolves. Prompt choice therefore shapes the contexts in which teacher supervision is applied, motivating the study of both how many prompts are needed and which prompts are useful.

Recent work improves transfer by aligning prompt content and templates with the teacher's post-training data (Li et al., 2026b). Concurrent work demonstrates substantial transfer from a single query (Fu et al., 2026) and shows that OPD can propagate strengths and weaknesses beyond the training domain, with generalization differing between same-origin and cross-origin pairs (Li et al., 2026c). Appendix A discusses related work in further detail. These findings motivate a distinction between prompt sufficiency and prompt interchangeability. Our central question is how prompt utility depends on the teacher–student pair and target capability. A related practical question is whether explicit prompt selection or source filtering yields consistent gains over uniform random sampling.

We study prompt quantity, source, teacher–student pairing, and selection across RL- and SFT-continuation pairs and cross-model settings. Figure 1 illustrates both low prompt requirements and sensitivity to prompt choice: four DAPO prompts match the observed mathematics score of 3,840 DeepMath prompts, while individual code prompts can produce positive or negative transfer. Figure 3(b) further demonstrates teacher-dependent source preferences: switching between Math-RL and Code-RL teachers reverses the relative effectiveness of mathematics and code supports across all four evaluations, with the initial student and both supports held fixed. For multi-domain teachers, source preferences vary across target capabilities, while broader mixtures at a fixed prompt count can improve some capabilities at the expense of others.

To understand how different prompt supports shape transfer, we analyze parameter updates, predictions on common prefixes, hidden-state representations, and generated behavior. Functional alignment with the teacher varies across supports and target tasks. When the teacher is obtained by further training the student's initial checkpoint, teacher-aligned functional changes can coexist with weak parameter alignment. Hidden-state interventions further probe the role of representations in unfavorable transfer, while continued OPD on effective supports can restore performance.

This data sensitivity does not imply consistent gains from targeted selection: uniform random sampling leads the JustRL–DeepMath study in four-task average and remains competitive in the code and 30B-to-4B comparisons. Table 5 shows no consistent gain from excluding GSM8K from a DeepMath/DAPO/GSM8K pool across three paired support draws, despite its poor standalone performance.

Our contributions are threefold:

- **Data requirements and preferences.** We systematically compare prompt counts and sources across teacher–student pairs, including controlled teacher swaps with the student initialization and prompt supports held fixed.
- **Functional transfer, failure, and recovery.** We combine parameter and functional diagnostics with hidden-state interventions and continued distillation to characterize unfavorable transfer and performance recovery.
- **Practical data selection.** We evaluate within-pool selection and mixed-pool source filtering against uniform random sampling, examining both transfer performance and selection cost.

Table 1: **Teacher–student pairs and OPD sources.** The first three teachers are trained by us; the others are publicly released models. We list constituent data sources; support construction, including mixtures and source filtering, is detailed in Appendix B.3.

Student	Teacher	OPD sources
RL-continuation pairs		
Qwen3-4B	Math GRPO-500 RL data: DeepMath L6	DeepMath L6,
	Code GRPO-300 RL data: Eurus-2 Code	Eurus Code
Qwen3-1.7B-Base + direct-code SFT	Code GRPO-400 RL data: Open-R1 Code	Open-R1 Code
DeepSeek-R1-Distill-Qwen-1.5B	JustRL-DeepSeek-1.5B RL data: DAPO-Math-17k	DeepMath L6, DAPO
	DeepScaleR-1.5B-Preview RL data: AIME, AMC, Omni-MATH, Still (∼40K)	DeepMath, DAPO
	Nemotron-Research-Reasoning-Qwen-1.5B (v1) RL data: DeepScaleR, Eurus-2, SCP-116K; Reasoning Gym, Llama-Nemotron IF	DeepMath L6, Eurus Code, TextbookReasoning
Qwen3-4B-Instruct-2507	Qwen3-4B-Inst-Mix RL data: LLM-Fusion-Train (Mix)	DeepMath L6, Eurus Code, TextbookReasoning, LLM-Fusion
SFT-continuation pairs		
DeepSeek-R1-Distill-Qwen-7B	Light-R1-7B-DS SFT data: Light stage2	DeepMath L6, DAPO, Light SFT-stage2
Cross-model pairs		
Qwen3-4B	Qwen3-30B-A3B-Instruct-2507	DeepMath L6, DAPO, GSM8K, Eurus Code, TextbookReasoning
Qwen3-1.7B		DeepMath L6, DAPO

2 Experimental Setup

2.1 On-policy distillation

Let π_T denote the frozen teacher and π_θ the student. We sample prompts from a fixed support $\mathcal{S}_M$ of M distinct prompts and generate responses with the rollout policy $\pi_{\bar{\theta}}$. OPD aligns student and teacher distributions at the resulting prefixes $s_t = (x, y_{<t})$ using reverse KL (Agarwal et al., 2024):

$$\mathcal{L}_{\mathrm{RKL}}(\theta; \bar{\theta}) = \mathbb{E}_{s\sim d_{\bar{\theta}}}\left[D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s) \,\|\, \pi_T(\cdot \mid s))\right], \tag{1}$$

where $d_{\bar{\theta}}$ is the induced distribution over valid response prefixes. Collected prefixes and $\bar{\theta}$ remain fixed during optimization. We use a sampled-token policy-gradient surrogate (Lu & Thinking Machines Lab, 2025); implementation and training details appear in Appendix B.2.

2.2 Model pairs and data

In a continuation pair, the teacher is obtained by RL or SFT from the student's exact initial checkpoint; cross-model teachers are not continuations of that checkpoint. We use the Qwen3 (Yang et al., 2025) and DeepSeek-R1-Distill (Guo et al., 2025) model families. Table 1 lists the pairs and data sources; construction details appear in Appendix B.

The support $\mathcal{S}_M$ is selected from a candidate pool $\mathcal{C}$ of size C. Fresh responses are generated throughout training, so fixing M does not fix token or compute budgets. Support construction and prompt templates are detailed in Appendices B.3 and B.5.

2.3 Evaluation

Math reports mean@16 averaged equally over AIME 2024/2025 and HMMT February/November. GPQA-Diamond (Rein et al., 2024), HumanEval+ (Liu et al., 2023), and LiveCodeBench v6 (Jain et al., 2025) use mean@8, with MBPP+ added for code comparisons; direct-code 1.7B comparisons use mean@16. Multi-domain evaluations additionally include IFEval (Zhou et al., 2023b), IFBench

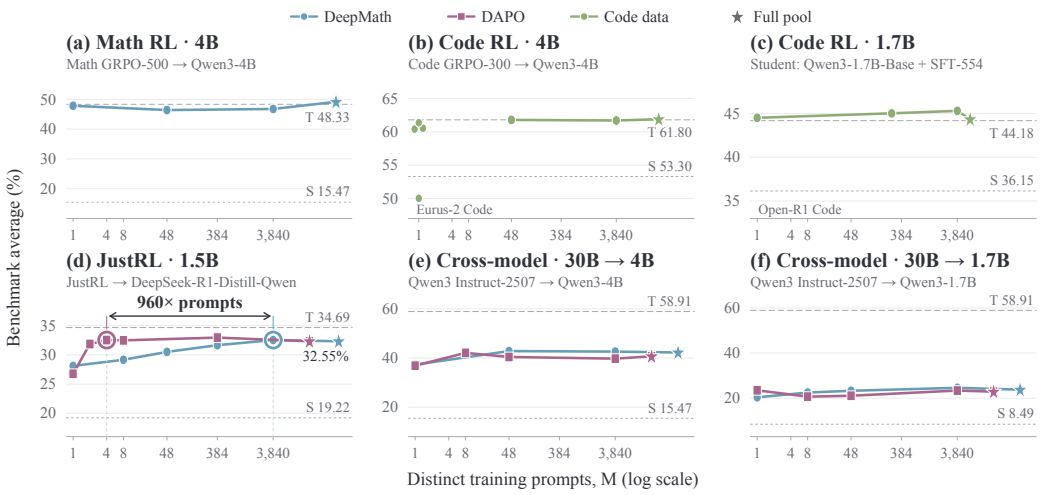


Figure 2: **Prompt-count scaling across six teacher–student pairs.** Stars mark full-pool OPD; S/T denote the initial student and teacher. Circled points in (d) attain the same observed Math score with 4 DAPO versus 3,840 DeepMath prompts. Training budgets are comparison-specific; full results appear in Appendix C.

(Pyatkin et al., 2025), and BFCL-v3 (Patil et al., 2025). Mean@k averages per-question correctness over k responses; macro averages are defined with the results. Decoding and scoring details appear in Appendix B.4.

For random selection, we report mean $\pm$ sample SD across independent support draws, varying selection seeds while holding the training seed fixed.

3 Data Requirements

Figure 2 examines how OPD performance varies with the number of distinct training prompts.

3.1 Few prompts can approach large-pool performance

In the Qwen3-4B Math-RL pair, one prompt reaches 44.74 on Math, versus 46.77 with 3,840 prompts, at a matched total of 3,840 rollouts. With Code-RL, 48 prompts come within 0.10 points of full-pool OPD with 22,618 prompts at step 100. Training curves are provided in Appendix C.3.

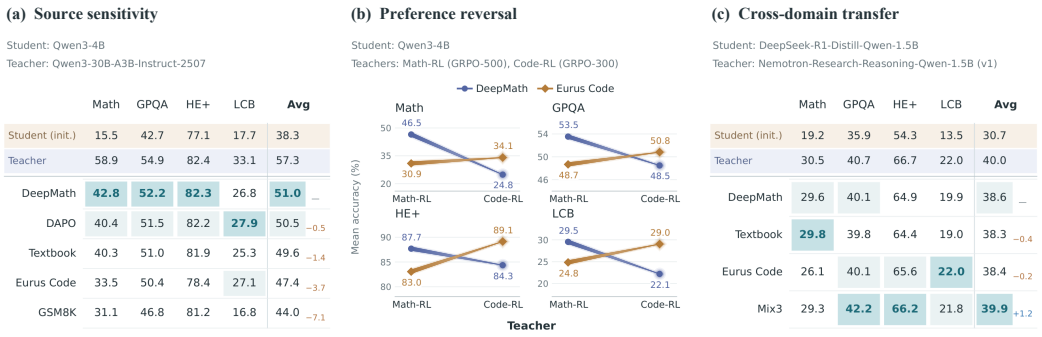
This pattern also appears beyond RL continuation. In 30B-Instruct-to-4B distillation, DeepMath48 performs comparably to DeepMath3840 at step 15. For the Light-R1 SFT-continuation pair, eight DAPO or DeepMath prompts achieve math scores of 40.83 and 40.63, respectively, compared with 40.99 using the original 3,533-prompt source.

However, selected single-prompt Code-RL runs range from 50.03 to 61.37, spanning the initial student's 53.30. We examine these unfavorable outcomes and their recovery in Section 5.

3.2 Prompt-count scaling depends on source and pair

Full-pool performance need not predict sensitivity to prompt count. With JustRL, DAPO and DeepMath achieve similar full-pool scores but diverge at $M = 8$. Appendix C reports the same observed Math score of 32.55 for four DAPO and 3,840 DeepMath prompts, a 960$\times$ difference in distinct prompt count.

Scaling also varies across students with the same teacher and source. With the 30B Instruct teacher and DAPO, the 4B student's score rises from 36.9 at $M = 1$ to 42.1 at $M = 8$, whereas the 1.7B student's score remains nearly unchanged from $M = 1$ to $M = 3{,}840$.



	Math	GPQA	HE+	LCB	Avg
Student (init.)	15.5	42.7	77.1	17.7	38.3
Teacher	58.9	54.9	82.4	33.1	57.3
DeepMath	**42.8**	**52.2**	**82.3**	26.8	**51.0** –
DAPO	40.4	51.5	82.2	**27.9**	50.5 −0.5
Textbook	40.3	51.0	81.9	25.3	49.6 −1.4
Eurus Code	33.5	50.4	78.4	27.1	47.4 −3.7
GSM8K	31.1	46.8	81.2	16.8	44.0 −7.1

	Math	GPQA	HE+	LCB	Avg
Student (init.)	19.2	35.9	54.3	13.5	30.7
Teacher	30.5	40.7	66.7	22.0	40.0
DeepMath	29.6	40.1	64.9	19.9	38.6 –
Textbook	**29.8**	39.8	64.4	19.0	38.3 −0.4
Eurus Code	26.1	40.1	65.6	**22.0**	38.4 −0.2
Mix3	29.3	**42.2**	**66.2**	21.8	**39.9** +1.2

Figure 3: **Data-source preferences across teacher–student pairs. (a)** Source sensitivity with a 30B Instruct teacher. **(b)** Preference reversals between Math-RL and Code-RL teachers with the same initial student. **(c)** Cross-domain transfer from the Nemotron teacher. Avg is the unweighted mean of the four displayed benchmark groups; side annotations report differences from DeepMath. Darker and lighter shading mark the highest and second-highest OPD scores within each column.

Since repeated prompts still generate fresh trajectories, Appendix G.1 separately examines the quality–cost trade-offs of reusing those trajectories.

TAKEAWAY 1

Few prompts can support effective OPD, but prompt sufficiency does not imply prompt interchangeability.

4 Data Preferences across Teacher–Student Pairs

At a fixed support size of $M = 48$, we compare prompt sources and mixtures within each teacher–student pair.

4.1 Source sensitivity at fixed support size

With equal prompt counts, DeepMath and GSM8K supports score 42.8 and 31.1 on Math, respectively, for a Qwen3-4B student and 30B Instruct teacher in Figure 3(a).

Table 2 and Appendix D.2 show GSM8K trailing DeepMath by 11.25 points on Math and 9.36 on LCB despite approximately matched cumulative supervision-token budgets. Supervision volume alone therefore does not explain the source gap.

Table 2: **Token control.** $M = 48$; Avg averages Math, GPQA, HE+, and LCB.

Data	Tokens (M)	Avg
GSM8K	1.19	44.01
+ token match	16.14	44.84
DeepMath	16.63	51.02

4.2 Teacher-dependent source preferences

Switching from the Math-RL to Code-RL teacher reverses source preferences on all four evaluations in Figure 3(b), with the Qwen3-4B initialization and both 48-prompt supports fixed. On Math, DeepMath outperforms Eurus Code under the Math-RL teacher (46.51 versus 30.89), whereas Eurus Code outperforms DeepMath under the Code-RL teacher (34.06 versus 24.84). The preferred source thus depends on the teacher even for the same student and target task. In Avg_4, the source-preference reversal in Figure 3(b) persists across all five tested support–training seed configurations in Appendix D.1, with a teacher–source difference-in-differences of 13.1–14.1 percentage points.

Table 3: **Data-source preferences with a multi-domain RL teacher.** Qwen3-4B-Inst-Mix → Qwen3-4B-Instruct-2507; $M = 48$. Teal marks the best and second-best OPD scores; mixture construction is detailed in Appendix B.3.

Model / OPD support	Math	GPQA-D	HumanEval+	LCB v6	IFEval	IFBench	BFCL-v3	Avg_6
Student (init.)	45.83	45.52	81.48	27.86	83.71	30.42	63.91	53.61
Teacher	48.18	45.27	79.57	30.21	86.02	39.96	70.81	56.17
DeepMath	48.28	47.35	81.86	29.71	83.90	29.04	64.25	54.65
Textbook	46.46	51.14	**83.23**	28.93	83.50	27.88	62.66	54.69
Eurus Code	48.49	**61.81**	81.17	31.93	83.13	28.67	66.50	**57.63**
IF	41.51	46.46	68.29	22.07	78.35	**42.56**	53.25	48.67
Agent	43.38	50.88	78.35	28.21	82.99	26.73	62.28	52.99
Mix3	**48.59**	59.34	82.01	**32.36**	**83.93**	28.79	64.28	57.16
Mix5	47.55	55.18	80.18	31.14	82.05	36.79	**69.56**	57.17

4.3 Task-specific transfer and mixture trade-offs

Data composition also changes which capabilities benefit. With the Mix-RL teacher, Code48's GPQA scores span 61.24–61.81 across three support draws, versus DeepMath48's 47.35 in Table 3. Even within instruction following, IF48 improves IFBench over the initial student while lowering IFEval.

