LOOK BACK TO MOVE FORWARD: REUSING RL-INDUCED POLICY SHIFTS FOR POLICY IMPROVEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) improves language model reasoning, but obtaining these gains often requires costly trajectory sampling under sparse outcome feedback. This raises the question: can the policy shift learned during an RL run be reused to further improve the same model? We propose Distillation with Implicit Reward (DIR), a framework that turns RL-induced policy shifts into reusable supervision for further policy improvement. Under idealized KL-regularized policy optimization, the log-policy ratio before and after RL is equivalent to the inducing reward up to scale and an additive constant. Motivated by this correspondence, DIR extracts a token-level implicit reward from a fixed checkpoint pair and reuses it in a second regularized optimization anchored at the post-RL policy, yielding an extrapolated distillation target. DIR then distills this target on-policy, providing dense token-level supervision on states visited by the evolving student without a separately trained reward model or an external teacher. Experiments on mathematical reasoning and agentic search demonstrate two complementary benefits. DIR improves performance beyond RLVR. When initialized from intermediate RL checkpoints, it also approaches later-stage RL performance at substantially lower training cost. These results suggest that RL-induced policy shifts can serve as reusable supervision, supporting both continued performance improvement and more compute-efficient post-training.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has substantially advanced the reasoning capabilities of large language models, from solving mathematical problems to conducting multi-turn complex agentic tasks (Shao et al., 2024; DeepSeek-AI et al., 2025; Yu et al., 2026b; Gao et al., 2025). By optimizing against verifiable outcomes, RL incentivizes effective problem-solving strategies beyond those demonstrated in the training data. Yet achieving these gains requires substantial computation for on-policy sampling and outcome verification, while supervision remains limited to sparse outcome-level rewards. Complementary to RL's sparse outcome supervision, on-policy distillation (OPD) provides dense token-level supervision by aligning the student's conditional output distribution with the teacher's at states visited by the student's own policy. Its effectiveness, however, depends on the supervision available from the teacher. These observations motivate a question at the intersection of RL and OPD: Can the learning signal acquired during RL be reused for more effective and efficient policy improvement?

The difference in LLM's prediction token preferences during RL provides a record of how reward feedback has shaped its policy. Comparing the final policy with its pre-RL counterpart reveals how RL has shifted the model's preferences over actions. This distinction matters because the post-RL distribution combines newly acquired preferences with those inherited from earlier training. Recent work exploits such model differences for weak-to-strong learning: Direct-OPD transfers RL-induced policy shifts as implicit rewards (Feng et al., 2026), and W2S-OPD uses contrasts between weaker models to construct a proxy teacher for a stronger student (Yu et al., 2026a). The key question is whether this policy shift remains useful after the model has already incorporated the original RL update. We study its reuse as supervision for subsequent training, both to improve final performance and to reduce training cost.

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.

Under idealized KL-regularized policy optimization, the log-ratio between the updated and reference policies recovers the reward up to a positive scaling factor and an input-dependent constant (Rafailov et al., 2023). For LLMs, this sequence-level ratio decomposes into token-level log-probability differences, providing a dense signal from the pre- and post-RL checkpoints. This connection suggests that the supervision acquired through RL can be reused beyond the update that produced it. Optimizing against this recovered signal from the post-RL policy yields an extrapolated target for further training, offering a way to improve performance while reducing reliance on additional RL optimization.

We propose **Distillation with Implicit Reward (`DIR`)**, a framework that reuses the policy change encoded by existing RL checkpoints to guide subsequent learning. `DIR` extracts an implicit reward from the pre-/post-RL contrast and uses it to construct a distillation target that further emphasizes the preferences learned during RL. By converting the policy shift learned during RL into dense supervision, `DIR` enables further improvement from reward information already acquired by the model. This reuse supports both higher performance and more compute-efficient training without relying on a stronger teacher.

`DIR` supports two complementary uses. After a completed RL run, it provides a further training stage aimed at improving beyond the existing checkpoint. When applied to an intermediate checkpoint, it offers a way to reuse the progress already made and reduce the subsequent RL computation needed to reach a given level of performance. We evaluate these settings on mathematical reasoning and agentic search. `DIR` improves the average score across seven mathematical reasoning benchmarks from 40.56% to 44.17% and raises BrowseComp-Plus accuracy from 36.14% to 38.80%. When initialized from an intermediate RL checkpoint, `DIR` achieves accuracy comparable to later-stage RL on BrowseComp-Plus (33.01% vs. 33.37%) using only 45.9% of the total GPU hours, including the initial RL training. Behavioral analysis shows that `DIR` encourages more extensive search, with increases in both search calls and response length during training.

Our contributions are summarized as follows:

- We investigate whether the policy shift learned during RL can serve as reusable supervision for further improvement of the same model (§4.2), connecting this setting to an implicit-reward view of policy optimization.
- We introduce `DIR`, which combines policy extrapolation with on-policy distillation to learn from the relative preferences encoded by an existing checkpoint pair (§3).
- We demonstrate gains beyond the evaluated RL checkpoints on mathematical reasoning and agentic search (§4.2), and show that reuse from intermediate checkpoints can approach later-stage RL performance at substantially lower computational cost (§4.3).

2 Related Work

2.1 RL with Verifiable Rewards

Reinforcement learning with verifiable rewards (RLVR) has become an effective approach to improving LLM performance through automatically checked outcomes. DeepSeekMath and DeepSeek-R1 demonstrate its potential for mathematical and extended reasoning (Shao et al., 2024; DeepSeek-AI et al., 2025), while DAPO improves large-scale RL training (Yu et al., 2026b) and ASearcher extends outcome-based RL to long-horizon search agents (Gao et al., 2025). P1 extends RLVR to Olympiad-level physics reasoning through multi-stage training with rule-based outcome verification (Chen et al., 2025a). These methods primarily improve how policies learn from newly sampled trajectories and their rewards. Our work addresses a complementary question: how can the policy change already learned by RL be reused for further improvement? `DIR` extracts a dense token-level signal from the pre-/post-RL log-policy ratio and uses it to construct an extrapolated distillation target, extending the use of verified supervision beyond the original RL training stage.

2.2 On-Policy and Self-Distillation

On-policy distillation aligns student and teacher distributions at states visited by the student (Agarwal et al., 2023). MiniLLM studies reverse-KL distillation (Gu et al., 2024), while Self-Distilled

Reasoner obtains teacher supervision by conditioning the student's initial model on reference solutions (Zhao et al., 2026). SimpleOPD enables cross-tokenizer distillation for long-context reasoning by aligning teacher and student tokens that span identical text segments (He et al., 2026). Concurrent work extends distillation to weak-to-strong transfer: Direct-OPD derives implicit rewards from a smaller model's RL-induced policy shift (Feng et al., 2026), and W2S-OPD constructs proxy teachers from contrasts between smaller models (Yu et al., 2026a). DIR reuses the policy shift from a model's own RL training to construct an extrapolated teacher for that model. We study whether this reuse can improve final performance and reduce training cost when applied to intermediate checkpoints.

3 METHOD

Distillation with Implicit Reward (DIR) reuses the policy shift learned during an RL stage to train the same model further. As illustrated in Figure 1(a), DIR extrapolates the pre-to-post RL logit change to construct a virtual teacher from two frozen checkpoints. Figure 1(b) contrasts on-policy distillation toward this teacher with continued RL from the same post-RL checkpoint. The student learns on its own trajectories, without outcome rewards or correctness filtering during the DIR stage. Figure 1(c) details the DIR training loop. A trainable student π_θ, initialized from π_{post}, samples on-policy trajectories and visits contexts s_t. The frozen pre- and post-RL checkpoints evaluate the same contexts, and their logits are combined to construct the extrapolated teacher. The student is then optimized by on-policy distillation toward π_T and updated to generate the next batch of trajectories.

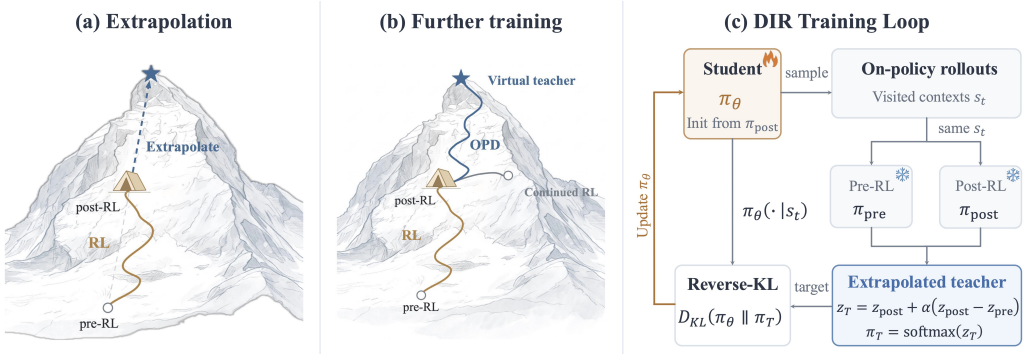

Figure 1: **Illustration of DIR. (a)** DIR extrapolates the RL-induced logit change between two frozen checkpoints to construct a virtual teacher (star). **(b)** Initialized from the post-RL checkpoint (tent), the student learns from this teacher through on-policy distillation (blue), with continued RL from the same checkpoint shown for comparison (gray). **(c) DIR training loop.** The trainable student π_θ generates on-policy trajectories and visits contexts s_t. The frozen pre- and post-RL checkpoints evaluate the same contexts to construct the extrapolated teacher π_T through logit extrapolation. Reverse-KL distillation updates only the student, which then generates new trajectories. The landscape and training paths in (a) and (b) are schematic.

