LEARNING TO HINT FOR REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Group Relative Policy Optimization (GRPO) often suffers from advantage collapse in reinforcement learning with verifiable rewards: when all rollouts in a group receive the same reward, their relative advantages vanish. Hint-based methods address all-incorrect groups by adding guidance that induces mixed outcomes and recovers a learning signal. However, creating a learning signal under the hinted input does not necessarily improve the no-hint policy used at test time. To this end, we propose **Hi**nt **L**earning for Reinforcement **L**earning (HiLL), a framework that learns to generate hints by considering both signal creation and signal transfer. HiLL trains a hinter alongside the reasoner, generating hints conditioned on the question, an incorrect rollout from the current reasoner, and a reference solution available only during training. We introduce *hint reliance*, which measures how strongly correct hinted trajectories depend on the hint by comparing their likelihoods with and without it. We derive a theoretical transfer bound relating hint reliance to hinted and no-hint success probabilities. Motivated by this result, we construct a transfer-weighted hinter reward that favors hints that recover informative GRPO groups while penalizing high hint reliance. Experiments across eight benchmarks with two backbone LLMs show that HiLL outperforms GRPO and hint-based baselines. Ablations further show that transfer weighting reduces hint reliance and improves no-hint performance, demonstrating the value of transfer-aware hint learning for RL.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become a standard approach to improve large language model (LLM) reasoning, especially in mathematics, where final-answer correctness can be checked reliably (Guo et al., 2025; Wen et al., 2025). Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is widely used for RLVR, estimating advantages from groups of sampled rollouts without a separate value critic. Under binary rewards, however, GRPO suffers from advantage collapse: when all rollouts in a group are correct or all are incorrect, their relative advantages vanish and the group provides no reward-based learning signal (Mroueh, 2025; Xiong et al., 2025; Le et al., 2026; Liao et al., 2026). This is particularly limiting for hard questions, where correct trajectories are rarely sampled and repeated all-incorrect groups prevent the reasoner from directly learning from these failures (Qu et al., 2026a; Chen et al., 2025).

Existing methods address this issue through adaptive sampling (Yao et al., 2025; Xiong et al., 2025) or filtering and reshaping uninformative groups (Zheng et al., 2025; Xu et al., 2025; Le et al., 2026; Yu et al., 2025). Hint-based methods instead add guidance, privileged prefixes, or reasoning scaffolds to induce mixed outcomes and recover non-zero GRPO advantages (Chen et al., 2025; Zhang et al., 2025b; Liao et al., 2026; Zhang et al., 2025d). This guidance can adapt during training through changes in hint strength or periodic hint generation with the current policy (Zhang et al., 2025d; Liao et al., 2026). Related work also learns teaching strategies or guidance generators alongside a learner (Fan et al., 2018; Qu et al., 2026b), or adjusts learner updates to use training-only feedback (Song et al., 2026).

In this paper, we focus on what makes a hint useful for learning, rather than only for solving the hinted question. A hint that restores a GRPO signal does not necessarily improve the no-hint policy used at test time. For example, supplying a key intermediate result may enable correct solutions while leaving the reasoner dependent on information absent from the original question. Recovering mixed outcomes alone does not distinguish such hints from guidance toward correct trajectories that

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.

remain plausible without the hint. The hint-learning objective should therefore account for both signal creation and signal transfer to the no-hint policy.

To this end, we propose **Hi**nt **L**earning for Reinforcement **L**earning (HiLL), a framework that learns to generate hints using both criteria. HiLL trains a hinter alongside the reasoner. For each question with an all-incorrect group, the hinter generates candidate hints conditioned on the question, an incorrect rollout from the current reasoner, and a reference solution available only during training. Conditioning on the failed rollout allows hint generation to target the reasoner's current errors, while joint training lets the hinter adapt as those errors evolve.

To guide hint learning, we introduce *hint reliance*, which measures how strongly correct hinted trajectories depend on the hint by comparing their likelihoods with and without it. We derive a theoretical transfer bound relating hint reliance to hinted and no-hint success probabilities. Motivated by this result, we define a transfer-weighted hinter reward that combines the recovery of mixed-outcome groups with a penalty for high hint reliance. The reward both trains the hinter and selects hinted groups for the reasoner update, making hint reliance an explicit consideration in generating and choosing training-time guidance.

We evaluate HiLL across eight benchmarks with two backbone LLMs, covering mathematical reasoning and out-of-distribution generalization. HiLL outperforms GRPO and hint-based baselines across both backbones, with gains that persist in compute-matched comparisons. Ablations further show that transfer weighting reduces hint reliance and improves no-hint performance over rewarding signal creation alone. In summary, we make the following contributions:

- We introduce hint reliance and derive a theoretical transfer bound relating hinted and no-hint success, providing a transfer-aware basis for evaluating training-time hints.
- We propose HiLL, which learns failure-conditioned hints through a transfer-weighted reward that accounts for both signal creation and signal transfer.
- We show on comprehensive benchmarks that HiLL consistently outperforms standard GRPO and hint-based baselines, demonstrating the value of adaptive and transfer-aware hint learning for RL.

2 Related Work

Adaptive hinting and learned guidance. Privileged hinting supports RLVR on difficult questions through solution prefixes (Qu et al., 2026a), progressive hints (Zhang et al., 2025b), and curriculum-based scaffolds (Chen et al., 2025). Scaf-GRPO selects teacher-generated hints of increasing strength when weaker guidance does not recover successful rollouts (Zhang et al., 2025d), while SAGE periodically refreshes self-generated hints using the current policy and adjusts their strength based on rollout outcomes (Liao et al., 2026). GHPO and SEELE adapt solution-prefix guidance to the learner's capability (Liu et al., 2025; Li et al., 2025), while ADHint combines adaptive hint ratios with difficulty-aware advantage estimation and token-level gradient controls (Zhang et al., 2025a). Learning the guidance itself also connects to Learning to Teach, which optimizes teaching strategies using feedback from the student's learning process (Fan et al., 2018). For LLM reasoning, RLAD jointly trains an abstraction generator and a solver, rewarding generated abstractions according to the solver's success rate (Qu et al., 2026b). HiLL takes a transfer-aware approach to learning hints, adapting guidance to the reasoner's evolving failures with the goal of improving no-hint reasoning. It considers not only whether hints help solve difficult questions, but also whether the learning signals they create remain useful when the hints are removed.

Learning from training-only guidance. Hint learning is also related to methods that use privileged information during training to improve the policy used without that information at inference. Critique-GRPO learns from initial responses and critique-guided refinements (Zhang et al., 2025c), while RLTF explicitly studies how training-only textual feedback can improve a no-feedback policy (Song et al., 2026). Related on-policy self-distillation methods provide token-level supervision on student-generated trajectories through teacher distributions conditioned on reference solutions, demonstrations, or textual feedback (Zhao et al., 2026; Hübotter et al., 2026; Shenfeld et al., 2026). Recent work further highlights how teacher–student information asymmetry can limit the transfer of privileged supervision (Yang et al., 2026; Yu et al., 2026; Kim et al., 2026). HiLL approaches this shared transfer challenge through learning to hint, adapting hints to the reasoner's evolving failures with the goal of improving no-hint reasoning. Our theoretical analysis of hint reliance relates