Mixtures likewise improve some tasks at the expense of others. Under Nemotron, Mix3 improves GPQA, HumanEval+, and LCB over DeepMath in Figure 3(c), with a small Math decrease. For Mix-RL, Mix5 reallocates the same 48-prompt budget across five domains, reselecting prompts within each pool as in Appendix B.3. Relative to Mix3, it improves IFBench by 8.00 points and BFCL by 5.28 points, while GPQA decreases by 4.16 points.

Takeaway 2

Source preferences are relational: the same source can be effective or ineffective depending on the teacher–student pair and target capability, while broader mixtures redistribute transfer gains.

5 Transfer Analysis

We examine how prompt supports shape the student's functional and behavioral changes, whether unfavorable transfer can be repaired, and what these results imply for prompt selection.

5.1 Functional alignment

For continuation pairs, we compare OPD and teacher changes from the shared student initialization. Parameter alignment measures weight displacements; functional alignment measures token log-probability changes on common frozen prefixes, defined in Appendix E.1. Functional cosine measures directional alignment; α is the teacher-direction projection coefficient. Diagnostic KL closure measures the reduction in the conditional KL gap to the teacher.

Figure 4(a) reports median parameter and functional cosines of 0.069 and 0.859 across 43 checkpoints from seven teacher families, including intermediate states. OPD can therefore reproduce teacher-aligned prediction changes without following the teacher's parameter-update direction.

We next examine support-dependent prediction changes across tasks in Figure 4(b,c). In 30B-to-4B distillation, DeepMath reduces the diagnostic teacher–student KL gap on frozen Math prefixes by 11.20%, whereas GSM8K widens it by 13.64%. GSM8K nevertheless has positive functional cosine (0.318), showing that directional alignment alone does not ensure closer predictions. Eurus Code closes only 1.13% of the Math gap but 42.30% of the LCB gap. Appendix E.3 details these diagnostics, which measure task-specific prediction changes, not benchmark gains.

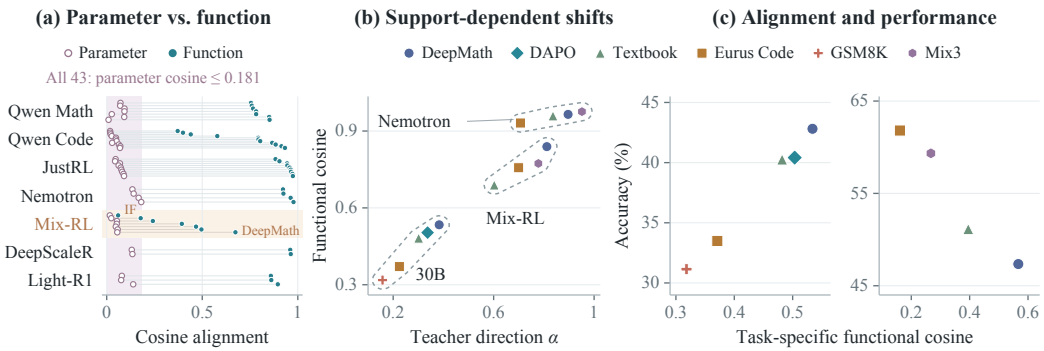


Figure 4: **Parameter and functional alignment. (a)** Parameter and all-domain functional cosines for 43 checkpoints; each line joins the same endpoint. **(b)** Teacher-direction projection α versus functional cosine on Math probes, grouped by model pair. **(c)** Functional cosine versus accuracy: 30B-to-4B Math (left) and Mix-RL-to-4B GPQA (right). Definitions: Appendix E.1.

TAKEAWAY 3

OPD can produce teacher-aligned functional changes through substantially different parameter updates; the extent of alignment depends on the support and task.

5.2 GENERATION BEHAVIOR

We next move from common-prefix predictions to freely generated responses in Figure 5. With Mix-RL, Eurus Code yields longer GPQA responses and higher accuracy than DeepMath; two additional Code support draws reproduce this pattern. DeepMath OPD also produces longer responses than the teacher in both 30B pairs, so successful transfer need not imitate the teacher's response length. Yet in 30B-to-1.7B distillation, DAPO produces longer responses than DeepMath with lower accuracy. Longer generation thus accompanies some gains but does not consistently indicate better transfer.

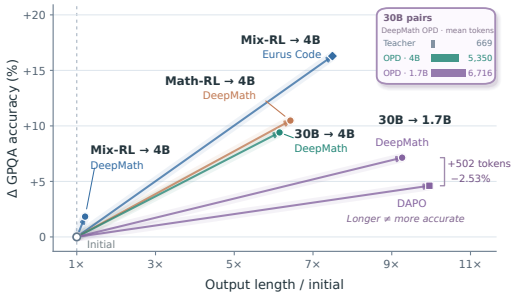


Figure 5: **GPQA accuracy and output-length changes.** All supports contain $M = 48$ prompts; changes are relative to each pair's initial student. Inset: mean output tokens.

For Mix-RL Code48 and Mix3, most additional text precedes the final answer. Among the 15–16% of outputs with an identifiable early answer, correct-to-wrong revisions outnumber wrong-to-correct revisions. The results link supports to generation behavior without establishing that extra tokens cause accuracy gains. Complete accuracy–length measurements and output analyses are in Table 21 and Appendix E.4.

5.3 FAILURE AND RECOVERY

Can interventions redirect these support-dependent changes? In the Code pair, we compare a degraded endpoint A', an effective single-prompt endpoint B, and native M48 trained directly on 48 prompts. Figure 6(a) shows similar layerwise CKA (Kornblith et al., 2019) profiles for A' and native M48 relative to full-pool OPD, despite their performance gap. We therefore test behavior directly through hidden-state replacement and continued distillation.

At each decoding step, we replace the recipient's current-position hidden vector after block 30 (zero-based) with the donor's. On the 20-question probe in Figure 6(b), with eight samples per question, $B \to A'$ raises accuracy from 18.75% to 24.38%, while $A' \to B$ lowers it from 28.13% to 21.25%. Self-replacement leaves accuracy unchanged; block 10 gives no aggregate improvement. The tested representation interface can thus alter the degraded model's behavior during evaluation.

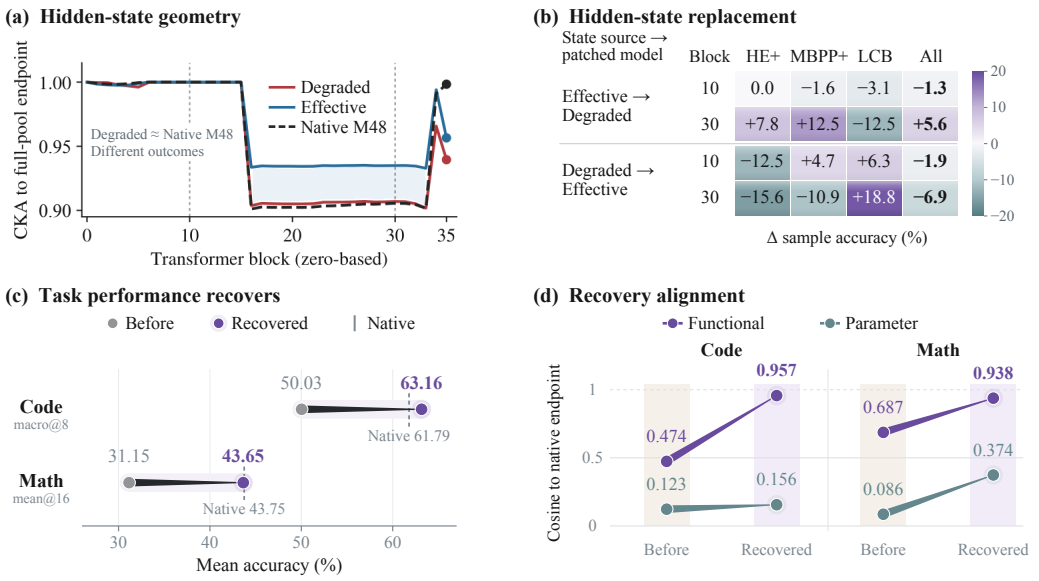


Figure 6: **Redirecting unfavorable transfer.** **(a)** CKA relative to full-pool OPD. **(b)** Bidirectional hidden-state replacement. **(c)** Performance recovery through continued distillation; vertical ticks mark native references. **(d)** Functional versus parameter alignment to native endpoints. Reference checkpoints and unequal Code update budgets are specified in Appendix E.5.

Figure 6(c) shows recovery through continued OPD. Switching the degraded Code model to an effective 48-prompt support raises Code Macro from 50.03 to 63.16, above the initial student's 53.30. Switching from GSM8K to DeepMath also recovers performance in the 30B-to-4B Math setting. Code recovery adds training and resets the optimizer; the Math comparison matches update counts, not necessarily token budgets.

A 15-update continuation control shows that DeepMath outperforms GSM8K from both the GSM8K-trained checkpoint (Avg_4: 51.75 versus 43.77) and the DeepMath-trained checkpoint (51.98 versus 46.38). The two DeepMath continuations reach close scores despite different training histories.

Figure 6(d) shows strong functional but weak parameter alignment between recovered models and native references. These results show that unfavorable training histories need not prevent strong subsequent transfer, even when parameter updates remain different. Appendix E.5 further shows that Math recovery persists under alternative answer scoring.

TAKEAWAY 4

Unfavorable OPD outcomes are recoverable: representation interventions can redirect behavior, while continued OPD on effective supports can restore performance without retracing the parameter path of successful training.

5.4 PROMPT SELECTION

These analyses characterize transfer after training. We now test whether selecting prompts beforehand improves transfer over uniform random sampling. Table 4 compares eight JustRL–DeepMath selectors. Uniform random sampling achieves the highest reported four-task average at both $M = 8$ and $M = 48$ (37.65 and 37.63), without additional GPU screening. Other selectors lead on individual tasks; method definitions are in Appendix F.1.

Random sampling remains competitive in the Code and 30B-to-4B comparisons in Appendices F.3 and F.4, although individual selectors achieve higher mean scores.

Initial-gradient diagnostics in Appendix F.5 may help explain random sampling's competitiveness. In JustRL–DeepMath ($C = 384$, $M = 8$), random supports preserve the candidate-pool mean direction (mean cosine 0.971; fifth percentile 0.952). However, Figure 7 shows that lower gradient approximation error does not consistently predict higher final accuracy.

Table 4: **Prompt selection within DeepMath.** JustRL-1.5B → DS-1.5B. Random selectors report mean ± sample SD over three sampled supports. Bold marks the column best within each M. Costs estimate selection-only GPU hours; 0 denotes no GPU screening.

Method	Math	GPQA-D	HE+	LCB	Avg_4	GPU·h
$M = 8, \quad C = 384$						
Uniform random	**31.09** ± 0.50	39.44 ± 1.24	62.40 ± 0.32	17.67 ± 0.86	**37.65** ± 0.38	0
Stratified random	30.36 ± 0.39	38.19 ± 0.58	61.74 ± 0.46	17.93 ± 0.41	37.06 ± 0.25	1.1
Semantic diversity	30.21	39.46	62.88	17.36	37.48	< 0.1
Hard selection	28.96	38.07	**63.57**	**18.21**	37.20	1.1
Shortest	28.75	39.08	61.28	16.86	36.49	2.2
Longest	29.82	**40.12**	62.48	17.14	37.39	2.2
T–S disagreement	28.53	38.64	62.91	17.68	36.94	3.3
Cost-D-opt	28.91	39.90	62.27	17.43	37.13	7.5
$M = 48, \quad C = 2{,}304$						
Uniform random	29.84 ± 0.31	**41.02** ± 0.48	62.40 ± 0.37	17.26 ± 0.28	**37.63** ± 0.21	0
Stratified random	**30.72** ± 0.28	37.95 ± 0.42	61.58 ± 0.35	17.83 ± 0.29	37.02 ± 0.19	6.6
Semantic diversity	29.16	38.83	61.72	17.25	36.74	< 0.1
Hard selection	27.91	39.42	62.15	**18.04**	36.88	6.6
Shortest	29.38	40.56	61.94	16.88	37.19	15.2
Longest	30.25	40.64	61.85	17.22	37.49	15.2
T–S disagreement	28.84	39.15	62.18	17.63	36.95	16.3
Cost-D-opt	30.45	38.74	**63.12**	16.93	37.31	47.7

Table 5 shows no consistent gain from removing GSM8K across three paired draws, despite its weak standalone performance. Appendix F.2 shows that replacing four of 48 GSM8K prompts with DeepMath improves transfer, while those same four alone achieve a similar mean Math score.

Table 5: **Source filtering in a mixed pool.** 30B Instruct → 4B, $M = 48$. Scores (%) are mean ± SD across three support draws.

Sampling strategy	Math mean@16	OOD mean@8
Mixed random	40.7 ± 2.3	54.7 ± 0.7
Source-filtered random	40.5 ± 1.0	54.5 ± 0.8
Filtered − Mixed	−0.3 ± 2.0	−0.2 ± 0.9

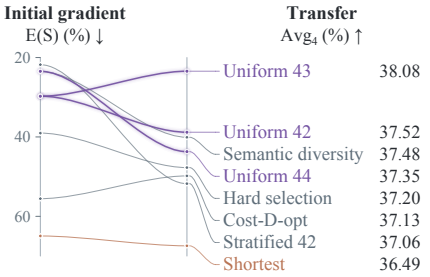


Figure 7: **Initial gradients and transfer.** JustRL–DeepMath, $C = 384$, $M = 8$.

TAKEAWAY 5

Strong transfer does not require a carefully curated prompt pool: uniform random sampling remains competitive, while targeted selection or source filtering did not consistently improve transfer in our experiments.

6 Discussion

Prompt efficiency does not imply interchangeability: prompt utility depends on the teacher–student pair and target capability. One possible interpretation is that simple prompts can support broadly elicitable behavioral changes, whereas more context-dependent reasoning behaviors benefit from supports that expose relevant trajectories. The contrasting prompt requirements of Qwen3-4B Math-RL and JustRL are consistent with this interpretation. Continued OPD on effective supports can restore performance after unfavorable transfer despite different parameter updates. Random sampling remains competitive, while filtering a source that performs poorly alone yields no consistent gain in our mixed-pool experiments. Selection strategies should therefore demonstrate gains beyond random sampling while accounting for screening cost.

AI USE STATEMENT

We used ChatGPT and Codex to assist with manuscript editing and language polishing. The authors take responsibility for the final manuscript.

ETHICS STATEMENT

Our experiments use released model checkpoints and existing research datasets. Data sources and preprocessing are documented in Appendix B. Reuse of these resources should respect their licenses and access restrictions. Distillation may transfer teacher biases, errors, or unsafe behaviors along with useful capabilities. Our reasoning and task-performance evaluations do not establish safety or fairness, and downstream deployment requires separate evaluation of these risks.

REPRODUCIBILITY STATEMENT

Appendix B documents model and data sources, training hyperparameters, support construction, selection seeds, prompt templates, and evaluation protocols. Appendices C and D report extended prompt-count and source-comparison results. Appendix E specifies the functional diagnostics, hidden-state interventions, and recovery protocols, while Appendix F details selection methods and per-support results. Appendix G reports supplementary training-efficiency protocols and results.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5be69a584901a26c521c2b51e40a4c20-Abstract-Conference.html.