3.1 RL-INDUCED POLICY SHIFTS AS IMPLICIT REWARDS

Let π_{pre} and π_{post} denote the policies before and after an RL stage. For a generation context s and a candidate token a, define

$$r_\Delta(s, a) = \log \pi_{\text{post}}(a \mid s) - \log \pi_{\text{pre}}(a \mid s). \tag{1}$$

This signal records how RL changed the model's token preferences. For an input x and response y, let $s_t = (x, y_{<t})$. Under autoregressive factorization, the sequence-level log-policy ratio decomposes as follows:

$$\Delta_{\text{RL}}(x, y) := \log \frac{\pi_{\text{post}}(y \mid x)}{\pi_{\text{pre}}(y \mid x)} = \sum_{t=1}^{|y|} r_\Delta(s_t, y_t). \tag{2}$$

The sequence-level shift has an implicit-reward interpretation. If π_{post} exactly maximizes expected reward $R(x, y)$ with a KL penalty of strength $\beta > 0$ relative to π_{pre}, then $\Delta_{\text{RL}}(x, y) = R(x, y)/\beta - \log Z(x)$, where $Z(x)$ is independent of y (Appendix B.1). Thus, under this idealized update, the shift preserves reward differences up to a positive scale. In practice, RL only approximates this relationship: we use r_Δ as the relative preference encoded by the realized update. Since these preferences are encoded in the pre-/post-RL log-policy ratio, we can obtain dense supervision directly from the frozen checkpoints without training a separate reward or value model.

3.2 Constructing an Extrapolated Teacher

DIR reuses this preference signal by constructing a teacher that extrapolates the RL-induced logit change:

$$\pi_T(\cdot \mid s) = \text{softmax}\big(z_{\text{post}}(s) + \alpha\big[z_{\text{post}}(s) - z_{\text{pre}}(s)\big]\big)\,, \tag{3}$$

where z_{pre} and z_{post} are the checkpoints' logits on the same context, and $\alpha \geq 0$ controls extrapolation strength. The teacher is a conditional target distribution constructed from these outputs; both checkpoints remain frozen.

For $\alpha > 0$, this construction is the unique solution to a local regularized reward-reuse problem:

$$\pi_T(\cdot \mid s) = \arg\max_q \left\{ \mathbb{E}_{a\sim q}[r_\Delta(s, a)] - \frac{1}{\alpha} D_{\text{KL}}(q \,\|\, \pi_{\text{post}}(\cdot \mid s)) \right\}. \tag{4}$$

Its closed-form solution is

$$\pi_T(a \mid s) = \frac{\pi_{\text{post}}(a \mid s) \exp\big(\alpha r_\Delta(s, a)\big)}{Z_\alpha(s)}, \tag{5}$$

which is algebraically equivalent to Eq. (3). The post-RL policy anchors the update, while the implicit reward emphasizes the relative preferences acquired during RL. At $\alpha = 0$, the teacher reduces to π_{post}. Appendix B.2 proves these identities and shows that reverse-KL matching to π_T optimizes this same local reward-regularization trade-off. This is a per-context construction; Appendix B.3 distinguishes it from sequence-level optimization.

3.3 On-Policy Distillation

We initialize π_θ with π_{post}. At iteration k, the current student π_{θ_k} generates trajectories, and both frozen checkpoints provide next-token distributions on the student-generated prefixes. Let d_k denote the distribution over valid generation contexts collected in this way. The underlying reverse-KL distillation objective is

$$\mathcal{L}_k(\theta) = \mathbb{E}_{s\sim d_k}\left[D_{\text{KL}}(\pi_\theta(\cdot \mid s) \,\|\, \pi_T(\cdot \mid s))\right]. \tag{6}$$

We update the student using the sampled prefixes and fixed teacher distributions, then collect new trajectories with the updated policy. Repeating this process keeps distillation aligned with the prefixes the student generates during training. For agentic search, these prefixes include tool observations, but the loss is computed only on assistant-generated tokens. Details of candidate-token selection, gradient computation, and loss masking are provided in Appendix C.

DIR can start from either a final or an intermediate RL checkpoint. A final checkpoint provides the starting point for further performance improvement, whereas an intermediate checkpoint allows us to reuse earlier RL progress to approach later-stage performance at lower total training cost. Both settings use the same teacher construction and on-policy distillation procedure.

4 Experiments

We evaluate DIR on mathematical reasoning and agentic search, organized around four questions. **RQ1: Performance Ceiling.** Can the policy shift induced by RL be reused to improve an already converged policy? **RQ2: Compute Efficiency.** Can a shift learned early in RL reduce the cost of reaching later-stage performance? **RQ3: Extrapolation Strength.** How sensitive is DIR to the extrapolation coefficient α? **RQ4: Behavioral Analysis.** Which behaviors are amplified by extrapolating the RL-induced policy shift?

4.1 EXPERIMENTAL SETUP

We evaluate DIR on both mathematical reasoning and multi-turn agentic search tasks. For **mathematical reasoning**, we initialize Qwen3-4B Base via GRPO on DAPO-Math-17k (Yu et al., 2026b). We then apply DIR on the same training set and evaluate on 7 math reasoning benchmarks including AIME 2024 (Li et al., 2024), AIME 2025 (Balunović et al., 2025), AMC (Li et al., 2024), HMMT 2025 (Dekoninck et al., 2026), MATH-500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), and Olympiad-Bench (He et al., 2024). We report **avg@**n, defined as the mean correctness over n sampled responses per problem, averaged across problems within each benchmark. We use n = 16 for AIME 2024, AIME 2025, AMC23, and HMMT 2025, and n = 4 for MATH-500, Minerva, and Olympiad-Bench. The aggregate score is the unweighted mean of the seven benchmark accuracies, which we refer to as macro-average accuracy. For all mathematical-reasoning experiments, we train DIR on DAPO-Math-17k with a batch size of 64 and a learning rate of 10^{-6}. Detailed training and evaluation settings are provided in Appendix D.

For **agentic search**, we use Qwen3-8B in non-thinking mode. We first train it with GRPO on the 13,285-example ASearcher training set (Gao et al., 2025), then apply DIR to the same data. Following the evaluation setup of Search-R1 (Jin et al., 2025), we evaluate multi-hop search on HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). Longer-horizon search is evaluated on BrowseComp-Plus (Chen et al., 2025b), BrowseComp-ZH (Zhou et al., 2025), and FRAMES (Krishna et al., 2025). We report answer accuracy on all seven benchmarks and retrieval recall on the three longer-horizon benchmarks. Unless noted otherwise, DIR is trained for two epochs with $\alpha = 1$ and a learning rate of 10^{-6}. Detailed training and evaluation settings are provided in Appendix F.

Within each benchmark, all compared methods use the same retrieval backend, decoding parameters, context-management procedure, and answer judge. For BrowseComp-Plus, we evaluate on the full set of 830 questions. For FRAMES, we use a fixed subset of 391 questions compatible with the 2018 Wikipedia snapshot. Both benchmarks use top-10 retrieval, a 40,960-token context window, up to 128 agent turns, and sampling with temperature 0.7 and top-p 0.8, with top-k filtering disabled. We evaluate BrowseComp-ZH on all 289 questions with a search budget of 32. The four multi-hop QA benchmarks comprise 3,125 questions in total and use top-3 retrieval, a 36,864-token context window, up to 100 agent turns, at most four queries per tool call, and greedy decoding with a single sample ($n = 1$). During evaluation, we follow Liu et al. (2026) and use model-generated summaries to manage long search histories. We use Qwen3-32B as an LLM judge to assess whether the extracted final answer is semantically equivalent to the reference answer, given the question. The judge receives no search history or retrieved documents.

4.2 PERFORMANCE CEILING (RQ1)

We first evaluate whether DIR can improve on the final GRPO checkpoint. In both mathematical reasoning and agentic search, we initialize the student from this checkpoint and construct the distillation target from the corresponding pre- and post-RL policies. We compare with the final GRPO model and with Continued RL, which performs further GRPO training from the same checkpoint using the same additional compute budget as DIR.

On mathematical reasoning, DIR raises macro-average accuracy across the seven benchmarks from 40.56% to 44.17% after only 20 additional updates, a gain of 3.61 percentage points, and improves over GRPO on every benchmark (Table 1). The largest gains are on AIME 2025, AMC, and Minerva, at 7.30, 5.78, and 4.69 percentage points, respectively. By comparison, Continued RL reaches 40.47%, with little change in the overall score. DIR also raises the macro-average accuracy of the Qwen3-4B-Base model above that of Qwen3-32B (44.17% vs. 42.50%). This comparison highlights the potential of reusing RL-induced policy shifts to improve the performance of smaller models.

Across HotpotQA, 2Wiki, MuSiQue, and Bamboogle, DIR improves average accuracy from 56.80% to 57.80%. The largest gain is on Bamboogle, where accuracy rises from 64.8% to 69.6%. Accuracy also increases from 36.14% to 38.80% on BCP+, from 19.03% to 22.84% on BCP-ZH, and from 47.57% to 48.85% on FRAMES, with retrieval recall gains of 6.99, 6.22, and 5.05 percentage points, respectively.

Table 1: **Main results on mathematical reasoning and agentic search (%).** Continued RL resumes GRPO from the same checkpoint with the same additional training compute as DIR. We report avg@n for mathematical reasoning; answer accuracy for HotpotQA, 2Wiki, MuSiQue, and Bamboogle; and accuracy and recall for BCP+, BCP-ZH, and FRAMES. Best results within each model family are in bold.

