Beyond Outcome Verification: Verifiable Process Reward Models for Structured Reasoning

Massimiliano Pronesti[1,2], **Anya Belz**[2], **Yufang Hou**[1,3]
[1]IBM Research, [2]Dublin City University,
[3]IT:U Interdisciplinary Transformation University Austria
Correspondence: massimiliano.pronesti@ibm.com, yufang.hou@it-u.at

Abstract

Recent work on reinforcement learning with verifiable rewards (RLVR) has shown that large language models (LLMs) can be substantially improved using outcome-level verification signals, such as unit tests for code or exact-match checks for mathematics. In parallel, process supervision has long been explored as a way to shape the intermediate reasoning behaviour of LLMs, but existing approaches rely on neural judges to score chain-of-thought steps, leaving them vulnerable to opacity, bias, and reward hacking. To address this gap, we introduce *Verifiable Process Reward Models* (VPRMs), a reinforcement-learning framework in which intermediate reasoning steps are checked by deterministic, rule-based verifiers. We apply VPRMs to risk-of-bias assessment for medical evidence synthesis, a domain where guideline-defined criteria and rule-based decision paths enable programmatic verification of reasoning traces. Across multiple datasets, we find that VPRMs generate reasoning that adheres closely to domain rules and achieve substantially higher coherence between step-level decisions and final labels. Results show that VPRMs achieve up to 20% higher F1 than state-of-the-art models and 6.5% higher than verifiable outcome rewards, with substantial gains in evidence grounding and logical coherence.

1 Introduction

Large language models (LLMs) have made remarkable progress in complex natural language processing tasks, including reasoning, planning, and structured decision making (Brown et al., 2020; OpenAI, 2024, 2025). Reinforcement learning with verifiable rewards (RLVR) has recently emerged as a robust alternative to preference-based reinforcement learning, enabling LLMs to improve using reward signals derived from deterministic checks such as program test suites or exact-match mathematical evaluation (Guo et al., 2025; Lambert et al., 2025). By grounding supervision in objective verifiers rather than learned reward models, RLVR avoids many of the alignment failures associated with neural reward hacking and has produced state-of-the-art performance in code generation and mathematical reasoning (Wang et al., 2025b; Zhang and Zuo, 2025; Guo et al., 2025; Yang et al., 2025a).

However, outcome-only RLVR provides rewards solely at the terminal step of reasoning, offering no guarantees about whether the model followed a valid intermediate process. To address this limitation, several extensions augment RLVR with structural or auxiliary signals, such as masked-and-reordered self-supervision (Wang et al., 2025c) or self-verification mechanisms (Zeng et al., 2025). These works strengthen RLVR but still operate fundamentally at the level of *outcome verification*. Most importantly, none of the above methods provide a fully verifiable form of process supervision, and existing approaches that score intermediate Chain-of-Thought (CoT) steps rely on neural judges (Lightman et al., 2024; Zhang et al., 2025; Zou et al., 2025) which reintroduce opacity, bias, and opportunities for reward hacking (Amodei et al., 2016; Skalse et al., 2022).

To date, no work has demonstrated that process rewards themselves can be made verifiable on tasks whose structure admits deterministic symbolic checking. Yet such a capability would be highly desirable: if intermediate reasoning steps can be validated explicitly, then reinforcement learning could optimise not only for correct outcomes, but also for transparent, logically sound reasoning. Crucially, verifiable step-level rewards would eliminate the aforementioned problems associated with neural process rewards.

This leaves a key open problem: can reinforcement learning be used to train models whose entire reasoning trajectory is rewarded only when each

Findings of the Association for Computational Linguistics: ACL 2026, pages 32187–32202

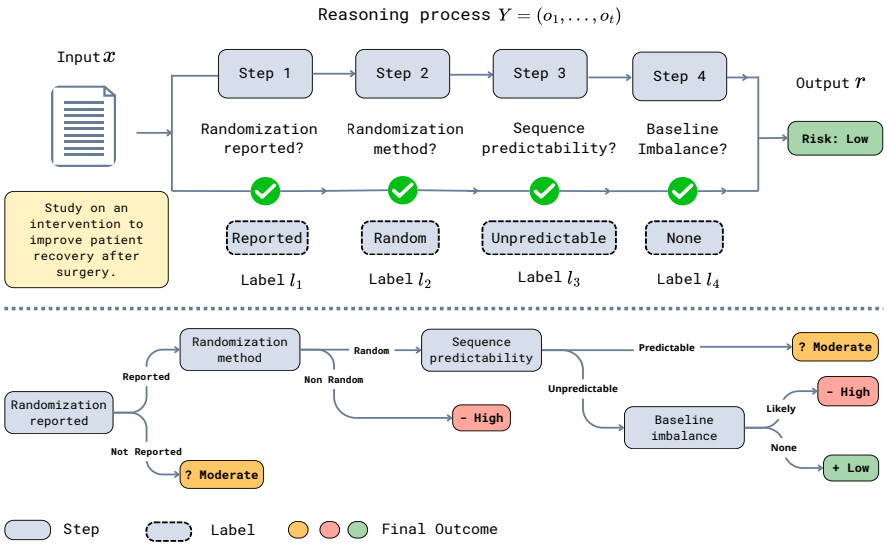

Figure 1: Illustration of the verifiable reasoning setup for risk-of-bias assessment (type A: bias arising from the randomisation process). Top: given an input study x, the model produces a structured reasoning trace $Y = (o_1, \ldots, o_T)$ with step-level labels $(\ell_1, \ell_2, \ell_3, \ell_4)$, each corresponding to a guideline-defined assessment question. Bottom: the corresponding rule-based decision tree, which deterministically maps each combination of step-level labels to low (**+**), high (**-**), or moderate (**?**) risk.

intermediate step satisfies domain-defined, rule-based criteria? To address this gap, we introduce **Verifiable Process Reward Models** (VPRMs), a reinforcement-learning framework in which each reasoning step is assessed by a deterministic verifier grounded in explicit task guidelines. VPRMs provide fine-grained, step-level reward signals that complement outcome-level verification, guiding optimisation toward reasoning traces that are both correct and aligned with domain logic. Crucially, we prove that under mild assumptions, VPRMs offer theoretical guarantees that gradient-based updates assign positive expected weight to correct reasoning trajectories and negative weight to inconsistent ones, thereby encouraging sound reasoning.

To evaluate this framework, we consider a challenging, real-world structured-reasoning task: risk-of-bias (RoB) assessment in clinical systematic reviews. In this setting, studies must be evaluated for susceptibility to systematic error (Chandler et al., 2019), and domain guidelines prescribe a rigid sequence of reasoning steps and decision rules that make the task uniquely amenable to verifiable process supervision. Figure 1 illustrates the verifiable reasoning process for assessing randomisation bias, one of the RoB domains defined in the Cochrane RoB tool for randomised trials (Sterne et al., 2019).

Across multiple models and RoB domains, we compare VPRMs against outcome-only RLVR, neural process-reward baselines, and pretrained LLMs prompted for Chain-of-Thought (CoT). Our results show that VPRM-trained models achieve substantially higher accuracy and more coherent reasoning traces, showing that verifiable process supervision offers a more reliable optimization signal for both result correctness and process soundness.

In summary, our contributions are as follows: (i) we propose a verifiable process reward framework that integrates deterministic step-level verification with reinforcement learning for dense, interpretable supervision over reasoning trajectories (Section 3); (ii) we show that, under mild reward-separation assumptions, verifiable process rewards encourage correct reasoning (Section 3.3, Appendix A); and (iii) we validate the approach on risk-of-bias assessment in medical systematic reviews, demonstrating significant improvements over outcome-only and neural process-reward baselines (Section 4).