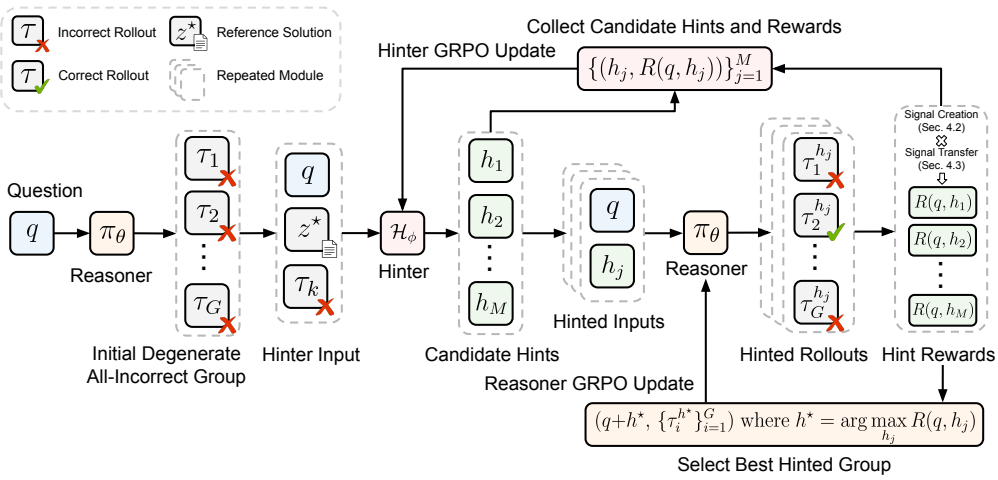

Figure 1: Overview of HiLL. Given a question q with an all-incorrect group, the hinter $\mathcal{H}_\phi$ takes the question, a failed rollout τ_k, and the reference solution $z^\star$ as input, and generates M candidate hints. The reasoner π_θ re-samples G rollouts under each hinted input $q{+}h_j$. Each hint is then scored by signal creation (Sec. 4.2) and signal transfer (Sec. 4.3). The best hinted group is selected for the reasoner GRPO update, while all candidate hints form the group for the hinter GRPO update.

hinted and no-hint success, providing a basis for learning hints whose successful trajectories remain plausible when the hints are removed.

3 Preliminaries

Consider reasoning with verifiable rewards. Given a question q, a reasoner policy π_θ samples a trajectory $\tau \sim \pi_\theta(\cdot \mid q)$, and a verifier returns a binary reward $r(\tau) \in \{0,1\}$ indicating whether the final answer is correct. We denote the per-question success probability by $p_\theta(q) = \Pr_{\tau\sim\pi_\theta(\cdot|q)}[r(\tau)=1]$. Our goal is to improve this no-hint success probability using hints available only during training.

GRPO and advantage collapse. Group Relative Policy Optimization (Shao et al., 2024) samples $G \geq 2$ independent trajectories from $\pi_\theta(\cdot \mid q)$, obtains rewards $r_i = r(\tau_i)$, and computes within-group normalized advantages:

$$A_i = \frac{r_i - \bar{r}}{\mathrm{std}(r_{1:G}) + \epsilon}, \qquad \bar{r} = \frac{1}{G}\sum_{i=1}^{G} r_i, \tag{1}$$

where $\epsilon > 0$ is a numerical stabilizer. When all rollouts receive the same reward, every advantage vanishes and the group contributes no reward-based policy-gradient signal.

For a question with success probability $p = p_\theta(q)$, the probability that a group contains both correct and incorrect outcomes is

$$s(p;\, G) = 1 - p^G - (1-p)^G. \tag{2}$$

This *non-degenerate probability* is maximized at $p = \frac{1}{2}$ and satisfies $s(p;G) \approx Gp$ for small p and fixed G. Hard questions with low success probabilities therefore frequently yield all-incorrect groups. Hinting can recover mixed outcomes, but whether the resulting training signal improves no-hint reasoning is a separate question.

4 HiLL: Hint Learning for Reinforcement Learning

HiLL learns training-time hints to recover learning signals from hard questions and improve no-hint reasoning. It intervenes only on all-incorrect GRPO groups, which produce no gradient under binary rewards. As illustrated in Figure 1 and summarized in Algorithm 1, the hinter generates candidate hints online, the reasoner re-samples under each hinted input, and hints are scored by both signal creation and signal transfer to guide hint selection and hinter training.

Algorithm 1 HiLL: Hint Learning for Reinforcement Learning

Require: Batch $\mathcal{B}$, reasoner π_θ, hinter $\mathcal{H}_\phi$, group size G, hint candidates M, temperature T, failure reward R_{fail}

For each $q \in \mathcal{B}$, sample G rollouts from $\pi_\theta(\cdot \mid q)$ and compute rewards
Identify all-incorrect questions $\mathcal{I} = \{q \in \mathcal{B} : \sum_{i=1}^{G} r_i(q) = 0\}$
for each $q \in \mathcal{I}$ **do**
  Form hinter input $c = (q,\, \tau_k,\, z^\star)$ using an incorrect rollout and reference solution
  Sample M candidate hints $\{h_j\}_{j=1}^{M} \sim \mathcal{H}_\phi(\cdot \mid c)$
  for each candidate hint h_j **do**
    if h_j is invalid **then**
      Set $R(q, h_j) \leftarrow R_{\text{fail}}$
    else
      Sample G rollouts from $\pi_\theta(\cdot \mid q{+}h_j)$ and compute rewards
      Estimate $\hat{p}_h = \frac{1}{G} \sum_{i=1}^{G} r_i^{(h_j)}$
      if $0 < \hat{p}_h < 1$ **then**
        Estimate $\hat{\rho}_c(q, h_j)$ from correct hinted trajectories via Eq. 8
      end if
      Compute $R(q, h_j)$ using Eq. 9
    end if
  end for
  Select $h^\star = \arg\max_{h_j} R(q, h_j)$
  if $R(q, h^\star) > 0$ **then**
    $(q,\, \{\tau_i\}_{i=1}^{G}) \leftarrow (q{+}h^\star,\, \{\tau_i^{h^\star}\}_{i=1}^{G})$
  end if
end for
Update reasoner π_θ using Eq. 10 on the final selected groups
Update hinter $\mathcal{H}_\phi$ using Eq. 11, where M hints per $q \in \mathcal{I}$ form the group

4.1 Failure-Conditioned Hint Generation

Given a training batch $\mathcal{B}$, the reasoner samples G rollouts $\{\tau_i\}_{i=1}^{G}$ per question and identifies the all-incorrect subset $\mathcal{I} = \{q \in \mathcal{B} : \sum_{i=1}^{G} r_i(q) = 0\}$. For each $q \in \mathcal{I}$, the hinter $\mathcal{H}_\phi$ generates M candidate hints conditioned on q, a randomly selected incorrect rollout τ_k, and a reference solution $z^\star$ available only during training:

$$\{h_j\}_{j=1}^{M} \;\sim\; \mathcal{H}_\phi(\cdot \mid q,\, \tau_k,\, z^\star). \tag{3}$$

The incorrect rollout exposes the reasoner's current error, while the reference solution guides targeted hint generation. We prompt the hinter to analyze the failure and produce a concise hint, appending only the extracted hint to q, as detailed in Appendix B.

A hint is invalid if it (i) violates the required output format, (ii) fails the final-answer leakage check, or (iii) causes $q{+}h$ to exceed the maximum input length. Validation details are provided in Appendix C. Invalid hints receive a fixed penalty $R_{\text{fail}} < 0$ as defined in Section 4.4.

4.2 Hinted Rollouts and Signal Creation

For each valid candidate hint h_j, the reasoner re-samples G rollouts from $\pi_\theta(\cdot \mid q{+}h_j)$ and computes rewards. Let $\hat{p}_h = \frac{1}{G} \sum_{i=1}^{G} r_i^{(h_j)}$ denote the estimated success rate under the hinted input. A hint recovers nonzero GRPO advantages when the resulting group is non-degenerate, i.e., contains both correct and incorrect rollouts. We use $s(\hat{p}_h;\, G) = 1{-}\hat{p}_h^{G}{-}(1{-}\hat{p}_h)^{G}$ from Eq. 2 as the signal-creation score, which vanishes when $\hat{p}_h \in \{0, 1\}$ and peaks at $\hat{p}_h = \frac{1}{2}$.

However, signal creation alone is not sufficient: a hint may produce mixed outcomes simply by making the problem much easier, without producing trajectories that remain useful once the hint is removed. We therefore introduce a second criterion that measures how strongly correct hinted trajectories depend on the hint.

