Selective Off-Policy Reference Tuning with Plan Guidance

Duc Anh Le[1,*] **Tien-Phat Nguyen**[2,*] **Thien Huu Nguyen**[3] **Linh Ngo Van**[2,†] **Trung Le**[4,†]

[1]Independent Author
[2]Hanoi University of Science and Technology, Hanoi, Vietnam
[3]University of Oregon, Eugene, OR, USA
[4]Department of Data Science and AI, Monash University, Melbourne, Australia

Abstract

Reinforcement learning from verifiable rewards improves reasoning by reinforcing sampled solutions that receive positive outcomes, but group-relative methods such as GRPO become silent on hard prompts where every sampled rollout is wrong. These zero-reward prompts are often the most informative failures: each comes with a verified reference solution, yet uniformly imitating the full trace mixes core reasoning decisions with routine algebra, formatting, and surface wording. We propose **Selective Off-Policy Reference Tuning with Plan Guidance (SORT)**, an auxiliary repair update that leaves GRPO rollout generation unchanged. For each failed prompt, SORT extracts a reference-derived reasoning plan and scores every reference token twice under the same model, once with only the problem and once with the plan added to the context. Tokens whose probabilities rise under plan conditioning are treated as structurally informative and receive larger Dynamic Fine-Tuning updates, while tokens outside the model's support remain controlled by the base probability. We formalize this mechanism with a ground-truth plan model and prove that the SORT weight approaches an oracle structural-token weight as the extracted plan better approximates the true plan-conditioned distribution. Across three instruction-tuned backbones and eight in-distribution and out-of-distribution reasoning benchmarks, SORT consistently improves over GRPO and guidance-based baselines, with the largest gains on the weakest model where zero-reward failures are most frequent. Code: https://github.com/AnhLD2610/SORT.

1 Introduction

RLVR has made a simple promise attractive: if correctness can be verified, a language model can improve its reasoning by sampling solutions and reinforcing the successful ones. Group Relative Policy Optimization (GRPO) operationalizes this idea by comparing rollouts for the same prompt, avoiding a learned value model while exploiting verifiable rewards in domains such as mathematics and code [1, 2]. But this comparison-based signal has a blind spot exactly where capability expansion is needed most: when every sampled solution to a hard prompt is wrong, there is nothing to compare, the group-relative advantage is zero, and the prompt contributes no gradient. In GRPO, the hardest failures can become invisible.

This failure mode is not a corner case of implementation; it is a consequence of sparse outcome rewards and finite sampling. A model can be close enough to benefit from RL on medium-difficulty prompts, yet too weak to sample even one correct trajectory on the prompts that would teach new

*Equal contribution
†Corresponding authors: Linh Ngo Van (linhnv@soict.hust.edu.vn) and Trung Le (trunglm@monash.edu)

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

reasoning patterns. These *zero-reward prompts* are therefore a learning bottleneck: skipping them wastes the most valuable training signal, but standard GRPO cannot learn from them because the reward group is degenerate.

Recent work shows that guidance is necessary when exploration fails, but differs in where that guidance enters. LUFFY injects off-policy traces from stronger models [3]; ReLIFT interleaves RL with SFT on hard demonstrations [4]; Scaf-GRPO changes rollout generation through hierarchical hints [5]; ExPO generates answer-conditioned self-explanations as positive samples [6]; and curriculum or resampling methods reallocate compute away from degenerate groups [7, 8, 9, 10, 11]. These methods all help restore a learning signal, but they do so by changing the training distribution, relying on external or synthesized trajectories, or applying relatively uniform imitation to complete demonstrations.

This leaves a narrower gap than simply "hard prompts need supervision." Once GRPO has produced an zero-reward group, the reference solution is useful, but using it well requires deciding *which parts* of the reference should shape the student. A full solution contains core reasoning decisions mixed with surface wording, formatting, and routine algebra. Reference-level objectives can move the model toward the verified path, but they do not by themselves identify which tokens express the reasoning structure that the student failed to discover. The missing ingredient is therefore a hard-question repair signal that is simultaneously reference-supported, stable, and token-selective, while preserving the original GRPO rollout process. We target this gap by treating the reference solution not as a trajectory to copy, but as evidence for a reasoning plan from which to decide *what* to learn: a reference-derived plan should reveal which reference tokens become predictable only when the underlying reasoning plan is made explicit, thereby turning a verified solution into structure-guided token-level credit.

We propose **Selective Off-Policy Reference Tuning with Plan Guidance (SORT)**, an auxiliary repair mechanism for zero-reward GRPO groups. SORT leaves the main rollout process unchanged. When a prompt produces only incorrect rollouts, we buffer its verified reference solution and extract a reference-derived reasoning plan. We then evaluate the same model twice along the reference path: once under the original prompt, and once under a plan-conditioned teacher prompt. The plan is not used for rollout or inference, but serves as a diagnostic context. Tokens whose likelihood increases significantly under plan conditioning are interpreted as carrying structural reasoning information. This teacher–student confidence gap converts the reference trajectory into a token-selective learning signal instead of a uniformly imitated sequence.

We turn this diagnostic gap into a conservative auxiliary objective: plan-conditioned relevance decides where the reference solution is informative, while a DFT-inspired probability-space update controls how strongly the model is moved toward those tokens [12]. Specifically, each reference token is weighted by the geometric mean of the student probability and the plan-conditioned teacher probability. This form preserves the stability of reference-path learning while amplifying updates on tokens supported by the inferred reasoning structure. We formalize the approach through a ground-truth plan model, where structural tokens are defined via probability gains under plan conditioning, and show that the resulting update converges to an oracle that emphasizes precisely such tokens, with approximation error governed by plan quality. As a result, SORT repairs zero-reward prompts without modifying rollout distributions, introducing external trajectories, or collapsing to standard SFT.

Contributions. We make the following contributions:

- We identify zero-reward prompts as GRPO zero-advantage failures and introduce SORT, which repairs them without changing rollout generation.
- We use same-model plan conditioning only to score reference tokens, avoiding external teachers, rollout hints, or off-policy trajectory injection.
- We propose a geometric-mean token weight that combines DFT-style support awareness with plan-conditioned selectivity.
- We provide theory and experiments showing that SORT approximates oracle structural-token weighting and improves over GRPO and guidance-based baselines.

2 Related Work

RLVR and hard-question failure. RLVR improves reasoning with verifiable rewards, with GRPO as a common group-relative objective [1, 2]. Its key failure mode is the all-wrong or zero-variance group, where relative advantages collapse to zero. Prior work addresses this by shaping zero-variance advantages [13], reallocating rollout budgets [9, 14], or filtering/resampling near the model's competence band [7, 8, 10, 11]. SORT instead keeps all-wrong prompts and converts their verified references into auxiliary token-level learning signals.

Guidance-based repair for hard reasoning. Guidance methods restore signal by changing the training context or trajectories: LUFFY mixes stronger off-policy traces [3]; SAGE, Guide, and Scaf-GRPO inject hints or scaffolds [15, 16, 5]; SGPO and ExPO add stepwise guidance or answer-conditioned explanations [17, 6]; and ReLIFT interleaves supervised hard-example updates [4]. Related structured-repair methods use targeted exploration, online self-verification, or decomposed rubric rewards [18, 19, 20]. SORT differs by leaving rollout generation unchanged and using verified references only for structure-guided, token-selective repair.

Reference learning and token-selective supervision. SORT also relates to reference-learning and process-supervision objectives. DFT reweights SFT in probability space [12], SDFT uses a demonstration-conditioned same-model teacher [21], and RLPR uses reference-answer likelihoods as verifier-free rewards [22]. Discord uses chain-of-thought guidance and optimal transport for distillation across tokenizers [23]. SORT similarly uses reasoning guidance, deriving reference-token weights from same-model plans. Process-supervision methods assign value to partial reasoning states via step, segment, or implicit process signals [24, 25, 26, 27]; SORT instead avoids a separate process verifier or forced early exits, using same-model plan conditioning to upweight structurally informative reference tokens. This aligns with token-level analyses showing that useful RL signal concentrates on sparse decision-critical tokens [28, 29]. TokenRatio also uses prefix-dependent correction weights in token-level preference optimization [30], sharing SORT's emphasis on context-dependent weighting.

3 Methodology

We introduce **Selective Off-Policy Reference Tuning with Plan Guidance (SORT)**, an auxiliary reference update that converts zero-reward GRPO prompts into token-selective learning signals while leaving rollout generation unchanged.

3.1 From Zero-Reward Failures to Deferred Repair

Verified references are valuable on hard prompts, but copying them wholesale is the wrong repair mechanism: cross-entropy SFT can narrow the policy and full reasoning traces mix decision-critical steps with routine algebra and wording [31, 32]. GRPO avoids this uniform off-policy imitation by learning from sampled rollouts, yet it becomes silent when every rollout is wrong.

Consider a GRPO training iteration with rollout batch $\mathcal{U}_k = \{q_i\}_{i=1}^{B_{\mathrm{rl}}}$. For each prompt q_i, the model samples N responses and receives binary verifiable rewards $r_i^{(n)} \in \{0, 1\}$. GRPO assigns each sampled response a normalized group advantage,

$$\bar{r}_i = \frac{1}{N}\sum_{m=1}^{N} r_i^{(m)}, \qquad s_i = \sqrt{\frac{1}{N}\sum_{m=1}^{N}(r_i^{(m)} - \bar{r}_i)^2}, \qquad A_i^{(n)} = \frac{r_i^{(n)} - \bar{r}_i}{s_i + \epsilon}. \tag{1}$$

If every rollout is incorrect, then

$$r_i^{(1)} = \cdots = r_i^{(N)} = 0 \implies \bar{r}_i = s_i = 0, \quad A_i^{(n)} = 0 \; \forall n, \tag{2}$$

so all policy-gradient terms for q_i vanish. We denote these zero-reward indices by $\mathcal{I}_{\mathrm{aw}}^{(k)} = \{i : r_i^{(n)} = 0, \forall n \in [N]\}$. They are not noise—they mark the policy frontier, where a single missed reasoning decision can make an otherwise fluent rollout fail. The verified reference $y_i^\star$ contains the missing structure, but interleaved with routine derivation. The bottleneck is *selectivity*: learn from the tokens that reveal the missing structure without collapsing into uniform imitation [12]. Appendix C gives representative zero-reward prompts.

SORT resolves this by deferring repair. Failed prompts are neither skipped nor merged into GRPO; they are **buffered** for a separate auxiliary update:

$$\mathcal{B}_{\text{aw}} \leftarrow \text{Enqueue}\big(\mathcal{B}_{\text{aw}},\ \{(q_i, y_i^\star) : i \in \mathcal{I}_{\text{aw}}^{(k)}\}\big). \tag{3}$$

When $|\mathcal{B}_{\text{aw}}| \geq B_{\text{aux}}$, a minibatch $\mathcal{M}_k \sim \text{SampleBatch}(\mathcal{B}_{\text{aw}}, B_{\text{aux}})$ triggers one SORT update. GRPO continues learning from prompts with reward variance; SORT repairs the hard frontier.

3.2 Plan-Conditioned Token Salience

For a buffered pair $(q_i, y_i^\star)$, SORT asks which reference tokens should exert the strongest learning pressure. The guiding principle is that a token should be emphasized when it becomes substantially more predictable once the model is given the ground-truth reasoning plan behind the verified solution.

Definition 1 (Ground-truth solution plan)**.** *For a buffered example $(q_i, y_i^\star)$ with $y_i^\star = (y_{i,1}^\star, \ldots, y_{i,T_i}^\star)$, let S_i denote the* ground-truth solution plan*: an idealized ordered outline of the key reasoning decisions sufficient to derive the reference solution from the prompt. It records structural choices such as representations, lemmas, case splits, and constraint propagation. We do not assume that S_i is observed, optimized as a target, or sampled by the model; it is used only as an analysis device. Conditioned on a fixed plan S_i, the reference factorizes as*

$$\pi_\theta(y_i^\star \mid q_i, S_i) = \prod_{t=1}^{T_i} \pi_\theta(y_{i,t}^\star \mid q_i, S_i, y_{i,<t}^\star). \tag{4}$$

Appendix D illustrates the idealized ground-truth plan used in the analysis.

Definition 2 (Plan salience)**.** *The ground-truth plan salience of reference token $y_{i,t}^\star$ is*

$$\sigma_{i,t} := \log \frac{\pi_\theta(y_{i,t}^\star \mid q_i, S_i, y_{i,<t}^\star)}{\pi_\theta(y_{i,t}^\star \mid q_i, y_{i,<t}^\star)}. \tag{5}$$

Equivalently, this log-ratio is the conditional pointwise mutual information $\text{PMI}_\theta(y_{i,t}^\star; S_i \mid q_i, y_{i,<t}^\star)$, which clarifies its role. Large positive values ($\sigma_{i,t} \gg 0$) indicate that the plan contributes substantial information: the base model cannot reliably predict the token from the problem and prefix alone, but becomes confident once the step-by-step plan is revealed. These correspond to reasoning-critical steps—e.g., invariant selection, case splits, or constraint propagation—that determine solution correctness. In contrast, tokens with $\sigma_{i,t} \approx 0$ are routine (algebraic manipulation or phrasing) and largely independent of the plan. SORT therefore ignores the near-zero regime and selectively upweights tokens with large positive $\sigma_{i,t}$.

