PACR: PROGRESSIVELY ASCENDING CONFIDENCE REWARD FOR LLM REASONING

Anonymous authors

Paper under double-blind review

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) has significantly improved LLM reasoning, but its sparse, outcome-based reward provides no guidance for intermediate steps, slowing exploration. We propose Progressively Ascending Confidence Reward (PACR), a dense, model-intrinsic reward computed directly from the model's evolving belief in the correct answer. PACR encodes the inductive bias that, along a well-formed reasoning trajectory, the probability of the ground-truth answer should have a generally ascending trend. We provide empirical and theoretical analysis validating that such an inductive bias constrains the exploration search space to regions richer in logically sound reasoning. We demonstrate that PACR accelerates exploration, reaches reward saturation with fewer trajectories, and yields improvements on multiple benchmarks. Our results suggest that dense, model-intrinsic shaping signals can make RLVR training more effective and reliable. Code will be released.

1 Introduction

Pre-trained large language models (LLMs) exhibit strong performance on complex, multistep reasoning tasks (Comanici et al., 2025; Yang et al., 2025a; Team, 2025). Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as a leading approach for further improving such capabilities, using a programmatically checkable terminal metric (e.g., exactmatch on the final answer) as the reward (Shao et al., 2024b; Guo et al., 2025). While effective, the standard RLVR formulation supplies a sparse terminal accuracy signal, offering no guidance for intermediate steps and thus exacerbating credit assignment. Alternative process-based supervision employs external reward mod-

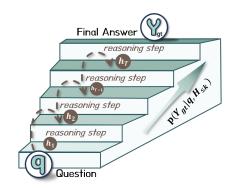


Figure 1: **Stepwise confidence growth.** For a question q, a well-formed sequence of reasoning steps h_1, \ldots, h_k should increase the model's probability of the ground-truth answer $Y_{\rm gt}$ across steps.

els to score intermediate reasoning, but is costly to train, data-hungry, and prone to misalignment (Cui et al., 2025; Cheng et al., 2025).

This work asks whether we can obtain *stepwise supervision* directly from the model. Psycholinguistic work shows that people interpret language incrementally, updating expectations with each word; as context accumulates, uncertainty falls and the correct interpretation becomes more likely (Hale, 2001; Levy, 2008). By the same logic, in tasks with a verifiable final answer, a correct intermediate step should typically raise the model's probability of the ground-truth answer. Concretely, given a question q, a reasoning prefix $H_{\leq k}$, and ground truth $Y_{\rm gt}$, we track the model's confidence $p(Y_{\rm gt} \mid q, H_{\leq k})$ and expect a general upward trend over steps (Figure 1).

Guided by this premise, we introduce the **Progressively Ascending Confidence Reward (PACR)**, a dense, model-intrinsic signal that converts confidence growth into stepwise supervision for LLM reasoning during reinforcement learning. During training, as the model produces a sequence of reasoning steps for a question with a verifiable answer, we evaluate at each step the log-probability assigned to the ground-truth answer and reward any positive change, effectively encouraging a consis-

tently upward trend in confidence. Because PACR is computed from the model's own probabilities, it requires no external reward model and is available at every step, improving credit assignment and steering search toward faithful trajectories. We pair PACR with the standard RLVR terminal accuracy reward so the objective remains anchored to verifiable correctness while the process signal shapes the reasoning path. In detail, our contributions can be summarized as follows:

- Empirical Validation of an Inductive Bias (Section 4.1). We provide extensive observational evidence that ground-truth confidence growth acts as a powerful inductive bias. Our analyses on open-source LLMs reveal three key findings: (1) a *consistent* confidence ascent strongly correlates with final answer correctness; (2) among correct answers, logically coherent reasoning paths exhibit an even *more consistent* ascent than spurious ones; and (3) the *magnitude* of the confidence gain effectively pinpoints pivotal reasoning steps.
- Theoretical Justification (Section 4.2 and 5). We provide a theoretical foundation for using confidence growth as a process reward. We prove that a reasoning step from an idealized oracle policy will, on average, increase or maintain the model's confidence in the ground truth, validating it as a strong inductive bias. Building on this, we formalize the Progressively Ascending Confidence Reward (PACR) and introduce two concrete methods for its implementation: Sparse-PACR for trajectory-level rewards and Dense-PACR for step-wise rewards.
- Experimental Results (Section 7). Across multiple reasoning benchmarks, augmenting RLVR with our PACR methods improves training dynamics and final performance. Our approach accelerates exploration and ultimately attains a higher, more consistent final score than the baseline, demonstrating a more effective and reliable training process.