Method	AIME24	AIME25	AMC	HMMT25	MATH500	Minerva	Olympiad	Avg.
Qwen3-32B	**29.37**	22.29	70.47	9.79	84.45	30.51	50.59	42.50
Qwen3-4B-Base	8.33	5.62	29.06	1.67	37.05	8.82	21.96	16.07
↪ GRPO	23.54	19.58	65.16	11.25	83.55	31.71	49.11	40.56
↪ Continued RL	24.38	20.62	65.78	10.83	83.00	29.50	49.15	40.47
↪ DIR	26.04	**26.88**	**70.94**	**11.46**	**84.70**	**36.40**	**52.78**	**44.17**

Method	HotpotQA	2Wiki	MuSiQue	Bamboogle	BCP+		BCP-ZH		FRAMES	
					Acc.	Recall	Acc.	Recall	Acc.	Recall
Qwen3-32B	49.4	28.5	19.2	47.2	12.77	23.43	11.07	34.95	26.09	64.70
Qwen3-8B	52.8	34.2	21.8	50.4	9.16	21.29	5.54	29.76	30.43	69.96
↪ GRPO	70.9	**56.8**	**34.7**	64.8	36.14	53.78	19.03	61.25	47.57	85.41
↪ Continued RL	67.9	53.5	34.5	64.0	**39.52**	58.48	19.72	63.54	46.04	86.96
↪ DIR	**71.0**	56.6	34.0	**69.6**	38.80	**60.77**	**22.84**	**67.47**	**48.85**	**90.46**

Relative to Continued RL, DIR achieves higher accuracy on HotpotQA, 2Wiki, and Bamboogle. It also improves both accuracy and recall on BCP-ZH and FRAMES. On BCP+, Continued RL has higher accuracy (39.52% versus 38.80%), but lower recall (58.48% versus 60.77%). Overall, DIR achieves stronger performance than simply extending RL training under the same compute budget. These results suggest that reusing the learned policy shift can raise the performance ceiling beyond additional RL training alone.

4.3 Compute Efficiency (RQ2)

We then apply DIR to intermediate RL checkpoints to assess its training efficiency. For mathematical reasoning, we start from RL step 100 and train for 10 or 20 additional steps. For agentic search, we start from RL step 96 and train for 48 or 96 steps. We compare the cumulative GPU hours required by DIR and GRPO to achieve comparable accuracy. For DIR, this includes both the initial RL training and the subsequent distillation stage.

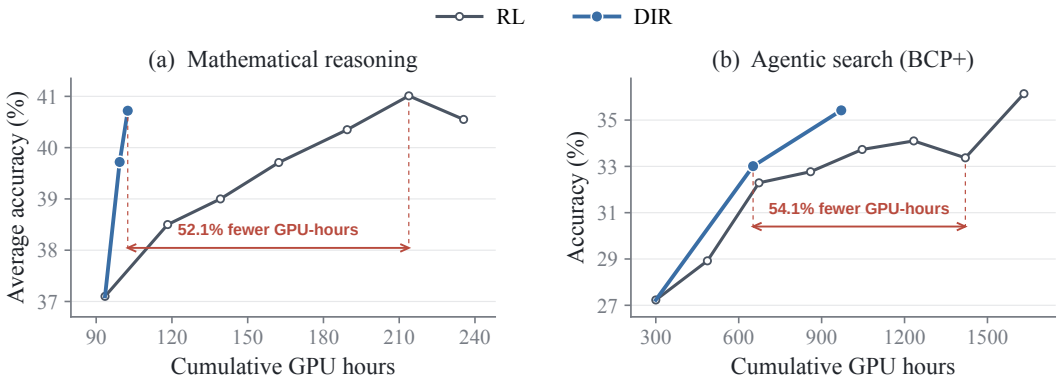

Figure 2: **Accuracy versus cumulative training cost.** Mathematical reasoning is shown on the left and BCP+ on the right. Gray curves trace checkpoints from the original RL runs. Each blue curve starts at its RL initialization checkpoint and connects the subsequent DIR checkpoints. Cumulative cost includes both the RL prefix used for initialization and the subsequent DIR stage.

For mathematical reasoning, DIR reaches an average score of 39.72% across seven benchmarks after 10 additional training steps (Figure 2). This matches GRPO at step 190 (39.71%) with 61.3% of the cumulative GPU hours. After 20 additional steps, DIR achieves 40.72%, comparable to GRPO at step 250, while using 47.9% of its cumulative GPU hours. Appendix E provides per-benchmark results for the GRPO checkpoints and the 20-update DIR run with $\alpha = 2$ shown in Figure 2.

On BCP+, 48 steps of DIR reach 33.01% accuracy, compared with 33.37% at RL step 288, using 45.9% of the cumulative GPU hours. With 96 steps, DIR reaches 35.42%, trailing RL at step 320 by 0.72 percentage points while using 59.5% of its compute. In both domains, applying DIR to an intermediate checkpoint approaches the performance of substantially longer RL training runs at lower total cost.

4.4 Extrapolation Strength (RQ3)

We assess whether the gains from DIR depend on precise tuning of the extrapolation strength. The coefficient α controls the amplification of the RL-induced policy shift, and Table 2 reports results for $\alpha \in \{0.5, 0.75, 1, 2\}$ in both domains.

Table 2: **Effect of the extrapolation strength α (%).** We report avg@n for mathematical reasoning; answer accuracy for HotpotQA, 2Wiki, MuSiQue, and Bamboogle; and accuracy and recall for BCP+, BCP-ZH, and FRAMES. Best results are in bold.

Method	AIME24	AIME25	AMC	HMMT25	MATH500	Minerva	Olympiad	Avg.
Qwen3-4B-Base	8.33	5.62	29.06	1.67	37.05	8.82	21.96	16.07
↪ GRPO	23.54	19.58	65.16	11.25	83.55	31.71	49.11	40.56
↪ DIR, $\alpha = .5$	**27.08**	24.38	70.16	11.04	85.30	**37.04**	51.82	43.83
↪ DIR, $\alpha = .75$	26.04	**26.88**	70.94	11.46	84.70	36.40	**52.78**	**44.17**
↪ DIR, $\alpha = 1$	26.67	26.04	71.25	**11.67**	**85.35**	34.47	51.97	43.92
↪ DIR, $\alpha = 2$	26.25	25.83	**72.97**	11.25	84.75	35.94	51.89	44.13

Method	HotpotQA	2Wiki	MuSiQue	Bamboogle	BCP+		BCP-ZH		FRAMES	
					Acc.	Recall	Acc.	Recall	Acc.	Recall
Qwen3-8B	52.8	34.2	21.8	50.4	9.16	21.29	5.54	29.76	30.43	69.96
↪ GRPO	70.9	56.8	34.7	64.8	36.14	53.78	19.03	61.25	47.57	85.41
↪ DIR, $\alpha = .5$	70.9	57.3	35.8	67.2	37.59	58.33	21.45	63.54	**48.85**	87.76
↪ DIR, $\alpha = .75$	**72.3**	**58.0**	**36.8**	65.6	38.31	60.94	22.15	66.73	48.08	87.67
↪ DIR, $\alpha = 1$	71.0	56.6	34.0	**69.6**	**38.80**	60.77	**22.84**	**67.47**	**48.85**	**90.46**
↪ DIR, $\alpha = 2$[1]	50.9	38.2	23.5	48.8	34.94	**61.89**	18.34	62.64	43.48	90.03

For math reasoning, average scores remain within 43.83%–44.17% across the tested range, compared with 40.56% for GRPO. Additional ablations initialized from step 100 are reported in Appendix E.2. For agentic search, $\alpha \in \{0.5, 0.75, 1\}$ yields closely matched average accuracy across HotpotQA, 2Wiki, MuSiQue, and Bamboogle (57.80%–58.18%, versus 56.80% for GRPO). These settings also consistently improve both accuracy and recall over GRPO on BCP+ and FRAMES.

4.5 Behavioral Analysis (RQ4)

To characterize what capability DIR actually incentivized during training, we analyze how DIR changes the reasoning and agentic search behavior. For mathematical reasoning, we analyze the allocation of generated tokens across functional reasoning activities. We collect one response per problem from π_{pre}, π_{post}, and DIR on 100 problems spanning AIME24, AIME25, and AMC23. Following the episode framework of Schoenfeld (Schoenfeld, 1985) and its adaptation in ThinkARM (Li et al., 2026), we automatically annotate response segments and map their labels to tokens using a

[1]For agentic search, training with $\alpha = 2$ became unstable and eventually collapsed under excessive extrapolation. We therefore report the first-epoch results for this setting, while all other settings are evaluated after two epochs.

Table 3: **Token allocation across eight problem-solving episodes.** Panel A reports the distributions in model-generated responses; Panel B reports the distribution of the top-1% implicit-reward tokens on π_{post}-generated traces. Percentages are normalized over classified tokens. Relative to the full π_{post} distribution, high-reward tokens are enriched in Plan, Explore, and Monitor. Darker shading indicates a larger percentage.

Model / token set	Read	Analyze	Plan	Implement	Explore	Verify	Monitor	Answer
A. Full generated responses								
Qwen3-4B-Base (π_{pre})	3.60	22.10	3.12	42.62	3.37	6.14	0.05	19.01
Qwen3-4B-GRPO (π_{post})	6.28	14.27	4.11	41.96	4.23	21.69	0.51	6.94
DIR ($\alpha = 2$, step 90)	7.24	11.40	3.73	41.50	4.74	17.32	0.61	13.46
B. Top-scoring tokens on the same π_{post} responses								
Implicit reward top 1%	9.30	13.17	9.52	23.53	14.86	21.42	3.72	4.48

ThinkARM-guided annotation pipeline. Table 3 reports pooled token allocations conditional on assignment to one of the eight episode categories. To examine the implicit reward underlying DIR, we additionally report the episode distribution of the globally top-ranked 1% of realized tokens on π_{post}-generated traces, ranked by implicit reward $r_{\Delta,t} = \log \pi_{\text{post}}(y_t \mid s_t) - \log \pi_{\text{pre}}(y_t \mid s_t)$ using identical full prefixes.

Table 3 shows that, among the classified tokens, π_{post} allocates more tokens to Verify than π_{pre} (21.69% vs. 6.14%). Relative to π_{post}, DIR retains a similar Implement share (41.50% vs. 41.96%), with smaller Analyze and Verify shares and a larger Answer share. Meanwhile, the top-1% implicit-reward tokens are enriched in Plan, Explore, and Monitor by approximately $2.3\times$, $3.5\times$, and $7.3\times$, respectively, relative to π_{post}'s classified-token distribution. Implement is underrepresented, whereas Verify remains near its baseline share. Thus, the implicit reward selectively emphasizes planning, exploration, and process-monitoring expressions on fixed traces generated by π_{post}.

