H^{2}SD: Hybrid Hindsight Self-Distillation

Qiye Cai*[1,2], Yichuan Ma*[1,3], Linyang Li[1,4], Peiji Li[1,3], Yongkang Chen[1], Qipeng Guo[1], Yicheng Zou[1], Tao Gui[3], Xiaocheng Feng†[2] and Bing Qin†[2]

[1]Shanghai Artificial Intelligence Laboratory, [2]Harbin Institute of Technology, [3]Fudan University, [4]The Chinese University of Hong Kong

Reinforcement learning with verifiable rewards (RLVR) has substantially improved the reasoning capabilities of large language models on tasks such as mathematical reasoning and code generation. However, most RLVR methods assign a scalar outcome reward to an entire trajectory, resulting in sparse supervision and limited token-level credit assignment. On-policy distillation (OPD) provides denser supervision by distilling token-level distributions from a stronger teacher model, but requires an additional teacher and typically assumes a shared vocabulary. On-policy self-distillation (OPSD) removes this dependency by conditioning the same model on privileged information to construct a teacher policy. However, directly matching the teacher distribution may cause information leakage and unstable optimization. RLSD avoids direct matching by using the teacher signal only to modulate update magnitudes, but it cannot provide an explicit correction direction when the sampled reasoning fails.

To address this tradeoff, we introduce H^{2}SD, a hybrid hindsight self distillation framework that uses the teacher differently according to trajectory correctness. For successful trajectories, the teacher receives the student response confirmed as correct together with a rephrasing instruction, and its probabilities on the original response tokens are used to modulate update magnitudes without changing the direction determined by the reward. For failed trajectories, we condition the teacher on a reference hint containing key reasoning steps and a verified answer, and minimize the reverse KL divergence from the student to the teacher. Experiments on multiple challenging reasoning benchmarks show that H^{2}SD consistently outperforms representative RLVR, OPSD, and RLSD baselines while maintaining stable optimization and favorable generation efficiency.

1. Introduction

Since the release of DeepSeek-R1 [3], reinforcement learning with verifiable rewards (RLVR), represented by GRPO [12], has been widely adopted to enhance the reasoning abilities of large language models (LLMs).

Meanwhile, the field of agentic RL has also rapidly progressed, where models interact with environments and improve through feedback and reward signals [9, 11]. By carefully designing environmental feedback and reward functions, general LLMs have achieved remarkable improvements in tool use, open-ended environment interaction, and other agentic tasks [15, 20].

Despite these advances, RLVR algorithms usually obtain a single feedback signal from the environment and use it as a scalar outcome reward for optimization, which creates a bottleneck in credit assignment, since the reward provides limited information about which token is responsible for success or failure. On-policy distillation (OPD) offers a possible solution to this problem [10, 16]. By introducing a stronger model as the teacher model and using the teacher's token-level logits to supervise trajectories sampled by the student model, OPD provides denser supervision over the entire response.

However, OPD relies on a stronger teacher model, and the teacher and student models often need to share the same tokenizer. These requirements restrict its applicability in broader settings. To address this issue, on-policy self-distillation (OPSD) proposes an alternative paradigm [6, 13, 21]. In OPSD, the same model is used as both the teacher and the student. The teacher model is provided with privileged information r, and, conditioned on this information, produces token-level supervision on trajectories sampled by the student. In this way, OPSD attempts to approximate the effect of OPD without actually requiring an external stronger teacher model.

† Corresponding authors: Xiaocheng Feng (xcfeng@ir.hit.edu.cn), Bing Qin (qinb@ir.hit.edu.cn)
* Equal contribution.

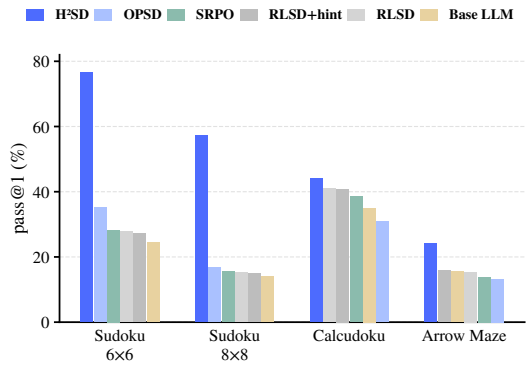

Figure 1: Overall performance comparison on representative logical reasoning benchmarks. H²SD consistently delivers strong performance across diverse tasks.