4.3 HINT RELIANCE AND SIGNAL TRANSFER

If correct hinted trajectories depend strongly on the hint, training on them may reinforce this dependence rather than improve no-hint reasoning. For example, "try factoring the left-hand side" suggests a strategy, whereas "note that $x^2-5x+6 = (x-2)(x-3)$" performs the factorization directly. We measure this dependence by comparing the same trajectory under hinted and original inputs.

Hint reliance. We define the *hint reliance* of a trajectory τ sampled under $q{+}h$ as

$$\rho(\tau;\, q, h) \;=\; \log \pi_\theta(\tau \mid q{+}h) \;-\; \log \pi_\theta(\tau \mid q). \tag{4}$$

A value near zero indicates similar likelihoods with and without the hint, while a large positive value indicates strong reliance on the hinted input. To compare trajectories of different lengths, we normalize reliance per token and average over correct hinted trajectories:

$$\rho_c(q, h) \;=\; \mathbb{E}_{\tau \sim Q_h}\left[\frac{\rho(\tau;\, q, h)}{|\tau|}\right], \tag{5}$$

where $Q_h = \pi_\theta(\cdot \mid q{+}h, r = 1)$ denotes the distribution of correct hinted trajectories. The following result relates hint reliance to hinted and no-hint success probabilities.

Proposition 1 (Transfer bound)**.** *For a question–hint pair (q, h) and fixed reasoner parameters θ, let $P_h(\tau) = \pi_\theta(\tau \mid q{+}h)$ and $P(\tau) = \pi_\theta(\tau \mid q)$ denote the rollout distributions under the hinted and original inputs. Their success probabilities are $p_h = P_h(r(\tau) = 1) > 0$ and $p = P(r(\tau) = 1) > 0$, respectively. For correct hinted trajectories drawn from $Q_h = P_h(\cdot \mid r = 1)$, let $L = |\tau|$ denote their length. Define the harmonic mean length ℓ_{H} and the squared coefficient of variation of inverse length $c^2_{1/L}$ under Q_h as*

$$\ell_{\mathrm{H}} = \left(\mathbb{E}_{Q_h}[L^{-1}]\right)^{-1}, \qquad c^2_{1/L} = \frac{\mathrm{Var}_{Q_h}(L^{-1})}{\left(\mathbb{E}_{Q_h}[L^{-1}]\right)^2}. \tag{6}$$

Assume these trajectories have positive, finite lengths and positive probability under P, with $\mathbb{E}_{Q_h}[|\rho(\tau;\, q, h)|/L] < \infty$. Then the no-hint success probability satisfies

$$p \;\geq\; \frac{p_h}{1 + c^2_{1/L}} \exp(-\ell_{\mathrm{H}} \rho_c(q, h)), \tag{7}$$

where $\rho_c(q, h)$ is the normalized population average defined in Eq. 5.

For a given hinted success probability and successful-trajectory length distribution, lower hint reliance yields a larger lower bound on no-hint success. This motivates learning hints that recover successful trajectories while limiting their dependence on the hinted input. The proof is provided in Appendix A.

Empirical estimation. We estimate $\rho_c(q, h)$ in Eq. 5 by averaging the same per-token reliance over the correct rollouts $\mathcal{C}$ from a mixed-outcome hinted group:

$$\hat{\rho}_c(q, h) \;=\; \frac{1}{|\mathcal{C}|} \sum_{\tau \in \mathcal{C}} \frac{\rho(\tau;\, q, h)}{|\tau|}. \tag{8}$$

Computing $\hat{\rho}_c$ requires scoring each correct hinted trajectory under both $q{+}h$ and q using two teacher-forced forward passes of π_θ.

4.4 TRANSFER-WEIGHTED HINTER REWARD

Motivated by the transfer bound, we combine signal creation and hint reliance to define the reward for hinter training:

$$R(q, h) \;=\; \begin{cases} R_{\mathrm{fail}}, & \text{if } h \text{ is invalid,} \\ \underbrace{s(\hat{p}_h;\, G)}_{\text{signal creation}} \cdot \underbrace{\exp\!\left(-\dfrac{\max\big(\hat{\rho}_c(q, h),\, 0\big)}{T}\right)}_{\text{transfer weight}}, & \text{otherwise,} \end{cases} \tag{9}$$

where $T > 0$ is a temperature parameter controlling the strength of the reliance penalty. The first term $s(\hat{p}_h;\, G)$ from Eq. 2 rewards hints that produce mixed-outcome groups. Valid hints producing all-correct or all-incorrect groups receive zero reward without estimating reliance. The second term uses $\hat{\rho}_c(q, h)$ from Eq. 8 to discount hints with high reliance. It equals 1 for nonpositive reliance and decays as positive reliance increases, without further rewarding negative reliance. Their product encourages hints that recover nonzero GRPO signal and support transfer to the no-hint setting.

4.5 JOINT REASONER-HINTER OPTIMIZATION

After scoring all candidates, we select the best hint $h^\star = \arg\max_{h_j} R(q, h_j)$ for each $q \in \mathcal{I}$. If $R(q, h^\star) > 0$, the original group is replaced by the hinted group $\{\tau_i^{h^\star}\}_{i=1}^G$ sampled under $q{+}h^\star$. Otherwise, the original input and group are kept. Both policies are updated after candidate scoring.

Reasoner update. Each question $q \in \mathcal{B}$ is paired with an input x_q, either q or $q{+}h^\star$, and its corresponding rollout group. Both rollout sampling and log-probability computation condition on the same input x_q. The reasoner is updated using GRPO, with the simplified policy-gradient loss

$$\mathcal{L}_{\mathrm{R}}(\theta) \;=\; -\mathbb{E}_{q\sim\mathcal{B}}\left[\sum_{i=1}^{G} A_i \sum_{t=1}^{|\tau_i|} \log \pi_\theta\big(y_{i,t} \mid x_q,\, y_{i,<t}\big)\right], \tag{10}$$

where $y_{i,t}$ is the t-th token of τ_i, and A_i are within-group normalized advantages computed from the verifier rewards as in Eq. 1. For simplicity, both update equations omit the PPO-style clipped surrogates used in the implementation.

Hinter update. For each $q \in \mathcal{I}$, the M candidate hints form a GRPO group with rewards $\{R(q, h_j)\}_{j=1}^M$. The corresponding simplified loss is

$$\mathcal{L}_{\mathrm{H}}(\phi) \;=\; -\mathbb{E}_{q\sim\mathcal{I}}\left[\sum_{j=1}^{M} A_j \sum_{t=1}^{|h_j|} \log \mathcal{H}_\phi(h_{j,t} \mid c_q,\, h_{j,<t})\right], \tag{11}$$

where A_j are group-normalized advantages from the hint rewards, and $c_q = (q, \tau_k, z^\star)$. Invalid candidates participate through R_{fail}, allowing the hinter to learn from differences in candidate validity and hint quality. Groups with identical rewards yield zero relative advantages. The hinter update is skipped when $\mathcal{I}$ is empty.

Co-training dynamics. The reasoner and hinter have separate parameters so that each can be optimized with its own objective. As the all-incorrect subset $\mathcal{I}$ changes during training, the hinter adapts to the reasoner's evolving failures. At inference, only the reasoner is used, without hints.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models. We evaluate our HiLL framework on two models as the reasoner policy: Llama-3.2-3B-Instruct (Meta AI, 2024) and Qwen2.5-7B-Instruct (Yang et al., 2025b). The hinter is initialized from Qwen3-4B-Instruct (Yang et al., 2025a) for both reasoners. We also evaluate a same-initialization configuration with both reasoner and hinter policies initialized from the same model. Both policies are trained jointly via GRPO as described in Section 4.

Training data. We use the same 15k-prompt subset of OpenR1-Math-220k (Hugging Face, 2025) curated by Liao et al. (2026), drawn from NuminaMath 1.5 (LI et al., 2024) together with ground-truth answers and reference solutions. No filtering is applied, so questions span a wide difficulty range.