For agentic search, Figure 3 tracks search calls and response length during DIR training, averaged over eight-step intervals. Both increase overall, showing that the agent searches more and generates longer responses as training proceeds. This pattern is consistent with the higher retrieval recall reported in Table 1. It does not establish that additional search alone explains the accuracy gains. We discuss evidence utilization and stopping behavior in Appendix H.

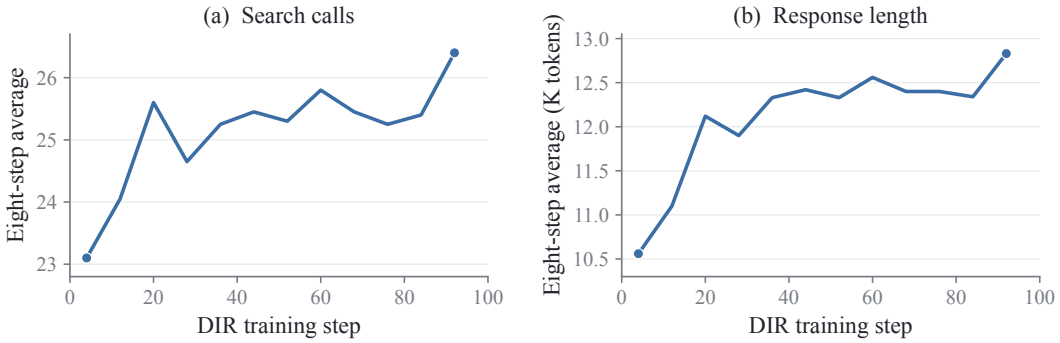

Figure 3: **Agentic search behavior during DIR training.** The number of search calls (left) and generated response length (right), averaged over eight-step training intervals. Both metrics exhibit an overall upward trend, consistent with more sustained evidence gathering.

Overall, the implicit reward emphasizes planning, exploration, and monitoring in mathematical reasoning, while DIR strengthens sustained evidence gathering in agentic search. These findings suggest that DIR builds on the problem-solving preferences acquired during RL to support further policy improvement.

5 DISCUSSION

5.1 INTERPRETATION OF THE IMPLICIT REWARD

The log-policy ratio in Eq. (1) should be interpreted as a record of how RL changed the model's preferences, rather than as a calibrated predictor of task success. In an idealized KL-regularized update, this ratio recovers the inducing reward up to state-dependent normalization. A practical RL run, however, also reflects finite sampling, optimization dynamics, and the structure of the training distribution. Consequently, a token assigned a large implicit reward is one that RL made more likely, but not necessarily one whose local contribution to the final outcome can be identified in isolation.

This distinction helps explain the agentic search results. RL training strengthens the multi-turn search pattern, and DIR further amplifies this preference, as reflected by the increase in search calls in Figure 3. Such a shift is useful on BCP+, whose questions typically require longer evidence-gathering trajectories than those encountered during training. In our setup, training trajectories commonly use on the order of ten search turns, whereas the harder evaluation questions can require more than twenty. Sustained search is therefore often necessary for success, even though searching for more turns is not sufficient by itself. DIR need not recover a token-level correctness signal to be useful: it can extrapolate a behavioral direction learned by RL when that direction remains beneficial on the evaluation distribution.

5.2 LIMITATIONS AND OUTLOOK

The effectiveness of implicit-reward extrapolation can depend on how well the RL-induced behavior transfers to the target task. If a policy change captures a training-specific shortcut, or if stronger expression of that behavior is less useful under distribution shift, the benefit of extrapolation may be smaller or less predictable. The search setting studied here is particularly illustrative because the harder evaluation benchmarks demand more sustained retrieval than the training questions. Our results on mathematical reasoning provide evidence that the approach is not limited to search, but broader evaluation across tasks, models, and RL objectives is needed to characterize when the recovered signal transfers reliably.

A natural extension is to combine the dense implicit reward with outcome-based validation. For example, extrapolation could be applied only to trajectories that pass a verifier, or its strength could be adjusted using a correctness signal. This would preserve the dense supervision available from the checkpoint pair while focusing extrapolation on policy changes that are supported by successful outcomes. More generally, separating transferable behavioral improvements from incidental changes in an RL update remains an important direction for making policy-shift reuse more reliable.

6 CONCLUSION

In this paper, we introduce Distillation with Implicit Reward (DIR), a framework for reusing RL-induced policy shifts to further train the same model. DIR constructs a virtual teacher by extrapolating the logit difference between frozen pre- and post-RL checkpoints, then trains a student initialized from the post-RL policy through on-policy reverse-KL distillation. The checkpoint pair provides dense token-level supervision on student-generated contexts, without outcome rewards, correctness filtering, or a separately trained reward model during the DIR stage. Experiments on mathematical reasoning and agentic search demonstrate performance gains beyond the evaluated RL endpoints. Starting from intermediate checkpoints also approaches later-stage RL performance with substantially fewer cumulative GPU hours, including the initial RL training cost.

These results suggest that an RL run can provide reusable supervision for subsequent policy improvement. The benefit depends on whether the learned preference shift remains useful as the student evolves: extrapolation may amplify undesirable preferences, and the local optimization interpretation does not guarantee improvement in the original task reward. Future work could investigate adaptive extrapolation, outcome-based validation, and broader evaluation across models, tasks, and RL objectives to better characterize when policy-shift reuse is effective.

AI USE STATEMENT

In this work, generative AI tools were used to raise questions and provide revision suggestions during manuscript preparation, and to assist with LaTeX table formatting. Generative AI was not used to formulate the research questions or hypotheses, design the method or experiments, implement the method, develop theoretical derivations, process data, analyze results, or draw conclusions. All AI-provided questions and suggestions were assessed by the authors, who wrote and revised the manuscript and take responsibility for the final content of this work.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. *arXiv preprint arXiv:2306.13649*, 2023. doi: 10.48550/arXiv.2306.13649. URL https://arxiv.org/abs/2306.13649.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions. *arXiv preprint arXiv:2505.23281*, 2025. doi: 10.48550/arXiv.2505.23281. URL https://arxiv.org/abs/2505.23281.

Jiacheng Chen, Qianjia Cheng, Fangchen Yu, Haiyuan Wan, Yuchen Zhang, Shenghe Zheng, Junchi Yao, Qingyang Zhang, Haonan He, Yun Luo, Yufeng Zhao, Futing Wang, Li Sheng, Chengxing Xie, Yuxin Zuo, Yizhuo Li, Wenxuan Zeng, Yulun Wu, Rui Huang, Dongzhan Zhou, Kai Chen, Yu Qiao, Lei Bai, Yu Cheng, Ning Ding, Bowen Zhou, Peng Ye, and Ganqu Cui. P1: Mastering physics olympiads with reinforcement learning. *arXiv preprint arXiv:2511.13612*, 2025a. doi: 10.48550/arXiv.2511.13612. URL https://arxiv.org/abs/2511.13612.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*, 2025b.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. doi: 10.48550/arXiv.2501.12948. URL https://arxiv.org/abs/2501.12948.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with LLMs. In *3rd AI for Math Workshop: Toward Self-Evolving Scientific Agents*, 2026. URL https://openreview.net/forum?id=DmPE4byHuN.

Shiyuan Feng, Huan-ang Gao, Haohan Chi, Hanlin Wu, Zhilong Zhang, Zheng Jiang, Bingxiang He, Wei-Ying Ma, Ya-Qin Zhang, and Hao Zhou. Weak-to-strong generalization via direct on-policy distillation. *arXiv preprint arXiv:2607.05394*, 2026. doi: 10.48550/arXiv.2607.05394. URL https://arxiv.org/abs/2607.05394.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous RL. *arXiv preprint arXiv:2508.07976*, 2025. doi: 10.48550/arXiv.2508.07976. URL https://arxiv.org/abs/2508.07976.

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://openreview.net/forum?id=5h0qf7IBZZ.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the*

62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Haonan He, Haodi Lei, Yun Luo, Haoran Zhang, Shunkai Zhang, Yizhuo Li, Shengji Tang, Zhilin Wang, Runzhe Zhan, Lei Bai, Ganqu Cui, Fangchen Yu, Yafu Li, Peng Ye, Ning Ding, and Yu Cheng. SimpleOPD: Simple tokenizer-agnostic on-policy distillation for long-context reasoning. *arXiv preprint arXiv:2608.14277*, 2026. doi: 10.48550/arXiv.2608.14277. URL https://arxiv.org/abs/2608.14277.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, 2020. URL https://arxiv.org/abs/2011.01060.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025. doi: 10.48550/arXiv.2503.09516. URL https://arxiv.org/abs/2503.09516.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, 2025. URL https://arxiv.org/abs/2409.12941.

Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=IFXTZERXdM7.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath. Hugging Face repository, 2024. URL https://huggingface.co/datasets/AI-MO/NuminaMath-CoT.

Ming Li, Chenrui Fan, Yize Cheng, Soheil Feizi, and Tianyi Zhou. Schoenfeld's anatomy of mathematical reasoning by language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32773–32802, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1513. URL https://aclanthology.org/2026.acl-long.1513/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Ziyuan Liu, Hengqi Liu, Zichuan Wang, Yang Qin, Jiachen Liang, Xu Chu, Shaowei Chen, Yuantao Gu, Zhaokai Luo, Yao Hu, and Mu Chuan. Iris: Climbing to the search frontier. *arXiv preprint arXiv:2609.04304*, 2026. doi: 10.48550/arXiv.2609.04304. URL https://arxiv.org/abs/2609.04304.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023. URL https://arxiv.org/abs/2210.03350.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023. doi: 10.48550/arXiv.2305.18290. URL https://arxiv.org/abs/2305.18290.