Art of Problem Solving. AIME problems and solutions. AoPS Wiki, n.d. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *Automata, Languages and Programming (ICALP)*, volume 2380 of *Lecture Notes in Computer Science*, pp. 693–703. Springer, 2002. doi: 10.1007/3-540-45465-9_59. URL https://link.springer.com/chapter/10.1007/3-540-45465-9_59.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025. URL https://arxiv.org/abs/2502.01456.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026. doi: 10.48550/arXiv.2606.19348. URL https://arxiv.org/abs/2606.19348.

Run-Ze Fan, Zengzhi Wang, and Pengfei Liu. MegaScience: Pushing the frontiers of post-training datasets for science reasoning. *arXiv preprint arXiv:2507.16812*, 2025. URL https://arxiv.org/abs/2507.16812.

Zixuan Fu, Bingxiang He, Yuxin Zuo, Haohuan Huang, Jinqian Zhang, Ruhang Xiao, Cheng Qian, Qinyu Luo, Huan-ang Gao, Yudong Wang, Zhiyuan Liu, Ning Ding, and Chaojun Xiao. Rethinking on-policy distillation of large language models II: One training example. *arXiv preprint arXiv:2609.04172*, 2026. URL https://arxiv.org/abs/2609.04172.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://www.nature.com/articles/s41586-025-09422-z.

Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, Ning Ding, and Zhiyuan Liu. JustRL: Scaling a 1.5B LLM with a simple RL recipe. *arXiv preprint arXiv:2512.16649*, 2025. URL https://arxiv.org/abs/2512.16649.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. DeepMath-103K: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kHB5Te5IWm.

HMMT. Archive of February 2025. Harvard–MIT Mathematics Tournament, 2025a. URL https://www.hmmt.org/www/archive/282.

HMMT. Archive of November 2025. Harvard–MIT Mathematics Tournament, 2025b. URL https://www.hmmt.org/www/archive/291.

Hugging Face. Open R1: A fully open reproduction of DeepSeek-R1. GitHub repository, 2025. URL https://github.com/huggingface/open-r1.

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 *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/94074dd5a072d28ff75a76dabed43767-Abstract-Conference.html.

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

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*, 2026. doi: 10.48550/arXiv.2603.11137. URL https://arxiv.org/abs/2603.11137.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Hynek Kydlíček. Math-Verify: Math verification library. Software, version 0.9.0, 2026. URL https://pypi.org/project/math-verify/0.9.0/.

Fengxiang Li, Han Zhang, Haoyang Huang, Jinghui Wang, Jinhua Hao, Kun Yuan, Mengtong Li, Minglei Zhang, Pengcheng Xu, Wenhao Zhuang, et al. KAT-Coder-V2 technical report. *arXiv preprint arXiv:2603.27703*, 2026a. doi: 10.48550/arXiv.2603.27703. URL https://arxiv.org/abs/2603.27703.

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

Zhaoyi Li, Deyang Kong, Yuan Wei, Evan Yang, Ranran Shen, Mahardika Krisna Ihsani, Ming Yang, Wei Zhang, Chuan Hao, Jian Yang, Ran Tao, Bryan Dai, Shikun Zhang, Wei Ye, Ying Wei, and Defu Lian. 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. URL https://arxiv.org/abs/2608.16647.

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

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract.html.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 20475–20508, 2025. doi: 10.52202/085713-0608. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1a22b912945fb7c0bdd079e792b31b6f-Abstract-Conference.html.

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

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. DeepScaleR-1.5B-Preview. Hugging Face model card, 2025. URL https://huggingface.co/agentica-org/DeepScaleR-1.5B-Preview.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The Berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392. PMLR, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1645. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/46499a0622ecf568b72d17b61e45dbd5-Abstract-Datasets_and_Benchmarks_Track.html.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Jiaxuan Wang, Xuan Ouyang, Zhiyu Chen, Yulan Hu, Zheng Pan, Xin Li, and Lan-Zhe Guo. TRACE: Distilling where it matters via token-routed self on-policy alignment. *arXiv preprint arXiv:2605.10194*, 2026. URL https://arxiv.org/abs/2605.10194.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example. In *Advances in Neural Information Processing Systems*, volume 38, pp. 136065–136108, 2025. doi: 10.52202/085713-4093. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/b1ea3f93167c55f3fc2999b68a170ff3-Abstract-Conference.html.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. Light-R1: Curriculum SFT, DPO and RL for long COT from scratch and beyond. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 318–327. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-industry.24. URL https://aclanthology.org/2025.acl-industry.24/.

Siye Wu, Kai Yang, Yuchen Cai, Xin Xu, Peng-Yuan Wang, Jiaxuan Wang, Jiashun Liu, Jiafei Lyu, Yangkun Chen, Saiyong Yang, and Yanghua Xiao. Consolidating RLVR capabilities across domains: A deep dive into fusion paradigms. *arXiv preprint arXiv:2608.27409*, 2026. URL https://arxiv.org/abs/2608.27409.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54104–54132. PMLR, 2024. URL https://proceedings.mlr.press/v235/xia24c.html.

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. URL https://arxiv.org/abs/2601.02780.

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. URL https://arxiv.org/abs/2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. GLM-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026. URL https://arxiv.org/abs/2602.15763.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025. URL https://arxiv.org/abs/2506.05176.

Mao Zheng, Zheng Li, Tao Chen, Mingyang Song, and Di Wang. HY-MT1.5 technical report. *arXiv preprint arXiv:2512.24092*, 2025. doi: 10.48550/arXiv.2512.24092. URL https://arxiv.org/abs/2512.24092.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ac662d74829e4407ce1d126477f4a03a-Abstract-Conference.html.

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. URL https://arxiv.org/abs/2311.07911.

A Related Work

On-policy distillation. Autoregressive distillation uses teacher supervision on student-generated sequences to address the mismatch between training and inference (Lin et al., 2020; Agarwal et al., 2024). Subsequent work develops divergence objectives and optimization strategies for language-model distillation (Gu et al., 2024; Ko et al., 2025; 2026), including selective supervision on critical spans in privileged-context self-distillation (Wang et al., 2026). These developments establish the algorithmic basis for our study, which examines how prompt quantity and composition affect transfer under a fixed OPD training procedure.

Understanding OPD transfer. Li et al. (2026b) identify compatible thinking patterns and additional transferable teacher capabilities as factors associated with successful OPD, and improve transfer through prompt-content and template alignment. Concurrent work demonstrates substantial transfer from a single query, relating its effectiveness to training-state coverage and studying the effects of query diversity and alignment dynamics (Fu et al., 2026). Other concurrent work shows that OPD can propagate both strengths and weaknesses across domains, with generalization depending on model origin and interactions among teachers (Li et al., 2026c). Our independently conducted study also finds substantial few-prompt and cross-domain transfer, and examines how prompt-count scaling and source preferences vary across teacher–student pairs. Controlled teacher comparisons and support-switch interventions connect these preferences to unfavorable transfer and recovery. Parameter and functional diagnostics further examine how recovery can emerge through different parameter updates.

Data requirements and selection. Small-data post-training has also been studied in supervised instruction tuning and reinforcement learning. LIMA demonstrates strong instruction-following performance from a small curated dataset (Zhou et al., 2023a), while one-shot RLVR obtains substantial reasoning gains from a single training example (Wang et al., 2025). Gradient-based methods such as LESS select instruction data according to its relevance to target capabilities (Xia et al., 2024). In OPD, a fixed prompt support supplies evolving student trajectories with teacher feedback, so distinct prompt count and supervision volume are separate quantities. We evaluate whether difficulty, length, semantic diversity, and gradient-based selection improve transfer over random sampling at fixed support sizes. We also test whether a source's poor standalone performance justifies removing it from a mixed candidate pool.

B Implementation and Evaluation Details

B.1 Model and data sources

Table 1 lists the student and teacher models and OPD data. The external JustRL-DeepSeek-1.5B teacher is an RL descendant of DeepSeek-R1-Distill-Qwen-1.5B, with reported RL data DAPO-Math-17k (He et al., 2025). DeepScaleR-1.5B-Preview also continues RL from DeepSeek-R1-Distill-Qwen-1.5B, using approximately 40K problems from AIME, AMC, Omni-MATH, and Still (Luo et al., 2025). Other released continuation teachers are Nemotron-Research-Reasoning-Qwen-1.5B (v1) (Liu et al., 2025), Qwen3-4B-Inst-Mix (Wu et al., 2026), and Light-R1-7B-DS (Wen et al., 2025). The Qwen3-30B-A3B-Instruct-2507 teacher is an official release. The Math-RL and Code-RL teachers differ in RL data and training history; the teacher-switch comparison does not isolate the effect of RL data domain.

Model construction. Math RL and code RL start from Qwen3-4B using DeepMath L6 (He et al., 2026) and Eurus-2 Code (Cui et al., 2025), respectively. The 1.7B branch starts from Qwen3-1.7B-Base, followed by our direct-code SFT checkpoint at step 554 and then code RL on cleaned Open-R1 Code (Hugging Face, 2025). Checkpoint steps below identify the selected models, not necessarily the total training duration. The direct-code OPD student is initialized at this SFT-554 checkpoint, and its teacher continues from the same checkpoint with code RL. This initialization is distinct from the official Qwen3-1.7B used in the cross-model comparisons. DeepMath L6 retains examples with level ≥ 6.

Direct-code SFT data. Our curated mixture combines NVIDIA OpenCodeInstruct, bigcode self-oss-instruct-sc2, Magicoder-OSS-Instruct-75K, and the synthetic and seed SFT subsets of rStar-Coder.

We standardize the examples to direct Python code without <think> traces and decontaminate the data against HumanEval+, MBPP+, and LiveCodeBench v6.

B.2 Training hyperparameters

Tables 6–10 report the key settings for teacher construction and OPD.

Sampled-token OPD implementation. We implement the distribution-matching objective in Equation 1 through a sampled-token policy-gradient surrogate (Lu & Thinking Machines Lab, 2025). For a token y_t sampled from the rollout policy $\pi_{\bar{\theta}}$ at prefix $s_t = (x, y_{<t})$, the K1 log-ratio and detached feedback are

$$\begin{aligned} \widehat{k}_t &= \log \pi_{\bar{\theta}}(y_t \mid s_t) - \log \pi_T(y_t \mid s_t), \\ \widehat{A}_t &= -\operatorname{sg}\Big[\operatorname{clip}(\widehat{k}_t, -10, 10)\Big], \end{aligned} \tag{2}$$

where sg denotes stop-gradient. The update averages over valid response tokens without accumulating future-token feedback. Task rewards are used for monitoring and reward-based selection baselines, but do not enter the OPD loss.

OPD training duration. Nemotron runs use 75 optimization steps; Mix-RL runs use 25. The `s100` label specifies the data-presentation schedule, not the number of optimization steps actually executed. The prompt-count comparisons in Section 3 report step-100 endpoints for Qwen3-4B Code-RL and JustRL, and step 15 for 30B Instruct to 4B with DeepMath. The Qwen3-4B Math-RL comparison matches the total number of rollouts at 3,840.

Comparison protocols. Prompt-count sweeps vary M within a pair and source under the specified rollout and update protocol. Source comparisons fix the pair and $M = 48$ while changing the source or mixture. The teacher-switch comparison fixes the initial Qwen3-4B student and both supports while swapping Math-RL and Code-RL teachers. Within-pool selection comparisons fix the candidate pool, support size, and training configuration while changing the selector. The mixed-pool comparison samples before and after excluding GSM8K from DeepMath–DAPO–GSM8K candidates. We record distinct prompts, fresh rollouts, optimization updates, and token budgets separately; fixing M does not match computational cost.

All lengths in the following tables are in tokens. PPO mini-batch sizes are configuration values; per-device forward and backward passes use dynamic token packing.

B.3 Prompt support construction

For the nested sweeps described below, each fixed candidate population and selection seed defines one ordered sequence. The support $\mathcal{S}_M$ contains its first M entries, so $\mathcal{S}_{M_1} \subset \mathcal{S}_{M_2}$ for $M_1 < M_2$; the construction scripts assert this nesting. The selection seed is 42 except for the Code48 redraws. In the data-presentation pipeline, the training seed is also 42: it controls presentation order without changing support membership. Each support is frozen before its run. Nesting applies within a pool and selection seed, not across different sources or redraw seeds.

Stratified nested prefixes. At each prefix extension, the construction chooses the stratum with the largest relative quota deficit. Within a stratum, a SHA-256 ordering keyed by the selection seed and example ID randomizes the examples. This produces covariate-balanced nested prefixes, not a uniform permutation of the entire pool.

For the main Qwen3-4B Math sweep, the population is the first 3,840 examples consumed by the corresponding full-data OPD run (batch 256, 15 updates), drawn from DeepMath-103K's `train_filtered_level6` split. Strata cross five baseline-step bins, four prompt-length bins, and binary answer correctness. Supports are nested at $M \in \{1, 48, 384, 3072, 3840\}$. The earlier batch-1,024, 25-update sweep instead uses a 25,600-example population and $M \in \{1, 48, 384, 3072, 25600\}$. The main DeepMath48 comes from the 3,840-example sequence, not a separate draw from the earlier 25,600-example population. These construction populations are distinct from source-level full-pool references, including the 57,046-prompt JustRL DeepMath reference.

Table 6: **Math RL (GRPO).**

Hyperparameter	Value
Train batch size	128
PPO mini-batch size	128
Rollout n	8
Max. prompt length	2,048
Max. response length	16,384
Temperature	1.0
Top-p	1.0
Learning rate	1×10^{-6}
KL coefficient	0
Checkpoint step	500

Table 7: **Code RL (GRPO).**

Hyperparameter	Value
Train batch size	128
PPO mini-batch size	128
Rollout n	8
Max. prompt length	2,048
Max. response length	8,192
Temperature	1.0
Top-p	1.0
Learning rate	1×10^{-6}
KL coefficient	0
Checkpoint step	300

Table 8: **Direct-code SFT.**

Hyperparameter	Value
Train batch size	128
Max. sequence length	4,096
Warm-up ratio	0.03
Learning rate	3×10^{-6}
Training epochs	1
Checkpoint step	554

Table 9: **Code RL after SFT (GRPO).**

Hyperparameter	Value
Train batch size	128
PPO mini-batch size	128
Rollout n	8
Max. prompt length	4,096
Max. response length	4,096
Temperature	0.8
Top-p	0.95
Learning rate	5×10^{-7}
KL coefficient	0.001
Checkpoint step	400

Table 10: **OPD hyperparameters.**

Hyperparameter	Value
Train batch size	256
PPO mini-batch size	256
Rollout n	1
Max. prompt length	2,048
Max. response length	16,384 (Math) 8,192 (Code)
Temperature	1.0
Top-p	1.0
Learning rate	1×10^{-5}
Optimization steps	Pair-specific
Distillation objective	K1 sampled reverse-KL + policy gradient
Log-ratio clamp	$[-10, 10]$

The Qwen3-4B Code sweep uses all 22,618 Eurus Code GRPO examples, stratified by `data_source` and six prompt-length quantiles, with $M \in \{1, 48, 3840, 22618\}$. Code48 redraws use the same population and algorithm with selection seeds 43–47. They are not nested with the seed-42 support. These redraws change support membership while keeping the training presentation seed at 42.

Uniform nested prefixes. The following controls instead take prefixes of one NumPy PCG64 permutation with selection seed 42, without difficulty or gradient screening. For the 30B teacher, cleaned DAPO-Math (Yu et al., 2025) has 17,170 candidates and $M \in \{1, 8, 48, 3840\}$; GSM8K train (Cobbe et al., 2021) has 7,473 candidates and $M \in \{1, 48\}$. The JustRL DAPO $M = 4$, 8, and 384 supports also use uniform nested prefixes of the cleaned 17,170-example pool. Science48 is a frozen 48-example prefix of 126,397 physics, chemistry, and biology examples from TextbookReasoning (Fan et al., 2025).