Although OPSD is effective, directly matching a teacher distribution conditioned on privileged information may cause training instability and privileged information leakage, as observed by RLSD [17]. RLSD mitigates these risks by using the teacher signal only to modulate token-level update magnitudes. SRPO further routes successful samples to GRPO and failed samples to self distillation [8]. However, this improved stability comes at the cost of weaker corrective supervision. Unlike OPSD, which transfers the teacher's fine-grained distributional preferences, RLSD mainly reweights the updates of tokens already generated by the student. Consequently, when the student follows an incorrect reasoning path, magnitude modulation alone may be insufficient to correct its underlying generation distribution.

To resolve this tension, we propose Hybrid Hindsight Self Distillation (H²SD), which assigns different roles to the teacher according to trajectory correctness. For successful trajectories, the reward already provides a reliable optimization direction. The teacher receives the student response confirmed as correct together with a rephrasing instruction, and its probabilities on the original response tokens are used only to modulate the update magnitudes in RLSD. For failed trajectories, we condition the teacher on a reference hint and minimize the reverse KL divergence from the student to the teacher. This preserves the direction determined by the reward for successful trajectories while providing explicit distributional correction when the sampled reasoning fails.

The quality of the teacher signal also depends on the privileged information it receives. We use a stronger language model to generate a reference hint for each problem, containing key intermediate reasoning steps and a final answer confirmed by the verifier. OPSD and H²SD use the same hints in our experiments, allowing us to compare how different objectives convert the same privileged information into learning signals.

With these designs, H²SD achieves the best overall performance on Sudoku, Calcudoku, and Arrow Maze among the evaluated methods, including GRPO, OPSD, RLSD, SDPO, and SRPO. These results show that reasoning can be improved by using teacher guidance to refine credit assignment on successful trajectories and by using a teacher distribution conditioned on reference hints to correct failed trajectories. Our main contributions are summarized as follows:

- We identify a trade-off in on-policy self-distillation between distribution-level correction and stable token-level credit assignment. We further show that the appropriate supervision strategy depends on whether the student-generated trajectory is correct.
- We propose H²SD, a correctness-aware hybrid self-distillation framework that applies magnitude-based supervision to correct trajectories and distribution-level distillation to incorrect trajectories, combining the complementary advantages of RLSD and OPSD.
- We conduct experiments on multiple challenging reasoning benchmarks, where H²SD consistently outperforms representative RLVR, OPSD, and RLSD baselines while maintaining stable optimization.

2. Related Work

Reinforcement Learning with Verifiable Rewards. RLVR improves LLM reasoning by optimizing policies with automatically verifiable outcome rewards. GRPO [12] estimates group-relative advantages from sampled trajectories without requiring an explicit value model, and subsequent methods such as DAPO and GSPO [19, 22] further enhance training stability, efficiency, and credit assignment.

On-Policy Distillation. On-policy distillation [1, 2, 10] mitigates the distribution shift problem of off-policy distillation by allowing the student model to generate its own trajectories and using a teacher model to evaluate these on-policy rollouts, providing dense token-level supervision. The advantage of on-policy distillation lies in its combination of on-policy exploration and dense supervision. However, such methods usually rely on an additional strong teacher, and logit-level distillation often requires the teacher and student to have compatible token spaces, which limits the choice of teacher models.

On-Policy Self-Distillation with Privileged Informa-

tion. To reduce reliance on external teachers, recent studies have explored building self-teachers with privileged information. OPSD [21] allows the same model to serve as both student and teacher, where the teacher branch receives privileged ground-truth solutions and guides the student through token-level distribution matching. SDPO [6] further broadens privileged information by incorporating environment feedback or correct rollouts as teacher signals. SD-Zero [4] introduces a reviser to refine responses based on model outputs and rewards. Other methods, such as OPCD [18] and GATES [14], extend privileged information beyond final answers to include richer auxiliary signals such as context, experience, and evidence.

Stable Self-Distillation for RLVR. Stable Self-Distillation for RLVR. Directly fitting a privileged teacher may introduce information leakage and training instability. RLSD [17] addresses this issue by letting the environment reward determine the update direction while using the teacher only to modulate token-level magnitudes. Related methods extend this reward-grounded formulation through entropy-aware confidence gating [7] and contrastive evidence reweighting [5], while SRPO [8] routes successful trajectories to GRPO and failed ones to self-distillation.

3. Preliminaries

Before introducing the proposed H²SD method, we first describe the implementation strategies of GRPO, OPD, OPSD, and RLSD. Throughout this section, x denotes a problem and $y = (y_1, \ldots, y_T)$ denotes a model response.