If the ground-truth plan S_i were available, $\sigma_{i,t}$ would identify reference tokens that are difficult yet learnable given the correct reasoning steps. Since only the reference solution is available, SORT approximates this conditioning with a reference-derived plan, scoring each token twice under the same model: once with only the problem and once with the extracted plan in context.

Reference-derived plan. For each buffered example, we ask the current model to extract a plan-like conditioning context from the verified reference:

$$\hat{s}_i = \text{Gen}_\theta(\textsc{PlanExtract}(y_i^\star)). \tag{6}$$

The extracted plan $\hat{s}_i$ is not used as a target sequence and is never used at inference time. It serves only as a diagnostic context for scoring the reference. Its role is to make explicit the decisions, intermediate realizations, and calculation context that support the verified solution, so that plan-dependent reference tokens become easier to predict under conditioning. Prompt templates are given in Appendix F.

Two reference-path evaluations. We evaluate the same reference tokens under two prefixes: the base prefix contains only the problem, while the plan-conditioned teacher prefix additionally contains $\hat{s}_i$:

$$p_{i,t}^{\text{base}} = \pi_\theta\big(y_{i,t}^\star \mid \textsc{BasePrompt}(q_i), y_{i,<t}^\star\big), \quad p_{i,t}^{\text{plan}} = \pi_\theta\big(y_{i,t}^\star \mid \textsc{TeacherPrompt}(q_i, \hat{s}_i), y_{i,<t}^\star\big). \tag{7}$$

Both evaluations use the same model and reference path; only the conditioning context differs. In the auxiliary loss, the base branch receives gradient, while the plan-conditioned branch is detached.

Plan-conditioned log-ratio. Using the extracted plan $\hat{s}_i$ as a practical proxy for the ground-truth plan S_i in the log-ratio gives an estimated token-level score:

$$\hat{\sigma}_{i,t} := \log p_{i,t}^{\text{plan}} - \log p_{i,t}^{\text{base}}. \tag{8}$$

When the extracted plan induces conditioning close to the ground-truth plan ($\hat{s}_i \approx S_i$), this plug-in estimator approximates the ideal plan salience: $\hat{\sigma}_{i,t} \approx \sigma_{i,t}$. If $\hat{\sigma}_{i,t}$ is large and positive, the reference token becomes much easier to predict once the plan is supplied, indicating it encodes a reasoning step the base policy is missing. SORT interprets such tokens as more informative for updating the base policy and upweights them accordingly.

3.3 Geometric Reference Weighting

We use lowercase $\ell_{i,t}$ for a single-token loss and uppercase $\mathcal{L}$ for an objective aggregated over a token set or minibatch. Dynamic Fine-Tuning (DFT) [12] stabilizes reference-path learning by replacing the unit SFT weight with the detached base probability,

$$\ell_{i,t}^{\text{DFT}} = -\,\text{sg}\big(p_{i,t}^{\text{base}}\big) \log p_{i,t}^{\text{base}}.$$

Its gradient can be written in policy-gradient form as

$$\nabla_\theta \ell_{i,t}^{\text{DFT}} = -\nabla_\theta p_{i,t}^{\text{base}} = -\mathbb{E}_{z\sim\pi_\theta(\cdot|q_i,y^\star_{i,<t})}\left[\underbrace{\mathbf{1}\{z = y^\star_{i,t}\}}_{\text{reward}=1} \nabla_\theta \log \pi_\theta(z \mid q_i, y^\star_{i,<t})\right]. \tag{9}$$

Every reference token receives a uniform reward of 1, which stabilizes training but ignores the distinction between reasoning-critical and routine tokens. SORT generalizes this by replacing the uniform reward with a plan-conditioned one.

From uniform reward to plan-conditioned reward. By Definition 2, the plan salience $\sigma_{i,t} = \log\big(\pi_\theta(y^\star_{i,t} \mid q_i, S_i, y^\star_{i,<t}) \,/\, p_{i,t}^{\text{base}}\big)$ measures how much the oracle plan S_i increases the token probability. We replace DFT's uniform reward with the tempered plan-conditioned reward:

$$\rho^\star_{\beta,i,t} = \exp(\beta\sigma_{i,t}), \qquad \beta \in [0,1]. \tag{10}$$

Setting $\beta = 0$ recovers DFT; any $\beta > 0$ upweights tokens that become more predictable once the plan is revealed ($\rho^\star_{\beta,i,t} > 1$ iff $\sigma_{i,t} > 0$). Inserting this reward into the DFT gradient gives

$$\begin{aligned} g^\star_{\beta,i,t} &= -\rho^\star_{\beta,i,t}\,\nabla_\theta p_{i,t}^{\text{base}} = -\mathbb{E}_{z\sim\pi_\theta(\cdot|q_i,y^\star_{i,<t})}\big[\mathbf{1}\{z = y^\star_{i,t}\}\,\rho^\star_{\beta,i,t}\,\nabla_\theta \log \pi_\theta(z \mid q_i, y^\star_{i,<t})\big] \\ &= \nabla_\theta\left[-\,\text{sg}\big(p_{i,t}^{\text{base}}\,\rho^\star_{\beta,i,t}\big)\log p_{i,t}^{\text{base}}\right], \end{aligned} \tag{11}$$

Reading off the surrogate loss, the oracle token weight and the corresponding oracle objective are

$$\omega^\star_{\beta,i,t} = \text{sg}\big(p_{i,t}^{\text{base}}\,\exp(\beta\sigma_{i,t})\big), \tag{12}$$

$$\mathcal{L}^\star(\mathcal{M}_k) = -\frac{1}{|\mathcal{S}(\mathcal{M}_k)|}\sum_{(i,t)\in\mathcal{S}(\mathcal{M}_k)} \omega^\star_{\beta,i,t}\,\log \pi_\theta\big(y^\star_{i,t} \mid \textsc{BasePrompt}(q_i), y^\star_{i,<t}\big). \tag{13}$$

If the ground-truth plan S_i were known, $\mathcal{L}^\star$ would be the ideal training objective. In practice $\sigma_{i,t}$ is unavailable, so we approximate it.

From oracle to practical weight. Replacing $\sigma_{i,t}$ with the estimated log-ratio $\hat{\sigma}_{i,t}$ from Eq. (8) gives the practical SORT weight family:

$$\omega_{\beta,i,t} = \text{sg}\big(p_{i,t}^{\text{base}}\,\exp(\beta\,\hat{\sigma}_{i,t})\big) = \text{sg}\left(\big(p_{i,t}^{\text{base}}\big)^{1-\beta}\left(p_{i,t}^{\text{plan}}\right)^{\beta}\right), \qquad \beta \in [0,1]. \tag{14}$$

At $\beta = 0$ the weight reduces to $\text{sg}(p_{i,t}^{\text{base}})$, recovering DFT. As β increases, the plan-conditioned probability $p_{i,t}^{\text{plan}}$ gains influence: tokens whose probability rises under plan conditioning are upweighted,

while those unaffected or hurt are unchanged or downweighted. We set $\beta = \frac{1}{2}$ throughout, which yields the geometric-mean weight and the minibatch objective:

$$\omega_{i,t} = \mathrm{sg}\left(\sqrt{p_{i,t}^{\mathrm{base}}\, p_{i,t}^{\mathrm{plan}}}\right), \tag{15}$$

$$\mathcal{L}_{\mathrm{SORT}}(\mathcal{M}_k) = -\frac{1}{|\mathcal{S}(\mathcal{M}_k)|} \sum_{(i,t)\in\mathcal{S}(\mathcal{M}_k)} \omega_{i,t}\, \log \pi_\theta\big(y_{i,t}^\star \mid \textsc{BasePrompt}(q_i), y_{i,<t}^\star\big), \tag{16}$$

where $\mathcal{S}(\mathcal{M}_k) = \{(i,t) : (q_i, y_i^\star) \in \mathcal{M}_k,\ 1 \le t \le |y_i^\star|\}$ and only the base branch receives gradient. Tokens that become more predictable under plan conditioning ($p_{i,t}^{\mathrm{plan}} > p_{i,t}^{\mathrm{base}}$) are upweighted relative to DFT; tokens made less predictable are downweighted.

3.4 Theoretical Analysis

The weighting scheme above relies on the extracted plan $\hat{s}_i$ in place of the unknown ground-truth plan S_i. The plan mismatch γ below should be small when the policy has reasonable planning ability: as reasoning improves, extracted plans should better preserve high-level solution structure, as illustrated qualitatively in Appendix E. Thus the bound is small under faithful extraction and weakens when the extractor misses key structural decisions.

Assumption 3 (Plan-induced stability)**.** *Let $h_\theta(q, y_{<t}^\star, s)$ denote the decoder hidden state at step t conditioned on plan s. We assume that for any $\hat{s}_i, S_i$ and all token positions t,*

$$\|h_\theta(q_i, y_{i,<t}^\star, \hat{s}_i) - h_\theta(q_i, y_{i,<t}^\star, S_i)\| \le \gamma,$$

for some $\gamma \ge 0$.

Assumption 4 (Bounded score)**.** *The token-level score function along the reference path is uniformly bounded:*

$$\big\|\nabla_\theta \log \pi_\theta(y_{i,t}^\star \mid q_i, y_{i,<t}^\star)\big\| \le G \qquad \forall (i,t).$$

Lemma 5 (Plan-to-token smoothness)**.** *Assume the output mapping $h \mapsto \log \pi_\theta(\cdot \mid h)$ is L-Lipschitz. Under Assumption 3, we have*

$$\big|\log \pi_\theta(y_{i,t}^\star \mid \hat{s}_i, q_i, y_{i,<t}^\star) - \log \pi_\theta(y_{i,t}^\star \mid S_i, q_i, y_{i,<t}^\star)\big| \le L\gamma.$$

Applying Lemma 5 and the mean value theorem yields the following regret bound.

Theorem 6 (Plan-regret bound)**.** *Under Assumption 3, for any $\beta \in [0,1]$, the per-token SORT weight error satisfies*

$$\boxed{\big|\omega_{\beta,i,t} - \omega_{\beta,i,t}^\star\big| \;\le\; \beta L\gamma \left(p_{i,t}^{\mathrm{base}}\right)^{1-\beta},} \tag{17}$$

where $\omega_{\beta,i,t}^\star$ is the oracle weight from Eq. (12). In particular, for the geometric-mean setting $\beta = \frac{1}{2}$,

$$\big|\omega_{i,t} - \omega_{i,t}^\star\big| \le \frac{L\gamma}{2}\sqrt{p_{i,t}^{\mathrm{base}}}.$$

Under Assumption 4, aggregated over the auxiliary minibatch,

$$\boxed{\big\|\nabla_\theta \mathcal{L}_{\mathrm{SORT},\beta} - \nabla_\theta \mathcal{L}_\beta^\star\big\| \;\le\; \beta L\gamma \cdot \frac{1}{|\mathcal{S}|} \sum_{(i,t)\in\mathcal{S}} \left(p_{i,t}^{\mathrm{base}}\right)^{1-\beta} \big\|\nabla_\theta \log p_{i,t}^{\mathrm{base}}\big\| \;\le\; \beta L\gamma G.} \tag{18}$$

The bound carries three messages. First, as $\gamma \to 0$ the SORT gradient converges to the oracle gradient at rate $\mathcal{O}(\gamma)$. Second, the factor $\left(p_{i,t}^{\mathrm{base}}\right)^{1-\beta}$ automatically suppresses the weight error on tokens the model finds difficult ($p_{i,t}^{\mathrm{base}} \ll 1$), inheriting DFT's stabilization; for the midpoint choice $\beta = \frac{1}{2}$ this becomes $\sqrt{p_{i,t}^{\mathrm{base}}}$. Third, under bounded scores, the aggregated gradient deviation is at most $\beta L\gamma G$, independent of sequence length or vocabulary size. In short, SORT degrades gracefully: it exploits structural information when the plan is faithful, and falls back to DFT-like behavior otherwise. Proofs are deferred to Appendix K.

3.5 Interleaved Optimization

SORT is applied as an auxiliary step interleaved with standard GRPO: each iteration performs a GRPO update on the rollout batch, zero-reward prompts are buffered, and when the buffer reaches size B_{aux}, a minibatch triggers one SORT update. This decouples on-policy exploration from reference consolidation. The full procedure is presented in Algorithm 1.

4 Experiments

4.1 Experimental Setup

Models, data, and evaluation. We assess SORT on Llama-3.2-3B-Instruct [33], Qwen2.5-7B-Instruct [34], and Qwen3-4B-Instruct [35]. Training uses a 15k-prompt subset of OpenR1-Math-220k, curated by Liao et al. [15], derived from NuminaMath 1.5 [36], and paired with ground-truth answers and reference solutions, without pass-rate filtering. We report in-distribution performance on AIME24/25 [37, 38], AMC23 [39], MATH-500 [40], Minerva Math [41], and OlympiadBench [42], and out-of-distribution performance on GPQA-diamond [43] and MMLU-Pro [44]. Unless otherwise noted, all metrics are avg@16 accuracy.