Alan H Schoenfeld. *Mathematical problem solving*. New York, 1985.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. doi: 10.48550/arXiv.2402.03300. URL https://arxiv.org/abs/2402.03300.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. URL https://arxiv.org/abs/2108.00573.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018. URL https://arxiv.org/abs/1809.09600.

Fangxu Yu, Weijia Xu, Michael Xu, Tianyi Zhou, and Zinan Lin. Weak-to-strong on-policy distillation. *arXiv preprint arXiv:2607.26246*, 2026a. doi: 10.48550/arXiv.2607.26246. URL https://arxiv.org/abs/2607.26246.

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

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026. doi: 10.48550/arXiv.2601.18734. URL https://arxiv.org/abs/2601.18734.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, Yuxin Gu, Sixin Hong, Jing Ren, Jian Chen, Chao Liu, and Yining Hua. BrowseComp-ZH: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*, 2025. URL https://arxiv.org/abs/2504.19314.

APPENDIX CONTENTS

A Major Notations and Definitions **14**

B Theoretical Interpretation of Policy-Shift Reuse **15**

B.1 Sequence-Level Reward Equivalence 15

B.2 Local Reward Reuse and Logit Extrapolation 15

B.3 Scope of the Interpretation 16

C Algorithm and Implementation Details **17**

C.1 Training Procedure and Gradient Flow 17

C.2 Candidate Supports and Target Normalization 17

C.3 Reverse-KL and Detached-Advantage Gradients 18

C.4 Token Masks and Checkpoint Choices 18

D Mathematical Reasoning Experimental Setup **19**

D.1 Training and Implementation 19

D.2 Evaluation Protocol 19

E Additional Mathematical Reasoning Results **22**

E.1 RL Checkpoint Scaling 22

E.2 Extrapolation Strength from an Early RL Checkpoint 22

F Search-Agent Experimental Setup **23**

F.1 Training and Implementation 23

F.2 Agent and Evaluation Protocol 23

G Additional Search-Agent Results **24**

G.1 Full Search-R1 Suite 24

G.2 RL Checkpoint Scaling on BCP+ 24

G.3 Optimization Controls 24

H Observation and Retrieval Diagnostics **25**

H.1 Tool Observation Format 25

H.2 Evidence Flow on BrowseComp-ZH 25

H.3 Stopping Behavior on FRAMES 25

A Major Notations and Definitions

Table 4 summarizes the major notations used in the main paper. We distinguish the frozen reference policies from the trainable student, and sequence-level quantities from their token-level counterparts. Unless a complete response y is specified, policy distributions refer to the next token conditioned on a generation context s.

Table 4: Major notations and definitions used in the main paper.

Notation	Definition
Inputs and generation contexts	
x	Input prompt or question.
y, y_t, $y_{<t}$, $\lvert y\rvert$	Generated response, its token at position t, the preceding response prefix, and the response length, respectively.
a	Candidate next token at a given generation context.
s, s_t	Generation context. For ordinary response generation, $s_t = (x, y_{<t})$; in agentic search, the context also includes tool observations and the visible interaction history.
Policies and logits	
π_{pre}	Pre-RL reference policy, frozen throughout the DIR stage.
π_{post}	Policy obtained after the selected RL stage, which may end at an intermediate or final checkpoint; frozen throughout DIR.
π_θ, π_{θ_k}	Trainable student with parameters θ, and the student used for on-policy sampling at iteration k, respectively. The student is initialized as $\pi_{\theta_0} = \pi_{\mathrm{post}}$.
$z_{\mathrm{pre}}(s)$, $z_{\mathrm{post}}(s)$	Next-token logit vectors produced by the two frozen reference policies on the same context s.
$z_T(s)$	Extrapolated teacher logits: $z_T(s) = z_{\mathrm{post}}(s) + \alpha[z_{\mathrm{post}}(s) - z_{\mathrm{pre}}(s)]$.
$\pi_T(\cdot \mid s)$	Virtual teacher distribution, $\pi_T(\cdot \mid s) = \mathrm{softmax}(z_T(s))$, constructed from the frozen checkpoint pair.
Policy shifts and reward reuse	
$r_\Delta(s, a)$	Token-level implicit reward: $\log \pi_{\mathrm{post}}(a \mid s) - \log \pi_{\mathrm{pre}}(a \mid s)$. It records the relative token preference change induced by RL.
$\Delta_{\mathrm{RL}}(x, y)$	Sequence-level log-policy shift: $\log[\pi_{\mathrm{post}}(y \mid x)/\pi_{\mathrm{pre}}(y \mid x)]$.
$r_{\Delta,t}$	Implicit reward evaluated at a realized response token: $r_{\Delta,t} = r_\Delta(s_t, y_t)$, as used in the behavioral analysis.
α	Nonnegative extrapolation coefficient. At $\alpha = 0$, the teacher reduces to π_{post}; positive values extrapolate the RL-induced logit shift.
$R(x, y)$	Sequence-level task reward in the idealized KL-regularized RL objective.
β	Positive KL-regularization coefficient in the idealized RL objective, distinct from the DIR extrapolation coefficient α.
$Z(x)$	Input-dependent normalizing constant in the idealized sequence-level optimum: $Z(x) = \sum_y \pi_{\mathrm{pre}}(y \mid x) \exp(R(x, y)/\beta)$.
$Z_\alpha(s)$	Context-dependent teacher normalizer: $Z_\alpha(s) = \sum_a \pi_{\mathrm{post}}(a \mid s) \exp(\alpha r_\Delta(s, a))$.
Local optimization and distillation	
q	Optimization variable in the main-text local reward-reuse problem: a next-token distribution at a fixed context s.
$D_{\mathrm{KL}}(p\Vert q)$	Kullback–Leibler divergence. DIR's underlying distillation objective uses the direction $D_{\mathrm{KL}}(\pi_\theta \Vert \pi_T)$.
d_k	Distribution over valid generation contexts collected using π_{θ_k}; held fixed during the corresponding student updates.
$\mathcal{L}_k(\theta)$	Underlying full-vocabulary on-policy distillation objective: $\mathbb{E}_{s\sim d_k}[D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s) \Vert \pi_T(\cdot \mid s))]$.

B Theoretical Interpretation of Policy-Shift Reuse

B.1 Sequence-Level Reward Equivalence

Fix an input x. Assume that the reference policy has positive probability on the response support under consideration and that the normalizing constants below are finite. Consider the idealized objective

$$J_x(\pi) = \mathbb{E}_{y\sim\pi(\cdot|x)}[R(x,y)] - \beta D_{\mathrm{KL}}(\pi(\cdot \mid x) \,\|\, \pi_{\mathrm{pre}}(\cdot \mid x)), \qquad \beta > 0. \tag{7}$$

Define

$$\pi^*(y \mid x) = \frac{\pi_{\mathrm{pre}}(y \mid x) e^{R(x,y)/\beta}}{Z(x)}, \qquad Z(x) = \sum_{y'} \pi_{\mathrm{pre}}(y' \mid x) e^{R(x,y')/\beta}. \tag{8}$$

Substitution gives

$$J_x(\pi) = \beta \log Z(x) - \beta D_{\mathrm{KL}}(\pi(\cdot \mid x) \,\|\, \pi^*(\cdot \mid x)). \tag{9}$$

Nonnegativity of KL divergence implies that π^* is the unique maximizer over distributions on this support. If $\pi_{\mathrm{post}} = \pi^*$, then

$$\Delta_{\mathrm{RL}}(x,y) = \frac{R(x,y)}{\beta} - \log Z(x). \tag{10}$$

The normalizer cancels when comparing two responses. Consequently, $\Delta_{\mathrm{RL}}(x,y) - \Delta_{\mathrm{RL}}(x,y') = [R(x,y) - R(x,y')]/\beta$: the shift preserves reward-induced preferences.

Independently of the optimization assumption, autoregressive factorization gives

$$\log \frac{\pi_{\mathrm{post}}(y \mid x)}{\pi_{\mathrm{pre}}(y \mid x)} = \sum_{t=1}^{|y|} \log \frac{\pi_{\mathrm{post}}(y_t \mid x, y_{<t})}{\pi_{\mathrm{pre}}(y_t \mid x, y_{<t})} = \sum_{t=1}^{|y|} r_\Delta(s_t, y_t). \tag{11}$$

This decomposition is exact for the realized checkpoint pair. It does not require the environment reward to possess a unique token-level decomposition.

B.2 Local Reward Reuse and Logit Extrapolation

Fix a context s and let $\alpha > 0$. Define

$$Z_\alpha(s) = \sum_a \pi_{\mathrm{post}}(a \mid s) e^{\alpha r_\Delta(s,a)} \tag{12}$$

and let π_T be the distribution in Eq. (5). For any distribution q,

$$D_{\mathrm{KL}}(q \,\|\, \pi_T(\cdot \mid s)) = D_{\mathrm{KL}}(q \,\|\, \pi_{\mathrm{post}}(\cdot \mid s)) - \alpha \mathbb{E}_{a\sim q}[r_\Delta(s,a)] + \log Z_\alpha(s). \tag{13}$$

Rearranging yields

$$\begin{aligned} &\mathbb{E}_{a\sim q}[r_\Delta(s,a)] - \frac{1}{\alpha} D_{\mathrm{KL}}(q \,\|\, \pi_{\mathrm{post}}(\cdot \mid s)) \\ &\qquad = \frac{\log Z_\alpha(s)}{\alpha} - \frac{1}{\alpha} D_{\mathrm{KL}}(q \,\|\, \pi_T(\cdot \mid s)). \end{aligned} \tag{14}$$

The right-hand side is uniquely maximized at $q = \pi_T$, proving Eq. (4). This identity also connects the distillation objective to reward reuse: at a fixed context, minimizing reverse KL to π_T is equivalent, up to an additive constant and positive scale, to maximizing the implicit reward with a KL penalty to the frozen post-RL policy. Across sampled contexts, parameter sharing constrains the student's approximation to these targets.