Evaluation. We report Average@16 accuracy on six math reasoning benchmarks: AIME24 (Mathematical Association of America, 2024), AIME25 (Mathematical Association of America, 2025), AMC23 (LI et al., 2024), MATH-500 (Hendrycks et al., 2021), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), and two non-math benchmarks: GPQA-diamond (Rein et al., 2024), MMLU-Pro (Wang et al., 2024) for generalization assessment. All reasoner models are evaluated with temperature 0.6, top-p 0.95, and maximum response length $8{,}192$. The hinter is never used at evaluation.

Baselines. We compare against the following baselines trained on the same 15k prompts: (1) Base, the initial reasoner; (2) GRPO (Shao et al., 2024), standard RL without hints; (3) LUFFY (Yan et al., 2025), which replaces one on-policy rollout with an off-policy trajectory from DeepSeek-R1; (4) Scaf-GRPO (Zhang et al., 2025d), which augments all-incorrect groups with hints from an external teacher model; and (5) SAGE (Liao et al., 2026), on-policy hinting with self-generated hints conditioned on reference solutions. The implementation builds on the official codebase of Liao et al. (2026), and

Table 1: Main results on in-distribution and out-of-distribution benchmarks, using a Qwen3-4B-Instruct hinter for HiLL. Best results are in bold and second-best are underlined.

Method	In-distribution						Out-of-distribution		
	AIME24 / 25	AMC23	MATH-500	Minerva	Olympiad	Avg.	GPQA	MMLU-Pro	Avg.
Llama-3.2-3B-Instruct									
Base	6.5 / 0.6	22.8	44.7	17.8	14.2	17.8	17.9	27.0	22.5
GRPO	8.9 / 0.9	29.8	53.2	18.7	21.2	22.1	24.7	38.4	31.5
LUFFY	4.4 / 0.4	18.6	38.9	14.3	11.9	14.7	16.0	26.7	21.4
Scaf-GRPO	7.7 / **2.3**	28.8	51.7	19.4	19.5	21.5	24.1	38.0	31.0
SAGE	**9.5** / 1.5	32.8	55.0	20.3	21.5	23.4	27.5	38.7	33.1
HiLL$_{\text{w/o TW}}$	6.9 / **2.3**	32.5	56.7	21.6	22.4	23.7	**28.0**	41.0	34.5
HiLL	8.5 / 1.7	**34.8**	**57.7**	**21.7**	**23.2**	**24.6**	27.8	**42.7**	**35.3**
Qwen2.5-7B-Instruct									
Base	13.8 / 6.7	53.4	75.7	38.1	39.2	37.8	37.1	56.4	46.7
GRPO	15.0 / 13.5	55.5	79.2	39.1	44.5	41.1	37.2	57.6	47.4
LUFFY	**17.1** / 13.5	55.2	81.3	39.0	44.2	41.7	38.1	59.1	48.6
Scaf-GRPO	14.6 / 12.7	58.8	78.0	39.8	42.0	41.0	36.6	58.4	47.5
SAGE	16.0 / 12.5	60.3	80.0	39.3	45.9	42.3	38.0	59.3	48.6
HiLL$_{\text{w/o TW}}$	15.2 / 14.4	60.5	80.5	39.8	45.7	42.7	38.6	59.9	49.2
HiLL	16.9 / **15.3**	**63.0**	**81.8**	**40.1**	**48.1**	**44.2**	**40.4**	**61.6**	**51.0**

Table 2: Same-initialization and matched-compute comparison with both HiLL policies initialized from Llama-3.2-3B-Instruct. Results are mean Average@16. Avg. is the eight-benchmark mean, and Time is per-step wall-clock time relative to GRPO with $G = 8$.

Method	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	GPQA	MMLU-Pro	Avg.	Time
GRPO ($G = 8$)	8.86	0.92	29.77	53.22	18.73	21.20	24.66	38.39	24.47	1.00×
GRPO ($G = 32$)	10.11	**1.88**	32.34	54.60	19.66	21.78	25.57	39.64	25.70	3.89×
SAGE	9.48	1.46	32.81	54.95	20.25	21.50	27.51	38.69	25.83	4.15×
HiLL$_{\text{w/o TW}}$	8.23	0.94	30.47	54.43	20.28	20.71	25.02	39.42	24.94	3.80×
HiLL	**11.04**	1.67	**35.16**	**56.09**	**21.33**	**22.66**	**28.18**	**41.02**	**27.14**	3.88×

we report baseline results from their paper. We additionally report HiLL$_{\text{w/o TW}}$, which removes the transfer weight from the hinter reward in Eq. 9, using only the signal-creation score.

Implementation details. We use verl (Sheng et al., 2025) for RL training and vLLM (Kwon et al., 2023) for rollout generation and all experiments run on 8×B200 GPUs. For reasoner policy training in all methods, we follow DAPO (Yu et al., 2025) to disable the KL penalty and apply Clip-Higher with ϵ_{low}=0.2, ϵ_{high}=0.28. The reasoner is trained for 500 steps with a batch size of 128, G=8 rollouts per prompt, and learning rate 10^{-6}. The maximum prompt and response lengths are set to 2,048 and 8,192 tokens. We evaluate every 50 steps and select the reasoner checkpoint with the best Average@16 validation accuracy to report the results. For hinter policy training in HiLL, the hinter generates M=4 candidate hints per all-incorrect question with a maximum prompt and response length being 10,240 and 1,024 tokens, transfer temperature T=0.3, failure penalty R_{fail}=−0.2, and the same learning rate and clipping as the reasoner. We use Ray (Moritz et al., 2018) to co-locate reasoner and hinter models on the same GPU nodes. Since the co-training pipeline executes each phase sequentially as in Algorithm 1, Ray idles one policy while the other is active, letting both FSDP-sharded models share the same devices without extra memory overhead compared to baseline training methods. Prompts used for reasoner and hinter are provided in Appendix B.

5.2 Results and Analysis

HiLL consistently boosts reasoner performance compared to baselines. Table 1 shows that HiLL achieves the highest average accuracy on in-distribution math reasoning tasks on both Llama-3.2-3B-Instruct and Qwen2.5-7B-Instruct. The gains over GRPO extend across AMC23, MATH-500, Minerva Math, and OlympiadBench, rather than being concentrated on a single benchmark. Among hint-based baselines, SAGE improves over standard GRPO on both backbones, demonstrating the benefit of adaptive guidance during training. HiLL further outperforms both SAGE and Scaf-GRPO

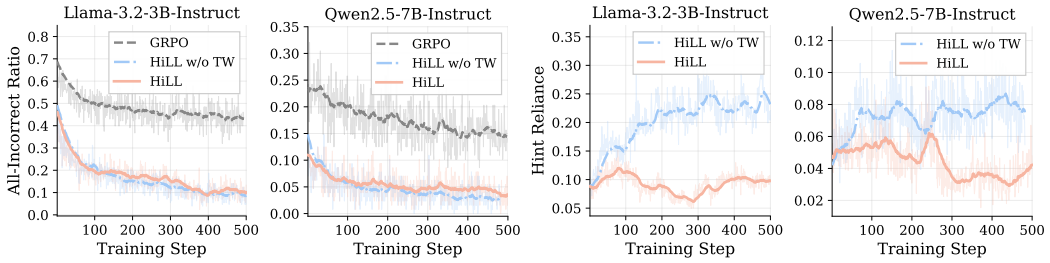

Figure 2: All-incorrect ratio (left two) and hint reliance (right two) over training steps. Both HiLL variants substantially reduce the fraction of degenerate all-incorrect groups compared to GRPO. Transfer weighting (HiLL vs. HiLL$_{\text{w/o TW}}$) keeps hint reliance consistently lower throughout training.

