SPEAR: Distilling General Reasoning Skeletons via Sequential Symbolic Alignment in Reinforcement Learning

Anonymous ACL submission

Abstract

Reinforcement learning-based knowledge distillation has the potential to transfer complex reasoning from teacher to student models, yet it currently faces a critical dilemma: researchers must choose between sparse outcome-based rewards, which provide insufficient logical guidance, or expensive neural Process Reward Models (PRMs) for dense signals. Recently, on-policy distillation uses token-level reverse KL-divergence which provides dense signals but often requires significant computations and white-box teacher distributions, limiting broader generalization. We resolve this by introducing **SPEAR** (**S**ymbolic **P**rocess **E**valuation and **A**lignment **R**eward), a training-free and plug-and-play process reward method for sequence-level on-policy distillation. SPEAR projects natural-language reasoning traces into domain-adaptive symbolic milestones, providing an efficient proxy for process-level reasoning alignment. By utilizing the longest common subsequence (LCS) to align student explorations with teacher milestones, SPEAR provides a dense, order-aware reward signal that enforces logical consistency without the need for an external neural verifier. Our experiments across math, science, and commonsense reasoning tasks demonstrate that SPEAR effectively bridges the reasoning gap between student and teacher models via sequence-level distillation with efficient dense process rewards.

1 Introduction

Distilling complex reasoning capabilities from proprietary Large Language Models (LLMs) to resource-efficient Small Language Models (SLMs) is essential for practical deployment of models. Traditionally, this is achieved via supervised fine-tuning (SFT) on teacher-generated rationales in off-policy distillation (Magister et al., 2023; Xu et al., 2024). However, off-policy distillation fundamentally encourages "style mimicry" (Gudibande et al., 2023) and suffers from "exposure bias" (Agarwal et al., 2024), where the student memorizes linguistic patterns without genuinely internalizing the underlying logical transitions, resulting in failures if the student makes an early error during inference that wasn't covered in the training data. To address this, reinforcement learning-based on-policy distillation (RL-KD) has emerged as a superior paradigm, allowing models to explore and optimize their own reasoning trajectories (Yang et al., 2025; DeepSeek, 2026). Yet, applying RL introduces a critical dilemma: relying solely on sparse outcome rewards (e.g., final answer verification) fails to guide the student through multi-step logic, while utilizing dense Process Reward Models (PRMs) or token-level divergence incurs massive computational overhead (Lightman et al., 2024).

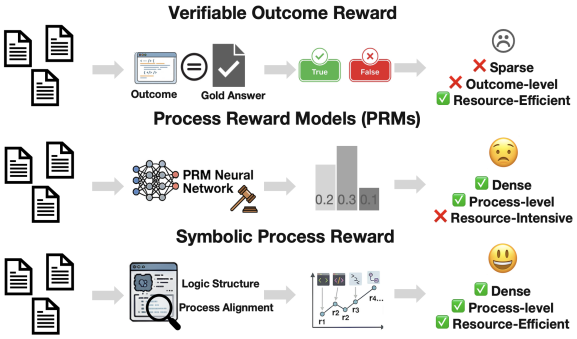

Figure 1: Comparison between SPEAR and other reward methods.

Existing process-supervision methods are largely limited to formal domains such as mathematics and programming, where intermediate steps and final answers are deterministically verifiable (Shao et al., 2024; Song et al., 2025; Yu et al., 2024). Extending process rewards to more general reasoning tasks (e.g., scientific and commonsense reasoning) remains challenging (Yu et al., 2023), as natural language reasoning lacks clear structural boundaries and is highly sensitive to linguistic variation compared to rigid reasoning-chain formats (OpenAI, 2024). Besides, prior work often overlooks leveraging the detailed reasoning traces from powerful

reasoning LLMs (Guo et al., 2025), despite their rich intermediate signals that can effectively guide student learning.

To bridge these gaps, we propose **SPEAR** (**S**ymbolic **P**rocess **E**valuation and **A**lignment **R**eward), a training-free, plug-and-play process reward framework for sequence-level on-policy distillation, as illustrated in Figure 2. Instead of forcing token-level matching or relying on heavy neural PRMs, SPEAR distills the logical structure of the teacher's rationale. We project high-dimensional reasoning traces into domain-adaptive *symbolic trajectories*—such as computational processes for math, causal dependencies for science, and entity state-transitions for commonsense. By employing the longest common subsequence (LCS) to align the student's explored symbolic path with the teacher's reference milestones, SPEAR provides a dense and order-aware reward. This enforces chronological logical consistency while granting SLMs the freedom to formulate their own response.

In summary, the contributions of our work are:

1. We introduce **SPEAR**, a novel, training-free process reward framework that enables highly efficient sequence-level on-policy distillation without the computational burden of process reward models.

2. SPEAR uses teacher thinking processes containing rich reasoning signals and projects these high-dimensional traces into **domain-adaptive symbolic anchors**, providing the student with a structural reference that successfully extends process supervision to reasoning tasks.

3. We conduct experiments across **comprehensive reasoning benchmarks**, including math, science, and commonsense tasks. Experimental results demonstrate that integrating SPEAR into standard RL pipelines consistently outperforms baselines, validating the efficacy of decoupling logical acquisition from linguistic expression.

2 Related Work

Reinforcement Learning for Reasoning Reinforcement learning with verifiable rewards (RLVR) has become the primary driver for inducing multi-step reasoning behaviors in LLMs (Luong et al., 2024; Guo et al., 2025). Algorithms like Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and its subsequent refinements—e.g., Dr. GRPO (Liu et al., 2025), which analyzes and mitigates length-related optimization bias in relative policy updates, and Dynamic Sampling Policy Optimization (DAPO) (Yu et al., 2025), which introduces dynamic trajectory reweighting to reduce sampling and length-induced biases, have substantially improved the efficiency and stability of RL-based reasoning training. However, these methods typically rely on sparse, binary outcome rewards. This sparsity can lead to "overthinking," where models generate thousands of redundant tokens to maximize rewards without valid logical progress (Agarwal et al., 2024; Song and Zheng, 2025).