2 Preliminaries

2.1 Policy Optimisation Algorithms

Group Relative Policy Optimization (GRPO) GRPO (Shao et al., 2024) is a widely-adopted group-based reinforcement learning method that optimises a policy by comparing multiple candidate completions for the same input.

For each passage $\mathbf{x}$, G candidate completions $\{y_i\}_{i=1}^{G} \sim \pi_{\text{old}}(\cdot \mid \mathbf{x})$ are sampled from the refer-

ence policy π_{old} to encourage robustness and diversity. These completions are scored using a reward model. The raw rewards R_i are then normalised across the group:

$$A_i = \frac{R_i - \mathbb{E}[R_j]}{\sqrt{\mathbb{V}[R_j]}}, \quad j \in \{i, ..., G\}$$

where $\mathbb{E}[R_j]$ and $\mathbb{V}[R_j]$ are respectively the mean and variance of the rewards for the group of responses. The policy is optimised using a clipped, KL-regularised objective that encourages agreement with high-reward behaviours while maintaining proximity to a reference model π_{ref}:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\Big(p_{i,t}(\theta)A_i, \text{clip}(p_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon)A_i\Big) - \beta\,\text{KL}[\pi_\theta \,\|\, \pi_{\text{ref}}]\Bigg]$$

where β governs the regularisation strength and $p_{i,t}(\theta)$ is the token-level probability ratio defined as follows:

$$p_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})}$$

Dynamic sAmpling Policy Optimization (DAPO) Building on GRPO, DAPO (Yu et al., 2025) removes the KL penalty, introduces a clip-higher mechanism, incorporates dynamic sampling, applies a token-level policy gradient loss, and adopts overlong reward shaping. The key improvement is dynamic sampling, by over-sampling and filtering out prompts with the accuracy equal to 1 and 0, leaving all prompts in the batch with effective gradients, avoiding dampening the gradient signals for model training with a larger variance in the gradient. This leads to the following maximisation objective:

$$\mathcal{L}_{\text{DAPO}}(\theta) = \mathbb{E}\Bigg[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\min\Big(p_{i,t}(\theta)A_i, \text{clip}(p_{i,t}(\theta), 1-\varepsilon_H, 1+\varepsilon_L)A_i\Big)\Bigg], \quad \text{s.t. } 0 < |\{y_i| \text{ is\_equivalent}(y_i, a)\}| < G$$

This ensures that for the same input, the sampled set contains both correct and incorrect answers.

2.2 Rule-based Reward Modeling

In rule-based reward modeling, the reward signal is defined by explicit, hand-crafted rules that verify whether a model output satisfies task-specific constraints. This approach is particularly effective for verifiable tasks with clear notions of correctness, such as mathematical problem solving, program synthesis, or logical reasoning. The reward is computed deterministically by a verifier and does not rely on learned preference models. In its simplest form, the reward is binary:

$$R(y) = \begin{cases} 1 & \text{if the output is verified as correct,} \\ 0 & \text{otherwise.} \end{cases}$$

This setting provides scalable, reliable supervision with minimal ambiguity.

2.3 Systematic Reviews

Systematic reviews provide a principled framework for aggregating empirical evidence through predefined search strategies, explicit inclusion criteria, and reproducible synthesis pipelines (Chandler et al., 2019). Their value lies in reducing subjective judgment in evidence collection and enabling structured comparison across heterogeneous studies. One of the gold-standard repositories of systematic reviews is the Cochrane library (), which contains curated reviews adhering to strict evidence-synthesis and bias-assessment protocols.

2.4 Risk of Bias Assessment

Risk of bias assessment is a core component of systematic reviews, providing structured evaluations of methodological flaws in primary studies that directly inform evidence synthesis and the credibility of review conclusions. Assessments are structured into a fixed set of domains of bias, focusing on different aspects of trial design, conduct and reporting (Chandler et al., 2019; Sterne et al., 2019). Within each domain, information about features of the trial that are relevant to risk of bias is collected and mapped to judgments of low, medium or high risk. This domain-based assessment identifies issues such as selection bias, measurement bias, and selective reporting, enabling subsequent evidence synthesis to appropriately weight studies not only in terms of their results, but also in terms of their risk of bias. Additional details on the different bias domains and the corresponding assessment reasoning process are provided in Appendix B.

3 Verifiable Process Reward Models

We introduce *Verifiable Process Reward Models* (VPRMs), a framework for process supervision

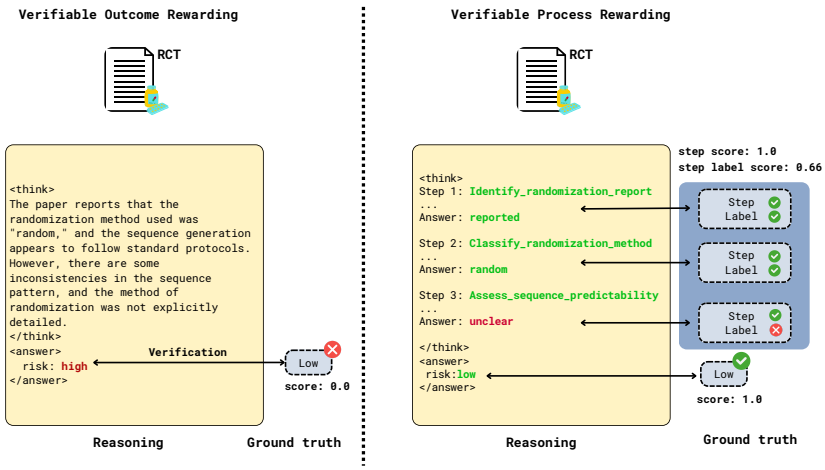

Figure 2: Comparison between verifiable outcome rewards (left), which evaluates only the final risk label, and verifiable process rewards (right), which additionally verifies each reasoning step and its associated label.

in which intermediate reasoning steps are evaluated by deterministic, externally checkable verifiers rather than learned neural judges. The framework is presented in the context of risk assessment tasks, which naturally admit structured reasoning steps, discrete labels, and rule-based transitions that can be validated against domain guidelines.

3.1 Reasoning Trajectories and Steps

For an input x, the model produces a reasoning trajectory $Y = (o_1, \ldots, o_T)$ with a stochastic policy

$$\pi_\theta(Y \mid x) = \prod_{t=1}^{T} \pi_\theta(o_t \mid o_{<t}, x).$$

Each step t contains two discrete outputs: (i) a step identifier $s_t \in \mathcal{S}$ and (ii) a step label $\hat{\ell}_t \in \mathcal{L}_t$, which represents the model's answer for that step. Domain guidelines specify, for each prefix $Y_{\le t}$, the *gold* step identifier $s_t^\star$ and gold step label $\ell_t^\star$ obtained by applying the rule-based logic of the task.

3.2 Verifiers and Process Rewards

Correctness is evaluated by two bounded scoring functions:

$$s_t^{\mathrm{n}}(s_t, s_t^\star), \qquad s_t^{\mathrm{l}}(\hat{\ell}_t, \ell_t^\star),$$

each mapping a model output and its corresponding gold value into $[0, 1]$. These provide a positive reward signal when the model selects the correct step identifier and the correct label according to the task rules. The instanteneous step-level verifiable reward is then defined as:

$$r_t(Y; x) = w_t^{\mathrm{n}}\, s_t^{\mathrm{n}}(s_t, s_t^\star) + w_t^{\mathrm{l}}\, s_t^{\mathrm{l}}(\hat{\ell}_t, \ell_t^\star),$$

where $w_t^{\mathrm{n}}, w_t^{\mathrm{l}} \ge 0$ are preset weights.