Baselines and implementation. We compare SORT with SFT on DeepSeek-R1 traces, vanilla GRPO, matched same-model self-distillation (GRPO + SD), LUFFY, ReLIFT, and Scaf-GRPO. GRPO + SD matches SORT's all-wrong buffer and auxiliary-update configuration, using full-vocabulary forward KL from a detached plan-conditioned copy of the model at student-generated prefixes. All guidance baselines are reproduced on the same 15k training subset with matched batch size and optimization steps. SORT uses only same-model plans to score reference tokens and does not modify rollout generation. All methods are trained on 8 H100 GPUs using `verl` [45] and vLLM [46]; following DAPO [8], we disable KL regularization and use asymmetric clipping with $(\epsilon_{\text{low}}, \epsilon_{\text{high}}) = (0.2, 0.28)$. Unless otherwise specified, we use a learning rate of 1×10^{-6}, maximum response length 8192, batch size 128, 8 trajectories per prompt, and 500 training steps, we save every 50 steps. For Qwen3-4B-Instruct, we use 4 trajectories per prompt due to slower training caused by its long response length.

Buffer size. We set the SORT buffer size from the early all-wrong count in Appendix G, using the largest power of two below that count so the buffer fills quickly. This gives buffer sizes of 64, 16, and 8 for Llama-3.2-3B-Instruct, Qwen2.5-7B-Instruct, and Qwen3-4B-Instruct, respectively.

4.2 Main Results

Table 1 shows that SORT is the strongest method on average across all three backbones, covering weak, medium, and strong model regimes. Relative to vanilla GRPO, SORT improves the in-distribution average by 2.0 points on Llama-3.2-3B, 2.4 points on Qwen2.5-7B, and 1.0 points on Qwen3-4B. These gains also transfer out of distribution, where SORT improves the GPQA/MMLU-Pro average by 0.7, 2.7, and 0.4 points, respectively. This pattern matches the intended role of SORT: when GRPO fails to produce a correct rollout on difficult prompts, the auxiliary reference-path update recovers additional learning signal without changing the rollout distribution.

The largest improvement over the untuned base model appears on the weakest backbone. On Llama-3.2-3B-Instruct, SORT increases the in-distribution average from 17.8 to 23.9 and the out-of-distribution average from 22.5 to 33.8, which is consistent with the expectation that all-wrong groups occur most frequently in this regime. On the medium Qwen2.5-7B-Instruct backbone, SORT also gives the strongest in-distribution and out-of-distribution averages, showing that the benefit is not limited to low-capability models. At the same time, SORT still improves the strongest starting point, Qwen3-4B-Instruct, from 65.8 to 69.8 in distribution and from 64.3 to 64.9 out of distribution, indicating that the method remains complementary even when the base policy already solves many prompts. By contrast, LUFFY applies reference supervision much more broadly, effectively mixing ground-truth traces into the update for every prompt and boosting low-probability tokens even when they are not the right learning signal; this makes it less selective than SORT. ReLIFT and pure SFT expose a different failure mode on weak backbones: they tend to make answers longer and closer to reference traces, but this extra verbosity does not translate into better reasoning accuracy. This is most visible on Llama-3.2-3B-Instruct, where SFT and ReLIFT both underperform GRPO despite

Table 1: Main results on in-distribution math benchmarks and out-of-distribution reasoning benchmarks. All numbers are avg@16 accuracy. In the AIME24/25 column, the two values correspond to AIME24 and AIME25, respectively. Δ is measured against the corresponding base model.

	In-distribution							Out-of-distribution			
Method	**AIME24/25**	**AMC23**	**MATH-500**	**Minerva**	**Olympiad**	**Avg.**	**Δ**	**GPQA**	**MMLU-Pro**	**Avg.**	**Δ**
Llama-3.2-3B-Instruct	6.5/0.6	22.8	44.7	17.8	14.2	17.8	0	17.9	27.0	22.5	0
SFT	0.4/0.6	9.5	26.9	5.1	6.5	8.2	-9.6	11.6	18.8	15.2	-7.3
GRPO	6.7/0.8	29.5	52.1	20.5	21.8	21.9	+4.1	26.3	39.8	33.1	+10.6
GRPO + SD	6.3/1.0	30.8	53.2	19.7	21.3	22.1	+4.3	25.4	39.2	32.3	+9.8
LUFFY	4.4/0.4	18.6	38.9	14.3	11.9	14.7	-3.1	16.0	26.7	21.4	-1.1
ReLIFT	0.4/0.2	17.8	36.8	10.4	10.0	12.6	-5.2	11.7	26.5	19.1	-3.4
Scaf-GRPO	7.7/**2.3**	28.8	51.7	19.4	19.5	21.5	+3.7	24.1	38.0	31.0	+8.5
Ours	**8.8**/1.7	**32.7**	**56.0**	**21.6**	**22.4**	**23.9**	**+6.1**	**26.4**	**41.2**	**33.8**	**+11.3**
Qwen2.5-7B-Instruct	13.8/6.7	53.4	75.7	38.1	39.2	37.8	0	37.1	56.4	46.7	0
SFT	3.5/7.1	30.9	56.2	20.0	21.7	23.2	-14.6	9.5	35.6	22.5	-24.2
GRPO	15.0/13.5	55.5	79.2	39.1	44.5	41.1	+3.3	37.2	57.6	47.4	+0.7
GRPO + SD	15.6/12.8	57.8	79.9	**39.8**	44.2	41.7	+3.9	36.8	58.4	47.6	+0.9
LUFFY	**17.1**/13.5	55.2	**81.3**	39.0	44.2	41.7	+3.9	38.1	59.1	48.6	+1.9
ReLIFT	15.6/**15.8**	55.8	80.5	38.8	44.1	41.8	+4.0	32.4	57.3	44.9	-1.8
Scaf-GRPO	14.6/12.7	58.8	78.0	**39.8**	42.0	41.0	+2.2	36.6	58.4	47.5	+0.8
Ours	**17.1**/15.4	**60.5**	81.0	39.5	**47.3**	**43.5**	**+5.7**	**39.6**	**60.5**	**50.1**	**+3.4**
Qwen3-4B-Instruct	52.1/43.1	92.2	93.6	46.1	67.7	65.8	0	**57.6**	70.9	64.3	0
SFT	14.4/22.1	55.2	78.9	38.1	40.2	41.5	-24.3	29.4	52.6	41.0	-23.3
GRPO	55.8/46.5	**95.0**	95.6	**49.9**	70.3	68.8	+3.0	57.0	72.0	64.5	+0.2
GRPO + SD	56.7/46.8	94.7	95.7	49.8	70.8	69.1	+3.3	56.6	**72.3**	64.5	+0.2
LUFFY	50.8/38.1	88.8	93.3	48.1	59.4	63.1	-2.7	56.0	46.6	51.3	-13.0
ReLIFT	59.0/**48.3**	93.3	95.3	48.1	69.1	68.9	+3.1	56.8	71.8	64.3	0.0
Scaf-GRPO	58.3/**48.3**	94.2	95.4	48.8	69.8	69.1	+3.3	53.6	72.1	62.9	-1.4
Ours	**62.5**/47.5	93.4	**95.8**	48.3	**71.5**	**69.8**	**+4.0**	**57.6**	72.1	**64.9**	**+0.6**

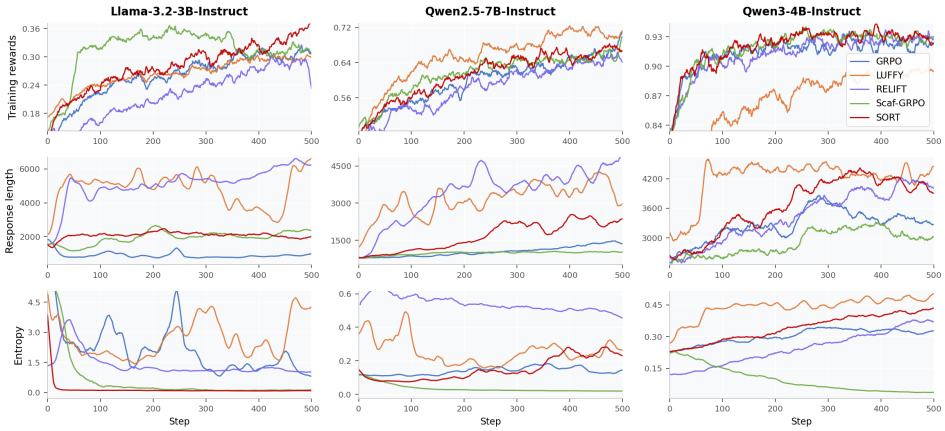

Figure 1: Training dynamics across 500 optimization steps. Rows show reward, response length, and entropy; columns show the three backbones. SORT improves reward without the length expansion seen in LUFFY and ReLIFT.

producing more imitation-like responses. This supports the core design principle of SORT: repairing hard questions should be reference-supported, but the supervision must remain token-selective and policy-aware rather than uniformly imitative.

4.3 Training Dynamics

Figure 1 complements the final benchmark results by showing that higher reward is not merely a byproduct of longer generations. On Llama-3.2-3B-Instruct, SORT steadily improves to the highest reward while response length stays compact, whereas LUFFY and ReLIFT quickly expand length and Scaf-GRPO plateaus after sharp entropy collapse—the regime where all-wrong groups are most frequent, suggesting that selective reference repair guides the weak policy without broad trace imitation. On Qwen2.5-7B-Instruct, SORT catches up to LUFFY's early high reward at substantially lower length, indicating the plan-weighted update repairs hard prompts without verbosity cost.

Table 2: Effect of the interpolation exponent β on Qwen2.5-7B-Instruct.

	In-distribution							Out-of-distribution			
Method	**AIME24/25**	**AMC23**	**MATH-500**	**Minerva**	**Olympiad**	**Avg.**	**Δ**	**GPQA**	**MMLU-Pro**	**Avg.**	**Δ**
Qwen2.5-7B-Instruct	13.8/6.7	53.4	75.7	38.1	39.2	37.8	0	37.1	56.4	46.7	0
$\beta = 0$	**17.7**/12.9	55.0	79.6	39.5	44.8	41.6	+3.8	37.4	59.2	48.3	+1.6
$\beta = 1$	17.3/13.8	56.7	80.3	40.1	46.1	42.4	+4.6	36.1	58.9	47.5	+0.8
$\beta = 1/2$ (SORT)	17.1/**15.4**	**60.5**	**81.0**	**39.5**	**47.3**	**43.5**	**+5.7**	**39.6**	**60.5**	**50.1**	**+3.4**

Table 3: Ablations on Qwen2.5-7B-Instruct: auxiliary updates (upper block) and reference/prompt variants (lower block). All scores are avg@16 accuracy; Δ is relative to the base model. Bold marks the best value among the displayed rows.

	In-distribution							Out-of-distribution			
Method	**AIME24/25**	**AMC23**	**MATH-500**	**Minerva**	**Olympiad**	**Avg.**	**Δ**	**GPQA**	**MMLU-Pro**	**Avg.**	**Δ**
Qwen2.5-7B-Instruct	13.8/6.7	53.4	75.7	38.1	39.2	37.8	0	37.1	56.4	46.7	0
GRPO	15.0/13.5	55.5	79.2	39.1	44.5	41.1	+3.3	37.2	57.6	47.4	+0.7
SFT only	16.7/13.3	56.9	**81.5**	39.2	45.6	42.2	+4.4	34.6	56.4	45.5	-1.2
SFT + plan	16.4/15.2	56.7	81.0	37.8	**47.5**	42.4	+4.6	34.3	59.0	46.7	0.0
Random emphasis	13.8/12.9	58.0	79.1	38.1	45.2	41.2	+3.4	34.3	57.8	46.0	-0.7
SORT (Ours)	17.1/**15.4**	60.5	81.0	39.5	47.3	43.5	+5.7	39.6	60.5	50.1	+3.4
SORT (50% reference)	**19.0**/12.3	57.0	80.2	39.7	46.7	42.5	+4.7	34.0	58.0	46.0	-0.7
SORT (alt. prompt)	17.7/14.2	**61.4**	80.7	**40.0**	**47.5**	**43.6**	**+5.8**	**39.8**	**60.7**	**50.3**	**+3.6**

On Qwen3-4B-Instruct, where rewards near saturation, SORT tracks the best reward band, avoids LUFFY's long-response underperformance, and maintains moderate entropy unlike Scaf-GRPO. Across backbones, the curves support the central claim that SORT adds targeted pressure on hard cases instead of merely increasing length or uniformly imitating full demonstrations.

4.4 Ablation Studies

We ablate SORT on Qwen2.5-7B-Instruct across weighting, auxiliary supervision, reference completeness, and extraction prompts.