LLMs Distillation Distillation is shifting from off-policy distillation via SFT toward on-policy distillation by reinforcement learning (Xu et al., 2025a; Lu and Lab, 2025; Xiao et al., 2026). Off-policy distillation asks students to learn by imitating a fixed teacher-generated dataset and can suffer from "exposure bias" because of the distribution mismatch between training and inference (Wen et al., 2025; Song and Zheng, 2026). On-policy distillation enables students to generate rollouts from current policy and learn under the teacher's guidance (Xu et al., 2025c; Li et al., 2026). Recent frameworks like RLKD (Xu et al., 2025b) utilize structural alignment to transfer the implicit multi-branch reasoning strategies of LLMs. Similarly, on-policy self-distillation (Zhao et al., 2026) uses privileged information to supervise exploration.

Process Supervision and Symbolic Alignment Evaluating intermediate reasoning steps is critical for mitigating reward hacking and reducing sample complexity (Lightman et al., 2024; Song et al., 2025; Li and Li, 2024). While neural PRMs offer dense supervision, they are computationally expensive and prone to hallucinations in domain-specific tasks (Yu et al., 2024; Su et al., 2025). Rule-based process rewards, such as RePAIR (Wang et al., 2026) and Logic-RL (Xie et al., 2025), provide a verifiable, training-free alternative. Specifically, the use of longest common subsequence (LCS) rewards has shown promise in identifying optimal reasoning paths across multiple tasks (Dong and Fan, 2025). SPEAR extends this by mapping diverse reasoning traces into symbolic, order-aware trajectories that provide a general alignment signal.

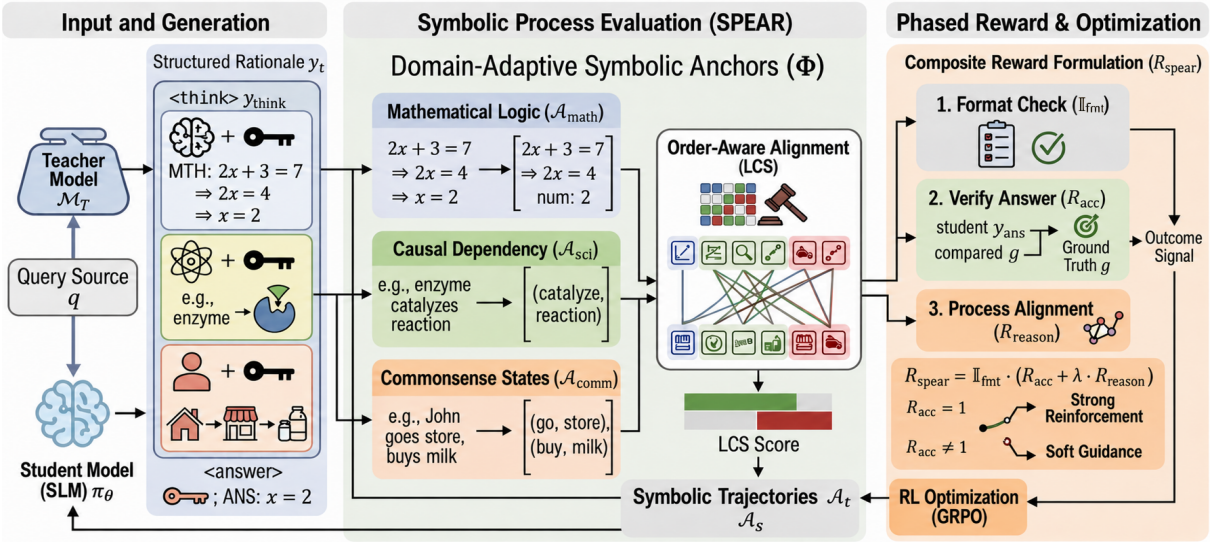

Figure 2: Overview of **SPEAR** (**S**ymbolic **P**rocess **E**valuation and **A**lignment **R**eward) method. It supervises the "logical skeleton" of reasoning tasks by aligning symbolic milestones extracted from teacher and student traces. By using an order-aware alignment (LCS) and an outcome-process combined reward, the framework enables small language models to internalize reasoning processes rather than token imitation from teacher's responses.

3 Method

Our approach departs from traditional token-level distillation by framing the reasoning process as a sequence of symbolic milestones. By supervising the student model through alignment with these milestones, SPEAR could internalize the underlying logical structure of a task while retaining the flexibility to explore their own linguistic paths.

3.1 Beyond Token-Level Distillation

While off-policy distillation via SFT is effective for style transfer, it often leads to *Mode Collapse* and *Style Mimicry* (Gudibande et al., 2023), where the student mimics the teacher's linguistic patterns without grounding the underlying logical transitions. We propose a shift to an on-policy distillation via an RL objective. We treat a high-capacity teacher model $\mathcal{M}_T$ as a "reasoning supervisor." For a query q, $\mathcal{M}_T$ provides a structured rationale $y^T = (y_{think}, y_{ans})$. The student policy π_θ is optimized using RL algorithms, such as Group Relative Policy Optimization (GRPO) (Guo et al., 2025) to maximize the expected reward:

$$\mathcal{J}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[\min\left(r(o)\hat{A}, \text{clip}(r(o), 1-\epsilon, 1+\epsilon)\hat{A}\right)\right] \quad (1)$$

where $r(o) = \frac{\pi_\theta(o|q)}{\pi_{\theta_{\text{old}}}(o|q)}$, and the advantage $\hat{A}$ is calculated from our composite SPEAR reward. This objective encourages the student to explore diverse reasoning paths and optimize for correctness and logical consistency, rather than merely reproducing the teacher's surface form.

3.2 Domain-Adaptive Symbolic Anchors

The core of SPEAR is the projection function $\Phi : \mathcal{Y} \to \mathcal{A}$, which maps a high-dimensional natural language trace y into a low-dimensional *Symbolic Trajectory* $\mathcal{A}_t = [a_1, a_2, \ldots, a_n], \quad a_i \in \mathcal{V}_t$. We define domain-specific anchors a_i grounded in symbolic logic and linguistic dependency theory.

3.2.1 Formal Logic and Mathematical States

In quantitative reasoning, the structure is defined by the evolution of symbolic expressions and variable assignments. Inspired by symbolic state-space search (Lample and Charton, 2019), we define the mathematical anchor extraction function Φ_{math} as a regex-based projection that isolates the calculus of the reasoning trace from its natural language prose.

$$\mathcal{V}_{\text{math}} := \mathcal{V}_{\text{latex}} \cup \mathcal{V}_{\text{assign}} \quad (2)$$

where $\mathcal{V}_{\text{latex}}$ captures expressions wrapped in LaTeX delimiters (\$...\$, \[...\]) and $\mathcal{V}_{\text{assign}}$ captures explicit assignments (e.g., "$x = 5$"). Notably, we exclude bare numerical values from the extraction to ensure that the alignment reward measures the **structural derivation** of the problem rather than the occurrence of constants. By focusing on these symbolic transitions, Φ_{math} captures the trajectory of the problem-solving process, ensuring the student model respects the necessary order of

operations, such as the isolation of variables prior to substitution.