To obtain the implemented form, substitute $r_\Delta = \log \pi_{\text{post}} - \log \pi_{\text{pre}}$:

$$\pi_T(a \mid s) \propto \pi_{\text{post}}(a \mid s)^{1+\alpha} \pi_{\text{pre}}(a \mid s)^{-\alpha}. \tag{15}$$

Since $\log \pi_j(a \mid s) = z_j(s,a) - \log \sum_b e^{z_j(s,b)}$ for $j \in \{\text{pre}, \text{post}\}$,

$$\log \pi_T(a \mid s) = z_{\text{post}}(s,a) + \alpha\big[z_{\text{post}}(s,a) - z_{\text{pre}}(s,a)\big] + C(s), \tag{16}$$

where $C(s)$ is independent of a. Normalizing over tokens therefore gives Eq. (3) exactly. This equivalence holds for any checkpoint pair with common positive support, whether or not the original RL stage solved Eq. (7) exactly. At $\alpha = 0$, the logit formula gives $\pi_T = \pi_{\text{post}}$ directly.

B.3 Scope of the Interpretation

Realized shifts versus exact reward recovery. The reward correspondence in Eq. (10) assumes an exact optimum over response distributions. Practical RL updates also reflect finite sampling, clipping, optimization dynamics, and model parameterization. `DIR` uses the preference shift encoded by the checkpoints; it does not require exact recovery of the environment reward. A large $r_\Delta(s,a)$ indicates an increase in token probability during RL, but does not identify that token's causal contribution to task success.

Local and sequence-level normalization. The implemented teacher normalizes separately at each generation context. For an autoregressive response,

$$\log \pi_T(y \mid x) = \log \pi_{\text{post}}(y \mid x) + \alpha \Delta_{\text{RL}}(x,y) - \sum_{t=1}^{|y|} \log Z_\alpha(s_t). \tag{17}$$

The final sum generally depends on the response through its prefixes. Thus, the local teacher need not equal the globally normalized distribution proportional to $\pi_{\text{post}}(y \mid x) e^{\alpha \Delta_{\text{RL}}(x,y)}$. The sequence-level result motivates the signal, while the local optimization defines the teacher actually used by `DIR`.

For interactive search, s includes the full visible history, including tool observations. The local construction and its optimization identity apply to the next assistant token at that history. They do not assume that tool observations are generated by the assistant policy.

On-policy realization. DIR combines a fixed preference signal with supervision evaluated on contexts generated by the evolving student. Although the checkpoint pair remains frozen, its next-token distributions are recomputed on newly sampled prefixes, allowing the same RL-induced shift to provide dense supervision throughout training. The local identities above characterize the teacher at each visited context, while on-policy distillation fits these targets under the student's current context distribution. This connects checkpoint-level policy-shift reuse to an iterative training procedure without requiring new outcome rewards. Our experiments examine the resulting performance gains and compute savings in mathematical reasoning and agentic search.

C Algorithm and Implementation Details

C.1 Training Procedure and Gradient Flow

Both reference checkpoints remain fixed throughout a `DIR` stage. The student is initialized from π_{post} and generates the trajectories used for its updates. The teacher need not generate separate responses: it is evaluated on the student's contexts. At each iteration, the sampled contexts, candidate-token indices, and teacher targets are held fixed during optimization.

Algorithm 1 Distillation with Implicit Reward (DIR)

Input: Prompt set $\mathcal{D}$; frozen checkpoints π_{pre} and π_{post}; extrapolation coefficient α; candidate-support rule; optimization budget B
Output: Trained student policy π_θ

Initialize $\pi_\theta \leftarrow \pi_{\text{post}}$
Keep π_{pre} and π_{post} frozen throughout training
while the optimization budget B is not exhausted **do**
  Sample a prompt batch $\mathcal{X}$ from $\mathcal{D}$
  Generate on-policy trajectories using π_θ on $\mathcal{X}$
  Extract valid assistant-generation contexts $\mathcal{C}$ and construct loss masks
  for each context $s \in \mathcal{C}$ **do**
    Obtain checkpoint outputs and select candidate support S_s according to the task-specific rule in Appendix C.2
    Construct teacher logits on S_s:

$$z_T(s,a) \leftarrow z_{\text{post}}(s,a) + \alpha\big(z_{\text{post}}(s,a) - z_{\text{pre}}(s,a)\big), \qquad a \in S_s$$

    Normalize and detach the teacher target:

$$q_s(a) \leftarrow \text{stopgrad}\left[\frac{\exp z_T(s,a)}{\sum_{b\in S_s}\exp z_T(s,b)}\right], \qquad a \in S_s$$

  end for
  Compute the task-specific distributional distillation loss using $\{q_s\}_{s\in\mathcal{C}}$
  Apply loss masks and the task-specific loss reduction
  Update only θ, holding sampled contexts, candidate supports, and teacher targets fixed
end while
return π_θ

The implicit reward is useful for interpreting the update; implementation can construct the teacher directly from logits. During a student update, neither the checkpoint parameters nor the context-sampling distribution is differentiated. The main `DIR` runs apply no correctness-based trajectory filter and use no outcome-reward term in the student loss.

C.2 Candidate Supports and Target Normalization

The full-vocabulary teacher is defined by Eq. (3). Computation uses a candidate support S_s at each context. Its restricted target is

$$\pi_T^{S_s}(a \mid s) = \frac{\exp z_T(s,a)}{\sum_{b\in S_s}\exp z_T(s,b)}, \qquad a \in S_s. \tag{18}$$

This is the full teacher conditioned on S_s; it is not an assertion that the full teacher assigns zero mass outside the candidate set. Restricting distributional supervision to a compact candidate set reduces the number of token scores retained and processed by the distillation loss. It can also reduce the storage and transfer requirements for teacher targets, while preserving the teacher's relative probabilities among the selected candidates. This retains soft supervision over multiple alternatives rather than reducing the target to a single token. Candidate supports used in the experiments are summarized below.

Setting	Support selection	Size
Mathematical reasoning	Top tokens under $z_T(s)$	32
Agentic search	Top tokens under $\pi_{\theta_k}(\cdot \mid s)$	16

In mathematical reasoning, the extrapolated logits are constructed over the full vocabulary before support selection. Student and teacher are renormalized on the same selected support. In search, both frozen checkpoints are evaluated on the candidate IDs selected by the pre-update student. Their full-vocabulary log probabilities can be combined as $(1+\alpha)\log \pi_{\text{post}} - \alpha \log \pi_{\text{pre}}$ and then normalized within this support. By Eq. (16), this gives the same target as Eq. (18).

C.3 Reverse-KL and Detached-Advantage Gradients

The mathematical-reasoning implementation directly differentiates the support-normalized reverse KL. For a fixed context and support, write $p_\theta(a) = \pi_\theta^{S_s}(a \mid s)$ and $q(a) = \pi_T^{S_s}(a \mid s)$. The token loss is

$$\ell_s(\theta) = \sum_{a \in S_s} p_\theta(a) \left[\log p_\theta(a) - \log q(a)\right]. \tag{19}$$

Gradients pass through both occurrences of p_θ, while q and S_s are fixed.

A detached-advantage representation clarifies the relation to distributional policy gradients. Let $p_k = p_{\theta_k}$, and define the detached quantities

$$A_k(a) = \log q(a) - \log p_k(a), \qquad p_k(a),\ A_k(a) \text{ fixed during differentiation.} \tag{20}$$

Because $\sum_a \nabla_\theta p_\theta(a) = 0$ on the normalized support,

$$\nabla_\theta \ell_s(\theta)|_{\theta=\theta_k} = -\sum_{a \in S_s} p_k(a) A_k(a)\ \nabla_\theta \log p_\theta(a)|_{\theta=\theta_k}. \tag{21}$$

An action-independent baseline can be subtracted from A_k without changing this identity. Equivalently, the detached surrogate

$$\ell_s^{\text{det}}(\theta) = -\sum_{a \in S_s} \text{sg}[p_k(a) A_k(a)] \log p_\theta(a) \tag{22}$$

has the same gradient as the direct reverse KL at θ_k, where sg denotes stop-gradient. This is a gradient identity at the pre-update policy, not an equality of losses for arbitrary θ. In particular, a sequence of updates with fixed advantages need not follow the same gradients as direct reverse-KL minimization after the student moves away from θ_k.

The search implementation detaches candidate IDs, pre-update student log probabilities, teacher targets, and the resulting advantages; only the current student receives gradients. The identity above describes the reverse-KL reference gradient for a shared normalized support and makes explicit the conditions needed to relate a detached-advantage update to this objective.

C.4 Token Masks and Checkpoint Choices

Mathematical reasoning applies the distillation loss to valid response tokens, excluding prompt and padding tokens. Search applies policy loss only to assistant-generated tokens, including assistant-generated tool calls; tool observations and other environment-generated tokens are excluded. Malformed calls and length- or turn-limit violations trigger trajectory masking in the search implementation. These validity masks do not depend on answer correctness. The search loss uses a per-sequence reduction over valid assistant tokens before aggregation; the mathematical-reasoning loss is averaged over valid response tokens.

For post-RL extension, the mathematical-reasoning student and frozen post-RL reference are initialized from RL step 279; the search counterparts use RL step 320. The compute-efficiency experiments instead use RL step 100 and RL step 96, respectively. The pre-RL reference remains the corresponding Qwen3 base model throughout. Within each `DIR` run, the checkpoint pair is fixed. Training prompts, optimizer settings, rollout budgets, and evaluation protocols are specified in the task-specific experimental appendices.

D MATHEMATICAL REASONING EXPERIMENTAL SETUP

D.1 TRAINING AND IMPLEMENTATION

RL and DIR use Qwen3-4B-Base in non-thinking mode, the same processed DAPO-Math-17k training data, and the same non-thinking chat template. After filtering prompts that exceed 1,024 tokens, the training set contains 17,909 examples. RL optimizes a binary answer-correctness reward with GRPO and runs for 279 updates. Table 5 summarizes the training configuration.