Shared supports across pairs. Qwen3-4B Math, 30B-to-4B DeepMath, Nemotron, and Mix-RL use the same 48 DeepMath IDs. JustRL's DeepMath scan directly reuses the frozen Qwen3-4B Math support files rather than drawing new supports. Light-R1's DeepMath $M = 8$ support uses the same schedule as JustRL's DeepMath random-8 support, drawn from `train_filtered_level6`

(level ≥ 6). The Code sweep, Nemotron, and Mix-RL likewise share the seed-42 Code48; 30B, Nemotron, and Mix-RL share the frozen Science48 indices. Shared membership does not imply identical presentation schedules or training budgets. Nemotron and Mix-RL differ in model pair, training endpoint, and DeepMath presentation schedule; source effects are compared within each pair.

Instruction-following and agent supports. IF48 and Agent48 come from the LLM-Fusion training parquet files in the `Siye01/llm-fusion` collection (Wu et al., 2026). The IF file, `IF/train-00000-of-00001.parquet`, contains 16,575 examples; strata cross `rm_type` and six prompt-length quantiles. All examples have type `ifevalg`, including all 48 selected prompts. The Agent file, `Agent/train-00000-of-00001.parquet`, contains 10,229 examples; strata cross ground-truth tool-call counts $0/1/2/3/4+$ and six prompt-length quantiles. Both use the stratified construction above with seed 42 and freeze $M = 48$. Agent48 contains 6, 28, 6, 2, and 6 examples in the five tool-count bins, respectively.

Mixtures. Mix3 (also called Mix48) takes the first 16 examples in the stored support-file order of each frozen DeepMath48, Code48, and Science48; it does not redraw them. Nemotron and Mix-RL share this 16+16+16 support. Mix5 instead selects 10 Math, 10 Science, 10 Code, 9 IF, and 9 Agent prompts from the corresponding frozen M48 supports. An example's identity hashes its `prompt`, `data_source`, and `reward_model` fields with SHA-256. Within each domain, examples are reranked by the SHA-256 hash of `42:{domain}:{identity}`, and the required number is retained. The combined presentation order uses the hash of `42:support:{domain}:{identity}`. Thus, each domain's Mix5 subset belongs to its frozen M48 parent but is not selected as a file-order prefix. Its Math, Code, and Science subsets overlap Mix3's corresponding 16-prompt subsets in 3, 2, and 1 prompts, respectively. Moving from Mix3 to Mix5 therefore changes both domain allocation and selected prompt membership.

B.4 Prompting and evaluation

Appendix B.5 gives the training and evaluation templates for math reasoning, 4B code OPD, and the direct-code SFT/GRPO/OPD family, as well as evaluation-only templates for GPQA-Diamond, IFEval/IFBench, and BFCL-v3. In the direct-code templates, HumanEval+/MBPP+ and LiveCodeBench identify evaluation task styles, not training data sources.

Generation format. Prompt instructions are distinct from generation mode: a non-thinking template followed by the instruction to think first can still produce reasoning text. The direct-code templates contain no `<think>` prefix and request no reasoning trace.

The math suite comprises AIME 2024/2025 (Art of Problem Solving, n.d.) and HMMT February/November 2025 (HMMT, 2025a;b).

Math-OOD evaluation uses GPQA-Diamond (Rein et al., 2024), HumanEval+, and LiveCodeBench v6. Both code settings use the EvalPlus benchmarks HumanEval+ and MBPP+ (Liu et al., 2023), and LiveCodeBench v6 (`release_v6`; Jain et al., 2025). Multi-domain evaluations additionally include IFEval (Zhou et al., 2023b), IFBench (Pyatkin et al., 2025), and BFCL-v3 (Patil et al., 2025). Table 11 reports the evaluation settings, which are distinct from the training budgets.

Table 11: **Evaluation configurations.** Lengths are in tokens; all settings report mean@k and pass@k, with $k = 16$ for Math and $k = 8$ for OOD.

Parameter	Math / OOD	Code (4B)	Direct-code
Samples per question (k)	16 / 8	8	16
Temperature	1.0	1.0	0.6
Top-p	1.0	1.0	0.95
Top-k / min-p	-1 / 0.0	-1 / 0.0	-1 / 0.0
Max. generation length	16,384	16,384	4,096

Aggregation. Mean@k averages the fraction of correct responses across questions; pass@k is the fraction of questions with at least one correct response among k samples. Initial-student, teacher, and full-pool OPD references are included where available. For each reported metric, we take an

unweighted average over the benchmarks included in the corresponding aggregate:

$$\text{Overall} = \frac{1}{B}\sum_{b=1}^{B}\text{score}_b,$$

where B is the number of included benchmarks and score_b is that metric's score on benchmark b. Each benchmark has equal weight, irrespective of its question count. For the multi-domain comparison in Table 3, Avg_6 averages six capability groups, with IFEval and IFBench first averaged within the instruction-following group; HumanEval+ and LCB each retain a separate weight. This is distinct from the four-task Avg in Figure 3a,c. The Eurus Code row uses the original support draw.

B.5 Prompt templates

Training and Evaluation Prompt Template for Math Reasoning

```
<|im_start|>user
{question}

Please reason step by step, and put your final answer within \boxed{}.<|im_end|>
<|im_start|>assistant
```

Training and Evaluation Prompt Template for Code Generation

````
<|im_start|>user
{question}

Write Python code to solve the problem. Present the code in
```python
Your code
```
at the end.
You need to think first then write the Python code.<|im_end|>
<|im_start|>assistant
````

Training and Evaluation Prompt Template for Direct Code Generation

Function-style — HumanEval+ / MBPP+

````
<|im_start|>user
{question}

Complete the requested Python function. Preserve the required function or class
interface exactly. Return only the final implementation inside one
```python
code block
```.<|im_end|>
<|im_start|>assistant
````

Competitive-programming-style — LiveCodeBench

````
<|im_start|>user
{question}

[Optional starter code:
```python
{starter_code}
```]
Solve the programming problem in Python. Preserve any required starter-code
interface. Return only the complete final program inside one
```python
code block
```.<|im_end|>
<|im_start|>assistant
````

Evaluation Prompt Template for GPQA-Diamond

```
<|im_start|>user
Answer the following multiple choice question. The last line of your response
should be of the following format: 'Answer: $LETTER' (without quotes) where
LETTER is one of ABCD. Think step by step before answering.

{question}

A) {choice_A}
B) {choice_B}
C) {choice_C}
D) {choice_D}<|im_end|>
<|im_start|>assistant
```

The final line must be `Answer: $A`, with `A` replaced by the selected letter (A–D); the dollar sign is literal.

Evaluation Prompt Template for Instruction Following

Query-only IFEval / IFBench

```
<|im_start|>user
{instruction_following_query}<|im_end|>
<|im_start|>assistant
```

No additional system prompt or shared instruction is appended; the dataset query already contains the instruction-following constraints.

Evaluation Prompt Template for BFCL-v3: Initial Turn

```
<|im_start|>system
You are a helpful assistant with access to a set of tools. Use the tools to
accomplish the user's request. Call one tool at a time and wait for its result
before deciding the next step. When the task is complete, reply to the user
without calling a tool.

# Tools

You may call one or more functions to assist with the user query.

You are provided with function signatures within <tools></tools> XML tags:
<tools>
{tool_schema_1}
{tool_schema_2}
...
{tool_schema_n}
</tools>

For each function call, return a json object with function name and arguments
within <tool_call></tool_call> XML tags:
<tool_call>
{"name": <function-name>, "arguments": <args-json-object>}
</tool_call><|im_end|>
<|im_start|>user
{user_request}<|im_end|>
<|im_start|>assistant
```

Each example exposes only the tool subset specified by `extra_info.tool_selection`, not the full tool inventory.

Evaluation Prompt Template for BFCL-v3: Subsequent Turns

Model tool call — Assistant

```
<|im_start|>assistant
<tool_call>
{"name": "{function_name}", "arguments": {arguments}}
</tool_call><|im_end|>
```

Environment execution result — User role

```
<|im_start|>user
<tool_response>
{tool_execution_result}
</tool_response><|im_end|>
<|im_start|>assistant
```

Task completion — Final response without a tool call

```
<|im_start|>assistant
{final_response}<|im_end|>
```

BFCL-v3 uses a multi-turn tool agent: one tool is called at a time, and its result is awaited before the next action.

C Complete Results on Data Requirements

C.1 Prompt-count sweeps

Tables 12–17 give benchmark-level prompt-count results for the six teacher–student pairs in Figure 2a–f. All scores are percentages, and M counts distinct training prompts. Math is the unweighted mean of AIME24, AIME25, HMMT-Feb, and HMMT-Nov; Code macro averages HumanEval+, MBPP+, and LCB-v6. A dash indicates that M does not apply to a reference model. Reported values and their precision are preserved.

Table 12: **Math-RL continuation.** Math GRPO-500 to Qwen3-4B (Figure 2a), with 3,840 rollouts per OPD run. Math reports mean@16; other tasks report mean@8.

Source	M	AIME24	AIME25	HMMT Feb	HMMT Nov	Math	GPQA-D	HE+	LCB
Initial	–	22.29	19.79	11.04	8.75	15.47	42.74	77.06	17.71
Teacher	–	62.71	56.46	34.38	39.79	48.33	52.65	83.92	20.07
DeepMath	1	55.21	54.17	32.29	37.29	44.74	51.70	82.39	29.93
DeepMath	48	60.21	53.12	33.33	38.96	46.41	53.22	83.23	27.21
DeepMath	3,840	61.25	55.83	32.92	37.08	46.77	52.15	85.14	29.79

Table 13: **Code-RL continuation.** Code GRPO-300 to Qwen3-4B (Figure 2b), at OPD step 100; all scores are mean@8. The principal $M = 1$ support is ID 5038 (B in Section 5); rows below the separator give additional single-prompt experiments.

Source / support	M	HE+	MBPP+	LCB	Code macro
Initial	–	76.60	65.15	18.14	53.30
Teacher	–	86.36	71.03	28.00	61.80
Eurus Code (ID 5038)	1	79.04	68.39	26.43	57.95
Eurus Code	48	86.66	70.14	28.57	61.79
Eurus Code	3,840	86.05	70.90	28.14	61.70
Eurus Code (full)	22,618	86.74	71.06	27.86	61.89
A' / ID 15240 (replication)	1	69.74	61.71	18.64	50.03
Selected M1: ID 187	1	83.23	70.07	27.93	60.41
Selected M1: ID 228	1	83.23	69.21	29.21	60.55
Selected M1: ID 19303	1	86.20	68.92	29.00	61.37

Table 14: **Direct-code RL continuation.** Code GRPO-400 to Qwen3-1.7B-Base after direct-code SFT at step 554 (Figure 2c). OPD endpoints are at step 50; all scores are mean@16.

Source / reference	M	HE+	MBPP+	LCB	Code macro
Initial: direct-code SFT-554	–	44.32	52.88	11.25	36.15
Teacher: Code GRPO-400	–	58.12	58.47	15.96	44.18
Open-R1 Code	1	58.82	58.74	15.96	44.51
Open-R1 Code	256	59.98	59.51	15.57	45.02
Open-R1 Code	3,840	59.68	59.62	16.61	45.30
Open-R1 Code (full)	6,519	58.73	58.80	15.50	44.34

Table 15: **JustRL continuation.** JustRL-DeepSeek-1.5B to DeepSeek-R1-Distill-Qwen-1.5B (Figure 2d), at OPD step 100. Math reports mean@16; other tasks report mean@8. Full denotes the available source pool; DeepMath $M = 8$ is the random support in the prompt-count sweep.

Source	M	AIME24	AIME25	HMMT Feb	HMMT Nov	Math	GPQA-D	HE+	LCB
Initial	–	27.50	25.63	12.50	11.25	19.22	35.92	54.34	13.50
Teacher	–	53.96	39.38	21.67	23.75	34.69	40.47	64.56	19.36
DAPO	1	42.71	30.42	16.88	17.08	26.77	38.13	60.98	16.29
DAPO	2	49.58	36.25	20.83	20.83	31.87	38.64	62.27	16.71
DAPO	4	47.71	36.04	22.08	24.38	32.55	41.10	63.87	17.50
DAPO	8	51.04	35.00	21.25	22.71	32.50	40.91	64.10	18.64
DAPO	384	49.38	39.17	21.04	22.29	32.97	39.14	63.57	19.21
DAPO (full)	17,170	50.21	35.83	21.46	21.88	32.34	40.28	64.25	19.21
DeepMath	1	44.38	30.83	20.62	16.67	28.12	38.45	61.13	16.57
DeepMath	8	45.42	34.38	17.92	18.96	29.17	38.83	63.34	17.43
DeepMath	48	46.67	33.96	20.42	21.04	30.52	38.51	64.02	17.21
DeepMath	384	49.17	35.21	22.29	20.00	31.67	38.83	63.64	18.79
DeepMath	3,840	49.79	37.92	20.83	21.67	32.55	40.97	63.80	18.64
DeepMath (full)	57,046	50.21	38.54	20.83	19.79	32.34	39.14	63.03	18.57

Table 16: **Cross-model distillation.** Qwen3-30B-A3B-Instruct-2507 to Qwen3-4B (Figure 2e), at OPD step 15. Math reports mean@16; other tasks report mean@8. Full-pool endpoints at other training durations appear in Table 18.

Source	M	AIME24	AIME25	HMMT Feb	HMMT Nov	Math	GPQA-D	HE+	LCB
Initial	–	22.29	19.79	11.04	8.75	15.47	42.74	77.13	17.71
Teacher	–	75.21	61.67	42.50	56.25	58.91	54.86	82.39	33.14
DeepMath	1	46.0	46.7	24.6	31.9	37.29	48.3	82.4	24.3
DeepMath	48	56.46	45.63	29.38	39.79	42.81	52.15	82.32	26.79
DeepMath	3,840	53.8	48.3	30.2	38.1	42.60	52.0	83.7	27.0
DAPO	1	49.4	40.4	26.2	31.5	36.88	49.7	83.5	26.3
DAPO	8	56.0	45.0	28.3	39.0	42.08	50.9	81.4	26.9
DAPO	48	54.4	45.6	27.7	34.0	40.42	51.5	82.2	27.9
DAPO	3,840	51.5	43.3	29.0	35.2	39.74	51.5	81.1	27.3

Table 17: **Cross-model distillation.** Qwen3-30B-A3B-Instruct-2507 to the official Qwen3-1.7B checkpoint (Figure 2f), not the SFT-554 initialization in Table 14. Math reports mean@16; other tasks report mean@8.

Source	M	AIME24	AIME25	HMMT Feb	HMMT Nov	Math	GPQA-D	HE+	LCB
Initial	–	13.33	10.00	5.00	5.63	8.49	29.10	59.83	12.36
Teacher	–	75.21	61.67	42.50	56.25	58.91	54.86	82.39	33.14
DeepMath	1	27.3	24.0	16.0	14.2	20.4	30.1	65.5	16.1
DeepMath	8	33.1	26.7	15.0	15.2	22.5	32.5	65.8	19.4
DeepMath	48	35.0	26.9	14.2	17.1	23.3	36.2	66.7	18.1
DeepMath	3,840	36.0	30.8	14.8	16.7	24.6	32.7	68.9	18.7
DAPO	1	33.5	29.2	15.0	16.2	23.5	35.6	71.2	18.2
DAPO	8	32.1	26.0	14.6	10.2	20.7	33.9	66.9	17.9
DAPO	48	30.2	27.3	11.7	15.4	21.1	33.7	64.1	16.9
DAPO	3,840	33.5	29.8	15.4	14.8	23.4	34.5	66.5	18.9

C.2 Additional full-pool references

The following endpoints use different training durations or budgets from the principal sweeps above. Full denotes the available sampling pool, not a guarantee that every prompt was consumed.