2 RELATED WORK

Outcome-based RL for LLM Reasoning Reinforcement Learning (RL) is increasingly used to fine-tune Large Language Models (LLMs). This is done not only to align models with human preferences via Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022; Li et al., 2024b; Bai et al., 2022) but also to enhance their reasoning abilities for complex problemsolving (Kumar et al.). To improve these reasoning capabilities, a recent prominent approach is Reinforcement Learning with Verifiable Reward (RLVR) (Guo et al., 2025; Yang et al., 2024; Shao et al., 2024a), which uses an outcome-based reward instead of a proxy reward model. For example, a reward of 1 is assigned for a correct answer and 0 (or -1) for an incorrect one. Then, the model generates multiple trajectories for a single problem. The reward for each trajectory is then compared against the average reward across all samples in the group, and this relative reward is used as an advantage to train the model. This outcome-based reward system is widely explored (Liu et al., 2025; Yu et al., 2025; Hu et al., 2025; Zeng et al., 2025) because it is easily scalable, and mitigates concerns about reward hacking by eliminating the need for a separate reward model (Guo et al., 2025). However, this approach has a significant limitation for complex reasoning tasks that require generating a long thought process (Zhang et al., 2025). In such cases, relying solely on the final outcome provides a sparse and noisy reward signal.

Dense Reward for LLMs Finetuning with RL To overcome the limitations of holistic, trajectory-level sparse rewards, various approaches for providing dense rewards have been explored. In RLHF, for instance, approaches include training an external reward model to assign token-level rewards using synthesized data (Yoon et al., 2024), utilizing a more mature external model as the reward model (Cao et al., 2024; Wu et al., 2023), and use implicit reward signal from reward model (Chan et al., 2024). Similarly, direct alignment algorithms (e.g., DPO (Rafailov et al., 2023)) have been adapted to provide dense rewards by re-framing DPO's implicit reward at a token level (Zeng et al.; Zhu et al.; Rafailov et al.) or by selectively using specific tokens for the reward signal (Yoon et al.; Liu et al.). For training a reasoning LLM with RL, previous approaches include training a Process Reward Model (PRM) for process-level rewards (Li & Li, 2025; Cheng et al., 2025; Zhang et al., 2025), or defining a DPO-like implicit reward at the token level (Cui et al., 2025; Yuan et al., 2024). However, these approaches typically require additional models, such as a reward model or a reference model, to generate the reward signal. In contrast, our work eliminates the need for any additional models. We instead use the current policy model itself to generate a dense reward signal that enhances reasoning capabilities.

3 BACKGROUND AND PROBLEM SETUP

This section introduces the notation for reasoning trajectories, how we segment and evaluate stepwise confidence in the ground-truth answer, and the RL objective we use in training.

Reasoning Trajectories and Notation. Given a question q, a policy π_{θ} generates a *sequence of reasoning steps* $H = (h_1, \dots, h_T)$ and a final answer \hat{Y} . Let $Y_{\rm gt}$ denote the verifiable ground-truth answer. We write $H_{\leq k} = (h_1, \dots, h_k)$ for the reasoning steps up to step k. We analyze and shape the reasoning process by tracking how the model's probability of $Y_{\rm gt}$ evolves with the prefix $H_{\leq k}$.