3.1. GRPO

In the GRPO algorithm, given a question x, GRPO samples G responses from the old policy $\pi_{\theta_{\text{old}}}$. We first sample G responses $\{y^{(i)}\}_{i=1}^{G}$ from the old policy, where $y^{(i)} = (y_1^{(i)}, \ldots, y_{T_i}^{(i)})$ and $y^{(i)} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$. A reward function is then used to assign a scalar reward to each response. We denote the reward of each response as $R(x, y^{(i)})$. For each sampled response $y^{(i)}$, GRPO computes a relative advantage within the current group:

$$A_i = \frac{R(x, y^{(i)}) - \mu}{\sigma}, \tag{1}$$

where μ denotes the mean reward of the current response group, and σ denotes the standard deviation of the rewards within the group.

Then, for the token at position t in the i-th response, GRPO optimizes the LLM by maximizing a clipped surrogate objective. Specifically, we first define the probability ratio between the current policy and the old policy as

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_t^{(i)} \mid x, y_{<t}^{(i)})}{\pi_{\theta_{\text{old}}}(y_t^{(i)} \mid x, y_{<t}^{(i)})}. \tag{2}$$

The token-level clipped surrogate objective is defined as

$$\ell_{i,t}(\theta) = \min\left(\rho_{i,t}(\theta)A_i,\ \text{clip}\left(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right) A_i\right). \tag{3}$$

Finally, the GRPO objective can be written as

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\left[\ell_{i,t}(\theta) - \beta D_{\text{KL}}\left(\pi_\theta \,\|\, \pi_{\text{ref}}\right)\right]. \tag{4}$$

In GRPO, all tokens within the same response are rewarded or penalized with the same magnitude, which is solely determined by the response-level relative advantage A_i.

3.2. OPD, OPSD, and RLSD

3.2.1. On-Policy Distillation and OPSD

As discussed above, GRPO assigns the same reward or penalty magnitude to all tokens within the same response. On-policy distillation (OPD) addresses this limitation by introducing a stronger model as the teacher model to provide more informative supervision for the student model. Within the same trajectory, OPD assigns diverse training signals, thereby improving the overall training performance.

Formally, let π_s denote the student model and π_t denote the teacher model. Given a question x, suppose that a trajectory $y = (y_1, \ldots, y_T)$ is sampled from the student model:

$$y \sim \pi_s(\cdot \mid x). \tag{5}$$

At each token position t, OPD uses the next-token distribution predicted by the teacher model as the supervision signal. To simplify notation, we define

$$p_t^{\text{T}}(\cdot) = \pi_t(\cdot \mid x, y_{<t}), \qquad p_t^{\text{S}}(\cdot) = \pi_s(\cdot \mid x, y_{<t}). \tag{6}$$

The OPD objective is then written as

$$\mathcal{L}_{\text{OPD}}(\pi_s) = \mathbb{E}_{x,y}\left[\sum_{t=1}^{T} D_{\text{KL}}\left(p_t^{\text{T}} \,\|\, p_t^{\text{S}}\right)\right], \tag{7}$$

where the expectation is taken over $x \sim \mathcal{D}$ and $y \sim \pi_s(\cdot \mid x)$. Compared with GRPO, which relies on a trajectory-level scalar reward provided by a rule-based reward function, OPD provides fine-grained supervision at each token position.

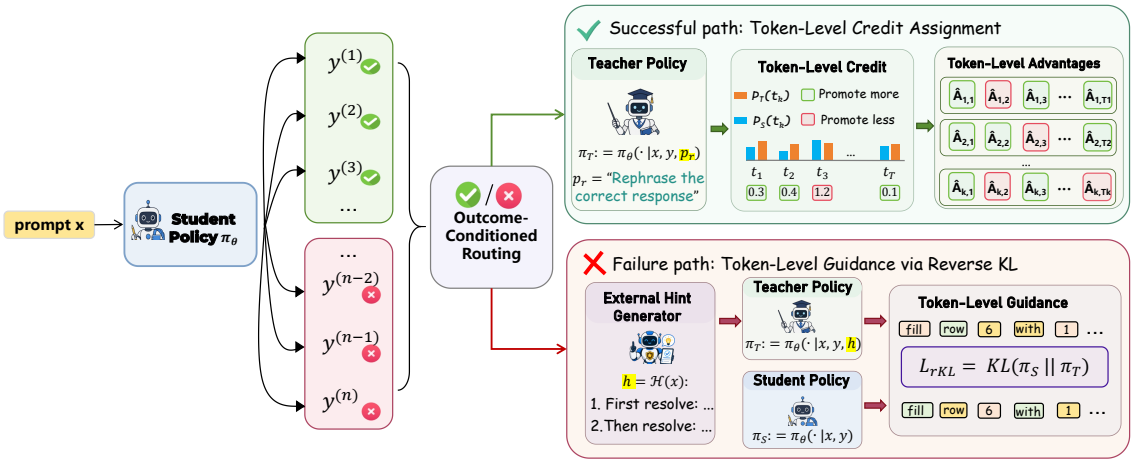

Figure 2: Overview of H²SD. The framework routes student-generated trajectories according to their correctness: successful trajectories receive fine-grained token-level credit assignment, while failed trajectories are corrected through hint-conditioned distributional guidance.