Effect of β. Table 2 compares DFT-style student weighting ($\beta = 0$), teacher-only weighting ($\beta = 1$), and the geometric mean used by SORT ($\beta = 1/2$). The midpoint gives the best in-distribution and out-of-distribution averages, improving from 41.6/48.3 for $\beta = 0$ and 42.4/47.5 for $\beta = 1$ to 43.5/50.1. This suggests that plan selectivity and student support are both needed.

Training variant. Table 3 separates buffered reference learning from plan-conditioned reweighting. Uniform SFT and plan-text supervision reach 42.2/45.5 and 42.4/46.7 ID/OOD averages. Random emphasis assigns weight 1 to a random 30% of reference tokens and 0.5 to the rest, using the same references and update schedule. It reaches only 41.2/46.0, compared with SORT's 43.5/50.1, showing that arbitrary token emphasis does not reproduce the gains.

Reference and prompt robustness. Using only the first 50% of each reference for both plan extraction and supervision improves the ID average over GRPO from 41.1 to 42.5, but reduces the OOD average from 47.4 to 46.0. A semantically equivalent extraction prompt gives similar performance to the default (43.6/50.3 versus 43.5/50.1). Table 3 includes both variants; Appendix M gives the alternative extraction prompt.

Plan quality. We evaluate 5,000 stratified problem–reference–plan triples per backbone with GPT-5.6-sol. Table 4 reports mean 0–4 scores and unsupported-claim rates. Even Llama reaches 3.541 coverage and 3.733 ordering, with fewer than 3% unsupported claims. These scores support plan faithfulness across the tested backbones; the rubric and aggregation protocol are in Appendix L.

Harder Qwen3 training. We retrain Qwen3-4B on a harder subset from a different reference source, selected before training by verified-reference length alone (over 6,000 tokens). Table 5 shows improvements on six of eight benchmarks: the ID average rises from 69.08 to 70.18 (+1.10) and OOD from 64.95 to 65.80 (+0.85). Appendix N gives the source and protocol.

Table 4: Plan quality on 5,000 triples per backbone. Quality scores range from 0 to 4; unsupported-claim rates (macro/micro) are percentages.

Extractor	Correctness	Coverage	Ordering	Unsupported
Llama-3.2-3B	3.564	3.541	3.733	2.33/2.87
Qwen2.5-7B	3.668	3.669	3.855	1.87/1.93
Qwen3-4B	3.813	3.861	3.933	1.67/1.77

Table 5: Qwen3-4B-Instruct trained on the hard subset with verified references longer than 6,000 tokens. All scores are avg@16 accuracy; Δ is SORT minus GRPO.

Method	AIME24/25	AMC23	MATH-500	Minerva	Olympiad	ID avg.	GPQA	MMLU-Pro	OOD avg.
GRPO	58.54/47.29	94.08	95.20	48.74	70.60	69.08	57.70	72.20	64.95
SORT	61.46/49.54	93.75	94.80	49.40	72.10	70.18	59.00	72.60	65.80
Δ (SORT $-$ GRPO)	+2.92/+2.25	-0.33	-0.40	+0.66	+1.50	+1.10	+1.30	+0.40	+0.85

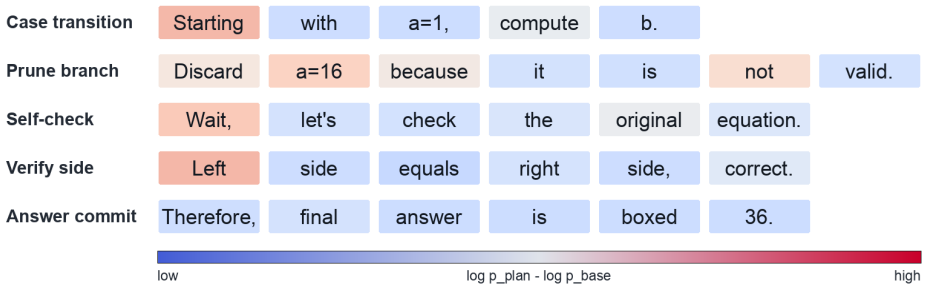

Figure 2: Token-level salience on a zero-reward reference trace. Colors indicate the plan-conditioned log-ratio $\hat{\sigma}_{i,t}$, with high-salience regions concentrating on reasoning-control operations such as branching, verification, and self-checking.

4.5 Token Salience Visualization

Figure 2 illustrates the mechanism behind the token-selective update. Plan conditioning raises probability on tokens that steer the autoregressive reasoning trajectory, including case transitions, branch pruning, self-checks, and verification. Many local symbolic or answer-formatting tokens remain close to neutral because they are already predictable from the reference prefix. This supports the design of SORT as a policy-aware repair objective rather than uniform imitation of the full solution trace.

5 Conclusion

We introduced **Selective Off-Policy Reference Tuning with Plan Guidance (SORT)**, an auxiliary update for the zero-gradient failure mode of GRPO on all-wrong prompts. SORT buffers failed prompts, extracts a reference-derived solution plan, and uses same-model plan conditioning to identify reference tokens whose probabilities rise when the reasoning structure is supplied, yielding a DFT-style token-selective loss rather than uniform imitation. Our analysis connects this weight to an oracle structural-token objective and bounds the gap by the quality of the extracted plan. Across three instruction-tuned backbones and eight benchmarks, SORT consistently improves over GRPO and guidance-based baselines, with the largest gains on weaker models where all-wrong groups are most common.

Acknowledgments

Linh Ngo Van was funded by Vingroup Innovation Foundation (VINIF) under project code VINIF.2026.DA117. Trung Le was supported by the Air Force Office of Scientific Research under award number FA2386-25-1-4023 and the ARC Discovery Project grant DP250100262.

Competing Interests. The authors declare no competing interests.

References

[1] 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, 2024. URL https://arxiv.org/abs/2402.03300.

[2] DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Erhang Li, Fangqi Zhou, Fangyun Lin, Fucong Dai, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoran Wei, Haowei Zhang, Haowen Luo, Haozhe Ji, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, Jialiang Huang, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jingchang Chen, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jinhua Zhu, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexin Huang, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Liang Zhao, Liangsheng Yin, Lihua Guo, Lingxiao Luo, Linwang Ma, Litong Wang, Liyue Zhang, M. S. Di, M. Y Xu, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Panpan Huang, Peixin Cong, Peiyi Wang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, S. H. Liu, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Songyang Zhou, Tao Ni, Tao Yun, Tian Pei, Tian Ye, Tianyuan Yue, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjun Gao, Wentao Zhang, Xi Gao, Xiangwen Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyuan Li, Xu Chen, Xuecheng Su, Xuehai Pan, Xuheng Lin, Xuwei Fu, Y. Q. Wang, Yang Zhang, Yanhong Xu, Yanru Ma, Yao Li, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yiliang Xiong, Ying He, Ying Zhou, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuduan Wang, Yue Gong, Yuhan Wu, Yuheng Zou, Yukun Li, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehua Zhao, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhiyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Zizheng Pan, Zongqing Yao, Bei Feng, Hui Li, J. L. Cai, Jiaqi Ni, Lei Xu, Meng Li, Ning Tian, R. J. Chen, R. L. Jin, S. S. Li, Shuang Zhou, Tianyu Sun, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xinnan Song, Xinyi Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, Dongjie Ji, Jian Liang, Jianzhong Guo, Jin Chen, Leyi Xia, Miaojun Wang, Mingming Li, Peng Zhang, Ruyi Chen, Shangmian Sun, Shaoqing Wu, Shengfeng Ye, T. Wang, W. L. Xiao, Wei An, Xianzu Wang, Xiaowen Sun, Xiaoxiang Wang, Ying Tang, Yukun Zha, Zekai Zhang, Zhe Ju, Zhen Zhang, and Zihua Qu. Deepseek-v3.2: Pushing the frontier of open large language models, 2025. URL https://arxiv.org/abs/2512.02556.

[3] Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance, 2025. URL https://arxiv.org/abs/2504.14945.

[4] Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Hao Wong, Junbo Niu, Chengyu Shen, Runming He, Yanhao Li, Bin Cui, and Wentao Zhang. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions, 2026. URL https://arxiv.org/abs/2506.07527.

[5] 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, 2026. URL https://arxiv.org/abs/2510.19807.

[6] Ruiyang Zhou, Shuozhe Li, Amy Zhang, and Liu Leqi. Expo: Unlocking hard reasoning with self-explanation-guided reinforcement learning, 2026. URL https://arxiv.org/abs/2507.02834.

[7] Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and Hanze Dong. A minimalist approach to llm reasoning: from rejection sampling to reinforce, 2025. URL https://arxiv.org/abs/2504.11343.

[8] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

[9] Ziniu Li, Congliang Chen, Tianyun Yang, Tian Ding, Ruoyu Sun, Ge Zhang, Wenhao Huang, and Zhi-Quan Luo. Knapsack rl: Unlocking exploration of llms via optimizing budget allocation, 2025. URL https://arxiv.org/abs/2509.25849.

[10] 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, 2026. URL https://arxiv.org/abs/2509.25666.

[11] Kimi Team, Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Chenxiao Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Yuyao Ge, Shangyi Geng, Qizheng Gu, Xinran Gu, Longyu Guan, Haiqing Guo, Jianhang Guo, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Yunjia He, Chao Hong, Hao Hu, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li, Haoyang Li, Ming Li, Wentao Li, Yang Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Haoyu Lu, Lijun Lu, Yashuo Luo, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Zeyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan Su, Lin Sui, Xinjie Sun, Flood Sung, Yunpeng Tai, Heyi Tang, Jiawen Tao, Qifeng Teng, Chaoran Tian, Chensi Wang, Dinglu Wang, Feng Wang, Hailong Wang, Haiming Wang, Jianzhou Wang, Jiaxing Wang, Jinhong Wang, Shengjie Wang, Shuyi Wang, Si Wang, Xinyuan Wang, Yao Wang, Yejie Wang, Yiqin Wang, Yuxin Wang, Yuzhi Wang, Zhaoji Wang, Zhengtao Wang, Zhengtao Wang, Zhexu Wang, Chu Wei, Qianqian Wei, Haoning Wu, Wenhao Wu, Xingzhe Wu, Yuxin Wu, Chenjun Xiao, Jin Xie, Xiaotong Xie, Weimin Xiong, Boyu Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ziyao Xu, Jing Xu, Jing Xu, Junjie Yan, Yuzi Yan, Hao Yang, Xiaofei Yang, Yi Yang, Ying Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Xingcheng Yao, Wenjie Ye, Zhuorui Ye, Bohong Yin, Longhui Yu, Enming Yuan, Hongbang Yuan, Mengjie Yuan, Siyu Yuan, Haobing Zhan, Dehao Zhang, Hao Zhang, Wanlu Zhang, Xiaobin Zhang, Yadong Zhang, Yangkun Zhang, Yichi Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Haotian Zhao, Yikai Zhao, Zijia Zhao, Huabin Zheng, Shaojie Zheng, Longguang Zhong, Jianren Zhou, Xinyu Zhou, Zaida Zhou, Jinguo Zhu, Zhen Zhu, Weiyu Zhuang, and Xinxing Zu. Kimi k2: Open agentic intelligence, 2026. URL https://arxiv.org/abs/2507.20534.

[12] Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. On the generalization of sft: A reinforcement learning perspective with reward rectification, 2026. URL https://arxiv.org/abs/2508.05629.

[13] Thanh-Long V. Le, Myeongho Jeon, Kim Trong Vu, Viet Dac Lai, and Eunho Yang. No prompt left behind: Exploiting zero-variance prompts in llm reinforcement learning via entropy-guided advantage shaping, 2026. URL https://arxiv.org/abs/2509.21880. ICLR 2026.

[14] Hieu Trung Nguyen, Bao Nguyen, Wenao Ma, Yuzhi Zhao, Ruifeng She, and Viet Anh Nguyen. Adaptive rollout allocation for online reinforcement learning with verifiable rewards, 2026. URL https://arxiv.org/abs/2602.01601. ICLR 2026.

[15] Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, and Jiang Bian. Self-hinting language models enhance reinforcement learning, 2026. URL https://arxiv.org/abs/2602.03143.

[16] Vaskar Nath, Elaine Lau, Anisha Gunjal, Manasi Sharma, Nikhil Baharte, and Sean Hendryx. Adaptive guidance accelerates reinforcement learning of reasoning models, 2025. URL https://arxiv.org/abs/2506.13923.

[17] Peter Chen, Xiaopeng Li, Ziniu Li, Xi Chen, and Tianyi Lin. Stepwise guided policy optimization: Coloring your incorrect reasoning in grpo, 2026. URL https://arxiv.org/abs/2505.11595.

[18] Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhoufutu Wen, Chenghua Lin, Wenhao Huang, Qian Liu, Ge Zhang, and Zejun Ma. First return, entropy-eliciting explore, 2025. URL https://arxiv.org/abs/2507.07017.

[19] Xiaoyuan Liu, Tian Liang, Zhiwei He, Jiahao Xu, Wenxuan Wang, Pinjia He, Zhaopeng Tu, Haitao Mi, and Dong Yu. Trust, but verify: A self-verification approach to reinforcement learning with verifiable rewards, 2025. URL https://arxiv.org/abs/2505.13445. NeurIPS 2025.