Segmenting the Reasoning Process and Stepwise Ground-truth Confidence. Similar to Yang et al. (2025b), we segment each generated reasoning trace into discrete steps $\{h_k\}_{k=1}^T$ using a simple, model-agnostic rule: start a new step at a newline ('\n') or at a period followed by a space ('. '); fragments shorter than five tokens are merged with the preceding step to avoid overly fine splits. To measure ground-truth-anchored confidence at step k, we standardize the answer format by appending a short prefix $y_{\rm gt}^0$ (e.g., 'So the final answer is \boxed{')} and evaluate the model's probability of the ground-truth answer $Y_{\rm gt} = (y_{\rm gt}^1, \ldots, y_{\rm gt}^L)$ under the current prefix $H_{\leq k}$. Writing $Y_{\rm gt} = (y_{\rm gt}^1, \ldots, y_{\rm gt}^L)$, we measure the **ground-truth confidence** at step k as

$$\log p(Y_{gt}|q, H_{\leq k}) = \sum_{l=1}^{L} \log p_{\theta}(y_{gt}^{l} | q, H_{\leq k}, y_{gt}^{0}, y_{gt}^{\leq l}), \tag{1}$$

where $y_{\text{gt}}^{< l}$ are preceding answer tokens. This measures the model's confidence in the ground truth answer at any given stage of its reasoning steps.

Group Relative Policy Optimization (GRPO) GRPO (Shao et al., 2024b) estimates advantages by comparing returns *within* a group of N samples rather than using a learned value function. For a given question q (with verifiable answer $Y_{\rm gt}$), the behavior policy $\pi_{\theta_{\rm old}}$ generates N trajectories

$$\{\tau^{(i)}\}_{i=1}^N, \qquad \tau^{(i)} = (H^{(i)}, \hat{Y}^{(i)}),$$
 (2)

where $H^{(i)}=(h_1^{(i)},\ldots,h_{T_i}^{(i)})$ are the reasoning steps, T_i is the number of steps for i-th trajectory and $\hat{Y}^{(i)}$ is the predicted answer for i-th trajectory.

For each sampled trajectory i, we compare the predicted answer $\hat{Y}^{(i)}$ with the ground truth Y_{gt} and assign a binary terminal accuracy reward:

$$R^{(i)} = \begin{cases} 1, & \text{is_equivalent}(\hat{Y}^{(i)}, Y_{\text{gt}}) \\ 0, & \text{otherwise.} \end{cases}$$
 (3)

Here is_equivalent performs task-specific normalization (e.g., stripping whitespace/punctuation, handling LaTeX boxing, case folding, and numeric tolerances) before exact match. The group-relative advantage for trajectory i is computed by centering (and optionally standardizing) its reward within the cohort of N samples:

$$A^{(i)} = \frac{R^{(i)} - \text{mean}(\{R^{(i)}\}_{i=1}^{N})}{\text{std}(\{R^{(i)}\}_{i=1}^{N})}.$$
 (4)

Similar to PPO (Schulman et al., 2017), GRPO adopts a clipping with KL penalty:

 $\mathcal{J}_{GRPO}(\theta) =$

$$\mathbb{E}_{\substack{(q,Y_{gt}) \sim \mathcal{D} \\ \{\tau^{(i)}\} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q)}} \left[\frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\tau^{(i)}|} \left(\min\left(\frac{\pi_{\theta}(\tau^{(i)} \mid q)}{\pi_{\theta_{\text{old}}}(\tau^{(i)} \mid q)}(\theta) A^{(i)}, \operatorname{clip}\left(\frac{\pi_{\theta}(\tau^{(i)} \mid q)}{\pi_{\theta_{\text{old}}}(\tau^{(i)} \mid q)}, 1 - \varepsilon, 1 + \varepsilon \right) A_{i} \right) - \beta D_{\text{KL}}(\pi_{\theta} || \pi_{\text{ref}}) \right) \right],$$
(6)

where \mathcal{D} is the training dataset and π_{ref} is a reference policy. In our work, we follow the Dr. GRPO (Liu et al., 2025) formulation, a bias-mitigated variant of GRPO. This approach modifies the standard GRPO algorithm by discarding the standard deviation from the advantage calculation and the length normalization from the loss function (the terms shown in green in Eq. 4 and Eq. 5).

163164

166 167

168

169

170

171

172 173

174

175 176

177 178

179

181

182

183

185

186

187

188

189

190

191

192

193

194

195

196

197

199 200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

4 IS GROUND-TRUTH CONFIDENCE GROWTH A USEFUL INDUCTIVE BIAS?

In this section, we present both empirical observations and a theoretical justification to assess whether ground-truth confidence growth constitutes a useful inductive bias for training LLM reasoning.