However, a major limitation of OPD is that it requires a stronger teacher model π_t, which should also share the same vocabulary with the student model. This strict requirement restricts the practical applicability of OPD. To address this issue, on-policy self-distillation (OPSD) uses the same model as both the student and the teacher, thereby avoiding the dependence on an external stronger model.

To enable the teacher to induce a better policy than the student, OPSD provides the teacher with additional privileged information. For example, given a question x, a trajectory y sampled by the student model, and privileged information r, OPSD regards the model conditioned on both the question x and the privileged feedback r as the teacher, while the model conditioned only on x is regarded as the student. We define

$$p^{\mathrm{T}}_{t,\theta}(\cdot) = \mathrm{sg}\left[\pi_\theta(\cdot \mid x, r, y_{<t})\right], \qquad p^{\mathrm{S}}_{t,\theta}(\cdot) = \pi_\theta(\cdot \mid x, y_{<t}). \tag{8}$$

Therefore, the OPSD objective can be written as

$$\mathcal{L}_{\mathrm{OPSD}}(\theta) = \mathbb{E}_{x,y}\left[\sum_{t=1}^{T} D_{\mathrm{KL}}\left(p^{\mathrm{T}}_{t,\theta} \,\|\, p^{\mathrm{S}}_{t,\theta}\right)\right], \tag{9}$$

OPSD still provides differentiated training signals for different token positions within a single trajectory, thereby alleviating the sparsity of scalar rewards in RLVR. Meanwhile, since the teacher and the student are instantiated by the same model, OPSD significantly reduces the dependence of OPD on vocabulary consistency.

3.2.2. RLSD

However, RLSD points out that directly optimizing with OPSD may suffer from privileged information leakage. Since the teacher model is conditioned on privileged information, its logits may implicitly encode information that is unavailable during inference. As a result, the student model may gradually learn to rely on such privileged information, leading to performance degradation.

To address this issue, RLSD reformulates the distribution matching problem in OPSD as a token-level credit assignment problem. Specifically, for a trajectory $y = (y_1, \ldots, y_T)$ sampled by the student model, RLSD computes the log-probability of each token under both the student context and the teacher context. The difference between the two log-probabilities is then used to measure the gain brought by the privileged information:

$$\Delta_t = \mathrm{sg}\left(\log P_T(y_t) - \log P_S(y_t)\right), \tag{10}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation. This value measures how much the privileged information r supports the current token. If $\Delta_t > 0$, the feedback r assigns higher probability to this token under the teacher context; if $\Delta_t < 0$, the feedback r provides less support for this token.

RLSD then constructs a token-level weight according to the sign of the trajectory-level advantage A:

$$w_t = \exp\left(\mathrm{sign}(A) \cdot \Delta_t\right) = \left(\frac{P_T(y_t)}{P_S(y_t)}\right)^{\mathrm{sign}(A)}. \tag{11}$$

When $A > 0$, the model more strongly reinforces tokens that are supported by the privileged information. When $A < 0$, the model more strongly penalizes

tokens that are contradicted by the privileged information. Finally, RLSD clips w_t to limit the influence of individual tokens on the update magnitude, thereby stabilizing training.

The token-level advantage is then defined as

$$\widehat{A}_t = A\left[(1-\lambda) + \lambda \, \mathrm{clip}\left(w_t, 1-\epsilon_w, 1+\epsilon_w\right)\right], \quad (12)$$

where λ controls the strength of self-distilled magnitude modulation. RLSD replaces the trajectory-level advantage A in the GRPO objective with $\widehat{A}_t$ while preserving its sign.

Unlike OPSD, RLSD uses the teacher signal to redistribute the strength of reward or penalty within the same trajectory. Therefore, RLSD can be viewed as a fine-grained credit assignment method based on privileged information, rather than a distribution distillation method.

Although RLSD effectively alleviates the issue of information leakage, this design also weakens its ability to perform distribution learning. RLSD does not directly learn the full output distribution of the teacher model; instead, it only uses token-wise probability differences to adjust the update magnitude. Consequently, the distributional structure contained in the teacher signal, including preferences over reasoning expressions and relative relations among candidate tokens, cannot be effectively transferred to the student model.

This limitation is most evident for failed responses, where magnitude modulation alone cannot provide an explicit correction direction.