Table 18: **Additional full-pool OPD endpoints.** Math reports mean@16; other tasks report mean@8. Math-RL runs use batch size 1,024, with 25,600 and 51,200 rollouts at OPD steps 25 and 50, respectively.

Source	Step	AIME24	AIME25	HMMT Feb	HMMT Nov	Math	GPQA-D	HE+	LCB
Math GRPO-500 → Qwen3-4B									
DeepMath	25	62.92	54.58	32.29	38.75	47.14	52.78	84.91	29.86
DeepMath	50	64.79	56.04	34.38	41.46	49.17	54.29	84.45	28.07
Qwen3-30B-A3B-Instruct-2507 → Qwen3-4B									
DeepMath	25	55.0	48.8	27.3	37.9	42.2	48.4	82.8	25.8
DeepMath	50	56.9	49.0	31.7	38.8	44.1	48.4	83.5	26.6
DAPO	25	53.1	46.7	27.9	35.2	40.7	49.2	80.9	25.9
Qwen3-30B-A3B-Instruct-2507 → Qwen3-1.7B									
DeepMath	25	34.0	27.3	17.3	16.7	23.8	34.3	66.8	17.9
DeepMath	50	34.6	29.0	16.9	16.7	24.3	35.2	67.8	18.6
DAPO	25	34.2	26.9	16.5	14.2	22.9	35.5	66.8	19.1

C.3 Support size and the learning process

Figure 8 complements the endpoint tables with training trajectories for the Qwen3-4B Math-RL continuation pair. The $M = 48$ and $M = 3{,}840$ runs closely track one another in rollout length and logged K1. The $M = 1$ run already improves validation accuracy at the first post-training check (step 3), but generates longer responses with larger step-to-step length fluctuations. At step 15, mean rollout lengths are approximately 7,293, 7,377, and 11,676 tokens for $M = 48$, 3,840, and 1, respectively. Similar endpoint performance therefore need not imply the same generation behavior during training.

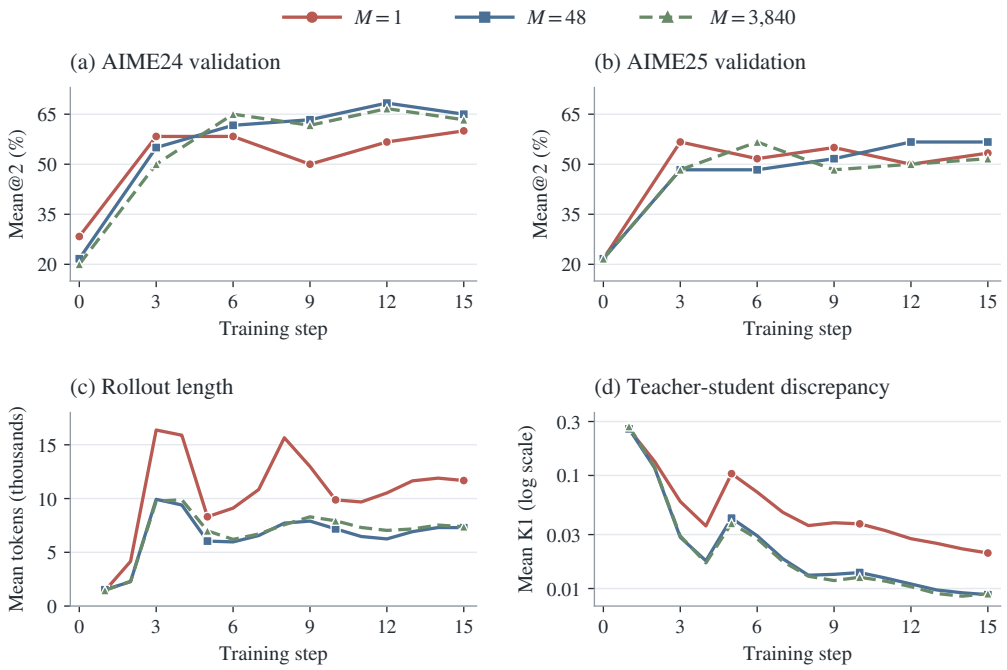


Figure 8: **Support size and learning dynamics.** All runs use the Qwen3-4B Math-RL continuation pair. **(a–b)** Training-time AIME24 and AIME25 validation accuracy (mean@2). **(c)** Mean rollout response length, in thousands of tokens. **(d)** Teacher–student discrepancy, reported as logged mean K1 on a logarithmic scale. These validation traces are distinct from the final mean@16/mean@8 evaluations in the complete result tables.

C.4 Single-prompt code supports

The following cards summarize the single-prompt supports compared in Figure 1b for the Qwen3-4B Code-RL setting.

P1 Conditional arithmetic — Index 15240

Given positive integers A and B, output $A + B$ if $A \mid B$; otherwise, output $B - A$.

P2 Largest perfect power — Index 187

Given an integer X with $1 \leq X \leq 1000$, find the largest perfect power b^p not exceeding X, where b is a positive integer and p is an integer satisfying $p \geq 2$.

P3 Fixed-start Hamiltonian paths — Index 228

Given an undirected graph with $N \leq 8$ vertices, count the paths that start at vertex 1 and visit every vertex exactly once; that is, count Hamiltonian paths with the starting vertex fixed to 1.

D Additional Data-Preference Results

Figure 3 and Table 3 report the fixed-support source comparisons, teacher swap, and multi-domain mixtures in Section 4. This appendix adds teacher–source learning curves and seed robustness, the benchmark-level supervision-token control, and the matched RL-instance overlap comparison. Support membership and mixture construction are specified in Appendix B.3.

D.1 Teacher and source effects

Figure 9 shows when teacher-dependent differences appear. With Math48 fixed, the Math-RL teacher yields higher AIME25 validation accuracy than the Code-RL teacher from the first post-training validation at step 5. With Code48 fixed, the Code-RL teacher yields higher TACO validation accuracy at the same check. The separation persists at subsequent validation checkpoints, even though logged mean K1 generally decreases in all four runs, with transient increases. A declining K1 trace thus coexists with distinct validation outcomes. The two validation tasks are kept separate rather than pooled: these curves describe the onset and persistence of teacher-dependent differences, while Figure 3b compares source preferences on the same final-evaluation benchmarks.

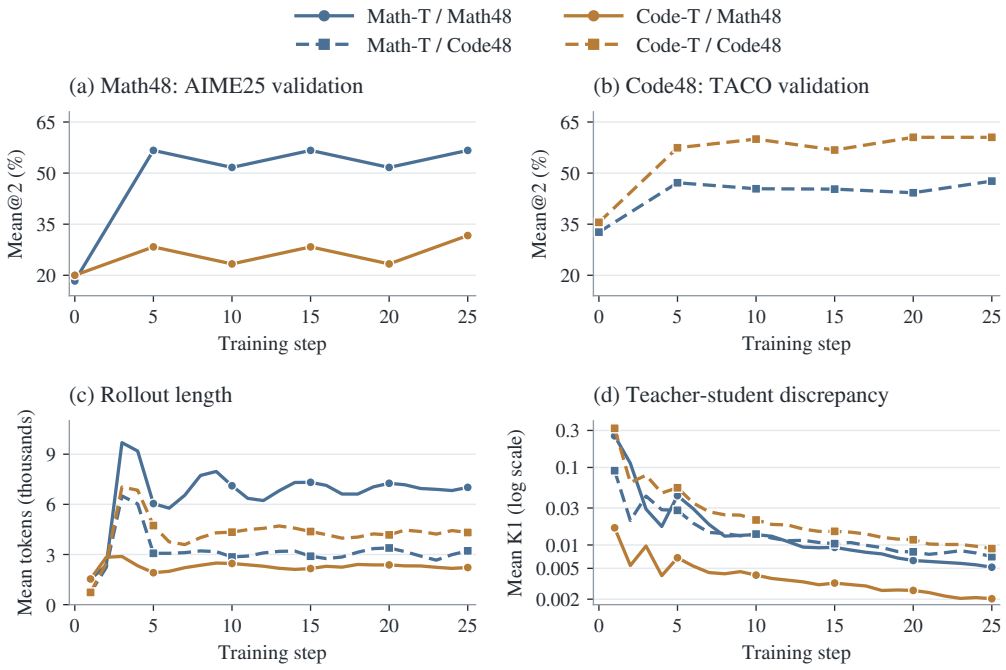


Figure 9: **Teacher and prompt-source effects during Qwen3-4B OPD.** Math-T and Code-T denote Math GRPO-500 and Code GRPO-300; Math48 and Code48 denote the fixed DeepMath and Eurus Code supports. **(a)** AIME25 validation with Math48. **(b)** TACO validation with Code48. Both report training-log mean@2. **(c)** Mean rollout response length. **(d)** Teacher–student discrepancy, reported as logged mean K1 (log scale). Panels (a–b) use different validation tasks; the common-benchmark source-preference reversal is shown separately in Figure 3b.

Seed robustness. We vary support-selection seeds over $\{42, 43, 44\}$ with training seed 42, and training seeds over $\{42, 43, 44\}$ with support-selection seed 42, keeping the student initialization fixed. Within each source and seed configuration, the two teacher conditions use identical supports ($M = 48$) and presentation schedules.

Table 19: **Seed robustness of teacher-dependent source preferences.** Avg_4 (%) across support-selection and training seeds.

	Math-RL teacher		Code-RL teacher	
Support / training seed	DeepMath48	Code48	DeepMath48	Code48
42 / 42	54.30	46.84	44.94	50.75
43 / 42	54.20	46.62	44.79	51.27
44 / 42	54.23	46.17	45.22	50.51
42 / 43	53.87	46.37	44.69	50.31
42 / 44	54.26	46.66	44.67	50.42

Across all five configurations, Math-RL favors DeepMath48 and Code-RL favors Code48 on each of the four benchmarks.

D.2 Supervision-token control

Table 20 gives the benchmark-level results underlying Table 2. All supports contain 48 prompts. Token counts are cumulative supervised tokens, in millions; the matched GSM8K run is close to, but not exactly at, the DeepMath budget. The main-text Avg is the unweighted mean of Math, GPQA, HumanEval+, and LCB.

Table 20: **Benchmark-level supervision-token control.** $M = 48$; tokens are cumulative supervised tokens in millions, and GSM8K token matching to DeepMath is approximate.

Data	Tokens (M)	Math	GPQA	HE+	LCB
GSM8K	1.19	31.15	46.84	81.25	16.79
+ token match	16.14	31.56	48.04	82.32	17.43
DeepMath	16.63	42.81	52.15	82.32	26.79

D.3 RL-instance overlap control

DAPO is also the JustRL teacher's RL pool, so its comparison with DeepMath cannot isolate exact training-instance reuse. The matched direct-code 1.7B control fixes source mixture, prompt template, and prompt length: RL-seen and RL-unseen sets yield close transfer point estimates (45.26 versus 45.28). This provides evidence that exact RL-instance reuse is not required for the observed transfer in this controlled comparison. The 30B teachers' RL instances are unknown, so their outcomes cannot identify an overlap effect.

E Functional Changes, Failure, and Recovery

This appendix specifies the measurements, probe scopes, and supporting evidence used in Section 5: finite endpoint prediction changes, conditional KL closure, generation behavior, and recovery interventions.

E.1 Finite endpoint prediction changes

Within each comparison, each model is loaded and fully forward-evaluated on the same frozen probe prefixes. The measured effects are finite changes in endpoint conditional predictions relative to the initial student. Freezing prefixes fixes the input states; it does not linearize the model with respect to its parameters. We do not compute $J_{\theta_0}\Delta\theta$ or use gradients at initialization to predict an endpoint. Fisher geometry is used to compare the measured prediction changes, not to replace the endpoint forward passes.

For the 30B-to-4B source comparison, the frozen probe comprises 128 initial-Qwen3-4B trajectories: 32 each from Math, GPQA, HumanEval+, and LCB, totaling 73,823 prediction positions. Action support is fixed to the initial student's top 128 tokens at each position. The initial-student anchor

and all seven evaluated models use FP32 forward passes with SDPA. Student and teacher tokenizer mappings agree, and retained support mass is at least 0.996 across the evaluated models and domains. The Code trajectory study likewise fixes 128 trajectories from its own initial student, the initial student's top-128 action support, and an FP32 anchor; each checkpoint is actually forward-evaluated. These protocol details do not identify an all-domain aggregate with a Math-only comparison.

Finite endpoint vectors. At frozen prediction position j, let S_j be the initial student's top-128 token set, and let $p_{a,j}(i)$ be model a's full-vocabulary probability of token i. Here $a \in \{0, T, D\}$ denotes the initial student, teacher, or distilled endpoint, respectively. Each model's conditional distribution on the same set is

$$m_{a,j} = \sum_{k \in S_j} p_{a,j}(k), \qquad q_{a,j}(i) = \frac{p_{a,j}(i)}{m_{a,j}}, \quad i \in S_j. \tag{3}$$

We center the actual endpoint-minus-initial log-probability changes under $q_{0,j}$ and embed them using the initial probabilities:

$$d_{a,j}(i) = \log p_{a,j}(i) - \log p_{0,j}(i), \qquad \bar{d}_{a,j} = \sum_{i \in S_j} q_{0,j}(i) d_{a,j}(i), \tag{4}$$

$$v_{a,j}(i) = \sqrt{p_{0,j}(i)}\,[d_{a,j}(i) - \bar{d}_{a,j}], \qquad a \in \{T, D\}. \tag{5}$$

Thus centering uses the renormalized $q_{0,j}$, while the Fisher inner-product weight is the unnormalized $p_{0,j}(i)$, not teacher or endpoint probabilities. The square-root embedding absorbs this weight: inner products of v below are ordinary Euclidean inner products, with no second Fisher weighting. These are finite endpoint log-probability changes, not probability differences or a parameter-local approximation.

Geometry and aggregation. For the evaluated position set $\mathcal{J}$, concatenate $v_a = \bigoplus_{j \in \mathcal{J}} v_{a,j}$. Functional cosine and projection are

$$\cos_F = \frac{\sum_{j \in \mathcal{J}} \langle v_{D,j}, v_{T,j} \rangle}{\sqrt{\sum_{j \in \mathcal{J}} \|v_{D,j}\|^2} \sqrt{\sum_{j \in \mathcal{J}} \|v_{T,j}\|^2}}, \qquad \alpha = \frac{\langle v_D, v_T \rangle}{\|v_T\|^2}. \tag{6}$$

Both norms must be nonzero for cosine, and projection requires a nonzero teacher direction. The orthogonal component $e_\perp = v_D - \alpha v_T$ is defined in the same embedded space; it is not a token-level training log-ratio.

For all-domain cosine, $\mathcal{J}$ contains all evaluated positions across domains. Each position enters once; longer trajectories and domains with more positions contribute more entries. There is no trajectory-equal or domain-equal averaging, nor an average of separately computed domain cosines. Confidence intervals use whole-trajectory or whole-question resampling, as specified for each comparison; this does not alter the point-estimate definition above. Task-specific comparisons restrict $\mathcal{J}$ to their stated probe scope.

Conditional forward KL and closure. The diagnostic uses teacher-first forward KL, with each model separately renormalized on S_j:

$$K_{T\|a,j} = D_{\mathrm{KL}}(q_{T,j} \| q_{a,j}) = \sum_{i \in S_j} q_{T,j}(i) \log \frac{q_{T,j}(i)}{q_{a,j}(i)}, \qquad a \in \{0, D\}. \tag{7}$$

This is conditional KL on the initial student's frozen top-128 support, not full-vocabulary KL. Probability outside S_j is neither included in this KL nor merged into an "other" token; retained support mass $m_{a,j}$ is reported separately. In particular, the conditional KL does not directly penalize changes in the omitted mass.