Ground-truth Confidence Growth. We first quantify confidence growth by defining the stepwise confidence gain, C_k , as the change in the log-probability of the ground-truth answer induced by the addition of reasoning step h_k :

$$C_k := \log \pi_{\theta}(Y_{\text{gt}} \mid q, H_{\leq k}) - \log \pi_{\theta}(Y_{\text{gt}} \mid q, H_{\leq k}), \tag{6}$$

where $H_{\leq k}=(h_1,\ldots,h_k)$ and $H_{< k}=(h_1,\ldots,h_{k-1})$. For $k=1,\,H_{< 1}$ is the empty prefix. Intuitively, $C_k>0$ indicates that step h_k makes the ground truth more probable, whereas $C_k<0$ indicates the opposite. (When indexing trajectories, we write $C_k^{(i)}$.) For brevity, we will hereafter use "confidence growth" and "ground-truth confidence growth" interchangeably.

4.1 Observing Ground-truth Confidence Growth on Reasoning Models

Observation 1: Consistent Confidence Ascent Correlates with Final Correctness. To empirically validate the connection between ground-truth confidence growth and final outcome accuracy, we analyzed trajectories generated by several strong open-source LLMs, such as Qwen2.5-Math.7B (Yang et al., 2024) and GPT-OSS-20B (OpenAI, 2025). Specifically, we prompted the models to generate a single reasoning trajectory (τ) for each question in a large set of reasoning tasks. This entire collection of trajectories was then partitioned into two distinct sets: the set of outcome-correct trajectores, $\mathcal{T}_{correct}$, where the final answer matched the ground truth, and the set of outcome-incorrect trajectories, $\mathcal{T}_{incorrect}$. We then calculated the **consistency of confidence growth**, defined as the proportion of positive-gain steps for each trajectory (i.e., $\frac{1}{T}\sum_{k=1}^{T}\mathbb{I}(C_k>0)$, where \mathbb{I} is the indicator function). Our findings reveal a clear distinction between the two groups. As illustrated in Figure 2, trajectories in $\mathcal{T}_{correct}$ exhibited a higher proportion of steps with positive confidence gain compared to trajectories in $\mathcal{T}_{incorrect}$. This

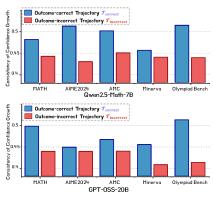


Figure 2: Consistency of confidence growth correlates with correctness. Outcome-correct trajectories ($\mathcal{T}_{correct}$) exhibit a higher proportion of steps with positive confidence gain ($C_k > 0$) compared to incorrect trajectories ($\mathcal{T}_{incorrect}$).

indicates that reasoning paths that result in a correct answer tend to be characterized by a more consistent, progressive increase in the model's belief in the ground-truth answer.

Observation 2: Coherent Reasoning Paths Exhibit More Consistent Confidence Ascent. While Observation 1 established a link between confidence growth and correct outcomes, we next sought to determine if this signal was also sensitive to the quality of the reasoning process. A trajectory can arrive at the correct answer through flawed or spurious steps, and a robust processlevel signal should be able to distinguish such cases. To investigate this, we focused exclusively on the set $\mathcal{T}_{correct}$. We employed a powerful external LLM evaluator (GPT-5) to further partition this set into two subgroups: those with logically coherent reasoning $\mathcal{T}_{coherent}$ and those with spurious reasoning $\mathcal{T}_{\text{spurious}}$, where the correct answer was reached via flawed logic or irrelevant steps (see Appendix A.6 for detailed evaluation prompts). As shown in Figure 3, the average proportion of positive-gain steps was significantly higher for $\mathcal{T}_{coherent}$ compared to $\mathcal{T}_{spurious}$. This

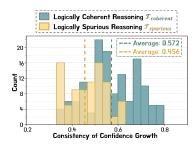


Figure 3: Consistency of confidence growth reflects reasoning quality. Coherent reasoning paths ($\mathcal{T}_{coherent}$) show a more consistent confidence ascent (higher proportion of $C_k > 0$ steps) than spurious paths ($\mathcal{T}_{spurious}$), even when both yield the correct final answer.

demonstrates that while both groups reached the correct final answer, the model's confidence grew

Figure 4: Qualitative example of a pivotal step. Among the reasoning steps, a critical insight at step h_7 (the introduction of Vieta's formulas for a cubic equation) results in a large, distinct spike in the ground-truth confidence gain ($C_7 = +4.83$). This is substantially larger than the gains from more routine algebraic steps. Further qualitative examples are provided in Appendix A.7.

more consistently when following a logically sound path. This finding suggests that confidence ascent is not merely an indicator of the final outcome but also a signal reflecting the internal quality of the reasoning trace itself.