4. H²SD: Hybrid Hindsight Self-Distillation

4.1. Hint Design

In RLSD and OPSD, privileged information for general reasoning tasks is typically derived from the ground-truth answer or feedback provided by a rule-based function. However, for more complex reasoning tasks, such answer-level privileged information can be too coarse-grained to reveal specific reasoning errors, missing key steps, or potential directions for correction.

To provide more informative supervision, we introduce a stronger LLM as a hint generator. Given a problem x, the hint generator produces a natural-language hint h that provides useful information for solving the problem:

$$h = \mathcal{H}(x). \quad (13)$$

The generated hint is used as privileged information that is available only during training to construct a self-teacher. The stronger model does not directly serve as the teacher and does not need to share the same vocabulary with the student model. It is only responsible for generating the natural-language hint. The actual teacher distribution is still produced by the current model conditioned on the hint:

$$\begin{aligned} \pi_T(\cdot \mid x, h, y_{<t}) &= \pi_\theta(\cdot \mid x, h, y_{<t}), \\ \pi_S(\cdot \mid x, y_{<t}) &= \pi_\theta(\cdot \mid x, y_{<t}). \end{aligned} \quad (14)$$

4.2. Hybrid Self-Distillation

A straightforward approach is to directly match the student distribution to the teacher distribution conditioned on the hint, as in OPSD. Although this provides supervision for the reasoning process, full distribution matching may expose the student to privileged answer information at every token and cause information leakage. RLSD avoids direct matching by using the teacher only to rescale updates determined by the reward. However, for failed trajectories, it only changes the strength with which sampled errors are penalized and does not indicate which alternatives should be preferred.

We therefore propose Hybrid Hindsight Self-Distillation (H²SD), which selects the distillation strategy according to trajectory correctness. For successful trajectories, the positive reward already provides a reliable optimization direction. The teacher receives the student response verified as correct together with an instruction to rephrase it, and its probabilities on the original response tokens are used to compute the magnitude weights in RLSD. This refines credit assignment across tokens without directly matching the teacher distribution or changing the direction determined by the reward.

For failed trajectories, the sampled reasoning path is unreliable. Although a negative reward discourages the sampled actions, it provides limited guidance on how they should be corrected. We therefore use the teacher distribution conditioned on the hint as corrective guidance and minimize the reverse KL divergence from the student to the teacher:

$$\mathcal{L}_{\mathrm{RKL}} = \frac{1}{T}\sum_{t=1}^{T} D_{\mathrm{KL}}\left(p_{t,\theta}^{S} \,\|\, \mathrm{sg}[p_{t,\theta}^{T,h}]\right), \quad (15)$$

where $p_{t,\theta}^{S} = \pi_\theta(\cdot \mid x, y_{<t})$ and $p_{t,\theta}^{T,h} = \pi_\theta(\cdot \mid x, h, y_{<t})$. The operator $\mathrm{sg}[\cdot]$ blocks gradients through the teacher distribution.

Let $m = \mathbf{1}[R(x,y) = 1]$ denote trajectory correctness. The overall objective is

$$\mathcal{L}_{\mathrm{H^2SD}} = \mathbb{E}_{x,y}\left[m\mathcal{L}_{\mathrm{RLSD}} + \gamma(1-m)\mathcal{L}_{\mathrm{RKL}}\right], \quad (16)$$

Method	Sudoku		Calcudoku			Arrow Maze						Overall
	6×6	8×8	5×5	6×6	Avg.	6×6	7×7	8×8	9×9	10×10	Avg.	
Base LLM	24.50	14.00	59.00	11.33	35.00	32.00	17.50	16.00	6.00	6.50	15.60	22.28
GRPO	26.25	16.75	66.33	13.67	40.00	34.50	16.00	14.00	7.50	6.50	15.70	24.68
SDPO	22.25	2.50	26.00	0.33	13.17	28.00	21.00	17.00	8.00	5.50	15.90	13.46
RLSD	27.75	15.25	66.67	**15.33**	41.00	34.00	18.00	16.00	5.00	4.00	15.40	24.85
OPSD	35.25	16.75	52.00	9.67	30.83	29.50	15.00	11.00	3.50	7.00	13.20	24.01
SRPO	28.25	15.50	63.00	14.00	38.50	28.00	16.50	12.50	5.50	6.50	13.80	24.01
RLSD+hint	27.25	15.00	66.33	**15.33**	40.83	34.00	17.00	15.00	7.00	6.00	15.80	24.72
H²SD	**76.50**	**57.25**	**73.30**	14.67	**44.00**	**52.00**	**29.50**	**21.00**	**9.00**	**9.50**	**24.20**	**50.49**

Table 1: Validation accuracy (%) on Sudoku, Calcudoku, and Arrow Maze benchmarks. Overall averages the two Sudoku results, the Calcudoku average, and the Arrow Maze average. Bold and underlined values indicate the best and second-best results, respectively. Tied results share the same formatting.

where $\mathcal{L}_{\text{RLSD}}$ is the loss form of the RLSD objective computed using the teacher prompted to rephrase, and γ controls the strength of corrective distillation on failed trajectories. Thus, H²SD refines credit assignment for successful trajectories and provides directional correction for failed trajectories.