A terminal outcome reward r_{label} evaluates whether the final risk value predicted from the full reasoning trace matches the gold risk value. The full verifiable process reward is then

$$R(Y; x) = \sum_{t=1}^{T} r_t(Y; x) + r_{\mathrm{label}}$$

As illustrated in Figure 2, this reward is fully computable using deterministic, rule-based checks, making all components of the reasoning trajectory verifiable.

3.3 Reward Separation and Optimisation Guarantee

An interesting consequence of combining VPRMs with GRPO or DAPO is that the resulting optimisation dynamics exhibit a clear structure: rule-consistent trajectories are, in expectation, pushed in a beneficial direction.

Let $R(Y)$ denote the verifiable process reward assigned to a response y, and let G be the number of responses sampled for the input x. Let $\mathcal{C}$ be the event that Y is *correct* according to the rule-based task semantics. Define the conditional expectations

$$\mu_c := \mathbb{E}[R(Y) \mid \mathcal{C}], \qquad \mu_i := \mathbb{E}[R(Y) \mid \mathcal{C}^c].$$

We assume the following mild conditions (see Appendix A for the full formal statement and discussion): (i) $R(Y)$ has finite variance, (ii) correct reasoning chains receive strictly larger expected reward than incorrect ones ($\mu_c > \mu_i$), and (iii) a sufficiently large G to ensure stable gradient updates.

Theorem 1. Under the above hypotheses, the expected GRPO and DAPO advantage $\mathbb{E}[\hat{A}(Y)]$ satisfies

$$\mathbb{E}[\hat{A}(Y) \mid \mathcal{C}] > 0, \qquad \mathbb{E}[\hat{A}(Y) \mid \mathcal{C}^c] < 0.$$

Thus, both GRPO and DAPO assign positive expected weight to correct reasoning trajectories and negative weight to incorrect ones, raising the likelihood of correct reasoning in expectation.

This follows from the theoretical results presented by Wen et al. (2026). A proof for both GRPO and DAPO objectives is provided in Appendix A.

4 Experiments

4.1 Training Dataset Creation

The first stage of our methodology involved the acquisition of a high-quality, human-aligned corpus suitable for training reward models.

To this end, we build on the COCHRANEFOREST (Pronesti et al., 2025a) and COCHRANEFORESTEXT (Pronesti et al., 2025b) datasets, which provide two essential components for our task: (i) the forest plots extracted from Cochrane systematic reviews, and (ii) the *full-text* primary studies corresponding to every trial included in those plots. The inclusion of full papers is critical, as risk-related signals often depend on methodological details available only in the complete manuscripts.

From these corpora, we retain exclusively the forest plots that contain an associated risk-of-bias map. Each such plot establishes an explicit correspondence between its set of included studies and their study-level bias assessments. We therefore define a single instance as a *paper-risk pair* consisting of a full-text study and its aligned risk-of-bias definition and label extracted from the map. The resulting dataset comprises 2,946 instances drawn from 104 systematic reviews, totalling 4M tokens.

4.2 Synthetic Data Annotation for Structural Reasoning Processes

Following the methodology described by Pronesti et al. (2025b), we enrich our dataset with step-level labels for RL using LlaMa 3.1 405B (Grattafiori et al., 2024) with the system prompt shown in Figure 4 (Appendix), temperature of 0.7 and 2,048 tokens generation limit. An example data instance is provided in Table 9 (Appendix); a human verification of the generated annotations in Appendix D.

Dataset	Train	Test	Total	Avg tokens
COCHRANEFORESTEXT	2651	295	2946	13,596.9
COCHRANEFOREST	–	1846	1846	12,722.8
RoBBR Cochrane	774	906	1680	9,084.6
RoBBR Non-Cochrane	–	2489	2489	7,940.7

Table 1: Datasets statistics. Train/test split only applies to COCHRANEFORESTEXT and RoBBR Cochrane. COCHRANEFOREST and RoBBR Non-Cochrane are used for testing.

4.3 Experimental Setup

Training and Evaluation Datasets. For training, we use the corpus constructed with the methods from Section 4.1, allocating 2,651 instances for training and 295 for validation. We also include the 774 training instances from the RoBBR Cochrane split (Wang et al., 2025a). All training data are augmented with step-level labels (Section 4.2).

For evaluation, we consider three datasets. The first is COCHRANEFOREST (Pronesti et al., 2025b), which contains 1,846 instances drawn from 48 Cochrane Systematic Reviews and 202 forest plots. The second consists of the two test sets from the RoBBR benchmark: RoBBR Cochrane, which contains 906 datapoints originating from 204 papers included in 58 Cochrane reviews; and RoBBR–Non-Cochrane, which contains 2,489 datapoints drawn from 496 non-Cochrane reviews that collectively assess 496 papers. Dataset statistics are summarised in Table 1, while per-risk-type statistics are reported in Table 7 (Appendix).

Evaluation Metrics. We evaluate all models on the main prediction task using Accuracy and macro–F1, computed over the discrete risk labels.

In addition, for analyses (Section 4.6), we report *Coherence* for the VPRM-trained models, defined as the proportion of datapoints for which the model's predicted risk is consistent with the conclusion implied by its own intermediate reasoning. Formally, let $\hat{r}_i \in \mathcal{R}$ denote the final risk value predicted by the model for datapoint i, and let $\hat{\ell}_{i,1}, \ldots, \hat{\ell}_{i,T}$ denote the sequence of step-level labels produced along the corresponding reasoning trace. Let $D : \mathcal{L}_1 \times \cdots \times \mathcal{L}_T \to \mathcal{R}$ be a fixed, externally specified decision function mapping step-level labels to a risk value. In our setting, this is the set of macros used in the RoB2 tool (Sterne et al., 2019) (See Figure 1). The coherence indicator for datapoint i is then

$$C_i := \mathbb{1}\Big\{ \hat{r}_i = D(\hat{\ell}_{i,1}, \ldots, \hat{\ell}_{i,T}) \Big\}$$

and the dataset-level Coherence is given by

$$\text{Coherence} := \frac{1}{N}\sum_{i=1}^{N} C_i$$

By construction, Coherence measures the degree to which the model's final conclusions are internally consistent with the reasoning signals expressed in its own intermediate steps.

Training Setup. We conduct our training using a compact instruct models of recent release: `Qwen2.5-7B` (Yang et al., 2025b). We study two methodological regimes: supervised fine-tuning (SFT) with reasoning traces augmentation and reinforcement learning (RL) with verifiable rewards. SFT is conducted for 5 epochs with a per-device batch size of 1, a learning rate of 5×10^{-5}, and the AdamW optimiser (Loshchilov and Hutter, 2017). For RL, we investigate two policy-optimisation algorithms, GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025), combined with two reward types: verifiable outcome reward and our verifiable process reward approach. All RL configurations are trained for 3 epochs with a learning rate of 1×10^{-6}, per-device batch size 1, and 16 sampled generations per batch. Further implementation details are provided in Appendix E.

Model Baselines. To validate our results, we compare a range of open- and closed-source models, with and without reasoning capabilities. models are evaluated in zero-shot settings with prompt and hyperparameters shown in Appendix C. We include three main model families: Qwen 2.5 (Yang et al., 2025b), Llama 3.1 (Grattafiori et al., 2024), and Granite 3.1 (Granite Team, 2024). In addition, we benchmark the distilled Qwen and Llama models derived from DeepSeek-R1 (Guo et al., 2025). Lastly, we include one closed-source and two open-source models from OpenAI (OpenAI et al., 2025).