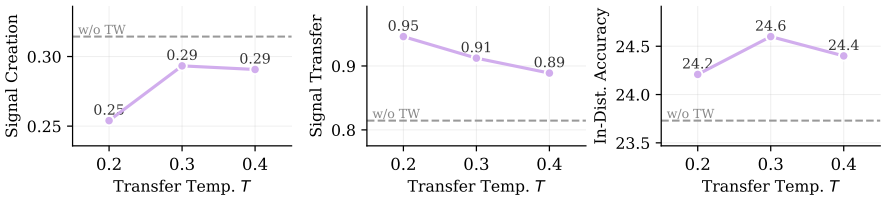

Figure 3: Effect of transfer temperature T on average signal creation, signal transfer, and in-distribution accuracy for HiLL with Llama-3.2-3B-Instruct. Dashed lines show HiLL$_{\text{w/o TW}}$.

in average accuracy, supporting the value of learning hints with transfer in mind. The gains also extend to GPQA and MMLU-Pro despite training exclusively on math data, with HiLL achieving the highest out-of-distribution average on both backbones. Since the hinter is absent at evaluation, these improvements reflect the reasoner's performance without guidance and suggest that the benefits of hint learning extend beyond mathematical reasoning.

HiLL remains effective with the same initialization for both policies. To examine whether these gains depend on the hinter initialization, we replace the Qwen3-4B hinter with a separate policy initialized from Llama-3.2-3B-Instruct, matching the reasoner. Table 2 shows that HiLL continues to outperform GRPO with $G = 8$ and SAGE on every evaluated benchmark. Its average math accuracy is also similar to that of the Qwen3-4B-hinter configuration in Table 1. These results show that the hinter can learn effective guidance from the reasoner's failures and reference solutions without requiring a larger or different-family initialization.

The gains of HiLL persist at comparable training cost. The same comparison also allows us to examine how HiLL uses its additional training computation. Candidate hint generation, hinted rollouts, reliance scoring, and hinter updates all add work beyond standard GRPO. We compare this allocation with increasing the number of unhinted rollouts, using GRPO with $G = 32$ instead of $G = 8$ while keeping the number of training steps fixed. As shown in Table 2, this brings GRPO's per-step wall-clock time close to that of HiLL. Although additional rollouts improve GRPO's accuracy, HiLL retains a higher overall average and outperforms this baseline on most benchmarks. HiLL also outperforms SAGE at slightly lower per-step cost. These results show that the benefits of transfer-aware hint learning persist when compared with spending a similar training budget on additional unhinted sampling. Standard deviations across training runs and additional statistical analysis are provided in Appendix D.1.

Transfer weighting keeps hint reliance low while recovering GRPO signals. To examine the contribution of transfer weighting to these gains, we compare HiLL with HiLL$_{\text{w/o TW}}$, which rewards signal creation alone. Tables 1 and 2 show that transfer weighting improves average no-hint accuracy across the evaluated reasoner and hinter configurations. Figure 2 provides a closer view through the all-incorrect ratio and hint reliance over training. Both HiLL variants substantially reduce the fraction of all-incorrect groups in the reasoner batch compared to standard GRPO. Their reliance curves differ, however: without transfer weighting, the reasoner's dependence on hints increases during training, whereas HiLL maintains lower reliance among correct hinted trajectories. Together with the accuracy improvements, these results support learning hints that do more than recover mixed outcomes.

Table 3: Example hints generated by HiLL and HiLL$_{\text{w/o TW}}$ on the same question. HiLL provides conceptual strategies while HiLL$_{\text{w/o TW}}$ tends to set up the computation directly.

Question (excerpt)	**Hint from HiLL**	**Hint from HiLL$_{\text{w/o TW}}$**
A line segment AB of length 1 moves on the x-axis. Point $P(0,1)$ is connected to A by a line, and point $Q(1,2)$ is connected to B by a line. Find the equation of the trajectory of the intersection point R.	Consider expressing the intersection in terms of a parameter a, then eliminate it to reveal the hidden trajectory. **(18 words, hinted pass rate 0.13)**	Let $A=(a,0)$ and $B=(a{+}1,0)$ for a real parameter $a \neq 0$. Write the equation of line PA, where $P=(0,1)$, and the equation of line QB, where $Q=(1,2)$, both in terms of a. Find the coordinates (x,y) of their intersection point R as a function of a.... **(108 words, hinted pass rate 0.38)**
In trapezoid $ABCD$, $AB \parallel CD$, AB=33, CD=21, the non-parallel sides are 10 and 14, and $\angle A$, $\angle B$ are both acute. Find the length of the shorter diagonal.	The shorter diagonal emerges from a triangle where 14, 12, and 10 form a critical configuration, with the angle between them yielding a result just above 25. **(27 words, hinted pass rate 0.13)**	Draw a line $DE \parallel BC$ from point D to side AB, intersecting AB at point E. This creates triangle ADE with $DE=10$, $AE=12$, and $AD=14$. Use the Law of Cosines in triangle ADE to find $\cos A$. Then apply the same angle $\angle A$ in ... **(97 words, hinted pass rate 0.50)**

Transfer temperature controls the signal creation–transfer trade-off. The preceding ablation removes transfer weighting entirely. We next vary the temperature T in Eq. 9 to examine how the strength of the reliance penalty affects hint learning. Figure 3 reports the average reduction in the all-incorrect ratio after hinting, the reliance-based transfer score $\exp(-\hat{\rho}_c)$ averaged over training, and in-distribution accuracy on Llama-3.2-3B-Instruct. A smaller T imposes a stronger penalty, yielding a higher transfer score but reducing signal creation. A larger T relaxes the penalty but yields a lower transfer score. The choice of T=0.3 balances signal creation and transfer, achieving the highest accuracy among the tested settings. All three tested values outperform HiLL$_{\text{w/o TW}}$, showing that the benefit of transfer weighting is not limited to a single temperature.

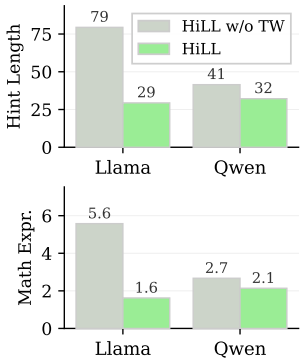

Figure 4: Hint length and math expressions per hint.

HiLL learns more concise hints with less explicit computation. Beyond its effects on training dynamics and accuracy, transfer weighting also shapes the generated hints. Figure 4 reports the average hint length in words and the average number of math expressions per hint, both computed from hints collected at 50-step intervals throughout training. HiLL produces shorter hints with fewer math expressions than HiLL$_{\text{w/o TW}}$ on both backbones. Table 3 illustrates this difference with two examples: HiLL hints summarize a strategy or geometric configuration, whereas HiLL$_{\text{w/o TW}}$ hints spell out more intermediate steps by defining coordinates, writing equations, or specifying theorem applications. This pattern is consistent with the transfer-weighted objective in Eq. 9, which penalizes hints whose successful trajectories depend strongly on the hinted input.

6 Conclusion

In this paper, we introduced HiLL, a framework that learns hints by considering both signal creation and signal transfer to no-hint reasoning. HiLL trains a hinter alongside the reasoner, generating hints conditioned on the reasoner's current failures and adapting its guidance as training progresses. We introduce hint reliance to measure how strongly correct hinted trajectories depend on the hint and derive a theoretical transfer bound relating this quantity to hinted and no-hint success probabilities. Motivated by this result, we define a transfer-weighted hinter reward that favors informative GRPO groups while limiting hint dependence. Experiments across eight benchmarks and two backbone LLMs show that HiLL outperforms GRPO and hint-based baselines, with gains that persist in same-initialization and compute-matched comparisons. Our analysis further shows that transfer weighting reduces hint reliance, improves no-hint performance, and produces shorter hints with less explicit computation. Together, these results demonstrate the value of adaptive and transfer-aware hint learning for reinforcement learning.

AI USE STATEMENT

We used generative AI tools to assist with drafting and editing the manuscript, including improving grammar, clarity, organization, and presentation. All AI-assisted content were reviewed and verified by the authors.

REFERENCES

Justin Chih-Yao Chen, Becky Xiangyu Peng, Prafulla Kumar Choubey, Kung-Hsiang Huang, Jiaxin Zhang, Mohit Bansal, and Chien-Sheng Wu. Nudging the boundaries of llm reasoning. *arXiv preprint arXiv:2509.25666*, 2025.