3.2.2 Causal Dependency Anchors for Science

For scientific reasoning, logic is embedded in interactions between domain-specific entities and actions. Standard overlap metrics fail here because they ignore the directionality of these relations. Grounded in the finding that syntactic structure is a primary driver of semantic role representation (Punyakanok et al., 2008), we implement Φ_{sci} using a dependency-parsing framework following successful applications in large-scale scientific relation extraction (Percha and Altman, 2018). This yields a de-duplicated relational sequence where each anchor a_i is a verb-centered relational tuple, with v, sbj, obj denoting the governing verb, subject, and object respectively.

$$\mathcal{V}_{\text{sci}} := \begin{cases} (\text{lemma}(v), \text{span}(obj)) & \text{if object } obj \text{ exists} \\ (\text{span}(sbj), \text{lemma}(v)) & \text{otherwise} \end{cases} \tag{3}$$

To prevent reward inflation through repetitive reasoning loops, $\mathcal{A}_{\text{sci}}$ enforces a uniqueness constraint: $\forall i \neq j : a_i \neq a_j$, ensuring SPEAR rewards progression through the teacher's causal chain rather than dwelling on a single repeated fact.

3.2.3 Commonsense State Transitions

In open-domain reasoning, such as commonsense, the logic typically tracks the movement of agents or the evolution of objects through temporal states. Grounded in *Script Theory* (Schank and Abelson, 2013), which posits that knowledge is organized around stereotypical event sequences, we implement Φ_{com} as a projection into a sequence of State-Action anchors. We extract these anchors by identifying noun phrases (noun chunks) and their governing syntactic heads. Formally, for each noun chunk c, $\mathcal{V}_{\text{com}}$ is defined as:

$$\mathcal{V}_{\text{com}} := \begin{cases} (\text{root}(c),\ \text{lemma}(\text{head}(c))) & \text{if head}(c) \in \text{VERB} \\ \text{root}(c) & \text{otherwise} \end{cases} \tag{4}$$

These anchors act as logical key-frames, capturing the student's ability to track world-state changes chronologically (e.g., *ball, throw"* $\rightarrow$ *window, break"*). By extracting the root of the noun phrase and the lemma of the action, it ensures that the alignment reward is invariant to modifiers and tense variations, focusing purely on the **state transition** intended by the teacher.

3.3 Sequential Alignment via LCS

Motivated by the success of longest common subsequence (LCS) rewards in identifying coherent reasoning trajectories across diverse tasks (Dong and Fan, 2025), we employ LCS to enforce chronological logical consistency between the student trajectory $\mathcal{A}_s$ and the teacher trajectory $\mathcal{A}_t$. Unlike unordered metrics such as Jaccard similarity or surface-level metrics like ROUGE (Lin, 2004), LCS is strictly **order-aware** and invariant to redundant linguistic filler that does not contribute to the symbolic path.

We define the reasoning process reward R_{reason} as the **alignment precision**:

$$R_{reason} = \frac{|\text{LCS}(\mathcal{A}_s, \mathcal{A}_t)|}{\max(1, |\mathcal{A}_s|)} \tag{5}$$

While most alignment objectives focus on recall (coverage of the teacher's steps), our precision-oriented formulation specifically addresses the phenomenon of *length bias*—a common failure mode in RL where models learn to generate excessively verbose reasoning to inflate reward signals. By normalizing by the student's trajectory length $|\mathcal{A}_s|$, it penalizes the generation of spurious or hallucinated milestones that diverge from the teacher's rationale. This forces the student model to maintain a high **logical density**, rewarding only the necessary steps required to bridge the query to the correct conclusion.

To ensure the reward is computable efficiently during real-time RL rollouts, we use a dynamic programming approach where the alignment state $L(i, j)$ is computed as follows:

$$L(i,j) = \begin{cases} L(i-1, j-1) + 1 & \text{if } a_{s,i} = a_{t,j} \\ \max(L(i, j-1), L(i-1, j)) & \text{otherwise} \end{cases} \tag{6}$$

This provides a dense reward signal that penalizes logical reversals while rewarding the correct chronological order of reasoning milestones.

3.4 Phased Composite Reward Formulation

To stabilize the RL process, we integrate SPEAR into a phased reward structure (Xie et al., 2025). We implement a strict gate for formatting compliance R_{fmt}, checking `<think>` and `<answer>` blocks. It follows by a weighted combination of accuracy R_{acc} and reasoning process alignment R_{reason}. The total reward R_{spear} is defined as:

$$R_{spear} = \mathbb{I}_{\text{fmt}} \cdot (R_{acc} + \lambda \cdot R_{reason}) \tag{7}$$

where $\mathbb{I}_{\text{fmt}} \in \{0, 1\}$ is an indicator function for format compliance. In our implementation, we set $\lambda = 0.5$. Unlike purely outcome-based supervision, which provides a sparse binary signal (0 or 1), R_{reason} serves as a **dense shaping signal** in $[0, 1]$. By awarding **partial credit** for reasoning milestones even when the final answer is incorrect, this design prevents gradient collapse during early training phases and encourages the model to refine its logical trajectory throughout the exploration.

Algorithm 1 SPEAR Reward Calculation

```
procedure COMPUTESPEAR(y_s, y_t, y_gold, τ)
  ▷ y_s: student response, y_t: teacher response,
  y_gold: gold answer, τ: task type
    y_think, y_ans ← ParseTags(y_s)
    if ¬ValidFormat(y_s) then
        return 0.0
    end if
    R_acc ← VerifyAnswer(y_ans, y_gold)
    A_s ← Φ(y_think, τ)
    A_t ← Φ(y_t, τ)
    if A_s = ∅ ∨ A_t = ∅ then
        return R_acc
    end if
    ℓ ← LCS(A_s, A_t)
    R_reason ← ℓ / max(1, |A_s|)
    return R_acc + λ · R_reason
end procedure
```

4 Experiments

4.1 Datasets

We focus on evaluating reasoning abilities with various datasets, including mathematical reasoning with GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021), scientific reasoning with GPQA (Rein et al., 2024), and commonsense reasoning with CommonsenseQA (Talmor et al., 2019). Datasets statistics are shown in Appendix A.