To contextualise the effectiveness of our verifiable reward formulation, we also evaluate neural process-reward baselines. At present, no pretrained PRM exists for risk-of-bias assessment or, more broadly, for non-mathematical scientific reasoning. Therefore, to approximate a general-purpose PRM, we follow prior work on using LLMs as step-level judges, prompting a model to assign correctness scores to each reasoning step (Song et al., 2025). This setup has been shown to deliver competitive process-level feedback in domains where explicit PRM training data is unavailable, and therefore serves as a reasonable baseline for comparison. In addition, we train a policy using MedPRM (Yun et al., 2025) as a reward model, one of the first open-source PRMs for general medical reasoning.

4.4 Main Results

Comparison with Pretrained Baselines. Table 2 presents a performance comparison of pretrained and fine-tuned language models on the CochraneForest and RoBBR benchmarks. Across all datasets, models trained with verifiable rewards substantially outperform pretrained models, including large reasoning-enabled systems. Reinforcement learning with verifiable outcome rewards already yields strong gains over supervised fine-tuning, while incorporating verifiable process rewards consistently leads to further improvements in both accuracy and macro-F1. On CochraneForest, `Qwen2.5-7B` trained with VPRM achieves the best overall performance, and similar improvements are observed on both RoBBR Cochrane and Non-Cochrane. The latter result indicates that the benefits of verifiable process supervision generalise beyond the training distribution, rather than exploiting dataset-specific regularities.

Comparison with Neural PRMs. Table 3 compares verifiable process rewards against neural process-reward baselines. In all settings, models also receive the same verifiable outcome reward; the comparison isolates only the effect of the process-level supervision. While neural PRMs substantially improve over outcome-only training, they are consistently outperformed by VPRM. This performance gap suggests that learned step-level judges introduce noise and misalignment that limit their effectiveness, whereas deterministic, guideline-based verification provides a cleaner and more reliable optimisation signal. These results support the central claim that verifiable process rewards offer a stronger and more robust alternative to neural process supervision for complex, structured reasoning tasks.

4.5 Ablation Studies

To assess the contribution of different components of our verifiable reward formulation, we conduct two ablation studies: the structure of process supervision and the inclusion of an outcome-level reward. For process supervision, we compare a *steps-only* reward that verifies whether the model follows the correct sequence of reasoning steps, irrespective

Model	Think	CochraneForest		RoBBR Cochrane		RoBBR Non-Cochrane	
		Acc	F1	Acc	F1	Acc	F1
Pretrained LLMs							
GPT-4-0125	✗	52.4	41.6	56.0	47.9	47.8	42.3
GPT-OSS-20B	✓	61.4	43.9	56.4	50.3	46.3	42.8
GPT-OSS-120B	✓	67.1	49.8	59.5	51.0	48.8	44.2
Qwen2.5-7B	✗	32.9	31.6	35.8	34.1	36.4	34.5
Qwen2.5-14B	✗	39.0	35.1	37.0	35.5	35.4	32.5
Qwen2.5-72B	✗	51.3	42.1	56.1	51.0	47.5	43.6
Llama-3.1-8B	✗	36.4	30.6	34.5	32.1	36.4	32.5
Llama-3.1-70B	✗	38.8	30.2	49.5	40.0	42.5	38.9
Llama-3.1-405B	✗	68.4	45.5	59.4	44.0	52.5	39.8
DeepSeek-Qwen-7B	✓	–	–	–	–	–	–
DeepSeek-Qwen-14B	✓	33.3	19.2	35.8	23.5	35.4	23.5
DeepSeek-Qwen-32B	✓	40.8	35.9	44.9	40.4	46.4	41.3
DeepSeek-Llama-8B	✓	–	–	–	–	–	–
DeepSeek-Llama-70B	✓	44.2	33.5	57.3	41.2	48.3	42.7
Granite-3.1-3B	✗	24.4	23.6	22.2	21.8	13.7	14.9
Granite-3.1-8B	✗	24.7	22.0	35.8	31.6	33.2	28.2
Granite-4.0-h-small (32B)	✗	48.2	33.5	45.4	41.2	40.9	33.1
Our Models							
Qwen2.5-7B-SFT	✓	45.1	36.9	38.6	32.4	38.3	31.9
Qwen2.5-7B-GRPO	✓	<u>81.5</u>	<u>70.2</u>	<u>63.1</u>	<u>58.0</u>	<u>56.8</u>	<u>45.1</u>
Qwen2.5-7B-DAPO	✓	76.8	57.3	60.2	45.4	55.8	43.6
Qwen2.5-7B-GRPO-VPRM	✓	**87.9**	**76.7**	**65.2**	**58.5**	**60.7**	**47.2**
Qwen2.5-7B-DAPO-VPRM	✓	79.2	60.6	60.7	48.9	57.1	45.3

Table 2: Evaluation results across models on three datasets, reporting Accuracy and macro-F1. "–" denotes unparsable or inconclusive outputs. Best results are bolded; second-best are underlined.

Method	Acc	F1
Neural PRMs		
Qwen2.5-7B-GRPO-PRM-GPT-OSS	78.2	56.1
Qwen2.5-7B-GRPO-MedPRM	76.8	53.4
Verifiable Rewards		
Qwen2.5-7B	32.9	31.6
Qwen2.5-7B-GRPO	81.5	70.2
Qwen2.5-7B-GRPO-VPRM	**87.9**	**76.7**

Table 3: Performance comparison between neural judges, rule-based rewarding and verifiable process rewarding on CochraneForest.

Setting	Acc	F1
w/o Outcome Reward		
Steps-only process reward	34.4	32.3
Full VPRM	40.2	35.3
w/ Outcome Reward		
Steps-only process reward	83.1	71.8
Full VPRM	87.9	76.7

Table 4: Ablation study on outcome and process reward components on CochraneForest. We report Accuracy (Acc) and F1; lower blocks compare models with and without outcome reward, and columns compare steps-only supervision to the full VPRM formulation.

of the correctness of their content, against the full VPRM, which additionally evaluates the correctness of each step and enforces consistency with the guideline-defined decision structure. For outcome supervision, we train each variant both with and without a verifiable outcome reward.

Table 4 shows that removing the outcome reward leads to substantial performance degradation, indicating that step-structure verification alone is insufficient to reliably optimize the task. Nevertheless, even in this setting, the full VPRM outperforms the steps-only variant, demonstrating the importance of verifying not just the presence but also the correctness and logical composition of intermediate reasoning steps. When the outcome reward is included, performance improves markedly, and the full VPRM consistently achieves the best results, showing that combining verifiable outcome supervision with fine-grained, correctness-aware process rewards yields the strongest learning signal.

Model	Coherence	CA
GPT-OSS-120B	36.2	28.5
Qwen2.5-72B	44.3	24.9
Llama-3.1-405B	50.7	27.1
Qwen2.5-7B-GRPO-VPRM	**89.5**	**75.0**
Qwen2.5-7B-DAPO-VPRM	<u>80.1</u>	<u>69.4</u>

Table 5: Coherence scores for Qwen models trained with DAPO and GRPO using verifiable process rewards compared with pretrained LLMs on COCHRANEFOREST. Best results are bolded; second-best underlined.

4.6 Analyses

Impact of Thought Process. Table 5 reports Coherence on the COCHRANEFOREST testset for the models trained with VPRMs compared with pretrained baselines prompted to output process labels. In addition, we report *Coherent Accuracy (CA)*, defined as the accuracy restricted to coherent instances. That is, among the datapoints for which the model's final prediction is consistent with the decision implied by its own reasoning steps (i.e., $C_i = 1$), we measure the proportion whose final predicted label is also correct. Formally, letting $\hat{y}_i$ and y_i denote the predicted and gold labels respectively, we define

$$\text{CA} = \frac{\sum_{i=1}^{N} \mathbb{1}[C_i = 1 \ \wedge \ \hat{y}_i = y_i]}{N}$$

CA therefore quantifies the reliability of the model's predictions conditioned on coherence.