Yang Fan, Fei Tian, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. Learning to teach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HJewuJWCZ.

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

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, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 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/.

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

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Jeonghye Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Rebellious student: Reversing teacher signals for reasoning exploration with self-distilled rlvr. *arXiv preprint arXiv:2605.10781*, 2026.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626, 2023.

Thanh-Long V. Le, Myeongho Jeon, Kim Vu, Viet Dac Lai, and Eunho Yang. No prompt left behind: Exploiting zero-variance prompts in LLM reinforcement learning via entropy-guided advantage shaping. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kiXFIESZKv.

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

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. https://huggingface.co/AI-MO/NuminaMath-CoT, 2024.

Ziheng Li, Zexu Sun, Jinman Zhao, Erxue Min, Yongcheng Zeng, Hui Wu, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Xu Chen, et al. Staying in the sweet spot: Responsive reasoning evolution via capability-adaptive hint scaffolding. *arXiv preprint arXiv:2509.06923*, 2025.

Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, and Jiang Bian. Self-hinting language models enhance reinforcement learning. *arXiv preprint arXiv:2602.03143*, 2026.

Ziru Liu, Cheng Gong, Xinyu Fu, Yaofang Liu, Ran Chen, Shoubo Hu, Suiyun Zhang, Rui Liu, Qingfu Zhang, and Dandan Tu. Ghpo: Adaptive guidance for stable and efficient llm reinforcement learning. *arXiv preprint arXiv:2507.10628*, 2025.

Mathematical Association of America. AIME Problems and Solutions, 2024. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Mathematical Association of America. AIME Problems and Solutions, 2025. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Meta AI. Introducing meta llama 3: The most capable openly available llm to date. https://ai.meta.com/blog/meta-llama-3/, 2024.

Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX symposium on operating systems design and implementation (OSDI 18)*, pages 561–577, 2018.

Youssef Mroueh. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025.

Yuxiao Qu, Amrith Setlur, Virginia Smith, Ruslan Salakhutdinov, and Aviral Kumar. Pope: Learning to reason on hard problems via privileged on-policy exploration. *arXiv preprint arXiv:2601.18779*, 2026a.

Yuxiao Qu, Anikait Singh, Yoonho Lee, Amrith Setlur, Ruslan Salakhutdinov, Chelsea Finn, and Aviral Kumar. RLAD: Training LLMs to discover abstractions for solving reasoning problems. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=fvJPjCioeR.

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.

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

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.

Yuda Song, Lili Chen, Fahim Tajwar, Remi Munos, Deepak Pathak, J Andrew Bagnell, Aarti Singh, and Andrea Zanette. Expanding the capabilities of reinforcement learning via text feedback. *arXiv preprint arXiv:2602.02482*, 2026.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.

Wei Xiong, Chenlu Ye, Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, Jiang Bian, Nan Jiang, and Tong Zhang. Reinforce-ada: An adaptive sampling framework under non-linear rl objectives. *arXiv preprint arXiv:2510.04996*, 2025.

Yixuan Even Xu, Yash Savani, Fei Fang, and J Zico Kolter. Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning. *arXiv preprint arXiv:2504.13818*, 2025.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025.

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

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and et al. Qwen2.5 technical report, 2025b. URL `https://arxiv.org/abs/2412.15115`.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026.

Jiarui Yao, Yifan Hao, Hanning Zhang, Hanze Dong, Wei Xiong, Nan Jiang, and Tong Zhang. Optimizing chain-of-thought reasoners via gradient variance minimization in rejection sampling and rl. *arXiv preprint arXiv:2505.02391*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Xinlei Yu, Gen Li, Qingyi Si, Guibin Zhang, Yuqi Xu, Congcong Wang, Shuai Dong, Kaiwen Tuo, Xiangyu Zeng, Kaituo Feng, et al. Dopd: Dual on-policy distillation. *arXiv preprint arXiv:2606.30626*, 2026.

Feng Zhang, Zezhong Tan, Xinhong Ma, Ziqiang Dong, Xi Leng, Jianfei Zhao, Xin Sun, Yang Yang, and Guanjun Jiang. Adhint: Adaptive hints with difficulty priors for reinforcement learning. *arXiv preprint arXiv:2512.13095*, 2025a.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. Stephint: Multi-level stepwise hints enhance reinforcement learning to reason. *arXiv preprint arXiv:2507.02841*, 2025b.

Xiaoying Zhang, Yipeng Zhang, Hao Sun, Kaituo Feng, Chaochao Lu, Chao Yang, and Helen Meng. Critique-grpo: Advancing llm reasoning with natural language and numerical feedback. *arXiv preprint arXiv:2506.03106*, 2025c.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. Scaf-grpo: Scaffolded group relative policy optimization for enhancing llm reasoning. *arXiv preprint arXiv:2510.19807*, 2025d.

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.

Haizhong Zheng, Yang Zhou, Brian R Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts. *arXiv preprint arXiv:2506.02177*, 2025.

A Proof of Proposition 1

We use the notation and assumptions of Proposition 1. Fix a question–hint pair (q, h) and reasoner parameters θ. Let $P_h(\tau) = \pi_\theta(\tau \mid q{+}h)$ and $P(\tau) = \pi_\theta(\tau \mid q)$ denote the hinted and no-hint rollout distributions. For the success event $\mathcal{S} = \{\tau : r(\tau) = 1\}$, their success probabilities are $p_h = P_h(\mathcal{S}) > 0$ and $p = P(\mathcal{S}) > 0$. The corresponding distributions of correct trajectories are

$$Q_h = P_h(\cdot \mid \mathcal{S}), \qquad Q = P(\cdot \mid \mathcal{S}).$$

For a trajectory of length $L(\tau) = |\tau|$, the normalized population reliance defined in Eq. 5 is

$$\rho_c(q, h) = \mathbb{E}_{Q_h}\left[\frac{\rho(\tau;\, q, h)}{L(\tau)}\right].$$

Every trajectory in the support of Q_h has positive probability under P and positive finite token length. We also assume $\mathbb{E}_{Q_h}[|\rho(\tau;\, q, h)|/L] < \infty$.

Proof. For each trajectory with $Q_h(\tau) > 0$, we have $P_h(\tau) = p_h Q_h(\tau)$ and $P(\tau) = pQ(\tau)$. Thus,

$$\rho(\tau;\, q, h) = \log \frac{p_h}{p} + \log \frac{Q_h(\tau)}{Q(\tau)}. \tag{12}$$

Inverse-length reweighting. Let $\ell_{\mathrm{H}} = (\mathbb{E}_{Q_h}[L^{-1}])^{-1}$ be the harmonic mean length of correct hinted trajectories. Since L is a positive finite token count, $0 < \mathbb{E}_{Q_h}[L^{-1}] \le 1$, so ℓ_{H} is finite. Define the inverse-length-reweighted distribution

$$Q_h^\dagger(\tau) = \frac{\ell_{\mathrm{H}}}{L(\tau)} Q_h(\tau). \tag{13}$$

This is a probability distribution because

$$\sum_{\tau \in \mathcal{S}} Q_h^\dagger(\tau) = \ell_{\mathrm{H}} \mathbb{E}_{Q_h}[L^{-1}] = 1.$$

Under this distribution, the expected trajectory-level reliance is

$$\mathbb{E}_{Q_h^\dagger}[\rho(\tau;\, q, h)] = \ell_{\mathrm{H}} \mathbb{E}_{Q_h}\left[\frac{\rho(\tau;\, q, h)}{L(\tau)}\right] = \ell_{\mathrm{H}} \rho_c(q, h). \tag{14}$$

Define the reweighting correction

$$\Delta_L = D_{\mathrm{KL}}(Q_h^\dagger \| Q_h) = \mathbb{E}_{Q_h^\dagger}\left[\log \frac{\ell_{\mathrm{H}}}{L}\right]. \tag{15}$$

The density ratio $Q_h^\dagger / Q_h = \ell_{\mathrm{H}}/L$ is bounded, so Δ_L is finite. The assumed integrability of ρ/L also gives $\mathbb{E}_{Q_h^\dagger}[|\rho|] < \infty$, ensuring that the expectations below are well defined.