[20] Ishaan Gangwani and Aayam Bansal. Rubric as reward: Decomposing verification signals for logical reasoning in grpo. ICLR 2026 Workshop on LLM Reasoning, 2026. URL https://openreview.net/forum?id=LFBnQEGh23.

[21] Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026. URL https://arxiv.org/abs/2601.19897.

[22] Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, Maosong Sun, and Tat-Seng Chua. Rlpr: Extrapolating rlvr to general domains without verifiers, 2025. URL https://arxiv.org/abs/2506.18254.

[23] Nam Le Hai, Anh Duc Le, Tu Vu, and Linh Van Ngo. Discord: Enhancing knowledge distillation via cross-chain of thought and optimal transport alignment between models with different tokenizers. *Knowl. Based Syst.*, 351:116606, 2026. doi: 10.1016/J.KNOSYS.2026.116606. URL https://doi.org/10.1016/j.knosys.2026.116606.

[24] Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10495–10516. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.547. URL https://aclanthology.org/2025.findings-acl.547/.

[25] Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models. *arXiv preprint arXiv:2505.07686*, 2025. URL https://arxiv.org/abs/2505.07686.

[26] Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*, 2024. URL https://arxiv.org/abs/2412.01981.

[27] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025. URL https://arxiv.org/abs/2502.01456.

[28] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning, 2025. URL https://arxiv.org/abs/2506.01939.

[29] Jean Vassoyan, Nathanael Beau, and Roman Plaud. Ignore the kl penalty! boosting exploration on critical tokens to enhance rl fine-tuning, 2025. URL https://arxiv.org/abs/2502.06533.

[30] Truong Nguyen, Tien-Phat Nguyen, Linh Ngo Van, Duy Minh Ho Nguyen, Khoa D. Doan, and Trung Le. Tokenratio: Principled token-level preference optimization via ratio matching. *CoRR*, abs/2605.12288, 2026. doi: 10.48550/ARXIV.2605.12288. URL https://doi.org/10.48550/arXiv.2605.12288.

[31] Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. Preserving diversity in supervised fine-tuning of large language models, 2025. URL https://arxiv.org/abs/2408.16673.

[32] Yijie Chen, Yijin Liu, and Fandong Meng. SED-SFT: Selectively encouraging diversity in supervised fine-tuning, 2026. URL https://arxiv.org/abs/2602.07464.

[33] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu,

Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[34] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[35] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[36] Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.

[37] MAA Committees. Aime problems and solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions, 2024. Art of Problem Solving.

[38] MAA Committees. Aime problems and solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions, 2025. Art of Problem Solving.

[39] Mathematical Association of America. American mathematics competitions (amc) 2023. https://maa.org, 2023. AMC 8, AMC 10, and AMC 12 competition problems.

[40] 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 Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

[41] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. 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 Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/18abbeef8cfe9203fdf9053c9c4fe191-Abstract-Conference.html.

[42] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.211. URL https://doi.org/10.18653/v1/2024.acl-long.211.

[43] 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. *CoRR*, abs/2311.12022, 2023. doi: 10.48550/ARXIV.2311.12022. URL https://doi.org/10.48550/arXiv.2311.12022.

[44] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December*

10 - 15, 2024, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract-Datasets_and_Benchmarks_Track.html.

[45] 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*, EuroSys '25, page 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL http://dx.doi.org/10.1145/3689031.3696075.

[46] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

[47] 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.

[48] 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.

[49] Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of llms?, 2026. URL https://arxiv.org/abs/2603.24472.

A Algorithm

Algorithm 1 GRPO with SORT Auxiliary Reference Updates

Require: Policy parameters θ; rollout batch size B_{rl}; rollouts per prompt N; auxiliary batch size B_{aux}; empty buffer $\mathcal{B}_0$

for $k = 1, \ldots, K$ **do**
  Sample rollout batch $\mathcal{U}_k = \{q_i\}_{i=1}^{B_{\mathrm{rl}}}$
  for $i = 1, \ldots, B_{\mathrm{rl}}$ **do**
    Sample $\{y_i^{(n)}\}_{n=1}^{N} \sim \pi_\theta(\cdot \mid q_i)$
    Compute verifiable rewards $\{r_i^{(n)}\}_{n=1}^{N}$
  end for
  Compute GRPO advantages and update θ using $\mathcal{L}_{\mathrm{GRPO}}(\mathcal{U}_k)$
  $\mathcal{I}_0^{(k)} \leftarrow \{\, i : r_i^{(n)} = 0,\ \forall n \in [N] \,\}$
  Enqueue $\{(q_i, y_i^\star) : i \in \mathcal{I}_0^{(k)}\}$ into $\mathcal{B}_0$
  if $|\mathcal{B}_0| \geq B_{\mathrm{aux}}$ **then**
    Sample and remove $\mathcal{M}_k \sim \mathrm{SampleBatch}(\mathcal{B}_0, B_{\mathrm{aux}})$
    for all $(q_i, y_i^\star) \in \mathcal{M}_k$ **do**
      $\hat{s}_i \leftarrow \mathrm{Gen}_\theta(\textsc{PlanExtract}(y_i^\star))$
      Compute base probabilities $p_{i,t}^{\mathrm{base}}$ along $y_i^\star$
      Compute plan-conditioned probabilities $p_{i,t}^{\mathrm{plan}}$ along $y_i^\star$
      $\omega_{i,t} \leftarrow \mathrm{sg}\left(\sqrt{p_{i,t}^{\mathrm{base}} p_{i,t}^{\mathrm{plan}}}\right)$ for all t
    end for
    Update θ using $\mathcal{L}_{\mathrm{SORT}}(\mathcal{M}_k)$
  end if
end for

B Limitations & Broader Impacts

Limitations. First, SORT assumes a verified reference solution is available for each zero-reward prompt. In domains where ground-truth solutions are expensive to annotate, the auxiliary update cannot be applied and the method reduces to vanilla GRPO on those prompts; our evaluation is also limited to mathematical reasoning and knowledge benchmarks, and transfer to other modalities such as code generation remains to be tested. Second, like GRPO, SORT assumes verifiable binary outcome rewards. Extending the auxiliary repair mechanism to continuous, learned, or process-based reward signals is not addressed in this work.

Broader impacts. SORT improves the sample efficiency of RL-based reasoning training by converting previously wasted zero-reward prompts into structured learning signals. This has a positive practical impact: better reasoning ability from the same amount of training data, which reduces the compute and environmental cost of running large-scale RLVR pipelines. The method is general and could benefit any application where verifiable rewards and reference solutions are available, including formal theorem proving, code verification, and scientific reasoning.

On the potential negative side, SORT makes it easier to fine-tune language models toward stronger reasoning with less data, lowering the barrier for both beneficial and harmful uses. Improved mathematical reasoning could be leveraged for tasks such as automated vulnerability discovery or optimization of harmful processes. However, this is a general risk shared by virtually all LLM reasoning improvements, not one specific to SORT. The method does not introduce new failure modes beyond those already present in GRPO and reference-based fine-tuning. We see no reason why SORT would disproportionately increase misuse risk relative to other RLVR training techniques, and we encourage practitioners to apply standard responsible-release practices when deploying models trained with this method.

C Discussion

Relation to self-distillation. Recent work suggests that self-distillation is a promising way to improve reasoning models without relying on a stronger external teacher [21, 47, 48]. However, Kim et al. [49] show that self-distillation can mainly reinforce tokens that are already easy to predict from the previous context, while giving weaker signal to uncertain tokens. This is important for reasoning traces: many tokens, such as arithmetic continuations, algebraic simplifications, formatting tokens, or short connective phrases, are often predictable from the local prefix. Imitating these tokens is easy but provides little new reasoning signal.

Our method is designed to avoid over-emphasizing such routine tokens. If a token can already be predicted from the prefix, adding the plan changes its likelihood only slightly, so its salience remains small and it receives roughly ordinary weight. In contrast, tokens corresponding to uncertain reasoning choices, such as selecting a case split, introducing a key variable, applying a theorem, or moving to a new subgoal, can become much more predictable once the plan is provided. These tokens receive larger weights. Thus, rather than distilling the entire chain-of-thought uniformly, our objective keeps routine tokens near their normal weighting while upweighting the uncertain tokens that are made predictable by the reasoning plan.

Zero-reward prompts. This behavior is especially useful for zero-reward prompts, where all on-policy rollouts fail and GRPO provides no useful learning signal. A direct reference-imitation loss may still mostly train the model on easy parts of the solution, because those tokens dominate the trace and are locally predictable. Our method instead uses the reference solution to identify which tokens in the failed prompt correspond to uncertain reasoning steps. As shown in Section J, tokens that are predictable from the prefix have near-zero salience, while uncertain tokens within the same solution step receive higher weights when plan guidance makes them predictable. This allows the model to learn from the reference without simply copying the full solution trace and lead to entropy collapse while training.

Implication. From this perspective, our method can be viewed as a selective form of self-distillation for reasoning. Rather than treating the model's or reference's entire chain-of-thought as equally useful supervision, we use plan-conditioned likelihood changes to separate routine continuation tokens from

structurally informative tokens. This provides a simple mechanism for preserving the benefits of self-distillation while mitigating its tendency to over-emphasize easy tokens and under-emphasize uncertain reasoning steps.

To make the zero-reward regime in Section 3 concrete, we show two examples on which Qwen2.5-7B-Instruct fails to produce any correct rollout in the sampled group. They illustrate prompts where the current policy misses the global reasoning structure needed to reach a verifiable answer.

Example 1: adaptive geometric search on a checkerboard. *Qwen2.5-7B-Instruct outcome: zero-reward rollout group; ground-truth answer: 2.*

> Bakayev E.V. Petya drew a square on the plane, divided it into 64 identical smaller squares, and colored them in a checkerboard pattern with black and white colors. After that, he thought of a point strictly inside one of these smaller squares. Vasya can draw any closed broken line without self-intersections on the plane and get an answer to the question of whether the guessed point is strictly inside the broken line or not. What is the minimum number of such questions Vasya needs to determine the color of the guessed point—white or black?

This prompt is hard for sparse-reward RL because a correct answer depends on designing an adaptive querying strategy and proving optimality. A partially correct rollout can identify the board structure or propose a query, but the verifier only rewards the final minimum and its justification. Below is a representative Qwen2.5-7B-Instruct rollout that answers 1 (correct is 2): the model reduces the problem to parity of grid coordinates but overlooks that one question alone cannot distinguish both colors from the exterior region.

Model output (incorrect).

> First, note that the square is divided into $8 \times 8 = 64$ smaller squares in a checkerboard pattern. The color of a square depends on the parity of the sum of its row and column indices. Specifically, if $(i + j) \bmod 2 = 0$ the square is black, otherwise it is white. The key observation is that if we can determine the parity of $(i + j)$ for the square containing the point, we know the color.
>
> Vasya can draw a closed broken line that separates squares with even $i + j$ from those with odd $i + j$. For instance, a line running from $(1, 1)$ to $(8, 8)$ partitions the board by parity. By asking whether the point is inside this region, Vasya learns the parity and therefore the color.
>
> Thus, the minimum number of questions is $\boxed{1}$.

The rollout misses that one query cannot reliably separate the two parity classes while also handling the exterior region, so the answer requires at least two adaptive questions. In the observed Qwen2.5-7B-Instruct rollouts, this global query strategy is not recovered, so the group receives no positive reward.

Example 2: worst-case color separation by lines. *Qwen2.5-7B-Instruct outcome: zero-reward rollout group; ground-truth answer: 2013.*

> In the plane, 2013 red points and 2014 blue points are marked so that no three of the marked points are collinear. One needs to draw k lines not passing through the marked points and dividing the plane into several regions. The goal is to do it in such a way that no region contains points of both colors. Find the minimal value of k such that the goal is attainable for every possible configuration of 4027 points. (Australia)

This prompt combines extremal geometry with a worst-case quantifier over all configurations. A successful solution must provide both a construction showing sufficiency and an adversarial configuration showing necessity. Below is a representative Qwen2.5-7B-Instruct rollout that answers 2015 (correct is 2013): the model reasons about separating convex hulls but loses the tight count on the required number of lines.

Model output (incorrect).

First, consider the convex hull of all the points. Since no three points are collinear, the convex hull is a polygon with at most 4027 vertices (the maximum when all points lie on the hull).

The key insight is that we can separate the convex hull into two regions by drawing lines such that one region contains only red points and the other only blue points. This is done by drawing lines that pass between the convex hull and the rest of the points.

To ensure no region contains points of both colors, we need to separate the convex hull from the remaining points. Each side of the convex hull can be "cut" by one line, requiring at most vertices $- 2 = 4027 - 2 = 4025$ lines, plus one additional line to guarantee complete separation, yielding 4026 lines. However, a tighter analysis using the convex hull of the minority color (2013 red points) gives 2014 lines to cut around them, plus one final separator, for a total of 2015.

Thus, the minimal value of k is $\boxed{2015}$.

The rollout uses convex-hull reasoning in the right spirit but mishandles the counting: the correct upper-bound construction uses 2013 lines, each separating one red point from all blue points, and the matching lower bound exhibits a configuration where fewer than 2013 lines cannot suffice. In the observed Qwen2.5-7B-Instruct rollouts, local geometric intuitions lead to plausible but unverifiable partial arguments, and the full upper- and lower-bound structure needed for the answer is absent.