Results show that pretrained models exhibit low coherence and very low CA, indicating that even when their step-level reasoning appears self-consistent, it rarely leads to correct final judgments. In contrast, VPRM-trained models achieve both substantially higher coherence and high CA, demonstrating that they not only follow the decision logic faithfully but also produce accurate conclusions when they do so, suggesting improved robustness and interpretability.

Reward Dynamics. Figure 3 shows that process and correctness rewards follow closely aligned trajectories: both rise sharply early, stabilise within the same oscillatory range, and peak at similar points. This alignment indicates that improved step-level reasoning directly improves final-label correctness. In contrast, the thought-format reward saturates quickly and remains flat, contributing little once formatting is learned. Overall, the strongly correlated shapes of the process and correctness curves highlight that VPRM training drives coherent, mutually reinforcing gains in intermediate and final reasoning behaviour.

Dataset	Baseline	Outcome-only	VPRM	Δ
CRiskEval	47.4	48.2	**53.6**	6.2
Gretel Financial Risk	49.1	49.8	**52.4**	3.3

Table 6: Out-of-Distribution Evaluation.

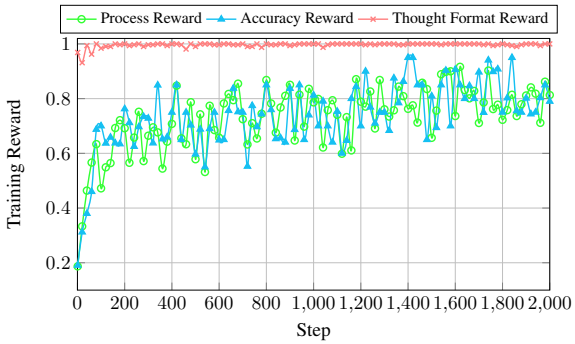

Figure 3: Reward dynamics. Format rewards plateau early, while accuracy and process rewards improve gradually, indicating that LLMs quickly learn structure but continue to refine quality.

Out-of-Distribution Evaluation. We further assess out-of-distribution generalisation of our models on CRiskEval (Shi and Xiong, 2025), a benchmark for AI safety risks, and on the Gretel financial risk dataset (GretelAI, 2024). Results (Table 6) show that our VPRM-trained models generalise well on out-of-distribution data, outperforming both the base instruct model and its outcome-only RL-tuned counterpart.

5 Related Work

Reinforcement Learning and Verifiable Rewards Several extensions of reinforcement learning with verifiable rewards (RLVR) go beyond outcome-only supervision by enriching the reward signal with structural information. Masked-and-reordered self-supervision provides auxiliary signals encouraging coherent intermediate reasoning (Wang et al., 2025c), while self-verification methods add progress-estimation or critique modules that guide models toward more reliable reasoning trajectories (Zeng et al., 2025). Theoretical analyses further show that verifiable rewards influence trajectory selection in predictable ways, steering models toward high-success modes under verifiable criteria (Wen et al., 2026). These approaches strengthen RLVR but remain fundamentally cen-

tered on terminal-outcome verification.

Complementary lines of research pursue process supervision, scoring CoT steps using neural judges (Lightman et al., 2024; Zelikman et al., 2022; Zhang et al., 2025; Zou et al., 2025). While such methods provide dense feedback unavailable to outcome-only RL, they depend on model-generated evaluations and therefore inherit issues of opacity, bias, and reward hacking (Amodei et al., 2016; Skalse et al., 2022). Crucially, their intermediate rewards are not verifiable.

Taken together, these works highlight two remaining gaps: existing approaches lack (i) *verifiability* of intermediate rewards and (ii) *fine-grained* step-level supervision grounded in deterministic rules. To date, no method provides reinforcement learning over reasoning trajectories where every step is evaluated by an externally checkable verifier. Verifiable Process Reward Models (VPRMs) address this gap by combining the robustness of RLVR with step-wise, rule-based verification, which not only enables transparent, structurally aligned reasoning but also removes the opportunities for reward hacking inherent in neural process rewards.

RoB Assessment and Automated Evidence Evaluation. Prior work on automated RoB assessment has largely relied on supervised modelling or prompted LLMs. Transformer-based systems such as RoBIn (Dias et al., 2025) frame RoB inference as a machine reading comprehension task and train classifiers directly on annotated evidence. Other approaches enhance pretrained LLMs with retrieval or auxiliary decision heads, as in RoBGuard (Ji et al., 2025). Several studies investigate LLM prompting for RoB assessment, reporting limited reliability when models operate without explicit procedural constraints (Huang et al., 2025; Šuster et al., 2024). Likewise, analyses of LLM-based critical appraisal highlight dependence on model pretraining and prompt sensitivity rather than verifiable optimisation (Wang et al., 2025a; Lai et al., 2025). Across these methodologies, existing systems employ prompting or supervised fine-tuning, but none leverage reinforcement learning for RoB assessment. Our work is, to our knowledge, the first to introduce RL-based training in this domain.

6 Conclusion

In this paper, we introduce verifiable process rewards that integrate deterministic step-level verification with reinforcement learning, provide theoretical guarantees under mild assumptions, and demonstrate substantial empirical gains on risk-of-bias assessment in medical systematic reviews.

Our results indicate that verifiable process supervision is a practical and robust approach to inducing reliable reasoning behaviour in large language models, opening the door to broader applications in structured scientific and decision-making tasks.

Limitations

While VPRMs offer strong guarantees for structured reasoning tasks, several limitations remain. First, the approach relies on the existence of deterministic, domain-specific rules; tasks lacking well-defined intermediate reasoning steps may not benefit directly. Second, our empirical evaluation is currently focused on risk-of-bias assessment; generalisation to other domains, particularly open-ended reasoning tasks, remains to be established.

Additionally, the approach assumes that the model can produce reasoning traces in a format compatible with the verifiers; misalignment between model output and verifier expectations could reduce reward effectiveness, especially in the context of smaller models. Finally, while VPRMs reduce reliance on neural reward models, they do not fully eliminate other sources of model bias or errors arising from incomplete guidelines. Addressing these challenges will be critical for deploying verifiable process supervision in broader, real-world applications.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Jacqueline Chandler, Miranda Cumpston, Tianjing Li, Matthew J. Page, and Vivian A. Welch. 2019. Cochrane handbook for systematic reviews of interventions. *Hoboken: Wiley*, 4.

Abel Corrêa Dias, Viviane Pereira Moreira, and João Luiz Dihl Comba. 2025. RoBIn: A transformer-based model for risk of bias inference with machine reading comprehension. *Journal of Biomedical Informatics*, 166:104819.

IBM Granite Team. 2024. Granite 3.0 language models. *URL: https://github.com/ibm-granite/granite-3.0-language-models*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

GretelAI. 2024. Synthetic financial risk analysis dataset. https://huggingface.co/gretelai/gretel-financial-risk-analysis-v1.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 180 others. 2025. Deepseek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jiajie Huang, Honghao Lai, Weilong Zhao, Danni Xia, Chunyang Bai, Mingyao Sun, Jianing Liu, Jiayi Liu, Bei Pan, Jinhui Tian, and Long Ge. 2025. Large language model–assisted risk-of-bias assessment in randomized controlled trials using the revised risk-of-bias tool: Usability study. *Journal of Medical Internet Research*, 27:e70450.