Observation 3: Large Stepwise Confidence Gains Pinpoint Pivotal Reasoning Steps. Beyond the overall trend of confidence, we investigated whether the *magnitude* of the stepwise gain, C_k , correlates with the importance of individual reasoning steps. Qualitatively, we observe that large, positive spikes in C_k often coincide with pivotal moments in the reasoning process, such as the application of a key theorem or a critical insight. For instance, as illustrated in Figure 4, a step introducing the sum of roots formula for a cubic equation yields a substantially larger confidence gain compared to adjacent steps involving routine algebraic manipulation. To validate this rigorously, we conducted a quantitative pairwise comparison. For trajectories in $\mathcal{T}_{correct}$, we randomly sampled pairs

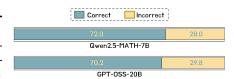


Figure 5: Quantitative validation of step importance. In a pairwise comparison, an LLM evaluator judged the step with the higher confidence gain $(C_i > C_j)$ as more critical with a win rate significantly above chance, confirming that gain magnitude correlates with step importance.

of reasoning steps, h_i and h_j , under the condition that $C_i > C_j$. We then prompted an LLM evaluator (GPT-5) to judge which of the two steps was more critical for reaching the final solution (see Appendix A.6 for detailed evaluation prompts). The step with the higher confidence gain, h_i , was frequently identified as more critical, achieving a win rate significantly above chance (Figure 5). This finding suggests that the magnitude of the confidence gain is not arbitrary; it is a meaningful signal that effectively pinpoints influential steps within a reasoning chain. This provides a strong rationale for using it as a training objective, as maximizing C_k would directly incentivize the model to generate these critical, problem-solving actions.

4.2 THEORETICAL JUSTIFICATION FOR GROUND-TRUTH CONFIDENCE GROWTH AS A PROCESS REWARD

Building on our empirical findings, we now provide a theoretical foundation for using confidence growth as a process reward. We prove that a reasoning step sampled from an ideal "oracle" policy (one that generates steps consistent with the ground truth, i.e., faithful steps) will, on average, increase or maintain the model's confidence in that ground truth.

The Oracle Policy Assumption Our theoretical analysis is built on the following assumption: a capable LLM, when conditioned on a correct final answer, is able to generate a faithful and logically sound reasoning path. This assumption is well-founded, as modern LLMs excel at rationalization; their training enables them to construct coherent explanations that bridge a given question and its answer. Based on this premise, we can construct an idealized model for analysis, which we term an oracle policy π_{oracle} . This policy is the model's own generative process given access to the ground-truth answer Y_{gt} , sampling the next step h_k from the distribution $\pi_{\theta}(h_k|q,Y_{\text{gt}},H_{< k})$.

 Proposition 1. Let C_k be the stepwise confidence gain at step k. The expected value of C_k is non-negative when the expectation is taken over reasoning steps h_k sampled from the oracle policy. Formally:

$$\mathbb{E}_{h_k \sim \pi_\theta(\cdot | q, Y_{\text{gt}}, H_{< k})}[C_k] \ge 0.$$

Proof. We begin with the definition of the ground-truth confidence growth C_k , as defined in Eq. 6:

$$\begin{split} C_k &= \log \pi_{\theta}(Y_{\mathrm{gt}}|q, H_{\leq k}) - \log \pi_{\theta}(Y_{\mathrm{gt}}|q, H_{< k}) \\ &= \log \frac{\pi_{\theta}(Y_{\mathrm{gt}}|q, h_k, H_{< k})}{\pi_{\theta}(Y_{\mathrm{gt}}|q, H_{< k})}. \end{split}$$