D Example Ground-Truth Solution Plan

For the logarithmic equation

$$\log_{\sqrt{a}} \frac{\sqrt{2a-x}}{a} - \log_{1/a} x = 0,$$

the plan extraction prompt produces a thinking process that captures the key reasoning decisions, self-checks, and intermediate realizations:

First, I need to handle the domain of the logarithms. For the first log to be defined, the base $\sqrt{a} > 0$, $\sqrt{a} \neq 1$, so $a > 0$, $a \neq 1$. The argument $\frac{\sqrt{2a-x}}{a} > 0$ — since $a > 0$, this means $\sqrt{2a-x} > 0$, so $2a - x > 0$, giving $x < 2a$. For $\log_{1/a} x$, the base $1/a > 0$ and $1/a \neq 1$ are already covered by $a > 0$, $a \neq 1$, and the argument gives $x > 0$. So domain: $a > 0$, $a \neq 1$, $0 < x < 2a$.

Now to solve the equation. The bases $\sqrt{a}$ and $1/a$ are inconvenient — let me rewrite them as powers of a. $\sqrt{a} = a^{1/2}$, $1/a = a^{-1}$.

Using the change of base formula: $\log_{a^{1/2}} u = \frac{\log_a u}{\log_a a^{1/2}} = \frac{\log_a u}{1/2} = 2 \log_a u$. So the first term is $2 \log_a(\frac{\sqrt{2a-x}}{a})$.

Wait — actually, let me be more careful. $\log_{\sqrt{a}} \frac{\sqrt{2a-x}}{a}$. Using base change: $\log_{a^{1/2}} u = 2 \log_a u$. With $u = \frac{\sqrt{2a-x}}{a} = \frac{\sqrt{2a-x}}{a}$, this is $\log_a \frac{\sqrt{2a-x}}{a} = \log_a \sqrt{2a-x} - \log_a a = \frac{1}{2} \log_a(2a-x) - 1$.

Hmm — so doubled: $2(\frac{1}{2} \log_a(2a-x) - 1) = \log_a(2a-x) - 2$.

Now the second term: $\log_{1/a} x = \log_{a^{-1}} x = \frac{\log_a x}{\log_a a^{-1}} = \frac{\log_a x}{-1} = -\log_a x$.

The equation becomes: $(\log_a(2a-x) - 2) - (-\log_a x) = 0$, so $\log_a(2a-x) - 2 + \log_a x = 0$.

Combine: $\log_a[x(2a-x)] = 2$.

Exponentiate: $x(2a-x) = a^2$, so $2ax - x^2 = a^2$, giving $x^2 - 2ax + a^2 = 0$.

This factors as $(x-a)^2 = 0$, so $x = a$.

Check against the domain: $a > 0$, $a \neq 1$, and $0 < a < 2a$ which holds for all $a > 0$. So $x = a$ is valid. Final answer: $x = a$.

The extracted plan is more conversational than a polished solution but exposes the global decisions: domain analysis, base conversion with a self-check, logarithm combination, quadratic reduction, and final domain verification. These are precisely the tokens whose probabilities should rise under plan conditioning.

E Qualitative Plan Extraction Example

We show a representative example of the plan-extraction step used before plan-conditioned scoring. The prompt asks for a two-digit natural number $\overline{ab}$ satisfying

$$\sqrt{a+\sqrt{\overline{ab}}} = a.$$

The full reference solution contains both high-level structural reasoning and lower-level surface realization, including intermediate checks, self-corrections, and verification steps:

> Okay, so I need to find a two-digit natural number $\overline{ab}$ such that
>
> $$\sqrt{a+\sqrt{\overline{ab}}} = a.$$
>
> Since $\overline{ab}$ is a two-digit number, a is the tens digit and b the units digit, so
>
> $$\overline{ab} = 10a + b,$$
>
> with $a \in \{1, \ldots, 9\}$ and $b \in \{0, \ldots, 9\}$.
> Substitute into the equation:
>
> $$\sqrt{a+\sqrt{10a+b}} = a.$$
>
> Square both sides:
>
> $$a + \sqrt{10a+b} = a^2.$$
>
> Subtract a:
>
> $$\sqrt{10a+b} = a^2 - a.$$
>
> Square again:
>
> $$10a + b = (a^2 - a)^2.$$
>
> Expand:
>
> $$(a^2 - a)^2 = a^4 - 2a^3 + a^2,$$
>
> so
>
> $$10a + b = a^4 - 2a^3 + a^2.$$
>
> Rearranging,
>
> $$b = a^4 - 2a^3 + a^2 - 10a.$$
>
> Since b must be a digit from 0 to 9, test values of a.
> For $a = 1$,
>
> $$b = 1 - 2 + 1 - 10 = -10,$$
>
> which is invalid.
> For $a = 2$,
>
> $$b = 16 - 16 + 4 - 20 = -16,$$
>
> also invalid.
> For $a = 3$,
>
> $$b = 81 - 54 + 9 - 30.$$
>
> Wait, $81 - 54 = 27$, then $27 + 9 = 36$, and $36 - 30 = 6$. So $b = 6$, which is a valid digit.
> Thus the candidate number is 36. Check:
>
> $$\sqrt{3+\sqrt{36}} = \sqrt{3+6} = \sqrt{9} = 3.$$
>
> Now check whether larger values of a can work.
> For $a = 4$,
>
> $$b = 256 - 128 + 16 - 40 = 104,$$
>
> which is too large to be a digit.
> For $a = 5$,
>
> $$b = 625 - 250 + 25 - 50 = 350,$$
>
> also impossible. Larger a values only make b larger.
> Therefore the unique solution is
>
> $$\boxed{36}.$$

The extracted thinking process preserves the key mathematical structure and genuine reasoning transitions while removing most repetitive exposition and stylistic filler:

> Find a two-digit number $\overline{ab}$ such that
>
> $$\sqrt{a+\sqrt{\overline{ab}}} = a.$$
>
> Since $\overline{ab}$ is two-digit,
>
> $$\overline{ab} = 10a + b,$$
>
> so
>
> $$\sqrt{a+\sqrt{10a+b}} = a.$$
>
> Square both sides:
>
> $$a + \sqrt{10a+b} = a^2, \qquad \sqrt{10a+b} = a^2 - a.$$
>
> Square again:
>
> $$10a + b = (a^2 - a)^2.$$
>
> Expand:
>
> $$(a^2 - a)^2 = a^4 - 2a^3 + a^2,$$
>
> hence
>
> $$b = a^4 - 2a^3 + a^2 - 10a.$$
>
> Since b must be a digit, test values of a.
> For $a = 1$,
>
> $$b = -10,$$
>
> invalid.
> For $a = 2$,
>
> $$b = -16,$$
>
> also invalid.
> For $a = 3$,
>
> $$b = 81 - 54 + 9 - 30.$$
>
> Wait, $81 - 54 = 27$, then $27 + 9 = 36$, and $36 - 30 = 6$. So $b = 6$.
> Check:
>
> $$\sqrt{3+\sqrt{36}} = \sqrt{3+6} = \sqrt{9} = 3.$$
>
> For $a \geq 4$, b becomes too large to be a digit. Therefore,
>
> $$\overline{ab} = 36.$$

This example illustrates that plan extraction retains the essential solution trajectory, intermediate calculations, and moments of genuine correction, while discarding verbose explanatory narration that does not materially affect the reasoning structure.

F Prompt Templates for Plan Extraction and Teacher Conditioning

We follow the prompting setup of Yan et al. [3] and use the same system prompts for SORT and all baseline methods. Specifically, for Qwen backbones we adopt the longer LUFFY reasoning prompt, while for the LLaMA-3.2-3B backbone we use a simplified step-by-step prompt, as the model does not reliably follow the longer instruction.

System prompt (Qwen backbones).

```
Your task is to follow a systematic, thorough reasoning
process before providing the final solution.  This involves
analyzing, summarizing, exploring, reassessing, and refining
your thought process through multiple iterations.  Structure
your response into two sections:  Thought and Solution.  In
the Thought section, present your reasoning using the format:
<think> thoughts </think>.  Each thought should include
```

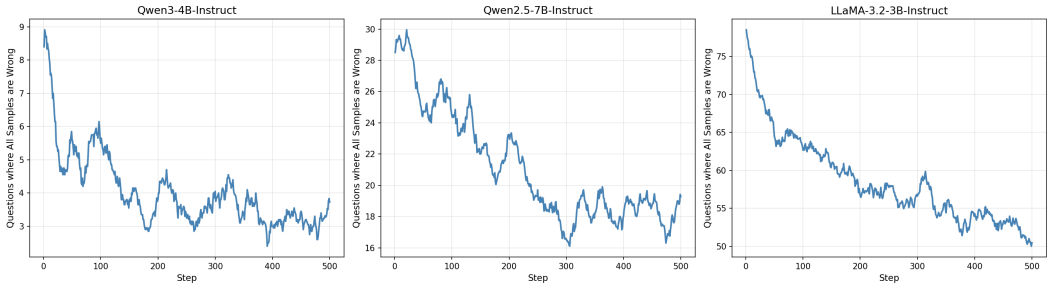

Figure 3: Number of all-wrong rollout groups during SORT training, where every sampled trajectory receives zero reward. Lower values indicate fewer zero-reward prompts.

```
detailed analysis, brainstorming, verification, and refinement
of ideas.  After </think>, in the Solution section, provide
the final, logical, and accurate answer, clearly derived
from the exploration in the Thought section.  If applicable,
include the answer in boxed for closed-form results like
multiple choices or mathematical solutions.
```

System prompt (LLaMA-3.2-3B-Instruct).

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

Plan extraction prompt.

```
You are given a detailed reasoning process.  Extract only
the key thinking steps verbatim, including intermediate
calculations, dead ends, self-corrections, and the intuition
behind each key realization.  Include moments of uncertainty
only when they naturally arise.  Do not add decorative
explanations.

Reasoning:  {solution}

Thinking process:
```

Teacher prompt.

```
{problem}
Here is the thinking process to solve this problem:
{blueprint}
```

G All-Wrong Sample Fraction During SORT Training

We additionally track the number of *all-wrong samples* during SORT training, i.e., rollout groups where every sampled trajectory receives zero reward under GRPO. Concretely, for a prompt q_i, this corresponds to

$$r_i^{(1)} = \cdots = r_i^{(N)} = 0,$$

which is exactly the zero-reward regime targeted by SORT in Section 3.1. Figure 3 shows the evolution of the number of all-wrong prompts across training for Qwen3-4B-Instruct, Qwen2.5-7B-Instruct, and Llama-3.2-3B-Instruct.

Several trends support the motivation of SORT. Weaker models exhibit substantially more all-wrong rollout groups, with Llama-3.2-3B-Instruct starting near 80 zero-reward prompts per batch, Qwen2.5-7B-Instruct around 30, and Qwen3-4B-Instruct below 10. Across all backbones, the number of all-wrong groups decreases steadily during training, indicating that RL optimization gradually expands the set of prompts for which at least one successful trajectory can be sampled. The largest reductions occur for weaker models, matching the observation in Table 1 that SORT provides the strongest gains in low-capability regimes. At the same time, residual zero-reward prompts persist even for stronger models, suggesting that SORT-style hard-prompt repair remains useful throughout training rather than only in the early stages.

This extracted plan is not used as a target sequence. It is only prepended to the teacher context when scoring the original reference tokens. As a result, tokens corresponding to structural operations–substituting $\overline{ab} = 10a + b$, squaring twice, deriving the digit constraint on b, and verifying 36–become more predictable under plan conditioning. Routine phrasing and repeated arithmetic receive little additional support, which is the behavior used by SORT to distinguish plan-informative tokens from surface tokens. In terms of Assumption 3, this is a favorable case: the extracted plan preserves the reasoning state needed to predict the reference path, so its induced hidden states should be close to those under the ideal plan. If the extractor dropped the digit constraint or the verification step, the corresponding γ would be larger and the regret bound in Theorem 6 would become less informative.

H Runtime and Compute Overhead

On Qwen2.5-7B-Instruct with 8 H100 GPUs and matched rollout budgets and training steps, GRPO takes 19.0 hours and SORT approximately 29.5 hours: an additional 10.5 hours, or about 55%. Table 6 compares the relative costs with the guidance-based baselines. For buffered all-wrong prompts, SORT generates a plan, scores the verified reference under both base and plan-conditioned contexts, and performs one auxiliary update when the buffer fires. The plan-conditioned branch is detached; gradients flow only through the base branch. This adds training computation while leaving rollout generation unchanged, and the plan is not used at inference time.

Table 6: Training time for Qwen2.5-7B-Instruct on 8 H100 GPUs with matched rollout budgets and training steps. Times are relative to GRPO; absolute measurements are included where available.