Hugging Face. 2025. Open R1: A fully open reproduction of DeepSeek-R1.

Changkai Ji, Bowen Zhao, Zhuoyao Wang, Yingwen Wang, Yuejie Zhang, Ying Cheng, Rui Feng, and Xiaobo Zhang. 2025. RoBGuard: Enhancing LLMs to assess risk of bias in clinical trial documents. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1258–1277.

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

Honghao Lai, Jiayi Liu, Chunyang Bai, Hui Liu, Bei Pan, Xufei Luo, Liangying Hou, Weilong Zhao, Danni Xia, Jinhui Tian, Yaolong Chen, Lu Zhang, Janne Estill, Jie Liu, Xing Liao, Nannan Shi, Xin Sun, Hongcai Shang, Zhaoxiang Bian, and 17 others. 2025. Language models for data extraction and risk of bias assessment in complementary medicine. *NPJ Digital Medicine*, 8.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, and 4 others. 2025. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*.

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 *The Twelfth International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

OpenAI. 2024. Introducing openai o1. https://openai.com/it-IT/o1/.

OpenAI. 2025. Introducing gpt-5. https://openai.com/it-IT/gpt-5/.

OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, and 107 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Massimiliano Pronesti, Joao H Bettencourt-Silva, Paul Flanagan, Alessandra Pascale, Oisín Redmond, Anya Belz, and Yufang Hou. 2025a. Query-driven document-level scientific evidence extraction from biomedical studies. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28034–28051, Vienna, Austria. Association for Computational Linguistics.

Massimiliano Pronesti, Michela Lorandi, Paul Flanagan, Oisín Redmond, Anya Belz, and Yufang Hou. 2025b. Enhancing study-level inference from clinical trial papers via reinforcement learning-based numeric reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 30345–30361, Suzhou, China. Association for Computational Linguistics.

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

Ling Shi and Deyi Xiong. 2025. CRiskEval: A Chinese multi-level risk evaluation benchmark dataset for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

13638–13659, Vienna, Austria. Association for Computational Linguistics.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems*, volume 35, pages 9460–9471.

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, Vienna, Austria. Association for Computational Linguistics.

Jonathan AC Sterne, Jelena Savović, Matthew J. Page, Roy G. Elbers, Natalie S. Blencowe, Isabelle Boutron, Christopher J. Cates, He Cheng, Mark S. Corbett, Sandra M. Eldridge, Miguel A. Hernán, Sally Hopewell, Asbjørn Hróbjartsson, Diana R. Junqueira, Peter Jüni, Jamie J. Kirkham, Toby Lasserson, Tianjing Li, Ann McAleenan, and 8 others. 2019. RoB 2: a revised tool for assessing risk of bias in randomised trials. *BMJ*, 366:l4898.

Simon Šuster, Timothy Baldwin, and Karin Verspoor. 2024. Zero-and few-shot prompting of generative large language models provides weak assessment of risk of bias in clinical trials. *Research Synthesis Methods*, 15(6):988–1000.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press.

Jianyou Wang, Weili Cao, Longtian Bao, Youze Zheng, Gil Pasternak, Kaicheng Wang, Xiaoyue Wang, Ramamohan Paturi, and Leon Bergen. 2025a. Measuring risk of bias in biomedical reports: The RoBBR benchmark. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3220–3248, Suzhou, China. Association for Computational Linguistics.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and yelong shen. 2025b. Reinforcement learning for reasoning in large language models with one training example. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zhen Wang, Zhifeng Gao, and Guolin Ke. 2025c. Masked-and-reordered self-supervision for reinforcement learning from verifiable rewards. *arXiv preprint arXiv:2511.17473*.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. 2026. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. In *The Fourteenth International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025b. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Jaehoon Yun, Jiwoong Sohn, Jungwoo Park, Hyunjae Kim, Xiangru Tang, Daniel Shao, Yong Hoe Koo, Ko Minhyeok, Qingyu Chen, Mark Gerstein, Michael Moor, and Jaewoo Kang. 2025. Med-PRM: Medical reasoning models with stepwise, guideline-verified process rewards. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16565–16582, Suzhou, China. Association for Computational Linguistics.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun MA, and Junxian He. 2025. SimpleRL-Zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *Second Conference on Language Modeling*.

Jixiao Zhang and Chunsheng Zuo. 2025. GRPO-LEAD: A difficulty-aware reinforcement learning approach for concise mathematical reasoning in language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5642–5665, Suzhou, China. Association for Computational Linguistics.

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,

Vienna, Austria. Association for Computational Linguistics.

Jiaru Zou, Ling Yang, Jingwen Gu, Jiahao Qiu, Ke Shen, Jingrui He, and Mengdi Wang. 2025. ReasonFlux-PRM: Trajectory-aware PRMs for long chain-of-thought reasoning in LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

A Theoretical Analysis

This section provides the proof of Theorem 1, which is an extension of prior results on reinforcement learning with verifiable rewards (RLVR) in base LLMs (Wen et al., 2026).

A.1 Setup and assumptions

Let $R(Y)$ denote the verifiable process reward assigned to a response Y, and let G be the number of responses sampled for the input x. Let $\mathcal{C}$ denote the correctness event for a trajectory Y. Define

$$\mu_c := \mathbb{E}[R(Y) \mid \mathcal{C}], \qquad \mu_i := \mathbb{E}[R(Y) \mid \mathcal{C}^c], \\ p := \mathbb{P}(\mathcal{C}),$$

and let $m := p\mu_c + (1-p)\mu_i$ be the unconditional expected reward.

Assumption A1. For fixed (x, r), the verifiable reward $R(Y)$ is a real-valued random variable with finite mean and nonnegative variance $\sigma_Y > 0$.

Assumption A2. Correct reasoning trajectories have higher probabilities to induce correct answers (the reward model is more likely to assign higher scores to correct responses than to incorrect ones): $\mu_c > \mu_i$.

Assumption A3. Let $\overline{R}$ and S denote the empirical mean and standard deviation of rewards within a sampled group. As the group size $G \to \infty$,

$$\overline{R} \xrightarrow{p} m, \qquad S \xrightarrow{p} \sigma := \sqrt{\mathrm{Var}[R(Y)]}.$$

A.2 Normalised advantages

GRPO uses the trajectory-level normalised advantage

$$\hat{A}(Y) = \frac{R(Y) - \overline{R}}{S}.$$

DAPO constructs token-level advantages by scaling the same trajectory advantage:

$$\hat{A}_{i,t} = c_{i,t}\, \hat{A}(Y_i),$$

where $c_{i,t} \geq 0$ and $\frac{1}{\sum_i |Y_i|} \sum_{i,t} c_{i,t} = 1$.

Following Wen et al., 2026 and without loss of generality, we consider a policy gradient update (Sutton et al., 1999)

$$\nabla J(\theta) \approx \frac{1}{G} \sum_{i=1}^{G} \hat{A}(Y_i) \nabla_\theta \, log \, \pi_\theta(Y_i \mid x).$$

A.3 Proof of Theorem 1

We prove the result for $\hat{A}(Y)$; the DAPO case follows immediately by the nonnegativity of the constants $c_{i,t}$.