Let $D_{T\|0}$ and $D_{T\|D}$ denote the baseline and remaining conditional KL, respectively, using the same prefix set and aggregation protocol. Diagnostic closure is

$$\mathrm{KLClosure} = 1 - \frac{D_{T\|D}}{D_{T\|0}}, \qquad D_{T\|0} > 0. \tag{8}$$

Reported percentages multiply this quantity by 100. A negative value denotes a larger remaining conditional KL gap, not a negative cosine or a percentage loss in accuracy. This teacher-first evaluation diagnostic is separate from the sampled reverse-KL OPD training objective. Neither closure nor directional cosine establishes equality of full prediction distributions or downstream behavior.

E.2 Probe scope

The 43 checkpoints across seven teacher families in Figure 4(a) include intermediate states, not 43 independent training runs. Their functional cosines use all-domain frozen-prefix probes, whereas the source comparisons in Figure 4(b) and the task-specific diagnostics in Figure 10 restrict the probe scope to the stated task. These scopes should not be treated as interchangeable.

E.3 Source-resolved functional diagnostics

Figure 10 supplements the source comparisons in Section 5.1 with task-specific diagnostic breakdowns. Marker shapes and colors identify the prompt sources in both the scatter plots and the matrices. The three scatter plots use different axis ranges. KL closure and directional alignment remain distinct diagnostics; the large negative closure values are not percentage decreases in benchmark accuracy.

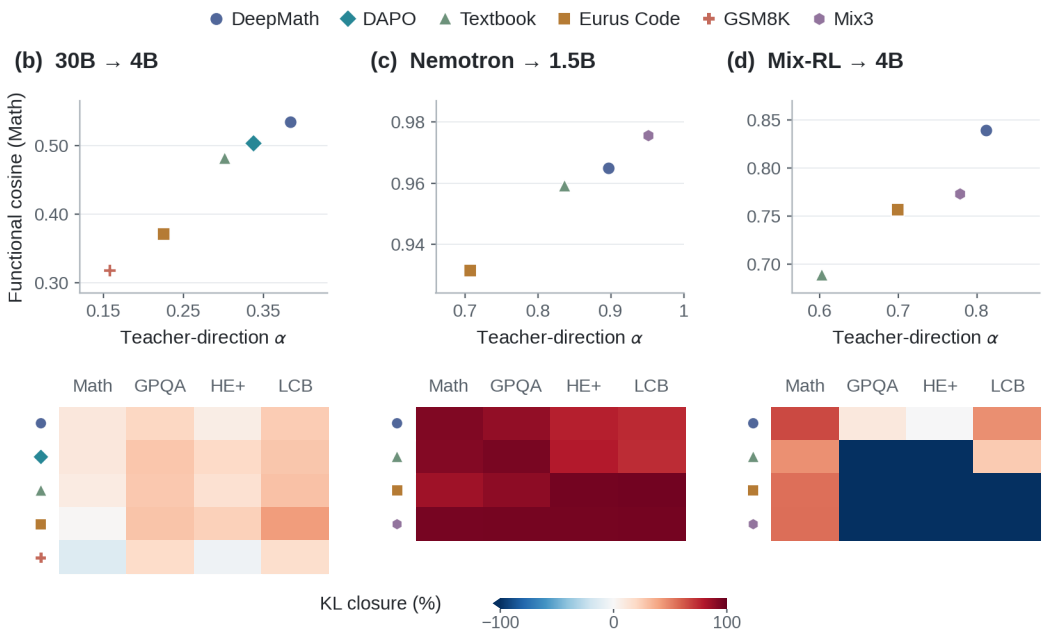


Figure 10: **Additional source-dependent functional diagnostics.** **(b)** 30B-to-4B. **(c)** Nemotron. **(d)** Mix-RL. Upper plots report Math functional cosine and teacher-direction coefficient α; lower matrices report task-specific KL closure (%). Closure uses teacher-first conditional KL on the initial student's frozen top-128 support, not a task-score recovery rate. Blue and red indicate negative and positive closure, respectively; colors saturate at -100% and 100%.

E.4 Additional generation-behavior evidence

Table 21 gives the complete accuracy and length results. Model identities and evaluation protocols are given in Table 1 and Appendix B.4, respectively.

Table 21: **Accuracy and output length for seven teacher–student pairs.** Each cell gives mean@8 accuracy (%) / mean output tokens; s denotes the OPD update step.

Model / OPD support	GPQA-D		HumanEval+		LCB v6	
(1) Math-RL: Qwen3-4B → Qwen3-4B						
Initial (Compact)	42.74 /	870	77.06 /	325	17.71 /	557
Teacher (GRPO-500)	52.65 /	2,619	83.92 /	700	20.07 /	2,670
DeepMath M48 (Compact)	53.22 /	5,590	83.23 /	2,328	27.21 /	9,567
(2) Qwen3-30B-A3B-Instruct-2507 → Qwen3-4B						
Initial	42.74 /	870	77.13 /	325	17.71 /	557
Teacher	54.86 /	669	82.39 /	609	33.14 /	3,784
DeepMath M48	52.15 /	5,350	82.32 /	1,532	26.79 /	7,599
DAPO M48	51.52 /	5,378	82.20 /	1,698	27.90 /	8,718
Textbook M48	51.01 /	6,000	81.90 /	1,585	25.30 /	7,814
Eurus Code M48	50.38 /	5,002	78.40 /	1,618	27.10 /	9,474
GSM8K M48	46.84 /	3,458	81.25 /	883	16.79 /	4,997
(3) Qwen3-30B-A3B-Instruct-2507 → Qwen3-1.7B						
Initial	29.10 /	725	59.83 /	198	12.36 /	422
Teacher	54.86 /	669	82.39 /	609	33.14 /	3,784
DeepMath M48	36.24 /	6,716	66.70 /	1,731	18.10 /	8,053
DAPO M48	33.71 /	7,218	64.10 /	2,077	16.90 /	9,015
(4) JustRL → DeepSeek-R1-Distill-Qwen-1.5B						
Initial	35.92 /	10,290	54.34 /	5,475	13.50 /	14,946
Teacher	40.47 /	7,217	64.56 /	5,252	19.36 /	11,783
DeepMath M8, s100	38.83 /	7,583	63.34 /	4,546	17.43 /	11,439
DAPO M8, s100	40.91 /	7,937	64.10 /	4,777	18.64 /	11,514
(5) Nemotron → DeepSeek-R1-Distill-Qwen-1.5B						
Initial	35.92 /	10,290	54.34 /	5,475	13.50 /	14,946
Teacher	40.66 /	5,185	66.69 /	4,760	22.00 /	6,113
DeepMath M48, s75	40.09 /	6,060	64.86 /	3,882	19.93 /	7,521
Textbook M48, s75	39.84 /	6,308	64.41 /	3,914	19.00 /	7,447
Eurus Code M48, s75	40.09 /	4,910	65.55 /	5,308	22.00 /	7,002
Mix3 M48, s75	42.17 /	5,933	66.16 /	4,752	21.79 /	7,008
(6) Mix-RL → Qwen3-4B-Instruct-2507						
Initial	45.52 /	458	81.48 /	976	27.86 /	2,944
Teacher	45.27 /	505	79.57 /	1,103	30.21 /	4,808
DeepMath M48, s25	47.35 /	555	81.86 /	1,207	29.71 /	3,371
Textbook M48, s25	51.14 /	952	83.23 /	1,332	28.93 /	4,051
Eurus Code M48, s25	61.81 /	3,432	81.17 /	1,452	31.93 /	7,940
IF M48, s25	46.46 /	473	68.29 /	133	22.07 /	1,769
Agent M48, s25	50.88 /	992	78.35 /	768	28.21 /	3,958
Mix3 M48, s25	59.34 /	2,804	82.01 /	1,535	32.36 /	7,770
Mix5 M48, s25	55.18 /	1,698	80.18 /	1,593	31.14 /	7,956
(7) Light-R1 → DeepSeek-R1-Distill-Qwen-7B						
Initial	49.05 /	8,414	81.78 /	3,782	23.36 /	11,998
Teacher	46.97 /	9,078	82.93 /	3,865	24.14 /	12,515
DAPO M8, s50	49.24 /	9,366	82.55 /	4,108	25.14 /	12,740
DeepMath M8, s50	49.49 /	9,283	82.62 /	3,814	24.21 /	12,180
Light3533, s50	50.63 /	9,630	81.78 /	4,249	24.00 /	12,617

Across three Mix-RL Code48 support draws, GPQA scores range from 61.24 to 61.81 and mean output length from 3,308 to 3,432 tokens. These are support-draw replications, not independent training-seed estimates.

Code48 and Mix3 use 3,428 and 2,800 pre-answer tokens and have repeated 8-gram rates of 2.13% and 1.83%, respectively. An early explicit answer candidate is identifiable in 244 and 261 responses, approximately 15–16% of the evaluated outputs. Within these subsets, wrong-to-correct revisions

number 5 and 8, compared with 10 and 11 correct-to-wrong revisions. The negative net counts refer only to these observable explicit changes; they do not measure implicit reasoning or establish a causal decomposition of the overall gain.

Similar output lengths can also accompany different accuracy: 30B-to-4B GSM8K48 and DeepMath48 produce 9,277 and 9,536 mean Math tokens but score 31.15 and 42.81, respectively. These observations concern free generation at evaluation, not the training-token consumption controlled in Section 4.1; they do not establish that longer test-time generation alone causes gains.

E.5 Failure trajectories and recovery protocols

Figure 11 highlights the geometry of Code recovery relative to native Code-M48. Continued distillation brings the recovered endpoint closer to this reference in function space even as its parameter-space distance increases. The distance to native Code-M48 decreases by approximately 71% in function space but increases by approximately 15% in parameter space.

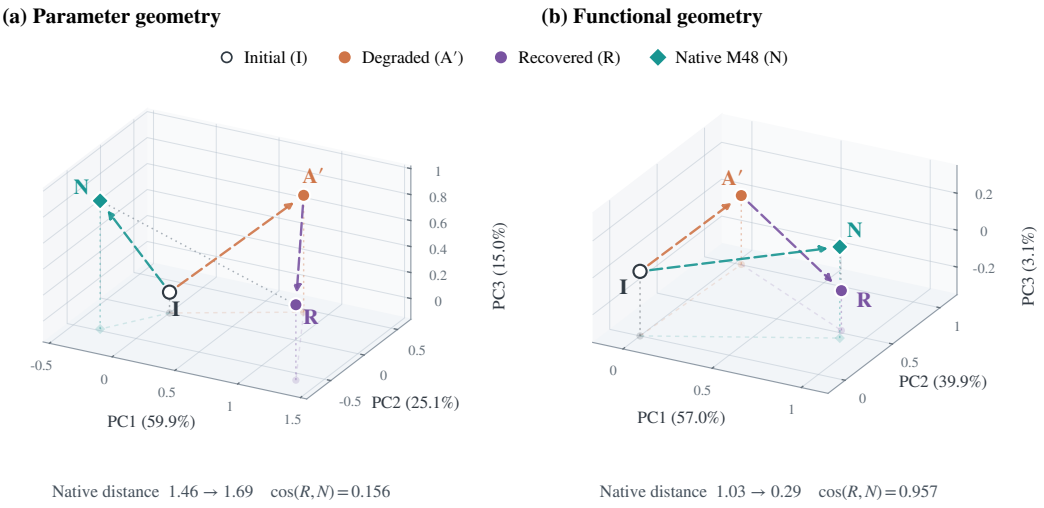


Figure 11: **Code recovery in parameter and function space.** PCA views of the initial student (I), degraded single-prompt endpoint (A'), recovered endpoint (R), and native Code-M48 endpoint (N). Native Code-M48 is trained directly from I; recovery continues from A' on an effective 48-prompt support. From A' to R, the reported distance to N increases from 1.46 to 1.69 in parameter space but decreases from 1.03 to 0.29 in function space. Axis percentages report explained variance; arrows connect endpoints rather than tracing intermediate updates.

Failure trajectories. Figure 12 traces how A' diverges during OPD and contrasts its functional alignment with the two-prompt support and prompt 187 alone. The increasing update magnitude is measured in function space, not parameter space.

For A' (prompt ID 15240), checkpoints 25, 50, 75, and 100 have Code Macro scores 55.20, 50.92, 49.66, and 50.03, compared with the initial student's 53.30. At the same checkpoints, functional cosine to the teacher is 0.5787, 0.4397, 0.3992, and 0.3713, and normalized functional-update magnitude is 1.041, 1.147, 1.218, and 1.268. Accuracy is not strictly monotonic: it rises slightly between steps 75 and 100 despite the overall degradation. For the joint $A' + 187$ support, the corresponding cosines are 0.7910, 0.7843, 0.7860, and 0.7934; for prompt 187 alone, they are 0.7967, 0.7934, 0.7966, and 0.8033. The joint support is trained from initialization, not used as a continuation of A'. The joint support achieves a Code Macro score of 59.63, compared with 60.41 for prompt 187 alone and 50.03 for prompt 15240 alone. Thus, a prompt that degrades performance in isolation need not prevent strong transfer when combined with an effective prompt.

Bidirectional representation intervention. The effective endpoint B is trained on prompt 5038; A' is the degraded endpoint above. The intervention replaces the output hidden state of zero-indexed

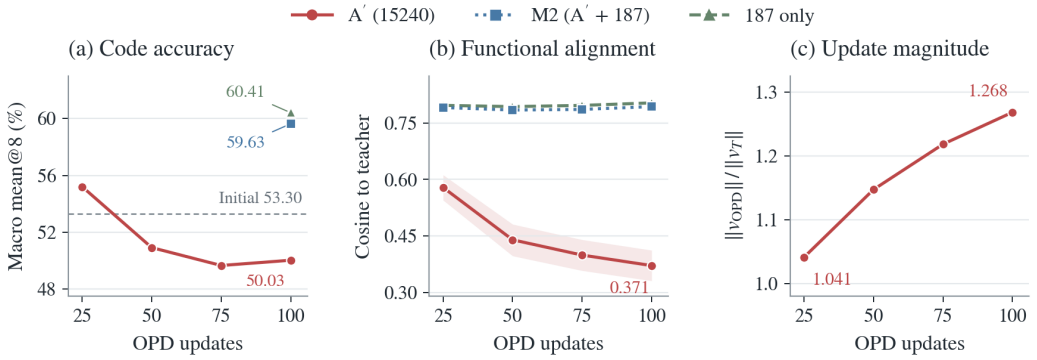


Figure 12: **Training trajectories under single-prompt and two-prompt supports.** All runs use the same Qwen3-4B initialization and Code GRPO-300 teacher. A' uses prompt 15240; M2 jointly uses prompts 15240 and 187 from initialization. **(a)** Code accuracy: the unweighted mean of mean@8 over HumanEval+, MBPP+, and LCB-v6. Markers for M2 and prompt 187 alone show only their step-100 endpoints. **(b)** Functional cosine to the teacher's change. **(c)** Normalized functional-update magnitude of A', $\|v_{\mathrm{OPD}}\|/\|v_T\|$.

block 30 (the 31st block), at the current prediction position on every autoregressive step. It copies the complete hidden vector, not weights or selected coordinates. Donor and recipient maintain separate, synchronously updated KV caches. The probe contains 20 questions with eight samples per question. $B \to A'$ changes the probe pass rate from 18.750% to 24.375%; $A' \to B$ changes it from 28.125% to 21.250%; $A' \to A'$ leaves it unchanged. These are probe rates, not the full Code Macro evaluations used for continuation. The intervention requires a donor at evaluation and is not a standalone student or an efficiency claim.

Figure 6(b) also reports replacement at zero-based block 10, with aggregate changes of -1.3 and -1.9 percentage points for $B \to A'$ and $A' \to B$, respectively. At block 30, $B \to A'$ improves HumanEval+ and MBPP+ but lowers LCB; the reverse intervention has the opposite benchmark-level signs. The tested interfaces show location- and task-dependent effects, not a uniquely responsible layer or uniform benefit across benchmarks.