4.2 Baselines

To evaluate the effectiveness of our method, we compare it against **SFT distillation** and the following RL baselines, which utilize the standard sparse **outcome-based reward**:

- **GRPO** (Guo et al., 2025): The standard Group Relative Policy Optimization optimized via outcome-based rewards.
- **Dr. GRPO** (Liu et al., 2025): A distilled-reasoning variant of GRPO optimized with outcome-based signals.
- **DAPO** (Yu et al., 2025): A direct alignment policy optimization framework restricted to terminal outcome rewards.

We also include rule-based reward baseline **Logic-RL** (Xie et al., 2025), which gives partial reward when the format or final answer is not fully correct.

4.3 Models

Since we aim to distill the teacher's thinking process, we employ DeepSeek-V3.2 (Guo et al., 2025) as the teacher LLM, owing to its strong thinking mode and relatively low cost. For student SLMs, we choose Llama-3-8B-Instruct (Dubey et al., 2024) for its original reasonable performance on the chosen benchmarks, as well as Qwen3-4B (Yang et al., 2025) to test the generalizability of the SPEAR method.
All evaluation results are based on the zero-shot test set. We use spaCy (Honnibal et al., 2020) and `en_core_web_sm` as tools to extract dependency anchors. We set $\lambda = 0.5$ in Equation 7 for the main experiments. More implementation details are in Appendix B.

4.4 Main Results

Main results are shown in Table 1.
Insights of Distillation Across both student architectures, distillation generally improves base performance, but the effect is **task-dependent**. The gains are most consistent on mathematical reasoning, where both SFT and RL-based distillation improve GSM8K and MATH. However, the improvement is smaller when the base model is already strong or near **saturation**. This is especially evident for Qwen3-4B on GSM8K (88.10%), leaving limited room for further improvement. Distillation is also not uniformly beneficial for commonsense reasoning: for example, Qwen3-4B drops from 83.95% to 80.90%, and Llama3-8B-Instruct drops from 75.67% to 74.75% on CommonsenseQA under SFT, suggesting that forcing the student to directly learn from teacher's natural language, which may contain implicit and less transferable patterns, can impair the distillation.
Comparison between RL and SFT Distillation The table reveals a clear trade-off between SFT and RL-based distillation. On the mathematical

Method	Mathematical		Scientific	Commonsense	Avg. Δ
	GSM8K	MATH	GPQA	CommonsenseQA	
Teacher LLM					
DeepSeek-V3.2	95.75	90.20	59.10	90.16	
Llama-3-8B-Instruct					
Zero-shot (Base)	67.10	26.80	25.75	75.67	
SFT (Distilled)	73.54	30.40	33.83	74.75	
GRPO	78.54	28.20	29.29	75.74	–
GRPO + Logic-RL (Xie et al., 2025)	79.00	29.00	29.80	74.59	+0.16%
GRPO + SPEAR (Ours)	**80.74**	**30.00**	**31.31**	**76.80**	**+1.77%**
Dr. GRPO	75.28	30.00	32.83	77.05	–
Dr. GRPO + Logic-RL (Xie et al., 2025)	75.97	31.20	32.32	77.46	+0.45%
Dr. GRPO + SPEAR (Ours)	**77.33**	**32.40**	**35.35**	**78.11**	**+2.01%**
DAPO	77.79	31.60	30.81	77.54	–
DAPO + Logic-RL (Xie et al., 2025)	78.24	32.40	31.31	77.05	+0.32%
DAPO + SPEAR (Ours)	**79.98**	**33.40**	**32.32**	**78.36**	**+1.58%**
Qwen3-4B					
Zero-shot (Base)	88.10	67.40	18.69	83.95	
SFT (Distilled)	90.98	70.20	24.24	80.90	
GRPO	92.11	71.40	20.20	86.98	–
GRPO + Logic-RL (Xie et al., 2025)	92.49	71.80	20.70	87.05	+0.34%
GRPO + SPEAR (Ours)	**93.86**	**73.40**	**23.23**	**88.20**	**+2.00%**
Dr. GRPO	91.50	72.00	22.73	85.01	–
Dr. GRPO + Logic-RL (Xie et al., 2025)	91.81	72.60	22.22	85.08	+0.12%
Dr. GRPO + SPEAR (Ours)	**92.87**	**73.80**	**24.75**	**86.07**	**+1.56%**
DAPO	92.49	72.80	23.74	86.00	–
DAPO + Logic-RL (Xie et al., 2025)	92.87	73.40	24.24	85.74	+0.31%
DAPO + SPEAR (Ours)	**94.31**	**74.60**	**25.76**	**87.05**	**+1.67%**

Table 1: Accuracy (%) across various reasoning tasks with different distillation methods. Results demonstrate that SPEAR consistently outperforms other outcome-only baselines across multiple distillation frameworks and student architectures. Avg. Δ denotes the average improvement over the corresponding base RL method. Best results for each student model are **bolded**.

benchmarks, RL is generally more effective than SFT, consistent with the fact that the **final answer is strongly coupled with stepwise reasoning in math problems**. By contrast, SFT remains highly competitive on GPQA, especially for Llama-3-8B-Instruct, where SFT achieves 33.83%, exceeding all three plain RL baselines. A similar pattern is also observed for Qwen3-4B. This suggests that for multi-choice scientific tasks such as GPQA, it needs complex and scientific reasoning processes, while it's still possible for models to **guess the final option correctly**. Thus, direct imitation of teacher explanations can sometimes transfer more useful domain knowledge than outcome-only exploration.

SPEAR Advantages over Baselines For all tasks, our SPEAR method consistently improves over the baselines across different SLMs and RL baselines. These gains are more evident on mathematical tasks, where our math-specific reward is better able to capture structured logic anchors in model responses. In contrast, CommonsenseQA shows the smallest improvement, suggesting that **commonsense reasoning is harder to analyze and reward using extracted anchors**, because it relies more on natural language variation and implicit structural patterns. Moreover, SPEAR consistently outperforms the reward baseline Logic-RL, indicating that format and final answer-based **partial reward signals are not sufficient for reliable distillation**. In fact, such weak supervision can even hurt performance when the rationale is loosely correlated with the final answer, as shown in the case of scientific and commonsense tasks.