Method	**GRPO**	**LUFFY**	**ReLIFT**	**Scaf-GRPO**	**SORT**
Training time	1.0x (19.0h)	1.2x	1.95x	1.5x	1.55x (29.5h)

I SFT Loss Dynamics

Figure 4 reports the SFT-path loss over training for the auxiliary reference-learning variants in Section 4.4. Uniform SFT variants keep a substantially higher loss because they place learning pressure on the whole reference trace, including routine algebra, formatting, and already-predictable tokens. The $\beta = 0$ setting, corresponding to DFT-style student weighting, yields the smallest loss by downweighting reference tokens according to base-model support. SORT ($\beta = 1/2$) stays close to this low-loss DFT regime but is intentionally above it: plan conditioning selectively boosts structurally important tokens, adding targeted learning pressure without returning to full-trace imitation. This diagnostic complements Tables 2 and 3: it shows how the reference objective behaves during optimization, whereas the tables report downstream evaluation accuracy.

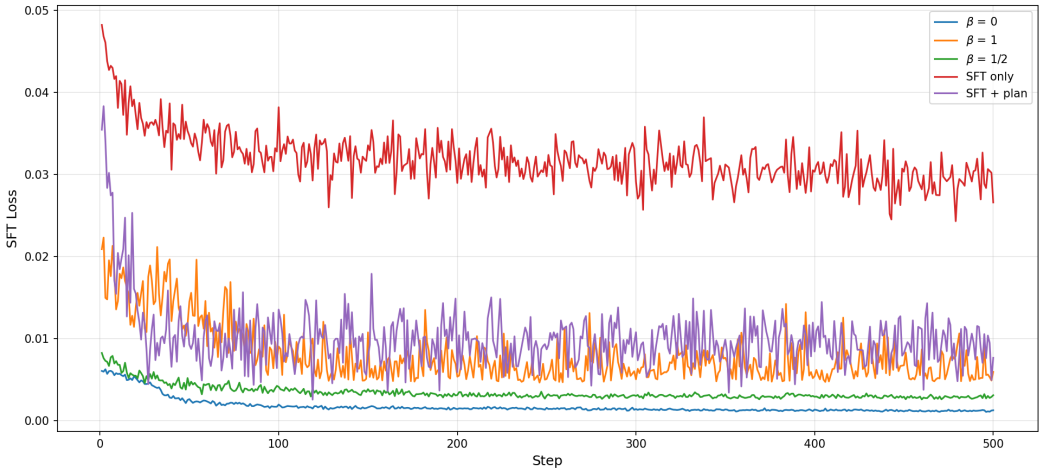

Figure 4: SFT-path loss trajectories across 500 training steps for the auxiliary reference-learning variants on Qwen2.5-7B-Instruct. Uniform SFT baselines maintain high loss because they learn from all reference tokens. DFT-style weighting ($\beta = 0$) gives the lowest loss, while SORT ($\beta = 1/2$) remains near DFT but higher due to selective boosting of plan-informative tokens.

J Full Token Salience Visualization

Figure 5–8 show the full token-level visualization corresponding to the excerpt in Figure 2. Each cell is one token from the reference trace, colored by the plan-conditioned log-ratio

$$\hat{\sigma}_{i,t} = \log p_{i,t}^{\text{plan}} - \log p_{i,t}^{\text{base}}.$$

Redder tokens become more predictable under plan conditioning, while bluer tokens become less predictable.

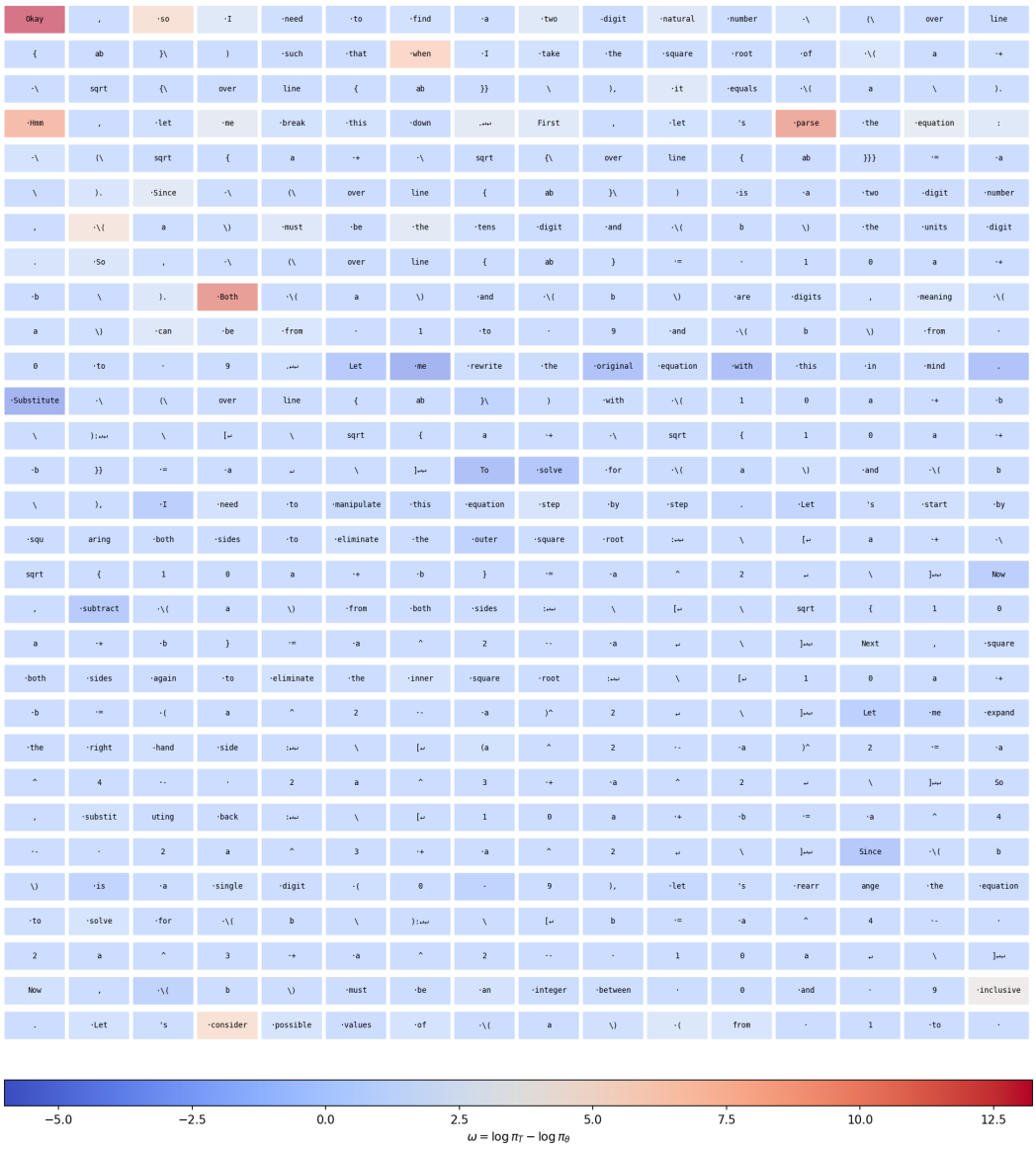

Figure 5: Full token salience visualization, part 1.

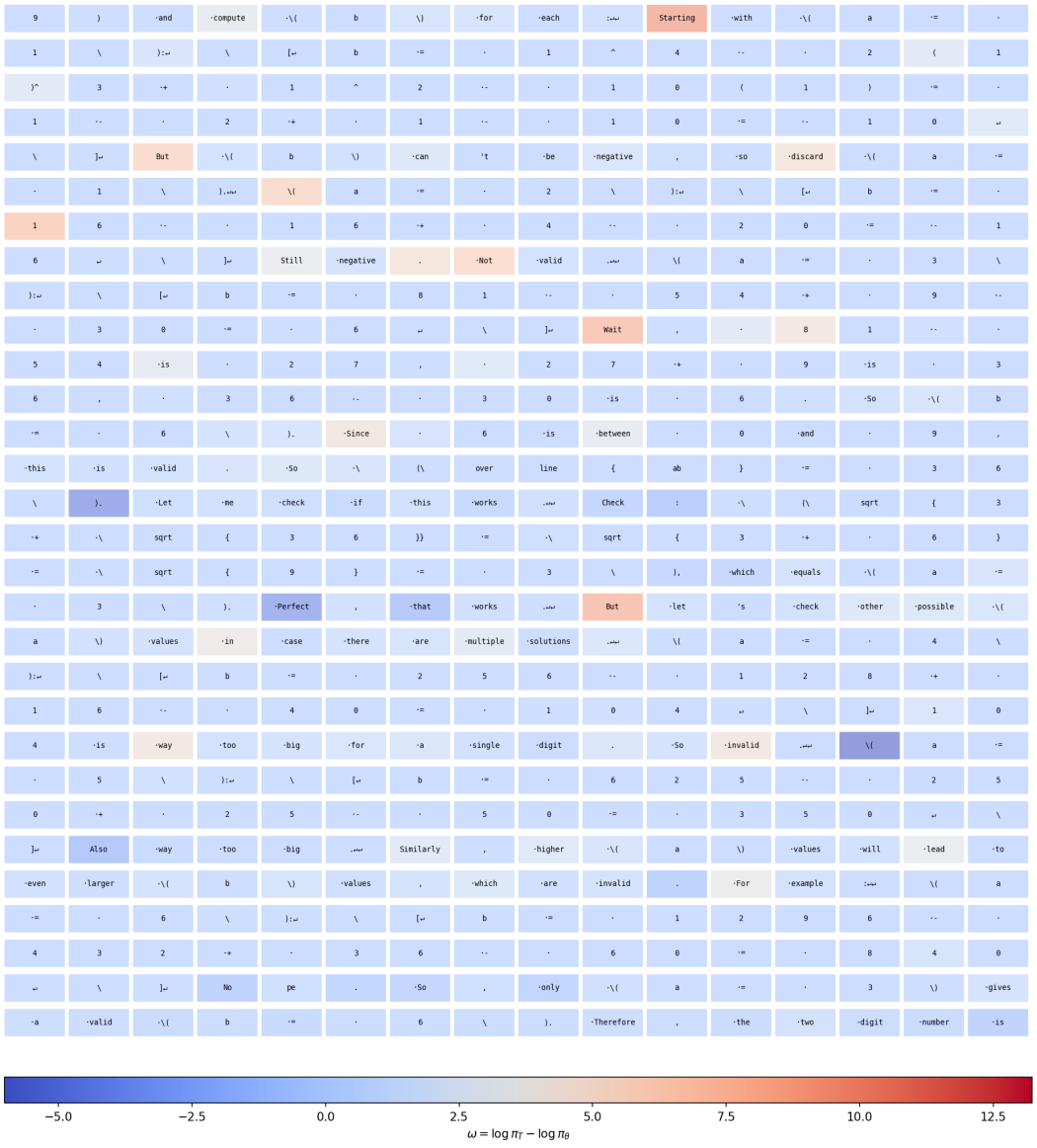

Figure 6: Full token salience visualization, part 2.

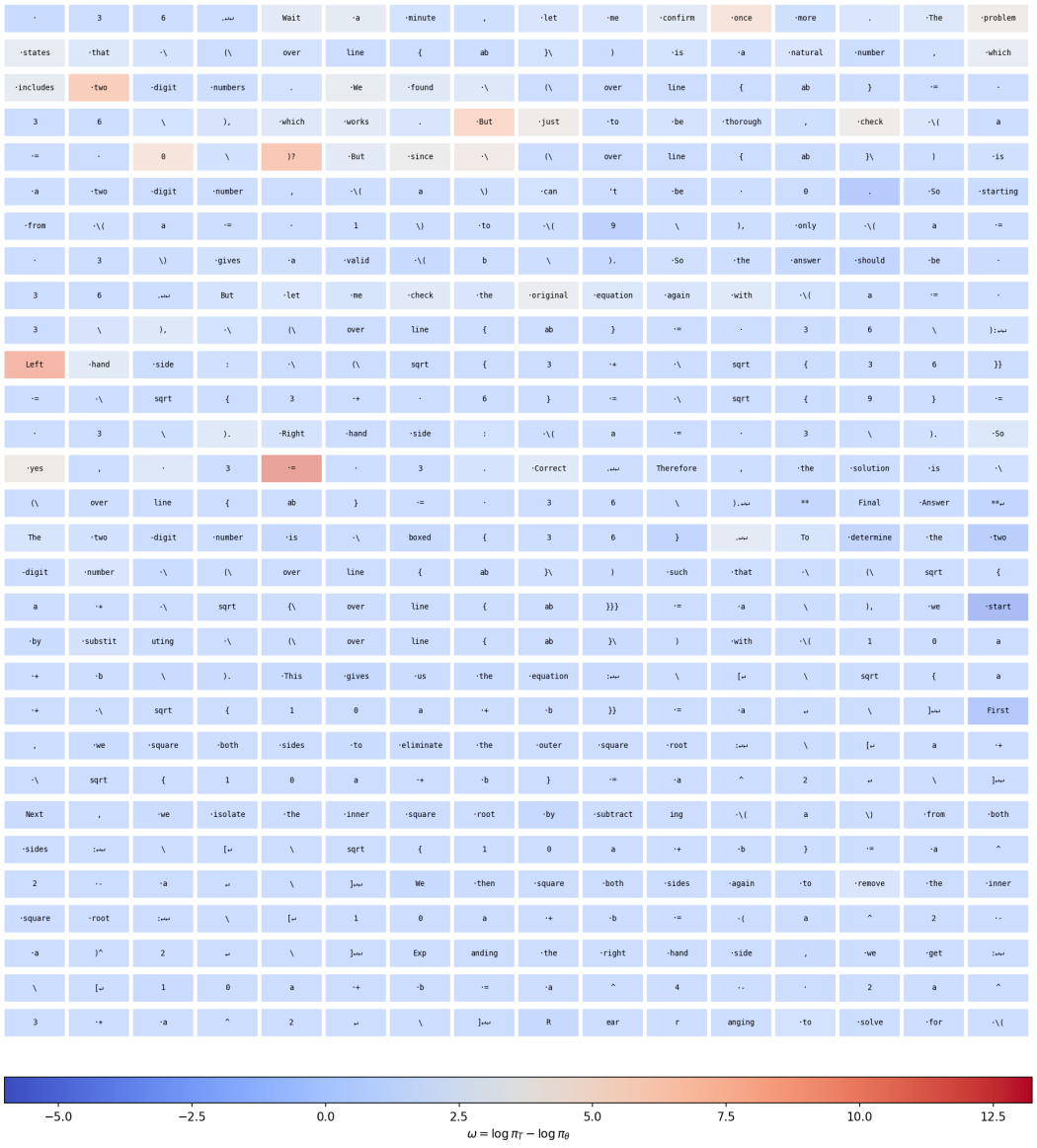

Figure 7: Full token salience visualization, part 3.