5. Experiments

5.1. Models and Benchmarks

We use Qwen3-30B-A3B-Instruct-2507 as the base model for all experiments. We evaluate our method on three representative logical reasoning benchmarks: Sudoku, Arrow Maze, and Calcudoku.

Specifically, Sudoku requires filling 6×6 and 8×8 grids with digits such that no digit is repeated within each row, column, or sub-grid (2×3 or 2×4). Arrow Maze requires assigning one of eight directional arrows to each empty cell such that the ray length starting from every numbered cell matches the corresponding value, while ensuring that all cells are covered by at least one ray. Calcudoku extends traditional grid-based constraint solving by introducing region-level arithmetic constraints, where numbers within each region must satisfy a specified operation and target value.

5.2. Implementation Details

For each training problem, Kimi-K2.6 generates a reference rationale (hint) containing key intermediate steps and a verifier-confirmed final answer. OPSD and H²SD use identical hints, while RLSD+hint replaces RLSD's answer-only privileged information with the same hints, isolating the effect of how the privileged context is used. Following their original settings,

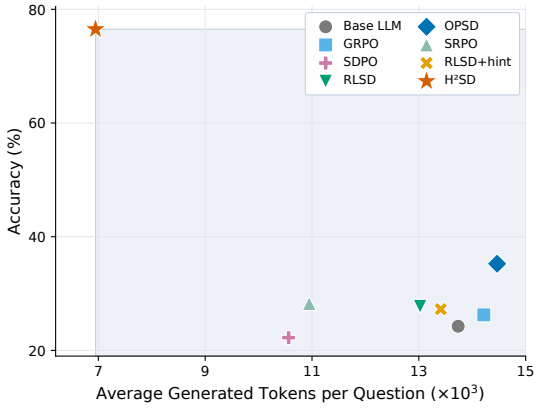

Figure 3: Accuracy–generation cost trade-off on Sudoku-6×6. Points toward the upper-left indicate better efficiency. H²SD achieves the highest pass@1 with the fewest generated tokens.

SDPO and SRPO use verified same-batch correct responses as feedback; SRPO routes correct trajectories to GRPO and failed ones with feedback to entropy-weighted SDPO.

We implement all methods in VERL with SGLang. To reduce the substantial memory overhead of full-vocabulary distillation for OPSD on long-chain reasoning tasks, we retain exact probabilities for the student's top-100 tokens and merge the remaining mass into a tail bucket. Both SDPO and the distillation branch of SRPO use generalized Jensen–Shannon divergence with $\alpha = 0.5$. Training uses four nodes, each with eight NVIDIA H200 140GB GPUs.

5.3. Main Results

Table 1 shows that H²SD achieves the strongest overall performance across the four benchmarks. Its advan-

Strategy	Sudoku		Calcudoku	Arrow Maze
	6×6	8×8	Avg.	Avg.
Magnitude Only	60.00	55.50	43.17	15.60
Reverse-KL Only	60.50	38.50	28.67	18.50
Reversed Routing	17.00	9.75	2.83	6.40
H²SD	**76.50**	**57.25**	**44.00**	**24.20**

Table 2: Accuracy (%) under different routing strategies. Calcudoku and Arrow Maze report averages across grid sizes. The "Only" variants apply one update to all trajectories, while Reversed Routing swaps H²SD's success/failure assignments.

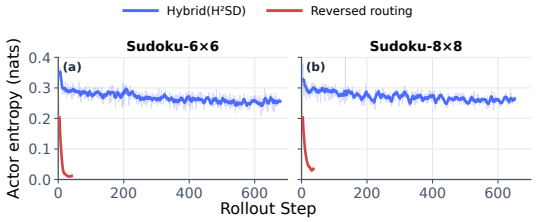

Figure 4: Actor-entropy dynamics under H²SD and Reversed Routing on Sudoku. Reversed Routing rapidly collapses toward zero entropy.

tage is particularly pronounced on Sudoku. Since OPSD, RLSD+hint, and H²SD use identical privileged hints, their performance differences suggest that the gains arise primarily from how the privileged information is converted into learning signals rather than from the information itself.