5 Discussion

5.1 Performance-Efficiency Analysis

To study the tradeoff between SPEAR and process reward models (PRMs), we compare against Qwen2.5-Math-PRM-7B (Zhang et al., 2025), a math-specialized PRM, and VersaPRM (Zeng et al., 2025), a multi-domain PRM. We run the same distillation setup on Qwen3-4B and only replace the reward signal with the average process reward produced by each PRM. The results are reported in Table 2.

Qwen2.5-Math-PRM-7B consistently achieves

the best performance on MATH, which is expected given its domain-specific supervision. However, its advantage over SPEAR is small, with SPEAR achieving comparable performance (within 1-2%) on all three RL backbones. In contrast, it transfers poorly to GPQA, where it underperforms SPEAR by an average of 3.87%. This suggests that a domain-specific PRM can have poor transferability for Out-Of-Distribution (OOD) conditions.

VersaPRM, as a multi-domain PRM, is substantially stronger on GPQA and generally provides better cross-domain supervision than the math-only PRM. It slightly exceeds SPEAR in some settings, but SPEAR still remains competitive across both MATH and GPQA. Importantly, SPEAR relies only on lightweight processing with spaCy and LCS, where the only external component is the small `en_core_web_sm` model ($\sim$12MB). This stands in sharp contrast to neural PRMs, which require an additional billions parameter model for process supervision. Overall, the result demonstrates that SPEAR provides a highly **favorable performance-efficiency tradeoff**, delivering competitive performance with significantly lower computational cost.

Qwen3-4B	MATH	GPQA	Overhead
GRPO + Qwen2.5-Math-PRM	**74.30**	20.20	7B Neural PRM
GRPO + VersaPRM	73.69	**24.24**	8B Neural PRM
GRPO + SPEAR	73.40	23.23	spaCy + LCS ($\sim$12MB)
Dr. GRPO + Qwen2.5-Math-PRM	**74.45**	21.21	7B Neural PRM
Dr. GRPO + VersaPRM	73.61	24.24	8B Neural PRM
Dr. GRPO + SPEAR	73.80	**24.75**	spaCy + LCS ($\sim$12MB)
DAPO + Qwen2.5-Math-PRM	**75.21**	20.70	7B Neural PRM
DAPO + VersaPRM	74.91	**26.77**	8B Neural PRM
DAPO + SPEAR	74.60	25.76	spaCy + LCS ($\sim$12MB)

Table 2: Performance–efficiency tradeoff between neural PRMs and SPEAR on Qwen3-4B. SPEAR achieves competitive performance while using only a lightweight `en_core_web_sm` ($\sim$12MB) parser, compared to billion-parameter neural PRMs.

5.2 Ablation of Reward Functions

Table 3 analyzes the contribution of each component in SPEAR.

Reward Components Removing the format gate, which enforces the `<think>` and `<answer>` structure, leads to a consistent drop on both benchmarks, indicating that structured outputs help stabilize training and enable reliable reward computation. In contrast, removing the accuracy reward results in the largest degradation (e.g., $30.00 \rightarrow 25.40$ on MATH), confirming that final-answer supervision remains the dominant learning signal in RL-based distillation.

Reasoning Weight We vary the coefficient λ in Equation 7, where $\lambda = 0.5$ is used in our main experiments. Setting $\lambda = 0$ removes the process reasoning reward and reduces performance on both tasks, showing that outcome-only supervision is insufficient. MATH performs best with $\lambda = 0.5$, while GPQA peaks at a higher value ($\lambda = 0.75$). However, further increasing λ to 1.0 degrades performance, suggesting that over-emphasizing reasoning alignment can harm optimization, and λ needs to be optimized for different tasks.

Alignment Design Replacing symbolic anchors with raw-token LCS degrades performance or removing order information with Jaccard overlap both lead to around 2% drop. This indicates that SPEAR benefits from both abstraction (symbolic anchors) and sequential structure (order-aware alignment), which together better capture the causal nature of reasoning processes.

Anchor Extraction Using generic surface-level anchors (e.g., logic markers such as "however", "wait", "thus") instead of our task-specific extraction also reduces 2.22% accuracy on average. This suggests that domain-aware milestone extraction is crucial for providing informative reasoning signals.

Overall, these results validate that SPEAR improves distillation through the joint effect of format constraints, answer supervision, and order-aware symbolic reasoning alignment, with each component playing a distinct and necessary role.

Llama-3-8B-Instruct	MATH	GPQA
SPEAR (GRPO, $\lambda = 0.5$)	**30.00**	31.31
Reward Components		
w/o Format Gate	28.80	30.30
w/o Accuracy Reward	25.40	26.77
Reasoning Weight (λ)		
$\lambda = 0.0$ (w/o process reasoning)	28.20	29.29
$\lambda = 0.25$	29.40	30.30
$\lambda = 0.75$	29.00	**31.82**
$\lambda = 1.0$	28.40	30.81
Alignment Design		
Raw Token LCS	28.00	28.79
Unordered Symbol Overlap (Jaccard)	28.20	29.80
Anchor Extraction		
Generic Surface Anchors	28.60	28.28

Table 3: Ablation of SPEAR reward design on Llama-3-8B-Instruct. Results highlight the importance of accuracy supervision, balanced process reasoning weight, and order-aware symbolic alignment.

5.3 Effect of Cold-Start SFT on RL Distillation

To further study the interaction between cold-start SFT and SPEAR-based RL in distillation, we vary the fraction of data used for SFT before RL and evaluate how the final performance changes. Specifically, we train Qwen3-4B on the MATH dataset using teacher-generated responses. We cold-start the model via SFT on progressively larger fractions of the training data (10% increments), and subsequently apply GRPO with SPEAR on the rest of the training data to evaluate performance improvements.

The result is shown in Figure 3. Starting from no SFT (100% SPEAR-based RL), introducing a small amount of supervised data consistently improves performance, as SFT helps the model learn basic reasoning patterns and output structure before RL optimization. The best performance is achieved when using a moderate fraction of SFT data (30%), suggesting that SFT provides a strong initialization that enhances the effectiveness of subsequent RL training. However, as the proportion of SFT data increases further, performance gradually declines. This is likely because excessive SFT reduces the diversity of exploration during RL, causing the model to overfit to teacher-generated responses. Overall, the results demonstrate that a balanced combination of SFT and SPEAR-based RL is crucial, where SFT provides a stable starting point, and RL refines reasoning through reward-driven optimization.