Next, we apply Bayes' rule to the numerator, $\pi_{\theta}(Y_{\text{gt}}|q, h_k, H_{< k})$:

$$\pi_{\theta}(Y_{\text{gt}}|q, h_k, H_{< k}) = \frac{\pi_{\theta}(h_k|q, Y_{\text{gt}}, H_{< k})\pi_{\theta}(Y_{\text{gt}}|q, H_{< k})}{\pi_{\theta}(h_k|q, H_{< k})}.$$

Substituting this back into the equation for C_k , the $\pi_{\theta}(Y_{\text{gt}}|q, H_{\leq k})$ terms cancel, yielding:

$$C_k = \log \frac{\pi_{\theta}(h_k|q, Y_{\text{gt}}, H_{< k})}{\pi_{\theta}(h_k|q, H_{< k})}.$$

Now, we take the expectation of C_k with respect to the oracle policy, $\pi_{\theta}(\cdot|q, Y_{\text{gt}}, H_{\leq k})$:

$$\mathbb{E}_{h_k \sim \pi_{\theta}(\cdot|q, Y_{\text{gt}}, H_{< k})}[C_k] = \sum_{h_k} \pi_{\theta}(h_k|q, Y_{\text{gt}}, H_{< k}) \log \frac{\pi_{\theta}(h_k|q, Y_{\text{gt}}, H_{< k})}{\pi_{\theta}(h_k|q, H_{< k})}$$

$$= D_{KL} \Big(\pi_{\theta}(\cdot|q, Y_{\text{gt}}, H_{< k}) \mid | \pi_{\theta}(\cdot|q, H_{< k}) \Big).$$

This expression is the Kullback-Leibler (KL) divergence between the probability distribution over the next reasoning step conditioned on the ground truth and the distribution without it. By the property of non-negativity of KL divergence, the proposition holds:

$$D_{KL}(\cdot||\cdot) \ge 0$$
 \therefore $\mathbb{E}[C_k] \ge 0$

Implication. This proof demonstrates that the expected confidence gain under an oracle policy is equivalent to the KL divergence between the ground-truth-conditioned policy and the standard policy. Since KL divergence is always non-negative, this provides the following theoretical guarantee: on average, a reasoning step consistent with the correct answer (i.e., faithful reasoning) will not decrease the model's confidence. This result validates the use of confidence growth as a strong inductive bias; encouraging the model to explore reasoning paths with non-decreasing confidence effectively constrains the search space to regions richer in logically sound reasoning.

5 METHOD: PROGRESSIVELY ASCENDING CONFIDENCE REWARD (PACR)

Based on our findings in Section 4, we now formalize how to incorporate the principle of ascending ground-truth confidence into the GRPO framework. To do this, we introduce the Progressively Ascending Confidence Reward (PACR), a procedural reward signal designed to complement the final outcome-based reward. We propose *two* variants: (1) Sparse-PACR, which applies a holistic, trajectory-level reward based on the consistency of confidence growth, and (2) Dense-PACR, which provides a fine-grained, step-wise reward based on the magnitude of each confidence change.

Sparse-PACR. In the Sparse setting, we compute a single procedural reward for an entire trajectory based on the consistency of its confidence growth. This reward, $C_{\text{sparse}}^{(i)}$, is the proportion of reasoning steps that produce a positive confidence gain. We calculate it using an indicator function, $\mathbb{I}(\cdot)$:

$$C_{\text{sparse}}^{(i)} = \frac{1}{T_i} \sum_{k=1}^{T_i} \mathbb{I}\left(C_k^{(i)} > 0\right),\tag{7}$$

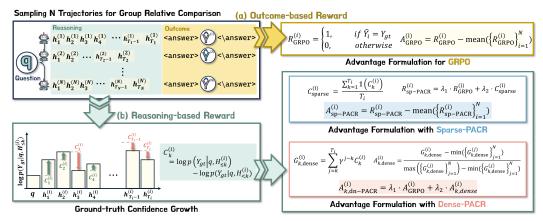


Figure 6: Overview of the PACR method and its integration with GRPO. Standard GRPO begins by sampling a group of N reasoning trajectories for a given question. (a) A standard outcome-based reward $(R_{\rm GRPO}^{(i)})$ is calculated based on the correctness of the final answer. (b) Our proposed reasoning-based reward is derived from the ground-truth confidence growth $(C_k^{(i)})$ at each step. This signal is integrated into the final advantage calculation in two ways: Sparse-PACR uses the consistency of confidence growth to compute a single reward for the entire trajectory, while Dense-PACR uses the magnitude of each step's gain to compute a fine-grained, per-step advantage.