H²SD also achieves the highest average performance on Calcudoku and consistently ranks first across all Arrow Maze sizes. Its strong results on the held-out Arrow Maze-8×8 and Arrow Maze-10×10 settings further demonstrate that the learned reasoning improvements generalize to unseen puzzle sizes. Moreover, although SRPO similarly routes trajectories according to correctness, H²SD consistently outperforms it; the controlled ablation in Table 2 further examines the effect of assigning different update rules to successful and failed trajectories.

Notably, SDPO underperforms the Base LLM overall and degrades sharply on the harder Sudoku and Calcudoku settings. We hypothesize that directly matching sibling-conditioned distributions over long reasoning trajectories may introduce misaligned token-level supervision, suppressing necessary reasoning steps and producing incomplete responses.

Figure 3 further shows that the gains of H²SD do not rely on longer responses. It achieves the highest accuracy with the lowest average generation cost, indicating that its improvements primarily come from more effective use of the available learning signals.

5.4. Does Outcome-Conditioned Routing Matter?

Table 2 evaluates alternative routing strategies across all three benchmarks. H²SD consistently outperforms the variants that uniformly apply magnitude modulation or reverse-KL distillation. In contrast, reversing the routing assignment reduces performance below the Base LLM in every reported setting. As shown in Figure 4, Reversed Routing also drives the actor entropy rapidly toward zero on both Sudoku sizes, indicating an early collapse of policy exploration.

These results support the outcome-conditioned design of H²SD. For successful trajectories already validated by the reward, preserving the policy-gradient direction while refining token-level credit assignment is sufficient. Failed trajectories instead benefit from the directional correction provided by the teacher distribution. Assigning these signals according to trajectory correctness therefore improves accuracy while preserving policy diversity.

5.5. Which Privileged Context Is Useful?

In the main experiments of H²SD, we use hints generated by an external hint generator as privileged information for the teacher policy. We further investigate the effectiveness of different types of privileged information in Table 3. Specifically, we keep the H²SD training recipe fixed and compare the following four types of privileged information: (1) **Programmatic feedback**, which provides fine-grained error feedback generated by a task-specific verifier based on the differences between the student response and the ground-truth solution; (2) **Ground truth**, which contains only the correct final answer without any intermediate reasoning; (3) **Sibling solution**, which uses a verified correct response sampled from multiple rollouts of the same problem as teacher guidance; and (4) **Hint**, which is a reference solution generated by an external strong model and contains both key intermediate reasoning steps and the final answer.

As shown in Table 3, ground-truth answers and programmatic verifier feedback provide only limited improvements. In contrast, privileged contexts containing richer process-level information lead to substantially larger gains. In particular, the sibling solution achieves accuracies of 61.75% and 48.50% on Sudoku-6×6 and Sudoku-8×8, respectively, and an average accuracy of 19.60% on Arrow Maze. While the The hint performs best in all three settings, reaching 76.50%, 57.25%, and 24.20% on Sudoku 6×6,

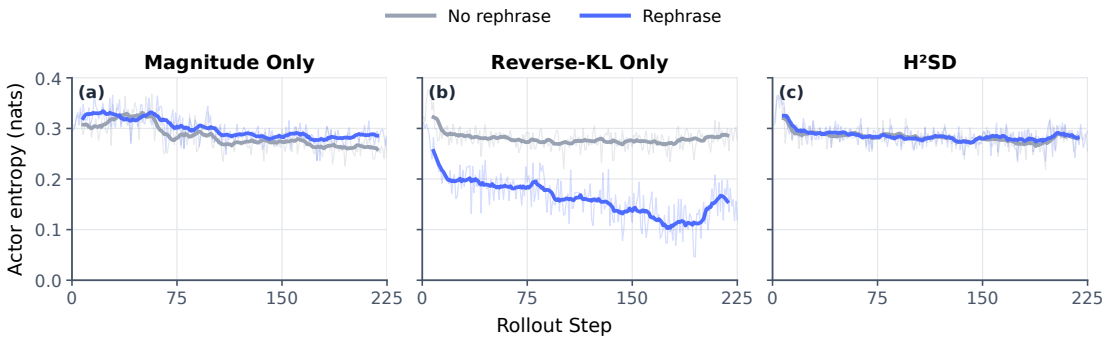

Figure 5: Actor entropy with and without successful-trajectory rephrasing on Sudoku-6×6. Rephrasing sharply reduces entropy under Reverse-KL Only, but not under Magnitude Only or H²SD.