For DIR, the pre-RL policy is Qwen3-4B-Base. The post-RL policy and the initial student are both the RL step-279 checkpoint for post-RL extension, or the RL step-100 checkpoint for the compute-efficiency experiments. Both reference policies remain frozen throughout each DIR run. We evaluate extrapolation coefficients $\alpha \in \{0.5, 0.75, 1, 2\}$ and save checkpoints every 10 updates, up to 100 DIR updates.

The mathematical-reasoning implementation constructs the extrapolated logits over the full vocabulary, and selects the top 32 tokens under this proxy at each response position. The student and proxy distributions are renormalized on this shared support, and the student directly minimizes their reverse KL divergence. The proxy targets and support indices are detached, and the loss is averaged over valid response tokens; prompt and padding tokens do not contribute.

Training uses verl with FSDP and vLLM rollouts. The RL run uses eight NVIDIA H20 GPUs, whereas each DIR run uses four; the actor and reference-model FSDP computations are configured in FP32. For same-checkpoint continued-RL controls, GRPO is initialized from the same merged RL checkpoint, with optimizer, scheduler, and data-loader state initialized anew.

Table 5: Mathematical-reasoning training configuration for the RL and DIR experiments. Lengths are measured in tokens. Checkpoint evaluation uses the settings described below; benchmark evaluation settings are listed in Table 6.

Setting	RL	DIR
Initialization	Qwen3-4B-Base	RL 279 / RL 100 steps
Training prompts after filtering	17,909	17,909
Prompts per update	64	64
Rollouts per prompt	8	1
Updates	279	100
Learning rate	10^{-6}	10^{-6}
Rollout sampling	$T = 1$, top-$p = 1$	$T = 1$, top-$p = 1$
Prompt / response limit	1,024 / 7,168	1,024 / 7,168
Checkpoint evaluation response limit	16,384	16,384
Checkpoint / evaluation interval	10 updates	10 updates
GPUs	8 H20	4 H20

D.2 EVALUATION PROTOCOL

We evaluate the initial model and subsequent checkpoints every 10 updates on 143 problems: 30 from AIME24, 30 from AIME25, and 83 from AMC. Checkpoint evaluation uses 16 responses per problem, temperature 1, top-p 0.95, and a maximum response length of 16,384 tokens. AIME24 and AIME25 are used in both checkpoint evaluation and benchmark evaluation.

Benchmark evaluation uses the problem counts and sampling counts in Table 6. The seven benchmarks reported in the main paper contain 1,576 problems in total. Each problem is augmented with the instruction "Please reason step by step, and put your final answer within \boxed{}." We then apply the model's non-thinking chat template and sample with temperature 0.7 and top-p 0.95. The context window is 32,768 tokens, and each response is capped at the remaining context budget: $\min(32{,}768, 32{,}768 - L_{\text{prompt}})$. Generation stops at either <|im_end|> or <|endoftext|>.

Answers are evaluated with a rule-based verifier that extracts the final answer and checks normalized string or symbolic equivalence to the reference answer. For each benchmark, we report mean correctness across its sampled responses. The aggregate score is the unweighted mean of the seven

benchmark accuracies. The competition benchmarks use sixteen samples per problem, whereas MATH-500, Minerva, and Olympiad-Bench use four samples.

Table 6: Mathematical-reasoning benchmark evaluation protocol. All benchmarks use non-thinking decoding with $T = 0.7$, top-$p = 0.95$, and a 32,768-token context window.

Benchmark	Problems	Samples per problem
AIME24	30	16
AIME25	30	16
AMC23	40	16
HMMT25	30	16
MATH-500	500	4
Minerva	272	4
Olympiad-Bench	674	4

Evaluation Metric: avg@n. For a benchmark containing N problems, we sample n responses per problem using the decoding settings described above. Let $c_{i,j} \in \{0, 1\}$ indicate whether the verifier judges the j-th response to the i-th problem as correct. We first average correctness over the n responses to each problem and then average across all N problems:

$$\text{avg}\,@n = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{n} \sum_{j=1}^{n} c_{i,j} \right). \tag{23}$$

We report this quantity as a percentage. This metric estimates single-sample accuracy (pass@1) under the specified sampling distribution, with n responses per problem used for estimation. Each response is scored individually, without best-of-n selection or majority voting.

We use $n = 16$ for AIME 2024, AIME 2025, AMC23, and HMMT25, and $n = 4$ for MATH-500, Minerva, and Olympiad-Bench. The aggregate score, denoted as *Macro Avg*, is the unweighted mean of the seven benchmark scores:

$$\text{MacroAvg} = \frac{1}{7} \sum_{b=1}^{7} \text{avg}\,@n_b(B_b), \tag{24}$$

where B_b denotes the b-th benchmark and n_b its sampling count. Each benchmark therefore contributes equally to the aggregate score, regardless of its number of problems or sampled responses.

Details of Evaluation Benchmarks To comprehensively evaluate the mathematical reasoning capabilities of our models, we utilize a diverse set of benchmarks spanning high-school mathematics, quantitative reasoning, and Olympiad-level problems. Below, we detail each dataset used in our experiments:

- **AIME 2024** (Li et al., 2024): The American Invitational Mathematics Examination (AIME) is a highly prestigious and challenging mathematics competition in the United States. It consists of 15 questions meant to be completed in 3 hours and is given to top-scoring students on the AMC exam. Each year, there are two versions of the exam which consist of distinct questions. The AIME 2024 benchmark consists of 30 problems from the 2024 iteration of this exam. As a recent competition-level dataset, it serves as a rigorous evaluation that assesses advanced multi-step deduction, logical reasoning, and complex algebraic manipulation.
- **AIME 2025** (Balunović et al., 2025): Following the same format as AIME 2024, this benchmark contains 30 problems from the 2025 AIME competition. Answers are integers between 0 and 999.
- **AMC** (Li et al., 2024): The American Mathematics Competitions (AMC) benchmark comprises competition-level questions given to top high school students. Checkpoint evaluation uses 83 problems from the 2022 and 2023 AMC12 exams. Benchmark evaluation uses the 40-problem AMC23 set listed in Table 6. Serving as a precursor to the AIME, the AMC benchmark tests a broad spectrum of high-school mathematics, including geometry, number theory, algebra, and combinatorics. It evaluates the model's foundational mathematical logic and problem-solving

stability across varying difficulty levels. Although the original problems are in multiple-choice format, the dataset presents modified versions of the problem which expect an integer solution.

- **HMMT 2025** (Dekoninck et al., 2026): The HMMT 2025 benchmark consists of 30 short-answer problems from the individual rounds of the February 2025 Harvard–MIT Mathematics Tournament, as curated by MathArena. It comprises three tests, Algebra and Number Theory, Combinatorics, and Geometry, with 10 problems each. These competition-level problems assess multi-step deduction, symbolic manipulation, combinatorial reasoning, and geometric problem solving. The reference answers include both numerical values and symbolic expressions, enabling evaluation based on the correctness of the final answer.
- **MATH-500** (Hendrycks et al., 2021; Lightman et al., 2024): MATH-500 is a curated, representative subset of 500 problems created by OpenAI (Lightman et al., 2024) by uniformly sampling from the expansive MATH dataset (Hendrycks et al., 2021). It contains competition math problems that cover seven distinct mathematical disciplines across five difficulty levels. This subset has been widely adopted in recent reasoning literature as it provides a robust and computationally efficient proxy for full MATH dataset evaluations, which is particularly crucial when executing computationally intensive decoding algorithms such as majority voting or tree-search strategies.
- **Minerva Math** (Lewkowycz et al., 2022): The Minerva benchmark derives from the quantitative evaluation suite introduced by Lewkowycz et al. (2022). It encompasses a broad spectrum of challenging STEM problems that necessitate profound mathematical understanding, precise symbolic manipulation, and the generation of exact numerical or algebraic final answers, pushing beyond simple arithmetic tasks.
- **Olympiad-Bench** (He et al., 2024): Olympiad-Bench is an expert-level, highly challenging benchmark designed to evaluate language models on problems sourced from international mathematics and physics Olympiads (e.g., the International Mathematical Olympiad). In our evaluation, we focus primarily on the mathematical subset. These problems require profound insight, formal logic, and rigorous proof-like multi-step derivations.

E Additional Mathematical Reasoning Results

E.1 RL Checkpoint Scaling

Table 7 reports performance across seven checkpoints from the mathematical-reasoning RL run. The macro-average accuracy increases from 37.10% at step 100 to 41.02% at step 250, a gain of 3.92 percentage points, before decreasing to 40.56% at step 279. Individual benchmarks also exhibit non-monotonic changes: Minerva attains its highest reported accuracy at step 190, AMC at step 220, and AIME25, MATH-500, and Olympiad-Bench at step 250. Thus, the final RL checkpoint is not the strongest checkpoint on every benchmark or in aggregate.

These results contextualize both the early-checkpoint and post-RL DIR experiments. RL step 100 provides the initialization for the compute-efficiency comparison, whereas the later checkpoints suggest that RL is approaching a performance plateau: the macro-average accuracy remains between 40.35% and 41.02% over steps 220–279, with no sustained improvement toward the end of training. Against this late-stage plateau, the further gains obtained by applying DIR to RL step 279 represent a non-trivial improvement, rather than an outcome that can be assumed from simply extending the earlier upward trend. This interpretation is consistent with the non-monotonic checkpoint behavior observed in the search-agent experiments, and cautions against interpreting DIR as a simple linear extension of the RL learning curve.

Table 7: Mathematical-reasoning performance across RL checkpoints (%). Macro Avg is the unweighted mean of the seven benchmark accuracies. Bold denotes the highest reported value in each column, including ties.