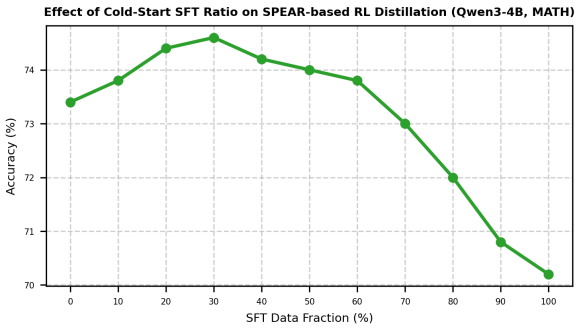

Figure 3: Effect of cold-start SFT ratio on SPEAR-based RL distillation (Qwen3-4B, MATH). A moderate amount of SFT data yields the best result, while excessive SFT reduces the subsequent RL performance.

5.4 Qualitative Case Study

To illustrate how SPEAR transfers reasoning structure, we present a representative GPQA example and compare the anchors extracted from the teacher and student rationales. For GPQA, our extractor forms anchors as verb-object pairs via dependency parsing, so the comparison focuses on whether the student preserves the teacher's ordered causal relations in its generated `<think>` trace.

Case Study: SPEAR Reward Signal on GPQA

Question: Trans-cinnamaldehyde was treated with methylmagnesium bromide (product 1); then with pyridinium chlorochromate (product 2); then with (dimethyl(oxo)-l6-sulfaneylidene)methane in DMSO at elevated temperature (product 3). How many carbon atoms are in product 3?

Teacher Response ($\mathcal{T}$): `<think>` Methylmagnesium bromide adds a methyl group to the aldehyde, giving a 10-carbon alcohol. Then, pyridinium chlorochromate oxidizes the alcohol to a ketone. Dimethyloxosulfonium methylide adds one carbon unit to form a cyclopropyl ring. `</think>` `<answer>` Thus, the product contains 11 carbon atoms. `</answer>`

Student Response ($\mathcal{S}$): `<think>` The Grignard reagent adds a methyl group to the aldehyde. Pyridinium chlorochromate oxidizes the alcohol to a ketone. The sulfonium ylide removes a methylene unit to the alkene. `</think>` `<answer>` 11 `</answer>`

The underlined anchors are compared in sequence :

$\mathcal{T}$ Anchors	$\mathcal{S}$ Anchors	Match
`add methyl group`	`add methyl group`	✓
`oxidize alcohol`	`oxidize alcohol`	✓
`add carbon unit`	`remove methylene unit`	×

$\mathrm{LCS}(\mathcal{T}, \mathcal{S}) = 2, \quad R_{\text{reason}} = \frac{|\mathrm{LCS}|}{|\mathcal{S}|} = \frac{2}{3} = 0.67$

Notably, verb-object anchors correctly absorb the subject paraphrase in step 1 and 2, yielding an exact match regardless of how the reagent is named. Step 3 fails (`remove methylene unit` vs. `add carbon unit`) due to lexical divergence in the object. Crucially, **both responses reach the correct answer**, so the accuracy reward assigns identical scores to both; SPEAR's combined reward penalizes the imprecise terminology in step 3, providing a finer training signal that accuracy alone cannot.

6 Conclusion

In this work, we introduce SPEAR, a training-free reward framework for sequence-level on-policy distillation. Across math, science, and commonsense tasks, SPEAR provides dense process supervision with a favorable efficiency–performance tradeoff. Our ablations also show that balanced reasoning weight, order-aware symbolic alignment, and a moderate SFT cold start are all important for effective distillation. We hope this work inspires efficient process supervision in distillation.

Limitations

Despite its effectiveness, SPEAR has several limitations. First, considering of the computation overhead, our method relies on symbolic anchor extraction and exact subsequence matching, which may fail to fully capture semantically equivalent but structurally different reasoning trajectories. Consequently, valid alternative reasoning paths that diverge from the teacher's symbolic ordering can receive lower rewards, particularly in open-ended reasoning tasks. Second, SPEAR assumes that teacher-generated rationales provide reliable reasoning trajectories, although LLM explanations may contain redundant or suboptimal intermediate steps. Additionally, our current implementation depends on rule-based extraction pipelines, such as regex matching and dependency parsing, which may be sensitive to parsing errors and formatting inconsistencies. Finally, due to computational constraints, we focus exclusively on sequence-level on-policy distillation and do not explore token-level reverse-KL distillation, which remain promising directions for future work.

Ethics Statement

Use of AI Assistants We used AI Assistants only for minor text polishing and part of code implementation. All ideas, experiments, analyses, and discussions were conducted solely by the authors. The AI Assistants did not contribute to the design and interpretation of our research.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence. https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf. Technical report.

Yubo Dong and Hehe Fan. 2025. Enhancing large language models through structured reasoning. *arXiv preprint arXiv:2506.20241*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*.

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

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

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, et al. 2020. spacy: Industrial-strength natural language processing in python.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Guillaume Lample and François Charton. 2019. Deep learning for symbolic mathematics. *arXiv preprint arXiv:1912.01412*.

Wendi Li and Yixuan Li. 2024. Process reward model with q-value rankings. *arXiv preprint arXiv:2410.11287*.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, et al. 2026. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Kevin Lu and Thinking Machines Lab. 2025. On-policy distillation. *Thinking Machines Lab: Connectionism*. Https://thinkingmachines.ai/blog/on-policy-distillation.

Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. Reft: Reasoning with reinforced fine-tuning. *arXiv preprint arXiv:2401.08967*.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781.

OpenAI. 2024. Learning to reason with llms. Accessed: 2026-02-27.

Bethany Percha and Russ B Altman. 2018. A global network of biomedical relationships derived from text. *Bioinformatics*, 34(15):2614–2624.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational linguistics*, 34(2):257–287.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Roger C Schank and Robert P Abelson. 2013. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology press.

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

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. 2025. Prmbench: A fine-grained and challenging benchmark for process-level reward models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25299–25346.

Mingyang Song and Mao Zheng. 2025. Walk before you run! concise llm reasoning via reinforcement learning. *arXiv preprint arXiv:2505.21178*.

Mingyang Song and Mao Zheng. 2026. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*.

Hongyuan Su, Yu Zheng, Yuan Yuan, Yuming Lin, Depeng Jin, and Yong Li. 2025. Reinforcement learning with adaptive reward modeling for expensive-to-evaluate systems. In *Forty-second International Conference on Machine Learning*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Xi Wang, Haoqi Zheng, Xinrui Shi, Meng Zhang, Yuhua Tang, Jing Ren, Liyang Xu, Wanrong Huang, and Ruochun Jin. 2026. Repair: A rule-based process-adaptive reinforcement for large language model training.