By Assumption A3 and Slutsky's theorem,

$$\hat{A}(Y) = \frac{R(Y) - \overline{R}}{S} \xrightarrow{d} \frac{R(Y) - m}{\sigma}.$$

Taking conditional expectations yields

$$\mathbb{E}[\hat{A}(Y) \mid \mathcal{C}] \xrightarrow{G\to\infty} \frac{\mu_c - m}{\sigma}, \\ \mathbb{E}[\hat{A}(Y) \mid \mathcal{C}^c] \xrightarrow{G\to\infty} \frac{\mu_i - m}{\sigma}.$$

Substituting $m = p\mu_c + (1-p)\mu_i$ gives

$$\mu_c - m = (1-p)(\mu_c - \mu_i) > 0, \\ \mu_i - m = -p(\mu_c - \mu_i) < 0,$$

establishing the sign separation:

$$\mathbb{E}[\hat{A}(Y) \mid \mathcal{C}] > 0, \\ \mathbb{E}[\hat{A}(Y) \mid \mathcal{C}^c] < 0.$$

For DAPO,

$$\mathbb{E}[\hat{A}_{i,t} \mid \mathcal{C}] = c_{i,t}\, \mathbb{E}[\hat{A}(Y) \mid \mathcal{C}] > 0, \\ \mathbb{E}[\hat{A}_{i,t} \mid \mathcal{C}^c] = c_{i,t}\, \mathbb{E}[\hat{A}(Y) \mid \mathcal{C}^c] < 0.$$

Thus, both GRPO and DAPO apply positive expected weight to correct traces and negative weight to incorrect ones, proving Theorem 1.

A.4 Verifiable Outcome Reward as a Special Case of VPRMs

Define the degenerate label spaces $\mathcal{L}_t^{(r)} = \varnothing$ for all $t < T$. Then $r_t(\cdot) = 0$ for $t < T$ and

$$R(Y; x, r) = r_{\text{label}}(Y; x, r).$$

Hence a Verifiable Outcome Reward Model is exactly the case of a VPRM with no intermediate verifiable labels. All the results above apply: they reduce to the original GRPO/DAPO statements where the scalar reward depends only on final outcome statistics.

B Risk of Bias Assessment

Risk-of-bias estimation evaluates the extent to which study findings may be systematically distorted. The process is organised into a set of domains that correspond to common sources of bias in randomized trials. For each domain, reviewers extract relevant information from the study report and translate it into qualitative judgments about the presence and potential impact of bias. Modern assessment tools, such as RoB 2.0 (Sterne et al., 2019), increasingly leverage automated decision rules to standardise these judgments. Algorithm 1 provides an example of a macro for risk of type A using the labels defined in this paper, which illustrates how extracted steps are mapped to specific risk levels. Below we outline the main domains considered in our work, together with the typical reasoning steps involved.

A. Random sequence generation This domain assesses whether the method used to generate the allocation sequence was truly random. Reviewers first check whether the study reports how randomization was carried out. If so, they evaluate the nature of the method (e.g., computer-generated sequence versus quasi-random methods such as alternation) and judge whether the sequence could have been predicted. Clearly reported and genuinely random procedures indicate low risk; quasi-random or non-random procedures, or a lack of information, increase concern.

B. Allocation concealment Here the question is whether the assignment to treatment groups was shielded from those enrolling participants. Reviewers determine whether concealment was reported and whether the method (e.g., sealed opaque envelopes, central allocation) prevented foreknowledge of upcoming assignments. Adequate concealment protects against selection bias, whereas inadequate or unclear procedures raise concerns.

C. Blinding of participants and personnel This domain considers whether participants and those administering interventions were aware of group assignments. Reviewers establish whether blinding was reported, whether it involved participants, personnel, or both, and whether the blinding approach was likely to have been effective. Lack of blinding, or ineffective procedures, may influence participants' behaviour or care delivery and thus introduce performance bias.

D. Blinding of outcome assessment Assessors may also be influenced by knowledge of treatment allocation. Reviewers check whether outcome assessors were blinded and whether blinding was likely to minimise biased measurement. Absence of blinding or unclear reporting raises the possibility that assessments were influenced by expectations or prior beliefs.

E. Incomplete outcome data This domain evaluates the extent and handling of missing data. Reviewers consider how much data is missing, whether reasons for missingness are reported and plausible, and whether the analysis appropriately accounts for missing data. High or unexplained attrition, or inadequate handling strategies, can produce biased estimates of effect.

F. Selective reporting Selective reporting bias arises when outcomes are reported inconsistently with the study protocol or when unplanned outcomes are introduced. Reviewers check whether a protocol is available, compare planned and reported outcomes, and assess whether omissions or additions suggest selective emphasis. Clear correspondence indicates low risk; discrepancies raise concern.

In addition to these core domains, we also consider supplementary aspects relevant to internal validity: similarity of baseline outcomes (G), similarity of baseline characteristics (H), and risk of contamination between study arms (I). These domains capture further sources of potential bias arising from imbalances at baseline or from unintended exposure to interventions across groups.

Algorithm 1 RoB A Macro

```
procedure PredictLabel-A(steps)
    if steps[IdentifyRandomizationReport] = NotReported then
        return Moderate
    end if
    if steps[ClassifyRandomizationMethod] = NonRandom then
        return High
    end if
    if steps[AssessSequencePredictability] = Predictable then
        return Moderate
    end if
    if steps[BaselineImbalance] = Likely then
        return High
    end if
    return Low
end procedure
```

C Prompts

The prompts used for synthetic data annotation and for training are shown in Figure 4 and 6, respectively. For training, a temperature of 0.7 and 2,048 tokens as maximum output length are used.

Prompt for synthetic data annotation

Articles: {articles}

Your task is to produce a structured reasoning trace for the following risk of bias domain to justify the ground truth value.

Comparison: {comparison}
Outcome: {outcome}
Bias: {bias_id} – {bias_definition}
Ground_truth: {bias_value}

You must follow the structured reasoning procedure defined for each risk-of-bias domain (A–I). For every domain, you must use the exact step names and allowable categorical labels listed below:

{steps_and_labels}

Follow this exact output structure:
{{
"step_name": "step_label",
"step_name_rationale": "your detailed rationale",
...
(repeat for all steps required by the bias domain)
}}

Figure 4: Prompt for synthetic data annotation.

Steps and labels

A — Random sequence generation
Identify_randomization_report → reported | not_reported
Classify_randomization_method → random | non_random
Assess_sequence_predictability → unpredictable | predictable
Baseline_imbalance → likely | none

B — Allocation concealment
Identify_concealment_report → reported | not_reported
Determine_concealment_method → adequate | inadequate
Assess_possibility_of_foreknowledge → no | possible

C — Blinding of participants and personnel
Identify_blinding_report → reported | not_reported
Assess_blinding_status → participants | personnel | both | none
Evaluate_blinding_effectiveness → effective | ineffective

D — Blinding of outcome assessment
Identify_outcome_blinding_report → reported | not_reported
Assess_assessor_blinding → yes | no
Evaluate_blinding_effect_on_measurement → no | possible

E — Incomplete outcome data
Quantify_missing_data → none | low | high
Identify_missing_data_reason → adequate | inadequate | not_reported
Assess_handling_of_missing_data → appropriate | inappropriate
Estimate_bias_due_to_missing_data → unlikely | likely

F — Selective reporting
Identify_protocol_availability → available | not_available
Compare_outcomes_reported → all | partial | none
Detect_unexpected_outcomes → none | added
Evaluate_reporting_selectivity → no | possible | yes

G — Baseline outcomes similar
Identify_baseline_outcomes_report → reported | not_reported
Compare_baseline_outcomes → similar | different
Evaluate_impact_of_differences → likely_impact | unlikely_impact

H — Baseline characteristics similar
Identify_baseline_characteristics_report → reported | not_reported
Compare_baseline_characteristics → similar | different
Evaluate_impact_of_differences → likely_impact | unlikely_impact

I — Contamination
Identify_contamination_risk_report → reported | not_reported
Assess_contamination_possibility → possible | unlikely
Assess_contamination_impact → likely_impact | unlikely_impact

Figure 5: Steps and labels.