Step	AIME24	AIME25	AMC	HMMT25	MATH-500	Minerva	Olympiad	Macro Avg
100	18.33	17.29	58.44	8.75	81.60	29.60	45.66	37.10
130	21.25	17.71	61.56	8.75	81.25	31.07	47.89	38.50
160	23.13	19.17	61.41	8.54	82.25	30.61	47.81	38.99
190	22.71	17.50	64.38	8.96	83.30	**33.55**	47.55	39.71
220	22.08	20.00	**65.63**	10.42	83.10	32.72	48.48	40.35
250	**23.54**	**21.25**	64.38	**11.25**	**83.85**	32.81	**50.04**	**41.02**
279	**23.54**	19.58	65.16	**11.25**	83.55	31.71	49.11	40.56

E.2 Extrapolation Strength from an Early RL Checkpoint

We complement the post-RL extrapolation-strength analysis in Section 4.4 by examining DIR initialized from an earlier RL checkpoint. Starting from RL step 100, we compare $\alpha \in \{0.5, 0.75, 1, 2\}$ after 20 DIR updates for each setting, using the training and evaluation protocol described in Appendix D. The pre-RL and post-RL reference policies are fixed to Qwen3-4B-Base and RL step 100, respectively. Table 8 reports the seven-benchmark results, with the base model and RL step 100 included as references.

All four DIR settings improve over RL step 100 on every benchmark. The macro-average accuracy increases from 37.10% to 39.11%–40.72%, corresponding to gains of approximately 2.01–3.62 percentage points. The macro-average increases with α over the tested values, although individual benchmark scores are not uniformly monotonic. The strongest setting in this comparison is $\alpha = 2$, which achieves 40.72% on average and the highest accuracy on five of the seven benchmarks; $\alpha = 1$ performs best on AIME24, while $\alpha = 0.75$ performs best on AMC23. The gains across all tested coefficients indicate that reward reuse remains effective before the late-stage RL plateau.

The contrast with Section 4.4 suggests that the preferred extrapolation strength may depend on the stage of RL training. For the later checkpoint, $\alpha = 0.75$ achieves the highest reported macro-average accuracy, but scores across coefficients span only 43.83%–44.17%, compared with 39.11%–40.72% for the early checkpoint. Together with the increasing macro-average accuracy as α grows in the early-checkpoint comparison, these observations suggest that stronger extrapolation may be more beneficial when reusing the policy shift learned early in RL, whereas later checkpoints are less sensitive to the coefficient. One possible explanation is that stronger extrapolation compensates for a less developed policy shift at an early training stage.

Table 8: Effect of extrapolation strength when DIR is initialized from RL step 100. All DIR settings are compared after 20 additional updates. Values are accuracies (%), and Macro Avg is the unweighted mean across the seven benchmarks. Sampling counts follow Appendix D.2. Bold denotes the highest accuracy in each column.

Method	AIME24	AIME25	AMC23	HMMT25	MATH-500	Minerva	Olympiad	Macro Avg
Qwen3-4B-Base	8.33	5.62	29.06	1.67	37.05	8.82	21.96	16.07
↪ RL 100 steps	18.33	17.29	58.44	8.75	81.60	29.60	45.66	37.10
↪ DIR, $\alpha = 0.5$	20.21	19.38	63.75	9.58	82.45	31.25	47.18	39.11
↪ DIR, $\alpha = 0.75$	20.42	20.00	**64.53**	9.38	82.15	32.35	46.11	39.28
↪ DIR, $\alpha = 1$	**23.33**	19.79	62.34	10.83	82.15	32.72	47.66	39.83
↪ DIR, $\alpha = 2$	22.92	**22.92**	63.13	**11.04**	**83.20**	**33.64**	**48.22**	**40.72**

F Search-Agent Experimental Setup

F.1 Training and Implementation

RL and DIR use the same Qwen3-8B agent, prompt, search interface, and multi-turn environment. Table 9 summarizes the main differences. RL optimizes answer correctness with GRPO. The main DIR runs do not filter trajectories by correctness.

Table 9: Search-agent training configuration.

Setting	**RL**	**DIR**
Initialization	Qwen3-8B	RL320
Training prompts	13,285	13,285
Prompts per update	128	256
Rollouts per prompt	8	1
Updates	320	96 (2 epochs)
Learning rate	10^{-6}	10^{-6}
Rollout sampling	$T = 1$, top-$p = 1$	$T = 1$, top-$p = 1$
Prompt / response length	4,096 / 32,768	4,096 / 32,768

For DIR, π_{pre} is Qwen3-8B Base, while both π_{post} and the initial student are RL320. The default setting uses $\alpha = 1$ and the top 16 student tokens at each position. Only assistant-generated tokens receive policy loss; tool observations and environment tokens are masked. Malformed calls and length or turn-limit violations trigger trajectory masking.

F.2 Agent and Evaluation Protocol

The agent issues up to four semantic queries per tool call and receives model-generated summaries of the retrieved documents. Summary generation uses temperature 0.6, top-p 0.95, top-k 20, and a maximum of 1,024 tokens. RL and DIR share the same prompt, tools, and summary module.

Table 10: Search-agent evaluation protocols. All settings use one trajectory per question and a deterministic Qwen3-32B answer-only judge.

Benchmark	**Questions**	**Retrieval**	**Context**	**Limit**	**Decoding**
BCP+	830	top-10	40,960	128 turns	$T = 0.7$, top-$p = 0.8$
Search-R1	6,125	top-3	36,864	100 turns	greedy
BrowseComp-ZH	289	IQS top-10	40,960	32 searches	$T = 0.7$, top-$p = 0.8$
FRAMES	391	Wiki18 top-10	40,960	128 turns	$T = 0.7$, top-$p = 0.8$

The main paper uses the four multi-hop Search-R1 subsets (3,125 questions); Appendix G additionally reports NQ, TriviaQA, and PopQA. FRAMES uses a fixed Wiki18-compatible subset. The judge receives only the question, reference answer, and extracted final answer.

G Additional Search-Agent Results

G.1 Full Search-R1 Suite

Table 11: Answer accuracy on the full Search-R1 suite (%).

Model	NQ	Trivia	PopQA	Hotpot	2Wiki	MuSiQue	Bamboogle
Qwen3-8B	52.0	67.4	38.8	52.8	34.2	21.8	50.4
↪ RL	62.2	**81.5**	**48.5**	70.9	**56.8**	**34.7**	64.8
↪ DIR	**62.4**	81.2	**48.5**	**71.0**	56.6	34.0	**69.6**

DIR maintains performance on the broad suite and produces its largest gain on Bamboogle. Average search calls rise from 13.24 for RL to 24.30 for DIR, consistent with Section 4.5.

G.2 RL Checkpoint Scaling on BCP+

Table 12: BCP+ performance across RL checkpoints.

Step	Accuracy	Recall	Avg. searches
96	27.23	45.29	21.08
128	28.92	46.10	17.73
160	32.29	48.78	18.16
192	32.77	50.97	19.79
224	33.73	49.93	19.23
256	34.10	52.37	23.22
288	33.37	50.77	21.40
320	**36.14**	**53.78**	24.23

Performance improves overall but is not monotonic across checkpoints. This variation cautions against viewing DIR as a simple linear extension of the RL learning curve.

G.3 Optimization Controls

Table 13: Additional BCP+ controls. The correctness-mask variant and the second epoch differ from the main method in only the indicated setting.

Variant	Acc.	Recall	Completion	Searches
RL320	36.14	53.78	90.12	24.23
DIR, step 48	37.83	60.59	71.69	32.39
DIR, step 96	38.80	60.77	73.37	31.52
DIR + correctness mask, step 48	38.43	**61.90**	73.37	31.82
Continued RL (32 steps)	**39.52**	58.48	81.33	27.33

Most of the DIR gain appears in the first epoch. Correctness masking provides a small additional gain at step 48, while pure DIR remains close without using answer correctness. Continued RL uses the same additional training compute as DIR. It resumes from the RL320 model weights without restoring the optimizer or scheduler state.

H Observation and Retrieval Diagnostics

H.1 Tool Observation Format

Table 14: Effect of tool observation format on BCP+ (%). Raw-3k supplies at most 3,000 characters of retrieved text.

Model	Observation	Acc.	Recall	Completion
RL320	Self-summary	36.14	53.78	90.12
RL320	No summary	18.67	36.92	37.71
RL320	Raw-3k	22.05	48.51	88.43
DIR	Self-summary	38.80	60.77	73.37
DIR	No summary	15.90	43.38	19.88
DIR	Raw-3k	26.87	57.13	74.58

Both policies are sensitive to how retrieved text enters the context. Raw-3k recovers much of the completion and recall lost without summaries, and DIR remains ahead of RL320 under this protocol. Because visible observation lengths differ, the three modes are not a strict single-variable ablation.

H.2 Evidence Flow on BrowseComp-ZH

Table 15: BrowseComp-ZH evidence-flow diagnostics (%). Retention and conversion are conditioned on retrieval evidence hit.

Model	Retrieval hit	Summary hit	Retention	Answer conversion
Qwen3-8B	29.76	7.61	25.58	18.60
RL320	61.25	30.10	49.15	30.51
DIR	**67.47**	**39.10**	**57.95**	**33.85**

DIR improves retrieval coverage, evidence retention in the generated summary, and conversion from retrieved evidence to the final answer. Evidence hit is a lexical proxy based on whether the normalized reference answer appears in the documents visible to the model; it may miss aliases and indirect multi-document evidence.

H.3 Stopping Behavior on FRAMES

DIR searches more than RL320 on 321 of 391 paired trajectories. Among these cases, DIR alone is correct on 22 questions and RL320 alone on 21. After answer evidence first appears in the summary, RL320 performs 5.90 more searches on average, compared with 13.40 for DIR. Thirteen DIR trajectories continue to the context limit after observing answer evidence, while none of the RL320 trajectories do so. This gap between evidence acquisition and stopping motivates explicit evidence-sufficiency or stopping objectives.