Context	Sudoku		Arrow Maze
	6×6	8×8	Avg.
Base LLM	24.50	14.00	15.60
Programmatic feedback	25.50	14.25	12.00
Ground truth	27.50	15.00	16.40
Sibling solution	61.75	48.50	19.60
Hint	**76.50**	**57.25**	**24.20**

Table 3: Validation accuracy (%) under different privileged contexts. The Arrow Maze column reports the average across the five evaluated grid sizes. Boldface indicates the best result in each column.

Update Strategy	Sudoku-6×6 (NR / R)	Sudoku-8×8 (NR / R)
Magnitude Only	36.25 / **60.00**	23.50 / **55.50**
Reverse-KL Only	**63.00** / 60.50	**42.00** / 38.50
H²SD	68.50 / **76.50**	51.50 / **57.25**

Table 4: Effect of successful-trajectory rephrasing on accuracy (%). NR and R denote training without and with rephrasing, respectively. Boldface indicates the better result within each pair.

Sudoku 8×8, and Arrow Maze, respectively. These results suggest that, for complex reasoning tasks, fine-grained process-level privileged information can provide more informative and directional corrective signals for failed trajectories.

5.6. When Does Rephrasing Help?

In H²SD, successful trajectories are routed to the Magnitude Only update, and the teacher receives the student response verified as correct together with a rephrasing instruction, and the resulting distribution is used for magnitude modulation. Table 4 examines how this design interacts with different update strategies. Rephrasing improves Magnitude Only by 23.75 and 32.00 percentage points on Sudoku-6×6 and Sudoku-8×8, respectively, and improves H²SD by 8.00 and 5.75 points. In contrast, it reduces the performance of Reverse-KL Only by 2.50 and 3.50 points.

This interaction suggests that rephrasing is most effective when the teacher signal is used to refine token-level credit rather than directly determine the target distribution. Tasks such as Calcudoku may admit multiple valid solutions, so an externally generated reference rationale may follow a different reasoning path from the student's correct trajectory. Rephrasing the student's own response preserves its valid reasoning path while removing redundant or irrelevant content. Under magnitude modulation, the resulting teacher signal emphasizes tokens that contribute to the correct solution without substantially changing the original policy distribution.

Direct reverse-KL matching, however, may overconstrain an already correct trajectory by forcing it toward the rephrased teacher distribution. Figure 5 supports this interpretation: adding rephrasing to Reverse-KL Only causes actor entropy to decrease sharply, indicating a substantial loss of exploration, whereas Magnitude Only maintains relatively stable entropy. H²SD largely preserves this stability by applying magnitude modulation to successful trajectories while reserving teacher-guided distribution alignment for failed ones.

6. Conclusion

We introduced H²SD, a hybrid self-distillation framework that provides token-level credit assignment for correct trajectories and corrective distributional guidance for incorrect ones. Across Sudoku, Calcudoku, and Arrow Maze, H²SD achieves the best overall performance among the evaluated methods and gener-

alizes to unseen puzzle sizes. Ablations show that outcome-based routing, informative reasoning hints, and successful-response rephrasing jointly drive these gains.

References

[1] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pages 21246–21263, 2024. 2

[2] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pages 32694–32717, 2024. 2

[3] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 1

[4] Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. Self-distillation zero: Self-revision turns binary rewards into dense supervision. *arXiv preprint arXiv:2604.12002*, 2026. 2

[5] Ahmed Heakl, Abdelrahman M Shaker, Youssef Mohamed, Rania Elbadry, Omar Fetouh, Fahad Shahbaz Khan, and Salman Khan. Cepo: Rlvr self-distillation using contrastive evidence policy optimization. *arXiv preprint arXiv:2605.19436*, 2026. 2

[6] 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. 1, 2

[7] Junlong Ke, Zichen Wen, Weijia Li, Conghui He, and Linfeng Zhang. Respecting self-uncertainty in on-policy self-distillation for efficient llm reasoning. *arXiv preprint arXiv:2605.13255*, 2026. 2

[8] Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. *arXiv preprint arXiv:2604.02288*, 2026. 1, 2

[9] Xuefeng Li, Haoyang Zou, and Pengfei Liu. Torl: Scaling tool-integrated rl. *arXiv preprint arXiv:2503.23383*, 2025. 1

[10] Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. https://thinkingmachines.ai/blog/on-policy-distillation. 1, 2

[11] Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *Advances in Neural Information Processing Systems*, 38:105523–105553, 2026. 1

[12] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 1, 2

[13] Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026. 1

[14] Alex Stein, Furong Huang, and Tom Goldstein. Gates: Self-distillation under privileged context with consensus gating. *arXiv preprint arXiv:2602.20574*, 2026. 2

[15] Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026. 1

[16] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 1

[17] Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026. 1, 2

[18] Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026. 2

[19] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026. 2

[20] Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026. 1

[21] 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. 1, 2

[22] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025. 2