Xiangyu Wen, Junhua Huang, Zeju Li, Min Li, Jianyuan Zhong, Zhijian Xu, Mingxuan Yuan, Yongxiang Huang, and Qiang Xu. 2025. Reasoning scaffolding: Distilling the flow of thought from llms. *arXiv preprint arXiv:2509.23619*.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. 2026. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*.

Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. 2025a. Kdrl: Post-training reasoning llms via unified knowledge distillation and reinforcement learning. *arXiv preprint arXiv:2506.02208*.

Shicheng Xu, Liang Pang, Yunchang Zhu, Jia Gu, Zihao Wei, Jingcheng Deng, Feiyang Pan, Huawei Shen, and Xueqi Cheng. 2025b. Distilling the implicit multi-branch structure in llms' reasoning via reinforcement learning. *arXiv preprint arXiv:2505.16142*.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.

Zhichao Xu, Zongyu Wu, Yun Zhou, Aosong Feng, Kang Zhou, Sangmin Woo, Kiran Ramnath, Yijun Tian, Xuan Qi, Weikang Qiu, et al. 2025c. Beyond correctness: Rewarding faithful reasoning in retrieval-augmented generation. *arXiv preprint arXiv:2510.13272*.

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

Junchi Yu, Ran He, and Rex Ying. 2023. Thought propagation: An analogical approach to complex reasoning with large language models. *arXiv preprint arXiv:2310.03965*.

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

Zhuohao Yu, Weizheng Gu, Yidong Wang, Xingru Jiang, Zhengran Zeng, Jindong Wang, Wei Ye, and Shikun Zhang. 2024. Reasoning through execution: Unifying process and outcome rewards for code generation. *arXiv preprint arXiv:2412.15118*.

Thomas Zeng, Shuibai Zhang, Shutong Wu, Christian Classen, Daewon Chae, Ethan Ewer, Minjae Lee, Heeju Kim, Wonjun Kang, Jackson Kunde, et al. 2025. Versaprm: Multi-domain process reward model via synthetic reasoning data. *arXiv preprint arXiv:2502.06737*.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10495–10516.

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

A Datasets Statistics

We download datasets GSM8K, GPQA, and CommonsenseQA from Huggingface, MATH from their official project website: https://github.com/hendrycks/math. GSM8K dataset is split according to the official original split ratio. We use the official training set for Math and MATH-500 for the test set due to its high representation and low cost. Since there is no official train/test split for GPQA, we use gpqa_main and gpqa_extended as the training set, gpqa_diamond as the test set. Table 4 shows the statistics of all datasets.

Dataset	Type	#Train	#Test
GSM8K	Mathematics	7473	1319
MATH	Mathematics	7500	500
GPQA	Science	994	198
CommonsenseQA	Commonsense	9740	1220

Table 4: Dataset statistics.

B Implementation Details

Primary experiments are conducted on four Nvidia A100-80GB GPUs.

B.1 Teacher LLM Parameters

DeepSeek-V3.2 (Guo et al., 2025) is required by the official API: https://api-docs.deepseek.com.

Parameter	Value
temperature	0
max new tokens	2048
do sample	True
model	deepseek-reasoner
response_format	{'type': 'text'}

Table 5: Teacher LLM parameter settings.

B.2 Student SLMs Parameters

Experiments are performed with the Huggingface trl framework. We use four Nvidia A100-80GB GPUs with FP16 and LoRA (Hu et al., 2022) for training and evaluation. The inference parameter settings across all datasets are shown in Table 6. The training hyperparameter settings across all datasets are shown in Table 7 and Table 8.

Parameter	Value
temperature	0.2
max_new_tokens	2048
top_p	0.9
top_k	50
do_sample	True
batch_size	16

Table 6: Student SLMs inference parameter settings.

Hyperparameter	Value
epoch	5
batch_size	8
learning_rate	2e-5
max_seq_length	2048
gradient_accumulation_steps	4
warmup_ratio	0.03
weight_decay	0.01
lr_scheduler_type	"linear"
max_grad_norm	1.0
lora_alpha	16
lora_dropout	0.1
r	8
task_type	"CAUSAL_LM"
target_modules	["q_proj", "v_proj"]

Table 7: Student SLMs SFT hyperparameter settings.

Hyperparameter	Value
epoch	1
batch_size	4
learning_rate	1e-5
max_completion_length	2048
beta	0.04
num_generations	4
gradient_accumulation_steps	4
scale_rewards	"group"
lora_alpha	32
lora_dropout	0.1
r	8
task_type	"CAUSAL_LM"
target_modules	["q_proj", "v_proj"]

Table 8: Student SLMs RL hyperparameter settings.

C Hyperparameter Tuning

We tuned λ on the full training set, ranging between [0, 1], and showed the performance on the final test set in Table 3. Moreover, for learning rate, epoch, and other hyperparameters, we tuned different values on the entire training set and compared their performance on the entire test set to prevent underfitting or overfitting.

D Prompt Templates

- System prompt for all models.

  "A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think><answer> answer here </answer>."

- Prompt templates for different benchmarks.

  – For benchmarks GSM8K:
  question + "Let's think step by step."

  – For benchmark MATH:
  question + "Let's think step by step and put the final answer in `\boxed{}`."

  – For benchmark GPQA:
  question + "Put the final answer in `\boxed{}`." + options

  – For benchmark CommonsenseQA:
  question + " Put the final answer in `\boxed{}`." + options

E The Performance of Out-of-Distribution (OOD) Scenarios

To evaluate the abilities of SPEAR on out-of-distribution (OOD) data, we conducted experiments using the inconsistent domain symbolic anchors extraction methods. We extract the causal dependency anchors for MATH, and the mathematical states anchors for GPQA. Table 9 highlights the performance of the SPEAR method on Qwen3-4B in OOD scenarios.

The results indicate a decrease in performance improvement compared to the original in-distribution scenarios in Table 1. Specifically, using non-corresponding anchor extraction can harm the original distillation results, even resulting in a poorer performance compared to the original outcome-based RL baselines. This is consistent with our domain-adaptive reward design concept, which proves that different reasoning tasks requires customized logical skeleton extraction approaches.