Code continuation: protocol and endpoints. Code recovery starts from the step-100 A' weights and trains for a further 100 updates on an effective 48-prompt support with the same Code GRPO-300 teacher. It uses batch size 256, learning rate 10^{-5}, sampled-token K1 policy-gradient feedback, and a fresh Adam optimizer. Training and evaluation use seed 42; evaluation reports mean@8 on HumanEval+, MBPP+, and LCB-v6 with temperature 1 and top-p 1. Only the prespecified continuation step-100 endpoint is evaluated. The initial student, teacher, and native M48 reference have Code Macro scores 53.30, 61.80, and 61.79, respectively; the recovered endpoint scores 63.16. Its 200 total updates are not matched to the native reference's 100. This combined intervention does not show that starting from an unsuccessful support is preferable.

Math continuation: protocol, endpoints, and controls. For Qwen3-30B-A3B-Instruct-2507 $\to$ Qwen3-4B, we branch each stage-1 checkpoint, trained for 15 updates on GSM8K (G) or DeepMath (D), into two continuations: a further 15 updates on G or on D. Thus, $G \to G$ and $G \to D$ share the same starting weights, as do $D \to G$ and $D \to D$; all four paths have 30 total updates. All four stage-2 runs use a fresh Adam optimizer and retain the learning rate, batch size, and training seed of the original 30B-to-4B configuration in Appendix B.2. Optimizer resets and additional update counts are therefore common to the paired continuation arms; token budgets need not be equal.

For Math mean@16, $G \to D$ reaches 43.65 from 31.15; its matched-update performance reference $D \to D$ scores 43.75. LCB remains lower after the GSM8K history (25.36 versus 26.93), so the similar Math aggregate does not erase every task-specific difference. Figure 13 further compares formal Avg_4 endpoints: continuing on DeepMath rather than GSM8K yields 51.75 versus 43.77 from

G, and 51.98 versus 46.38 from D. Thus, the continuation support matters at matched update counts, while the two DeepMath continuations reach close scores.

The figure's left panel uses AIME 2024/2025 validation mean@2; at stage-2 step 0, each pair of branches reuses its shared stage-1 validation result, not a measurement before OPD. The middle panel's Avg_4 averages Math mean@16 and GPQA-D, HumanEval+, and LCB-v6 mean@8. The right panel's parameter references are specified below.

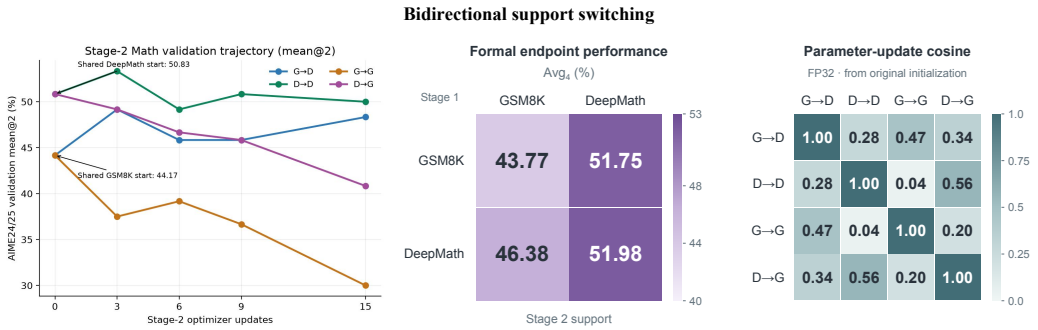


Figure 13: **Bidirectional support switching.** Qwen3-30B-A3B-Instruct-2507 → Qwen3-4B, with 15 updates in each stage; G denotes GSM8K and D denotes DeepMath. Left: AIME 2024/2025 validation mean@2 during continuation. Middle: formal endpoint Avg_4 (%). Right: cosine between endpoint parameter displacements from the original initialization.

Geometry references and parameter paths. Figure 6(d) uses native Code-M48 at step 100 as the Code reference and native DeepMath at step 15 as the Math reference, not the step-30 Math performance reference. Relative to these fixed references, functional cosine rises from 0.474 to 0.957 for Code and from 0.687 to 0.938 for Math. Parameter-displacement cosine changes from 0.123 to 0.156 for Code and from 0.086 to 0.374 for Math. These functional comparisons use a native OPD endpoint, whereas the failure-trajectory diagnostic compares with the teacher's change.

Endpoint parameter displacements in these cosine comparisons are measured from the original initialization θ_0, not the stage-2 starting checkpoint. The right panel of Figure 13 compares FP32 endpoint displacements and gives 0.28 for $G \to D$ versus $D \to D$ at step 30. The Math value 0.374 in Figure 6(d) instead uses DeepMath step 15 as the comparator; the two values are not interchangeable.

Let $\Delta_{\text{bad}} = \theta_{\text{bad}} - \theta_0$ and $U = \theta_{\text{recovered}} - \theta_{\text{bad}}$. Then $\cos(U, -\Delta_{\text{bad}})$ is 0.208 for Code and 0.178 for Math: continuation does not simply reverse the earlier parameter displacement. These comparisons do not establish exact removal of earlier functional changes. Likewise, high functional cosine indicates directional agreement, not equal amplitudes, conditional distributions, or predictions.

Robustness to answer scoring. We rescore all 9,600 outputs from five checkpoints across four mathematics benchmarks using math-verify 0.9.0 (Kydlíček, 2026) and conservative final-answer anchors. Under this alternative scoring pipeline, $G \to D$ improves over G by 12.24 percentage points (95% CI $[8.70, 16.04]$) and over $G \to G$ by 15.16 points (95% CI $[11.41, 19.17]$). Its difference from $D \to D$ is -0.16 points (95% CI $[-2.34, +2.08]$). A blinded audit of 152 stratified cases identifies 46 explicit final answers missed by the original parser, of which only six are correct. These counts describe the audited sample, not the prevalence of missed answers across all outputs. These results support recovery in evaluated performance that is robust to the tested scoring procedure; missed final-answer extraction is insufficient to explain the main gain. They do not establish recovery of an underlying reasoning mechanism.

F Prompt Selection and Mixed-Pool Sampling

This appendix details the within-pool selectors and paired mixed-pool comparison in Section 5.4, together with additional selection experiments on code and 30B-to-4B distillation. It also specifies

the initial-gradient representativeness diagnostic. Table 4 reports the selector results and selection costs; Table 5 reports the source-filtering comparison.

F.1 PROMPT-SELECTION METHODS

Table 4 compares eight selectors for distilling JustRL-DeepSeek-1.5B into DeepSeek-R1-Distill-Qwen-1.5B, using DeepMath candidates. Let $\mathcal{C}$ be a candidate pool of size C and $\mathcal{S}$ the selected support of size M. The two reported settings use $(M, C) = (8, 384)$ and $(48, 2{,}304)$. A nested ranking refers to prefixes within the same fixed candidate pool, not nesting between these two different candidate-pool settings. Selection is performed before OPD, using the initial student wherever student-dependent features are needed.

Uniform random. We sample M prompts uniformly without replacement from $\mathcal{C}$. This baseline tests whether a small unscored sample is sufficient. The random selectors in Table 4 report the mean and sample standard deviation across three independently sampled supports; these are not standard errors or training-seed variations. For uniform random at $M = 8$, $C = 384$, the support-selection seeds are 42, 43, and 44, with the training seed held fixed.

Stratified random. We jointly stratify candidates by data source, prompt character-length quartile, and initial-student difficulty. Difficulty is measured by the mean official reward of the first eight discovery rollouts:

$$\widehat{p}_i = \frac{1}{8} \sum_{r=1}^{8} R(y_{ir}), \tag{9}$$

where R is the official correctness reward. The four difficulty bins are zero ($\widehat{p}_i = 0$), low ($0 < \widehat{p}_i < 0.5$), medium ($0.5 \le \widehat{p}_i < 1$), and perfect ($\widehat{p}_i = 1$). All candidates are from DeepMath-103K, so the source dimension introduces no further subdivision; the joint partition has up to 4×4 length–difficulty strata.

Stratum quotas are proportional to their candidate-pool frequencies, not equal across difficulty bins. A cumulative quota-deficit rule selects one stratum at a time, so successive prefixes approximate the original joint distribution. Prompts within each stratum follow a seeded SHA-256 permutation; the reference ranking uses seed 42. This selector tests whether preserving length and difficulty composition helps avoid omissions in a small support.

Semantic diversity. We embed the full prompt messages, serialized as `<role>: <content>`, with Qwen/Qwen3-Embedding-0.6B (Zhang et al., 2025). Inputs are truncated to 2,048 tokens. The final non-padding token's hidden state is L2-normalized to obtain e_i. Selection uses deterministic farthest-first traversal, not k-means: the first prompt has the highest cosine similarity to the global embedding center, and each subsequent prompt maximizes

$$i^* = \arg\max_{i \in \mathcal{C} \setminus \mathcal{S}} \min_{j \in \mathcal{S}} \big(1 - \cos(e_i, e_j)\big). \tag{10}$$

The resulting order supplies nested prefixes within a fixed pool. There are no semantic-cluster quotas, and answers, rewards, and teacher feedback are not used. The method tests broad semantic coverage with reduced redundancy.

Hard selection. We rank prompts by increasing initial-student pass rate $\widehat{p}_i$ from Equation 9 and retain the first M. Ties follow a seed-42 SHA-256 order. The teacher does not contribute to this difficulty ranking; it supplies token-level feedback during subsequent distillation. This tests whether prompts that the initial student finds harder provide more useful training supervision.

Shortest. We rank candidates by the initial student's mean generated response length and retain the M shortest. This tests selection for cheaper generated trajectories rather than shorter prompt text.

Longest. We use the same initial-student response-length statistic but retain the M longest candidates. This tests whether longer trajectories offer more opportunities for distillation.

Teacher–student disagreement. The initial student generates $K = 16$ discovery responses per candidate, and the teacher scores the same sampled tokens. We rank prompts by a length-normalized, clipped Monte Carlo reverse-KL score. For sampled token y_{irt} at prefix h_{irt}, define

$$q_{irt} = \mathrm{clip}(\log p_0(y_{irt} \mid h_{irt}) - \log p_T(y_{irt} \mid h_{irt}), -10, 10)\,, \tag{11}$$

$$D_i = \frac{1}{K}\sum_{r=1}^{K}\left[\frac{1}{T_{ir}}\sum_{t=1}^{T_{ir}} q_{irt}\right]. \tag{12}$$

Here p_0 denotes the initial student, p_T the teacher, and T_{ir} the response length in tokens. We select the M largest D_i values. Tokens are averaged within each response before the 16 response means are averaged, so longer responses do not receive greater weight solely because of their length. This uses signed student-minus-teacher log-ratios, not absolute differences or a full-vocabulary KL computation.

Cost-D-opt. This selector favors complementary gradient directions while accounting for response-length cost. Its base construction is D-optimal selection; the cost-aware greedy score, rather than the gradient features, distinguishes Cost-D-opt from pure D-opt.

For each candidate i, the initial student generates 16 discovery responses. The frozen teacher scores the same sampled tokens. At prefix h_{irt} in response r, the detached token feedback is

$$a_{irt} = \mathrm{sg}[\mathrm{clip}(\log p_T(y_{irt} \mid h_{irt}) - \log p_0(y_{irt} \mid h_{irt}), -10, 10)]\,, \tag{13}$$

where p_0 is the initial student and sg denotes stop-gradient. The response-level token-feedback loss is

$$\ell_{ir}(\theta) = -\frac{1}{T_{ir}}\sum_{t=1}^{T_{ir}} a_{irt}\log p_\theta(y_{irt} \mid h_{irt}), \tag{14}$$

with gradients evaluated at the initial student. The implementation also applies rollout importance correction before gradient sketching; this factor is omitted from Equation 14 to display the token-feedback component. Three CountSketch projections (Charikar et al., 2002) avoid retaining full parameter gradients. Let $\widetilde{g}_{ir}$ denote the resulting correction-applied, sketched response gradient. Prompt-level aggregation is token-weighted:

$$\widetilde{g}_i = \frac{\sum_{r=1}^{16} T_{ir}\widetilde{g}_{ir}}{\sum_{r=1}^{16} T_{ir}}, \qquad L_i = \sum_{r=1}^{16} T_{ir}. \tag{15}$$

The 16 responses are split into two fixed eight-response subsets. Their cross-covariance provides a noise correction, retaining positive directions reproducible across the two subsets. The retained stable space has dimension $d \le 8$, and z_i denotes the stable coordinates of prompt i. D-opt uses $x_i = \sqrt{L_i}\, z_i$ and the objective

$$F(\mathcal{S}) = \log\det\left(\lambda I + \sum_{i\in\mathcal{S}} x_i x_i^\top\right), \qquad \lambda = 10^{-3}\frac{\mathrm{tr}(X^\top X)}{d}, \tag{16}$$

where row i of X is $x_i^\top$. Starting from $A = \lambda I$, the marginal gain of candidate i is

$$\Delta_i = \log\bigl(1 + x_i^\top A^{-1} x_i\bigr)\,. \tag{17}$$

Pure D-opt selects the largest Δ_i; Cost-D-opt selects the largest Δ_i/c_i, where $c_i > 0$ is the relative response-length cost. After selecting i, both update $A \leftarrow A + x_i x_i^\top$ and repeat until M prompts have been selected.

Pure D-opt uses neither correctness nor semantic embeddings, and it does not penalize response length. Indeed, L_i gives token-rich prompts greater information weight, potentially favoring long responses. Cost-D-opt additionally normalizes the greedy gain by length cost. Table 4 reports the cost-aware variant; pure D-opt is described here only to specify its underlying objective. Both share the gradient-statistics computation and its GPU cost.

F.2 MIXED-POOL AND SOURCE-FILTERED SAMPLING

For the three-source filtering comparison, the student is Qwen3-4B, trained with `enable_thinking=False`, and the teacher is Qwen3-30B-A3B-Instruct-2507. Each run in this comparison uses $M = 48$ prompts and 25 optimization steps. We vary the support-selection seed across 42/43/44 while holding the training seed fixed. Evaluation uses thinking off, temperature 1.0, top-$p = 1.0$, and a 16,384-token response limit, with 16 samples per Math question and eight per OOD question. Other training and evaluation settings follow the 30B-Instruct-to-4B protocols in Appendices B.2 and B.4.

Support construction. Candidates are formed by first drawing 3,840 prompts uniformly from each deduplicated source (DeepMath, DAPO and GSM8K; 11,520 in total, shared across selection seeds). Each selection seed then draws a single uniform per-prompt ordering over this pool; Mixed takes its first 48 prompts, and Filtered removes GSM8K from the same ordering before taking the first 48. Table 22 reports the source counts and overlap.

Table 22: **Mixed-pool support composition and overlap.** Shared counts prompt instances selected by both arms under the same random ordering, not shared sampled responses.

Selection seed	Mixed DeepMath / DAPO / GSM8K	Filtered DeepMath / DAPO / GSM8K	Shared prompts
42	17 / 15 / 16	21 / 27 / 0	32
43	13 / 15 / 20	22 / 26 / 0	28
44	20 / 13 / 15	31 / 17 / 0	33

Per-seed and per-benchmark results. Tables 23 and 24 report the complete results; OOD averages GPQA-Diamond, HumanEval+, and LiveCodeBench v6 equally. Paired summaries use unrounded scores, so displayed differences need not equal calculations from rounded entries.

Table 23: **Math results for mixed-pool and source-filtered sampling.** 30B Instruct → Qwen3-4B, $M = 48$, 25 optimization steps; selection seeds 42/43/44 share a fixed training seed. Scores are mean@16 (%); Math average is the unweighted mean of the four benchmarks. Δ denotes paired Filtered − Mixed differences; means, SDs, and differences use unrounded scores.