Transfer identity. Taking the expectation of Eq. 12 under $Q_h^\dagger$ gives

$$\begin{aligned}
\ell_{\mathrm{H}} \rho_c(q, h) &= \log \frac{p_h}{p} + \mathbb{E}_{Q_h^\dagger}\left[\log \frac{Q_h}{Q}\right] \\
&= \log \frac{p_h}{p} + \mathbb{E}_{Q_h^\dagger}\left[\log \frac{Q_h^\dagger}{Q} - \log \frac{Q_h^\dagger}{Q_h}\right] \\
&= \log \frac{p_h}{p} + D_{\mathrm{KL}}(Q_h^\dagger \| Q) - \Delta_L.
\end{aligned}$$

Therefore,

$$\ell_{\mathrm{H}} \rho_c(q, h) + \Delta_L = \log \frac{p_h}{p} + D_{\mathrm{KL}}(Q_h^\dagger \| Q). \tag{16}$$

The nonnegativity of the final KL divergence yields

$$p \ge p_h \exp(-\ell_{\mathrm{H}} \rho_c(q, h) - \Delta_L). \tag{17}$$

Equality in Eq. 17 holds if and only if $Q = Q_h^\dagger$.

Length-dispersion bound. Let $W = \ell_{\mathrm{H}}/L$, so that $Q_h^\dagger = WQ_h$ and $\mathbb{E}_{Q_h}[W] = 1$. By Jensen's inequality,

$$\begin{aligned}\Delta_L &= \mathbb{E}_{Q_h^\dagger}[\log W] \\ &\le \log \mathbb{E}_{Q_h^\dagger}[W] \\ &= \log \mathbb{E}_{Q_h}[W^2] \\ &= \log \frac{\mathbb{E}_{Q_h}[L^{-2}]}{(\mathbb{E}_{Q_h}[L^{-1}])^2} \\ &= \log\left(1 + \frac{\mathrm{Var}_{Q_h}(L^{-1})}{(\mathbb{E}_{Q_h}[L^{-1}])^2}\right) = \log\left(1 + c_{1/L}^2\right).\end{aligned}$$

Together with the nonnegativity of KL divergence, this establishes

$$0 \le \Delta_L \le \log\left(1 + c_{1/L}^2\right). \tag{18}$$

Substituting Eq. 18 into Eq. 17 gives

$$p \ge \frac{p_h}{1 + c_{1/L}^2} \exp(-\ell_{\mathrm{H}} \rho_c(q,h)),$$

which proves Proposition 1. □

A.1 Bounded Trajectory Lengths

The following corollary bounds the length-dispersion correction using only the range of successful hinted trajectory lengths.

Corollary 1 (Bounded-length transfer)**.** *Under the assumptions of Proposition 1, suppose correct hinted trajectories satisfy $L_{\min} \le L \le L_{\max}$ with probability one under Q_h, where $0 < L_{\min} \le L_{\max} < \infty$. Then*

$$p \ge \frac{4L_{\min}L_{\max}}{(L_{\min} + L_{\max})^2} p_h \exp(-\ell_{\mathrm{H}} \rho_c(q,h)). \tag{19}$$

In particular, if $L \in [(1-\delta)L_0, (1+\delta)L_0]$ for some $L_0 > 0$ and $0 \le \delta < 1$, then

$$p \ge (1 - \delta^2)\, p_h \exp(-\ell_{\mathrm{H}} \rho_c(q,h)). \tag{20}$$

Proof. Let $U = L^{-1}$ and $\mu = \mathbb{E}_{Q_h}[U]$. Then $a \le U \le b$, where $a = L_{\max}^{-1}$ and $b = L_{\min}^{-1}$. The pointwise inequality $(U-a)(U-b) \le 0$ implies

$$\mathbb{E}_{Q_h}[U^2] \le (a+b)\mu - ab.$$

Consequently,

$$\begin{aligned}1 + c_{1/L}^2 &= \frac{\mathbb{E}_{Q_h}[U^2]}{\mu^2} \\ &\le \frac{a+b}{\mu} - \frac{ab}{\mu^2} \\ &\le \frac{(a+b)^2}{4ab} = \frac{(L_{\min} + L_{\max})^2}{4L_{\min}L_{\max}}.\end{aligned}$$

The last inequality follows from $((a+b)\mu - 2ab)^2 \ge 0$. Substituting this bound into Proposition 1 proves Eq. 19. Setting $L_{\min} = (1-\delta)L_0$ and $L_{\max} = (1+\delta)L_0$ gives

$$\frac{4L_{\min}L_{\max}}{(L_{\min} + L_{\max})^2} = 1 - \delta^2,$$

which proves Eq. 20. □

When all correct hinted trajectories have the same length L_0, $\ell_{\mathrm{H}} = L_0$ and $c_{1/L}^2 = 0$, so the bound reduces to $p \ge p_h \exp(-L_0 \rho_c(q,h))$.

B PROMPTS FOR HINTER AND REASONER

This section lists all prompt templates used in HiLL. We do not set a system prompt for the hinter. Note that for the hinted reasoner input, we simply append the hint after the original question without adding explicit bridging prose such as "Here is a hint to help you:" as used in Liao et al. (2026). We empirically observed that such phrasing increases hint reliance by inducing reasoner outputs like "Given the hint, . . . " which are less likely under the no-hint input and thus less transferable.

User Prompt for Hint Generation

You are a Pedagogical Hint Generator for a Mathematical Reasoner.

TASK: GENERATE A MINIMAL HINT

OBJECTIVE
The Reasoner failed the Original Question. Your goal is to generate a minimal but useful hint that helps the Reasoner find the correct logical path. Provide a strategic nudge, not a solution. The Reasoner must still perform the deduction themselves.

INPUTS
1) Original Question:
`{original_question}`

2) Reasoner's Failed Attempt:
`{rollout_responses}`

3) Ground Truth Solution (Hidden from Reasoner; For Your Reference Only):
`{ground_truth_solution}`

GUIDELINES
1. Analyze the failure to identify the missing insight or misconception.
2. Provide a conceptual pointer, such as a relevant theorem, a structural property, an alternative representation, or an intermediate goal.
3. Do not reveal the final answer or any specific numerical values, formulas with substituted numbers, or computed intermediate steps.

OUTPUT FORMAT (Follow strictly)
<analysis>
[Briefly identify the logic breakdown and the necessary insight.]
</analysis>

<hint>
[Your concise hint — 1 to 3 sentences maximum.]
</hint>

System Prompt for Reasoner

Please reason step by step, and put your final answer within \boxed{}.

User Prompt for Reasoner

`{question}`

User Prompt for Reasoner with Hint

```
{question}

{hint}
```

C HINT VALIDATION DETAILS

Each candidate hint passes through three programmatic gates. A hint failing any gate receives the failure penalty R_{fail}.

Format gate. We extract the hint by matching the pattern `<hint>...</hint>` in the hinter's output with a case-insensitive multi-line regex. The hint is rejected if no match is found or the captured content is empty after whitespace stripping.

Final-answer leakage gate. Let $a^\star$ denote the gold final answer string. The hint is rejected if it contains the literal substring `\boxed{`$a^\star$`}`, or if $a^\star$ is detected by the following rule:

- If $|a^\star| \leq 2$, the hint is rejected if it contains any of the substrings "`= `$a^\star$", "`equals `$a^\star$", "`is `$a^\star$", "`answer is `$a^\star$", "`answer: `$a^\star$", "`result is `$a^\star$", "`value is `$a^\star$", or "`equal to `$a^\star$" (case-insensitive). This pattern restriction avoids false positives where short numerical answers appear incidentally (e.g., $a^\star$="2" matching "Consider 2 variables").
- If $|a^\star| > 2$, the hint is rejected if $a^\star$ appears as a substring of the hint, matched first case-sensitively (to respect mathematical notation) and then case-insensitively (to catch word-based answers).