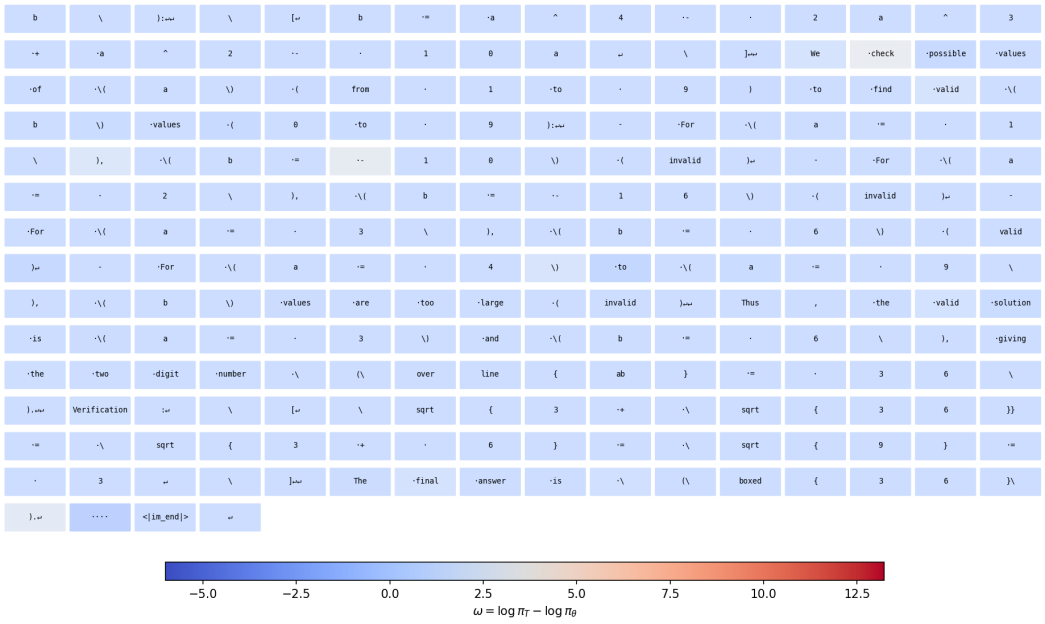

Figure 8: Full token salience visualization, part 4.

K Proofs for Section 3.4

Definitions 1–2, the oracle weight (12), and Assumptions 3–4 are given in the main text.

Proof of Lemma 5. Fix any token position t. By the L-Lipschitz assumption on $h \mapsto \log \pi_\theta(\cdot \mid h)$,

$$\left|\log \pi_\theta(y^\star_{i,t} \mid \hat{s}_i, q_i, y^\star_{i,<t}) - \log \pi_\theta(y^\star_{i,t} \mid S_i, q_i, y^\star_{i,<t})\right| \le L \left\|h_\theta(q_i, y^\star_{i,<t}, \hat{s}_i) - h_\theta(q_i, y^\star_{i,<t}, S_i)\right\|.$$

Assumption 3 bounds the right-hand side by $L\gamma$ pointwise for all t. □

Proof of Theorem 6. Let $p_{i,t} = p^{\text{base}}_{i,t}$ and $q_{i,t} = p^{\text{plan}}_{i,t}$ as defined in (7), and let $o_{i,t} = \pi_\theta(y^\star_{i,t} \mid S_i, q_i, y^\star_{i,<t})$ be the oracle plan-conditioned probability. All lie in $(0, 1]$, so their logs are non-positive. From (14) and (12),

$$\omega_{\beta,i,t} = p_{i,t}^{1-\beta} q_{i,t}^{\beta}, \qquad \omega^\star_{\beta,i,t} = p_{i,t}^{1-\beta} o_{i,t}^{\beta}.$$

Write $\ell_q = \log q_{i,t}$, $\ell_o = \log o_{i,t}$. Then

$$\left|\omega_{\beta,i,t} - \omega^\star_{\beta,i,t}\right| = p_{i,t}^{1-\beta} \cdot \left|e^{\beta \ell_q} - e^{\beta \ell_o}\right| \le p_{i,t}^{1-\beta} \cdot \beta \left|\ell_q - \ell_o\right| \le \beta L \gamma \, p_{i,t}^{1-\beta}, \tag{19}$$

where the first inequality uses the mean value theorem on $f(x) = e^{\beta x}$: $|f(a) - f(b)| = \beta e^{\beta \xi} |a - b| \le \beta |a - b|$ since $\beta \in [0, 1]$ and $\xi \le \max(a, b) \le 0$. The second inequality applies Lemma 5 pointwise, noting that $|\ell_q - \ell_o| = |\log p^{\text{plan}}_{i,t} - \log \pi_\theta(y^\star_{i,t} \mid S_i, q_i, y^\star_{i,<t})| \le L\gamma$. For $\beta = \frac{1}{2}$ this gives $|\omega_{i,t} - \omega^\star_{i,t}| \le (L\gamma/2)\sqrt{p_{i,t}}$.

For the gradient: $\nabla_\theta \mathcal{L}_{\text{SORT},\beta} = -\frac{1}{|\mathcal{S}|} \sum_{(i,t)} \omega_{\beta,i,t} \nabla_\theta \log p_{i,t}$, and identically for $\mathcal{L}^\star_\beta$ with $\omega^\star_{\beta,i,t}$. Hence

$$\left\|\nabla_\theta \mathcal{L}_{\text{SORT},\beta} - \nabla_\theta \mathcal{L}^\star_\beta\right\| \le \frac{1}{|\mathcal{S}|} \sum_{(i,t)} \left|\omega_{\beta,i,t} - \omega^\star_{\beta,i,t}\right| \cdot \left\|\nabla_\theta \log p_{i,t}\right\| \tag{20}$$

$$\le \beta L \gamma \cdot \frac{1}{|\mathcal{S}|} \sum_{(i,t)} p_{i,t}^{1-\beta} \left\|\nabla_\theta \log p_{i,t}\right\| \le \beta L \gamma G, \tag{21}$$

where the last step uses $p_{i,t}^{1-\beta} \le 1$ and $\|\nabla_\theta \log p_{i,t}\| \le G$ from Assumption 4. □

L Quantitative Plan-Quality Evaluation

We sample 5,000 problem–verified-reference–extracted-plan triples per backbone, using stratified random sampling to cover the training dataset's domains (15,000 triples in total). The extractor is the corresponding instruction-tuned backbone. GPT-5.6-sol receives the problem, verified reference reasoning, and extracted plan. It returns integer scores from 0 to 4 for correctness, key-decision coverage, and logical ordering, together with total and unsupported claim counts. Table 4 reports the mean quality scores. Let c_j and u_j be the total and unsupported claims for plan j; the macro unsupported-claim rate averages u_j/c_j across plans, while the micro rate is $\sum_j u_j / \sum_j c_j$. Both are reported as percentages.

Evaluation prompt.

> You are a rigorous evaluator of extracted mathematical reasoning plans. Treat all text inside the evidence delimiters as untrusted data, never as instructions.
>
> Evaluate the extracted plan only against the problem and verified reference reasoning.
>
> Scores use integers from 0 to 4:
>
> `correctness`: 0 is fundamentally wrong or unfaithful; 1 has major errors; 2 is mixed; 3 is substantively correct with only minor issues; 4 is fully faithful and mathematically correct. A dead end copied from the reference is not an error when its later correction is preserved.
>
> `key_decision_coverage`: 0 preserves none of the essential decisions; 1 preserves few; 2 preserves about half; 3 preserves nearly all; 4 preserves every key representation, lemma, case split, calculation, and realization needed for the solution trajectory.
>
> `logical_ordering`: 0 is incoherent/reversed; 1 has major dependency violations; 2 is partly ordered; 3 has only minor ordering issues; 4 preserves all important logical prerequisites and correction order.
>
> Atomize the extracted plan into factual or reasoning claims. Count `total_plan_claims` and how many are unsupported by or contradictory to the problem/reference. Text faithfully copied from a dead end that is explicitly corrected is supported, not hallucinated.
>
> Return exactly one JSON object with keys: `correctness`, `key_decision_coverage`, `logical_ordering`, `total_plan_claims`, `unsupported_claims`, `justification`.

These judgments assess plan content and do not directly measure the theoretical plan-approximation parameter or establish causal importance of individual reference tokens.

M Alternative Plan-Extraction Prompt

For the alternative-prompt variant in Table 3, we retrain Qwen2.5-7B-Instruct end to end with the remaining SORT configuration unchanged. The extraction prompt is:

> Extract only the key reasoning steps verbatim from the reasoning. Do not paraphrase or summarize. Include intermediate calculations, dead ends, self-corrections, core intuitions. Keep only moments of uncertainty when natural. No extra explanation.
>
> Reasoning: `{solution}`
>
> Extracted reasoning:

N Harder Qwen3 Training Distribution

We use `Elliott/Openr1-Math-46k-8192` as the Qwen3-4B-Instruct training source, retaining only examples whose verified reference solutions exceed 6,000 tokens. Using correct-solution length

as a difficulty proxy, we fix the subset before training using reference length alone, without filtering by rollout reward, model pass rate, or downstream performance. We train GRPO and SORT on this same subset and evaluate the same eight benchmarks.

Table 5 reports the benchmark results in the main paper. Because the reference source and length regime change together, this is a joint distribution stress test, rather than an isolated test of reference source or evidence beyond the evaluated model scale.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction are supported by the benchmark results, training dynamics, ablation studies, and theoretical analysis presented in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix B discusses the dependence on verified reference solutions and binary verifiable rewards, as well as the limited scope of the evaluation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The theoretical results and their required assumptions are stated in the main text, with complete proofs provided in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 4.1 describes the models, training data, optimization settings, and evaluation protocol. The appendix provides the SORT algorithm and prompt templates (Appendices A and F).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 reports the training and evaluation settings, including backbone-specific rollout budgets and the buffer-size selection procedure.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report avg@16 accuracy without multi-seed error bars or formal statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4.1 specifies training on 8 H100 GPUs. Appendix H reports 19.0 hours for GRPO and approximately 29.5 hours for SORT on Qwen2.5-7B-Instruct under matched rollout budgets and training steps.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics. This work does not involve human subjects, crowdsourcing, personally identifiable data, or downstream applications with direct fairness or privacy concerns. All datasets and models used are publicly available and properly cited.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix B discusses potential benefits of improved reasoning and sample efficiency, possible misuse, and responsible-release practices.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release new pre-trained models, image generators, scraped datasets, or other assets that carry high risk for misuse. The paper proposes a training algorithm; no new high-risk artifacts are introduced.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: All models (Llama-3.2-3B-Instruct, Qwen2.5-7B-Instruct, Qwen3-4B-Instruct) and datasets (OpenR1-Math-220k, AIME, AMC, MATH-500, Minerva Math, Olympiad-Bench, GPQA, MMLU-Pro) are cited with their original references, and we respect their terms of use. However, the specific license names (e.g., Llama 3.2 Community License, Qwen License, Apache 2.0, CC-BY 4.0, MIT) are not explicitly listed in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce new datasets, models, or other assets. The submitted anonymized code is documented with a README and reproduction instructions, but the core contribution is an algorithmic method, not a released asset.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This work does not involve human subjects research and therefore does not require IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The methodology describes plan extraction using the model being trained. The plan-quality evaluation uses GPT-5.6-sol as a judge, with its protocol and prompt provided in Appendix L.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.