Benchmark	Selection seed	Mixed	Filtered	Δ
Math average	42 / 43 / 44	43.2 / 40.3 / 38.8	41.4 / 39.3 / 40.7	−1.8 / −0.9 / +1.9
	Mean ± SD	40.7 ± 2.3	40.5 ± 1.0	−0.3 ± 2.0
AIME24	42 / 43 / 44	55.8 / 54.2 / 49.2	56.5 / 51.3 / 49.8	+0.6 / −2.9 / +0.6
	Mean ± SD	53.1 ± 3.5	52.5 ± 3.5	−0.6 ± 2.0
AIME25	42 / 43 / 44	50.4 / 48.1 / 43.8	44.4 / 46.7 / 46.9	−6.0 / −1.5 / +3.1
	Mean ± SD	47.4 ± 3.4	46.0 ± 1.4	−1.5 ± 4.6
HMMT-Feb	42 / 43 / 44	28.8 / 24.6 / 29.0	27.7 / 26.5 / 30.0	−1.0 / +1.9 / +1.0
	Mean ± SD	27.4 ± 2.5	28.1 ± 1.8	+0.6 ± 1.5
HMMT-Nov	42 / 43 / 44	37.7 / 34.2 / 33.1	36.9 / 32.9 / 36.0	−0.8 / −1.3 / +2.9
	Mean ± SD	35.0 ± 2.4	35.3 ± 2.1	+0.3 ± 2.3

Source composition and a four-prompt control. Figure 14 shows that replacing four of 48 GSM8K prompts with DeepMath improves transfer. Using only those same four DeepMath prompts yields a similar mean Math score and higher mean OOD and Avg_4 scores than the mixture. Thus, the gains over GSM8K-only do not establish a benefit from combining the sources. Changing support size may also change prompt repetition. The sweep and control use the same 30B-to-4B pair and 25-update configuration as above, with the training seed, batch size, learning rate, and thinking-off training and evaluation settings unchanged.

Table 24: **OOD results for mixed-pool and source-filtered sampling.** 30B Instruct → Qwen3-4B, $M = 48$, 25 optimization steps; selection seeds 42/43/44 share a fixed training seed. Scores are mean@8 (%); OOD average is the unweighted mean of GPQA-D, HE+, and LCB. Δ denotes paired Filtered − Mixed differences; means, SDs, and differences use unrounded scores.

Benchmark	Selection seed	Mixed	Filtered	Δ
OOD average	42 / 43 / 44	55.2 / 54.9 / 53.9	55.2 / 53.7 / 54.6	0.0 / −1.2 / +0.7
	Mean ± SD	54.7 ± 0.7	54.5 ± 0.8	−0.2 ± 0.9
GPQA-D	42 / 43 / 44	51.5 / 50.7 / 50.0	51.0 / 50.4 / 51.2	−0.4 / −0.3 / +1.2
	Mean ± SD	50.7 ± 0.7	50.9 ± 0.4	+0.2 ± 0.9
HE+	42 / 43 / 44	87.0 / 87.0 / 86.1	88.1 / 85.3 / 86.1	+1.1 / −1.8 / +0.1
	Mean ± SD	86.7 ± 0.6	86.5 ± 1.4	−0.2 ± 1.4
LCB	42 / 43 / 44	27.1 / 26.9 / 25.7	26.4 / 25.4 / 26.4	−0.6 / −1.5 / +0.7
	Mean ± SD	26.5 ± 0.7	26.1 ± 0.6	−0.5 ± 1.1

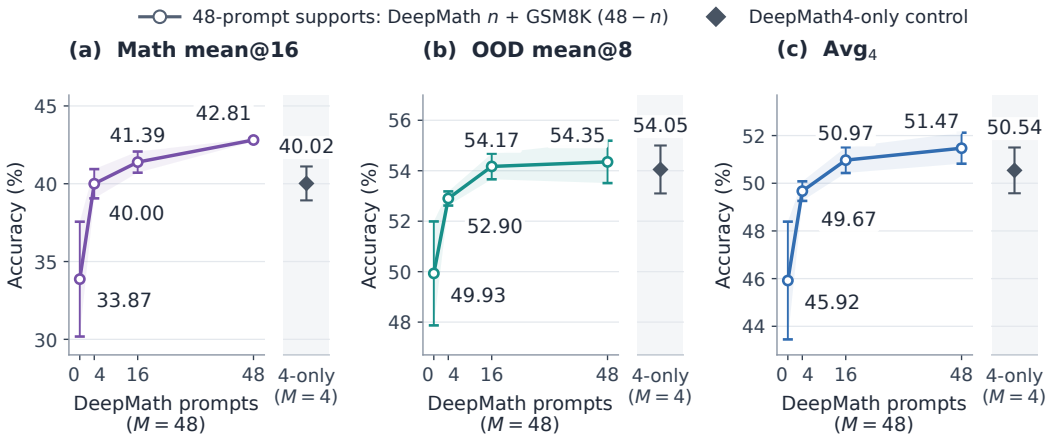


Figure 14: **Source composition and a four-prompt control.** DeepMath/GSM8K composition sweep (circles, $M = 48$) and DeepMath4-only control (diamonds, $M = 4$). Within each seed, the control reuses exactly the four DeepMath prompts in the corresponding 4/44 mixture. Error bars and bands show sample SD across selection seeds 42/43/44.

F.3 Prompt selection within Eurus Code

We additionally compare five selectors for the Qwen3-4B code-RL continuation pair, distilling the Code-RL teacher (GRPO-300) into Qwen3-4B. Each method selects $M = 48$ prompts from $C = 2{,}304$ Eurus Code candidates and trains for 100 optimization steps. Uniform random reports the mean and standard deviation across three independently sampled supports. The method names follow Appendix F.1.

Semantic diversity leads Code Macro at low screening cost; greater screening cost does not reliably produce a higher score in this comparison.

F.4 Prompt selection in cross-model distillation

We compare five selection methods for Qwen3-30B-A3B-Instruct-2507 → Qwen3-4B distillation. Each method selects $M = 48$ prompts from the same DeepMath L6 candidate pool of $C = 2{,}304$ prompts, followed by 25 OPD updates. Hard selection uses the lowest initial-student pass@8, and Shortest uses the lowest mean response length over eight initial-student samples. Random selectors use selection seeds 42/43/44; selection costs are approximate.

Table 25: **Prompt selection within Eurus Code.** Code-RL 4B → Qwen3-4B; $M = 48$, $C = 2{,}304$, 100 updates. Scores (%) use mean@8; Code Macro averages the three benchmarks. Uniform random reports mean ± sample SD across three support draws. Bold marks column-best scores. Costs cover selection only; 0 denotes no GPU screening.

Method	HE+	MBPP+	LCB	Code Macro	GPU·h
Uniform random	88.26 ± 0.32	71.33 ± 0.34	27.50 ± 0.43	62.36 ± 0.22	0
Semantic diversity	88.03	**71.83**	**28.14**	**62.67**	0.1
Hard selection	88.03	71.56	**28.14**	62.58	6.9
T–S disagreement	87.58	70.92	27.86	62.12	8.6
Cost-D-opt	**88.42**	71.17	27.64	62.41	48.6

Table 26: **Prompt selection within DeepMath L6.** 30B Instruct → Qwen3-4B; $M = 48$, $C = 2{,}304$, 25 updates. Scores (%) use mean@16 for Math and mean@8 otherwise; Avg_4 averages the four benchmarks. Random selectors report mean ± sample SD across three support draws. Bold marks column-best scores. Costs cover selection only; 0 denotes no GPU screening.

Method	Math	GPQA-D	HE+	LCB	Avg_4	GPU·h
Uniform random	40.56 ± 1.07	49.87 ± 0.55	86.16 ± 0.52	25.74 ± 0.52	50.58 ± 0.61	0
Stratified random	**41.79** ± 1.13	**50.65** ± 1.03	86.81 ± 1.47	26.81 ± 0.36	**51.52** ± 0.83	5.7
Semantic diversity	40.21	50.44	86.20	**27.36**	51.05	< 0.1
Hard selection	40.00	49.87	**87.65**	26.64	51.04	5.7
Shortest	35.05	46.53	83.61	20.21	46.35	5.7

Stratified random sampling achieves the highest reported four-task average, 51.52 ± 0.83, compared with 50.58 ± 0.61 for uniform random sampling; selecting the shortest responses yields 46.35 (Table 26).

F.5 Initial-gradient representativeness

We diagnose the JustRL–DeepMath selection study with $C = 384$ and $M = 8$, using the initial-student gradients from Appendix F.1. We use the discovery half and concatenate three independent, 32,768-dimensional CountSketch projections before SVD. No SVD truncation, PCA, or top-r projection is applied; these are not the stable coordinates used by Cost-D-opt.

Let $\widetilde{g}_{ir}$ concatenate the three token-mean gradient sketches for discovery rollout r of prompt i, and let T_{ir} count its valid response tokens. Both within-prompt and across-prompt aggregation are token-weighted:

$$L_i = \sum_{r \in \mathcal{R}_i^{\text{disc}}} T_{ir}, \qquad \widetilde{g}_i = \frac{\sum_{r \in \mathcal{R}_i^{\text{disc}}} T_{ir} \widetilde{g}_{ir}}{L_i}, \tag{18}$$

$$\bar{g}_{\mathcal{A}} = \frac{\sum_{i \in \mathcal{A}} L_i \widetilde{g}_i}{\sum_{i \in \mathcal{A}} L_i}, \qquad \mathcal{A} \in \{\mathcal{S}, \mathcal{C}\}. \tag{19}$$

The relative approximation error and directional alignment are

$$E(\mathcal{S}) = \frac{\|\bar{g}_{\mathcal{S}} - \bar{g}_{\mathcal{C}}\|_2}{\|\bar{g}_{\mathcal{C}}\|_2} = \sqrt{\frac{\sum_{q=1}^{3} \|\bar{g}_{\mathcal{S}}^{(q)} - \bar{g}_{\mathcal{C}}^{(q)}\|_2^2}{\sum_{q=1}^{3} \|\bar{g}_{\mathcal{C}}^{(q)}\|_2^2}}, \tag{20}$$

$$\cos(\bar{g}_{\mathcal{S}}, \bar{g}_{\mathcal{C}}) = \frac{\langle \bar{g}_{\mathcal{S}}, \bar{g}_{\mathcal{C}} \rangle}{\|\bar{g}_{\mathcal{S}}\|_2 \|\bar{g}_{\mathcal{C}}\|_2}. \tag{21}$$

The error is reported as $100E(\mathcal{S})$ in percent. The three projection contributions are summed before taking norms or computing cosine; their individual error ratios and cosines are not averaged. In Figure 7, each line connects the same support's initial-gradient approximation error and final four-task average; it does not trace a training trajectory.

We evaluate 2,000 random $M = 8$ supports in this same pre-SVD, token-weighted geometry. Their cosine mean is 0.97053344; the 5th, 50th, and 95th percentiles are 0.95150783, 0.97278932, and 0.98164110. These are diagnostic draws, distinct from the three trained uniform-random supports (selection seeds 42/43/44).

Table 27: **Per-support gradient representativeness and transfer.** JustRL–DeepMath, $C = 384$, $M = 8$. Each row is one support in Figure 7, not a mean across selection seeds. $E(\mathcal{S})$ and Avg_4 are percentages; cosine is dimensionless.

Method	Selection seed	$E(\mathcal{S})$ (%)	Cosine	Avg_4 (%)
Uniform random	42	29.7109	0.9670	37.5227
Uniform random	43	29.7557	0.9565	38.0760
Uniform random	44	23.4674	0.9722	37.3466
Stratified random	42	21.7985	0.9833	37.0561
Semantic diversity	42	23.5230	0.9721	37.4771
Hard selection	42	39.0314	0.9606	37.2021
Cost-D-opt	42	55.5506	0.9515	37.1266
Shortest	42	64.9447	0.8412	36.4925

Across the eight evaluated supports, Pearson and Spearman correlations between $E(\mathcal{S})$ and Avg_4 are -0.638 and -0.262, respectively, computed from the unrounded values with each support counted once. Excluding Shortest changes them to -0.255 and 0.107. These descriptive comparisons do not provide a consistent ranking of final accuracy: stratified seed 42 has the lowest error, whereas uniform seed 43 has the highest accuracy.

G Supplementary Training Efficiency

Low distinct-prompt requirements do not automatically reduce training cost. This appendix presents trajectory-reuse results and defines cost accounting. These address the resources consumed during training, separately from the main-text questions of how many distinct prompts are needed and how to select them.

G.1 Trajectory Reuse

We compare fresh-only OPD, which updates once per fresh rollout batch, with two reuse strategies. Full-batch reuse optimizes each batch for four passes. Uniform token replay performs one full-batch update followed by three updates, each using a uniformly sampled 25% subset of response tokens.

Table 28 shows setting-dependent quality–cost trade-offs. For DeepSeek 1.5B, reuse achieves 38.63–38.93 Avg_4 in 205–226 minutes, compared with 38.45 in 590.5 minutes for 67-batch fresh-only OPD. Reducing fresh-only training to 20 batches is faster still, but yields a lower score of 38.02. For 30B→4B, increasing updates from 15 to 60 at a fixed 15 fresh batches lowers Avg_4 from 51.84 to 50.37 with uniform replay and 48.46 with full-batch reuse. Thus, low distinct-prompt requirements do not by themselves imply that repeated optimization of existing trajectories preserves transfer quality.

G.2 Cost–quality accounting

Quality is considered alongside fresh generation, cumulative training tokens, and wall-clock time. Costs at the same quality target and quality at the same budget answer different questions and must be distinguished.

Generated tokens count newly sampled response tokens; training tokens count valid response tokens processed across all optimization passes. Rollout collection and optimizer updates are recorded separately. Teacher-scored tokens and prompt-prefill work are tracked separately where relevant; wall-clock measurements include teacher scoring. Evaluation samples and generation limits are excluded from the training budget. Prompt and response lengths are reported separately. Matched-

Table 28: **Performance and training cost with trajectory reuse.** F/U denotes fresh rollout batches / optimizer updates; generated and loss tokens are in millions (including repeated exposures for loss), and time is in minutes. Math reports mean@16 and the other benchmarks mean@8; Avg_4 is their unweighted mean.

	Training cost				Performance (%)				
Method	F/U	Generated	Loss	Time	Math	GPQA-D	HE+	LCB	Avg_4
DeepSeek 1.5B									
Fresh-only	67/67	128.19	128.19	590.5	32.55	39.02	63.64	18.57	38.45
Fresh-only	20/20	39.13	39.13	178.8	31.09	38.83	63.80	18.36	38.02
Uniform token replay	17/68	32.78	57.64	205.1	32.92	39.65	62.73	19.21	38.63
Full-batch reuse	17/68	32.60	130.40	226.0	32.14	40.15	64.94	18.50	38.93
30B Instruct → Qwen3-4B									
Fresh-only	15/15	24.59	24.59	148.7	41.41	50.82	88.34	26.79	51.84
Uniform token replay	15/60	22.82	40.00	142.4	39.64	49.81	85.52	26.50	50.37
Full-batch reuse	15/60	23.94	95.76	190.2	40.05	48.42	78.35	27.00	48.46

token and matched-time controls are identified explicitly, and historical runs retain their run-specific configurations.

H Future Work

Our findings motivate three directions for future work. First, prompt selection could adapt to the teacher, evolving student, and target capability, with gains evaluated against random sampling after accounting for screening cost. Second, controlled interventions on source composition, prompt exposure, and switching time could clarify the mechanisms and limits of recovery and tolerance to individually ineffective supports. Finally, our preliminary experiments in Appendix G.1 demonstrate the potential of trajectory replay to improve training efficiency. Future work could explore adaptive replay strategies that balance fresh trajectory generation and reuse while maintaining transfer performance.