Prompt for training and inference

Articles: {articles}

Question: Based on the given article, what is the risk of bias for the following Comparison and Outcome?
Comparison: {comparison}
Outcome: {outcome}
The bias you have to assess is defined as follows: {bias}
You must follow the structured reasoning procedure defined for each risk-of-bias domain (A–I). For every domain, you must use the exact step names and allowable categorical labels listed below:

{steps_and_labels}

Follow this exact structure for your reasoning:
<think>
Step 1: step_name
...your thought process here...
Answer: step_label

Step 2: step_name
...your thought process here...
Answer: step_label

(repeat for all steps required by the bias domain)
</think>
<answer>
risk: high | low | moderate
</answer>

Figure 6: Prompt for VPRM-training and -inference.

Dataset	A	B	C	D	E	F	G	H	I
CochraneForestExt	498	498	498	498	498	273	61	61	61
CochraneForest	330	330	330	330	330	112	28	28	28
RoBBR Cochrane	125	125	198	133	206	119	0	0	0
RoBBR Non-Cochrane	412	474	467	472	478	186	0	0	0

Table 7: Datasets statistics per risk type.

D Silver Steps and Labels Manual Verification

To assess the quality of the automatically generated reasoning steps and silver labels used for VPRM training, we selected a random sample of 20 instances from the full dataset and manually evaluated their correctness. The evaluation was conducted by two master's students in NLP familiar with the task. For each instance, annotators inspected the complete step-level reasoning trace and verified two properties: (i) whether the sequence of steps constituted a valid decision path for the target risk-of-bias domain, according to Cochrane's domain-specific guidelines; and (ii) whether each step and label was valid given the underlying paper. For each item, we recorded whether the overall reasoning trace was coherent, and for each individual step we recorded whether the step and its label were valid.

Table 8 summarises the proportion of coherent instances, correct steps, and correct labels observed in this manual evaluation. Results of the manual verification show that the automatically generated

Metric	Fraction
Coherent instances	100.0 %
Correct steps	100.0 %
Correct labels	96.7%

Table 8: Manual verification of 20 randomly sampled silver-labelled reasoning traces.

steps and labels used for VPRM training are of consistently high quality. All inspected traces follow the correct decision structure, and step-level labels are almost always accurate, providing a solid ground for model training.

E Hyperparameters and APIs

We executed all the experiments either via API or on our own cluster. We used the paid-for OpenAI API to access GPT-4. On the other hand, we hosted and trained the open-source models used in this paper on a distributed cluster.

SFT is performed for 5 epochs with a batch size of 1 (due to the large size of the input data) using a learning rate of 5×10^{-5} and the AdamW optimiser (Loshchilov and Hutter, 2017).

For the RL setups, we adopt the GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025) algorithms, training for 3 epochs with a learning rate of 1×10^{-6}, per-device batch size 1, and 16 sampled generations per batch. Both training protocols leverage gradient accumulation with 8 accumulation steps. All experiments are conducted using the Open-R1 framework (Hugging Face, 2025) on 8 NVIDIA A100 GPUs, each equipped with 80GB of memory. Models have been served for inference with the vLLM framework (Kwon et al., 2023).

F Scientific Artefacts and Licensing

In this work, we used the following scientific artefacts. LLaMa 3.1 is licensed under a commercial license[1]. GPT-4 is licensed under a commercial license[2]. Qwen2.5 is licensed under the Apache 2.0 license[3]. Granite 3.1 is licensed under the Apache 2.0 license[4]. DeepSeek models are licensed under the MIT license[5]. Mining text and data from the Cochrane library is permitted for non-commercial research through the Wiley API.[6]. The usage of the listed artefacts is consistent with their licenses.

[1] https://llama.meta.com/doc/overview

[2] https://openai.com/policies/terms-of-use

[3] https://qwenlm.github.io/blog/qwen3

[4] https://www.ibm.com/architectures/product-guides/granite-31

[5] https://api-docs.deepseek.com/news/news250120

[6] https://www.cochranelibrary.com/help/access

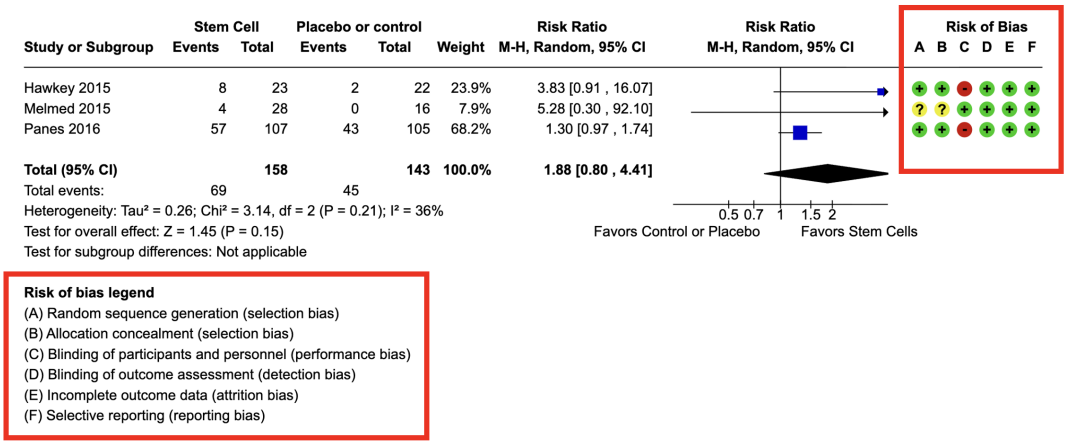

Figure 7: Example of forest plot containing three studies and their corresponding risk of bias map.

Hawkey 2015	
Text	Autologous Hematopoetic Stem Cell Transplantation for Refractory Crohn Disease: A Randomised Clinical Trial **Importance** Case reports and series suggest hematopoietic stem cell transplantation (HSCT) may benefit some patients with Crohn disease. ...
Outcome **Comparison** **Bias type** **Bias name** **Bias definition** **Bias value**	Clinical Remission Stem Cell vs Placebo or control A Random sequence generation (selection bias) Bias arising from the randomization process **+**
Annotation	"Identify_randomization_report": "reported", "Identify_randomization_report_rationale": "The methods section of the trial explicitly states that participants were allocated to groups using a computer-generated random sequence. The authors describe the randomisation process in sufficient detail, indicating that a statistician who was not involved in participant recruitment generated the sequence.", "Classify_randomization_method": "random", "Classify_randomization_method_rationale": "A computer-generated sequence (e.g., using a random number generator or statistical software) is a truly random method. It does not rely on any deterministic or quasi-random technique such as alternation, date of birth, or admission order, therefore it is classified as 'random'.", "Assess_sequence_predictability": "unpredictable", "Assess_sequence_predictability_rationale": "Because the sequence was generated by a computer algorithm and concealed from the investigators enrolling participants (allocation was performed via sealed, opaque envelopes prepared after the sequence was generated), neither the investigators nor the participants could foresee the upcoming allocation. This makes the sequence unpredictable.", "Baseline_Imbalance": "none", "Baseline_Imbalance_rationale": "Baseline tables show that demographic and clinical characteristics (age, sex, disease severity, etc.) are comparable across groups with no statistically significant differences, indicating that the randomisation produced balanced groups and is unlikely to have introduced bias."

Table 9: The corresponding entry in CochraneForestExt for Hawkey 2015 in Figure 7. Rationales are not used for RL training.