Length gate. The hinted prompt (system prompt + question + hint, formatted with the reasoner's chat template) is tokenized and rejected if its length exceeds the maximum prompt length, set to $2{,}048$ tokens in our experiments.

D ADDITIONAL EXPERIMENTAL RESULTS

This section provides additional results for the same-initialization setting, statistical analysis of the matched-compute comparison, and ablations on intervention targets, policy sharing, and the failure penalty.

D.1 SAME-INITIALIZATION RESULTS AND STATISTICAL ANALYSIS

Variation across training runs. Table 4 reports the complete mean and standard deviation for the same-initialization experiments in Table 2. Each method is trained for 500 steps with two independent seeds and evaluated at the final checkpoint using Average@16. HiLL shows a consistent aggregate advantage over SAGE, HiLL$_{\text{w/o TW}}$, and GRPO.

Table 4: Same-initialization results at step 500. Entries are mean $\pm$ standard deviation of Average@16 accuracy over two training runs.

Benchmark	GRPO ($G = 8$)	GRPO ($G = 32$)	SAGE	HiLL$_{\text{w/o TW}}$	HiLL
AIME24	8.86 ± 0.74	10.11 ± 0.15	9.48 ± 0.14	8.23 ± 1.91	11.04 ± 1.47
AIME25	0.92 ± 0.17	1.88 ± 0.00	1.46 ± 0.30	0.94 ± 0.74	1.67 ± 0.00
AMC23	29.77 ± 1.21	32.34 ± 0.66	32.81 ± 0.45	30.47 ± 0.66	35.16 ± 2.65
MATH-500	53.22 ± 0.75	54.60 ± 1.05	54.95 ± 0.30	54.43 ± 0.44	56.09 ± 0.16
Minerva Math	18.73 ± 0.23	19.66 ± 0.41	20.25 ± 0.28	20.28 ± 0.54	21.33 ± 0.13
OlympiadBench	21.20 ± 0.36	21.78 ± 0.88	21.50 ± 0.22	20.71 ± 0.33	22.66 ± 0.27
GPQA	24.66 ± 1.21	25.57 ± 0.08	27.51 ± 0.34	25.02 ± 1.68	28.18 ± 0.98
MMLU-Pro	38.39 ± 1.12	39.64 ± 0.32	38.69 ± 1.46	39.42 ± 0.35	41.02 ± 0.05
Overall	24.47 ± 0.34	25.70 ± 0.33	25.83 ± 0.15	24.94 ± 0.40	27.14 ± 0.39

Matched-compute significance analysis. We further evaluate the same HiLL and GRPO ($G = 32$) runs with Average@64 on five benchmarks and compute paired-bootstrap 95% confidence intervals

Table 5: Average@64 comparison between HiLL and compute-matched GRPO. Δ denotes the HiLL minus GRPO accuracy difference with paired-bootstrap 95% confidence intervals.

Method	AIME24	AIME25	AMC23	MATH-500	Olympiad	Question-pooled	Benchmark-macro
HiLL	10.26	1.72	34.45	55.78	22.45	35.12	24.93
GRPO ($G = 32$)	9.53	1.51	31.82	54.87	21.68	34.25	23.88
Δ (95% CI)	$+0.73$ $[-0.99, +3.05]$	$+0.21$ $[-0.39, +1.09]$	$+2.64$ $[-2.38, +7.77]$	$+0.91$ $[-0.17, +1.98]$	$+0.76$ $[+0.04, +1.46]$	$+0.87$ $[+0.27, +1.48]$	$+1.05$ $[-0.05, +2.24]$

Table 6: Average hint reliance over training in the same-initialization setting.

Method	1–100	101–200	201–300	301–400	401–500
$\text{HiLL}_{\text{w/o TW}}$	0.11	0.10	0.15	0.21	0.20
HiLL	0.03	0.05	0.08	0.09	0.09

Table 7: All-correct-only intervention with Qwen2.5-7B-Instruct at step 200.

Method	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	Avg.
GRPO	13.54	12.08	54.22	78.10	38.69	43.19	39.97
HiLL (all-correct only)	13.96	11.88	55.31	78.00	37.57	44.31	40.17
HiLL	14.38	12.29	57.81	79.07	39.20	45.11	41.31

Table 8: Training dynamics for the all-correct-only intervention.

Steps	Avg. hint tokens	Hint reliance	All-correct $\rightarrow$ mixed (%)
1–50	43.2	0.035	45.3
51–100	31.7	0.017	42.2
101–150	24.2	0.011	36.9
151–200	19.5	0.008	40.2

for the two-seed mean differences. Table 5 shows higher mean accuracy for HiLL on all five benchmarks. The confidence interval excludes zero on OlympiadBench and for the question-pooled comparison, while the benchmark-macro interval narrowly includes zero.

Hint reliance over training. Table 6 compares HiLL and $\text{HiLL}_{\text{w/o TW}}$ in the same-initialization setting. Transfer weighting maintains lower hint reliance throughout training, consistent with the dynamics observed in Figure 2.

D.2 Intervention on All-Correct Groups

HiLL intervenes on all-incorrect groups, where no successful rollout is observed. To examine the alternative all-correct failure mode of advantage collapse, we apply the HiLL intervention only to initially all-correct groups. We use Qwen2.5-7B-Instruct and train all methods for 200 steps with matched step-200 evaluation.

As shown in Table 7, the all-correct-only variant performs similarly to GRPO and below standard HiLL. This suggests that recovering outcome variation on already-solved questions provides little benefit compared with targeting observed failures.

Table 8 shows that hinted resampling converts a substantial fraction of all-correct groups into mixed groups, but this rate does not increase over training. At the same time, hint length and reliance steadily decrease. The resulting signals therefore do not translate into the same performance gains as intervention on all-incorrect groups.

D.3 Shared and Separate Policies

HiLL uses separate reasoner and hinter policies so that each role can be optimized with its own objective. We compare this design with direct full-weight sharing, where a single Llama-3.2-3B-

Table 9: Training dynamics with shared and separate reasoner–hinter policies.

Steps	Reasoner accuracy		Reasoner tokens		Hinter tokens	
	Shared	Separate	Shared	Separate	Shared	Separate
1–50	0.214	0.211	1,135	1,363	102	110
51–100	0.264	0.271	1,060	1,327	533	122
101–150	0.282	0.300	1,376	1,852	753	139

Table 10: Invalid-hint rate (%) for different failure penalties during early training.

R_{fail}	1–3	4–6	7–12	13–18	19–25
-0.01	11.6	1.7	0.7	0.8	0.4
-0.2	10.4	1.5	1.4	0.8	0.7
-0.5	11.7	1.0	1.0	0.9	0.4

Instruct policy serves as both reasoner and hinter and receives both GRPO updates. All other settings match the separate-policy configuration, and we train for 150 steps.

Table 9 shows similar early reasoner accuracy under the two configurations, but clear divergence in hint generation. The shared hinter becomes progressively longer and frequently reaches the generation limit, while the separate hinter remains concise.

Inspection of the shared-policy outputs shows two recurring behaviors: repeated paraphrasing of similar hints and continuation into reasoner-style solutions. For example, the hinter repeatedly produces variants such as "Explore alternative methods and consider clever substitutions" and "Consider a more insightful substitution," sometimes continuing until the generation limit. In other cases, a valid hint is followed by a step-by-step solution. These patterns indicate cross-role interference under direct full-weight sharing and support the use of separate policies in our setting.

D.4 Sensitivity to the Failure Penalty

The failure penalty affects the hinter advantages when valid and invalid hints coexist. We evaluate $R_{\text{fail}} \in \{-0.01, -0.2, -0.5\}$ during the first 25 training steps with both policies initialized from Llama-3.2-3B-Instruct.

Table 10 shows that invalid hints rapidly become rare under all three values, with no consistent dependence on the penalty magnitude. This supports using $R_{\text{fail}} = -0.2$ as a moderate penalty within the valid-reward scale.