Method	MATH	GPQA
GRPO+SPEAR	71.00	20.70
Dr. GRPO+SPEAR	71.60	21.72
DAPO+SPEAR	73.00	23.23

Table 9: The performance of SPEAR on Qwen3-4B in out-of-distribution (OOD) scenarios. Specifically, we conducted the non-corresponding domain anchor extraction methods on MATH and GPQA datasets.

F Case Study of Distillation Impact on Student LM's Output

Table 10, 11, 12 provides the comparisons of student model Llama-3-8B-Instruct behavior difference before and after the SPEAR distillation across benchmarks. It is evident that the <think> and <answer> parts from teacher LLMs help the student LM with reasoning think and finally reach the correct answer.

MATH Example

Question	Express $\frac{165_7}{11_2} + \frac{121_6}{21_3}$ in base 10. **Gold answer:** 39
Student's output before distillation	First, convert the numbers into base 10. $165_7 = 1 \times 7^2 + 6 \times 7 + 5 = 96$ $11_2 = 2$ So, $\frac{165_7}{11_2} = \frac{96}{2} = 48$ Next, $121_6 = 36 + 12 + 1 = 49$ $21_3 = 2 \times 3 + 1 = 7$ Thus, $\frac{121_6}{21_3} = \frac{49}{7} = 7$ Adding them together gives $48 + 7 = \boxed{55}$.
Teachers' response	<think>The expression is evaluated by converting each number from its given base to base 10: $165_7 = 1 \times 7^2 + 6 \times 7^1 + 5 \times 7^0 = 49 + 42 + 5 = 96$ $11_2 = 1 \times 2^1 + 1 \times 2^0 = 2 + 1 = 3$ $\frac{165_7}{11_2} = \frac{96}{3} = 32$ Next, $121_6 = 1 \times 6^2 + 2 \times 6^1 + 1 \times 6^0 = 36 + 12 + 1 = 49$ Thus, $21_3 = 2 \times 3^1 + 1 \times 3^0 = 6 + 1 = 7$ $\frac{121_6}{21_3} = \frac{49}{7} = 7$ Adding the results: $32 + 7 = 39$. </think> <answer>Thus, the expression in base 10 is $\boxed{39}$.</answer>
Student's output after distillation	<think> Convert each number into base 10 before evaluating the fractions. $165_7 = 1 \times 7^2 + 6 \times 7^1 + 5 \times 7^0 = 49 + 42 + 5 = 96$ $11_2 = 1 \times 2^1 + 1 \times 2^0 = 2 + 1 = 3$ $\frac{165_7}{11_2} = \frac{96}{3} = 32$ $121_6 = 1 \times 6^2 + 2 \times 6^1 + 1 \times 6^0 = 36 + 12 + 1 = 49$ $21_3 = 2 \times 3^1 + 1 \times 3^0 = 6 + 1 = 7$ $\frac{121_6}{21_3} = \frac{49}{7} = 7$ Finally, $32 + 7 = 39$ </think> <answer> Therefore, the expression in base 10 is $\boxed{39}$.</answer>

Table 10: Case study of student LM Llama-3-8B-Instruct output on MATH before and after distillation.

GPQA Example	
Question	What is the major outcome of the reaction between cyclopent-1-enol and bromine? **Options:** 2-bromocyclopentanone; 3-bromocyclopentanone; (1R,2R)-1,2-dibromocyclopentanol; (1R,2S)-1,2-dibromocyclopentanol
Student's output before distillation	Bromine usually adds across double bonds in alkenes. Since cyclopent-1-enol contains a double bond, bromine reacts by forming a dibrominated product across the alkene. Therefore, the major product is (1R,2R)-1,2-dibromocyclopentanol.
Teachers' response	<think> Cyclopent-1-enol is an enol, meaning it contains both a carbon-carbon double bond and an alcohol group directly attached to the alkene carbon. Enols are generally unstable and readily undergo keto-enol tautomerization. Under bromination conditions, the enol form reacts as a nucleophilic alkene. Bromine electrophilically attacks the double bond, generating an α-brominated carbonyl compound after tautomerization. First, cyclopent-1-enol tautomerizes to cyclopentanone: $$\text{cyclopent-1-enol} \rightleftharpoons \text{cyclopentanone}$$ Then, bromination occurs at the α-position adjacent to the carbonyl group through enolization of cyclopentanone. This produces the α-brominated ketone: $$\text{cyclopentanone} + Br_2 \rightarrow \text{2-bromocyclopentanone}$$ Therefore, the reaction does not mainly give dibromination across the double bond. Instead, the major product is the brominated ketone formed via keto-enol tautomerization and α-substitution chemistry. </think> <answer> Thus, the major product of the reaction is 2-bromocyclopentanone. </answer>
Student's output after distillation	<think> Cyclopent-1-enol is an enol, which can tautomerize into cyclopentanone. In the presence of bromine, ketones typically undergo α-bromination at the carbon adjacent to the carbonyl group. After tautomerization to cyclopentanone, bromination occurs at the α-position, producing the brominated ketone. </think> <answer> Therefore, the major product is 2-bromocyclopentanone. </answer>

Table 11: Case study of student LM Llama-3-8B-Instruct output on GPQA before and after distillation.

CommonsenseQA Example

Question	What is a great place to lay in the sun? **Options:** A. in the basement; B. west; C. solar system; D. beach; E. beans
Student's output before distillation	A good place to lay in the sun should be somewhere related to sunlight or space. The solar system contains the sun and receives sunlight directly. Therefore, the best answer is C. solar system.
Teachers' response	<think> The question asks for a place where people commonly relax or sunbathe. Evaluate each option: - A. in the basement: basements are underground and usually dark, so they are not good places to lay in the sun. - B. west: west is a direction, not a physical place where someone would lay down. - C. solar system: although the sun is part of the solar system, people cannot physically lay there. This option is too broad and unrealistic. - D. beach: beaches are commonly associated with relaxing, sunbathing, and laying in the sun. People frequently go to beaches specifically to enjoy sunlight. - E. beans: beans are food and not a place. Among all options, "beach" is the only realistic and commonsense location associated with laying in the sun. </think> <answer> Therefore, the correct answer is D. beach. </answer>
Student's output after distillation	<think> The question asks for a realistic place where people commonly lay in the sun. A basement is dark, west is only a direction, the solar system is not a practical location, and beans are food. A beach is a common place for relaxing and sunbathing. </think> <answer> Thus, the correct answer is D. beach. </answer>

Table 12: Case study of student LM Llama-3-8B-Instruct output on CommonsenseQA before and after distillation.