where $C_k^{(i)}$ is the confidence gain in Eq. 6. The final reward for trajectory i, $R_{\text{sp-PACR}}^{(i)}$, is a weighted combination of the standard outcome-based reward, $R_{\text{GRPO}}^{(i)}$, and our sparse procedural reward:

$$R_{\text{sp-PACR}}^{(i)} = \lambda_1 \cdot R_{\text{GRPO}}^{(i)} + \lambda_2 \cdot C_{\text{sparse}}^{(i)}.$$
 (8)

This combined reward is then used to calculate the trajectory's advantage, $A_{\text{sp-PACR}}^{(i)}$, within the GRPO framework by centering it against the group average:

$$A_{\text{sp-PACR}}^{(i)} = R_{\text{sp-PACR}}^{(i)} - \text{mean} \left(\{ R_{\text{sp-PACR}}^{(j)} \}_{j=1}^{N} \right). \tag{9}$$

Dense-PACR. The Dense setting provides a more granular, step-wise reward signal. At each reasoning step k in trajectory i, we use the ground-truth confidence gain, $C_k^{(i)}$, as an immediate reward. From this, we compute the discounted return for that step, $G_{k,\mathrm{dense}}^{(i)}$, by summing the rewards from that point forward:

$$G_{k,\text{dense}}^{(i)} = \sum_{j=k}^{T_i} \gamma^{j-k} C_j^{(i)}, \tag{10}$$

where γ is a discount factor. To create a stable, step-wise advantage signal, $A_{k,\mathrm{dense}}^{(i)}$, we normalize these returns across the group at each step k. Specifically, we use Min-Max scaling to map the returns to a [0,1] range. This creates a purely positive signal that only incentivizes confidence growth without penalizing steps that do not, a design choice we validate in our ablations (Section 7.3). To handle trajectories of varying lengths, the discounted return $G_{k,\mathrm{dense}}^{(i)}$ is treated as zero for any step k that does not exist in trajectory i. The resulting advantage for a step k in trajectory i is then:

$$A_{k,\text{dense}}^{(i)} = \frac{G_{k,\text{dense}}^{(i)} - \min_{j}(\{G_{k,\text{dense}}^{(j)}\}_{j=1}^{N})}{\max_{j}(\{G_{k,\text{dense}}^{(j)}\}_{j=1}^{N}) - \min_{j}(\{G_{k,\text{dense}}^{(j)}\}_{j=1}^{N})}.$$
(11)

Finally, the total advantage at each step, $A_{k,\text{dn-PACR}}^{(i)}$, is the weighted sum of the trajectory-level GRPO advantage and our dense, step-wise advantage:

$$A_{k,\text{dn-PACR}}^{(i)} = \lambda_1 \cdot A_{\text{GRPO}}^{(i)} + \lambda_2 \cdot A_{k,\text{dense}}^{(i)}, \tag{12}$$

where $A_{\rm GRPO}^{(i)}=R_{\rm GRPO}^{(i)}-{\rm mean}(\{R_{\rm GRPO}^{(j)}\}_{j=1}^N).$ This final advantage is then used to update the policy.

6 EXPERIMENTAL SETUP

Models and Baselines. We experiment with three open-source LLMs: Qwen2.5-Math-1.5B, Qwen2.5-Math-7B (Yang et al., 2024), and Qwen3-4B¹ (Yang et al., 2025a). Our baseline for all experiments is Dr.GRPO (Liu et al., 2025), a bias-mitigated version of GRPO (Shao et al., 2024b), which we implement using the OAT framework (Liu et al., 2024). We compare this baseline against our two proposed methods, Sparse-PACR and Dense-PACR.

Datasets and Evaluation. For training, we use the MATH dataset (Hendrycks et al.). Following prior work (Liu et al., 2025), we use the full dataset for the 1.5B model and filter for the more challenging levels (3-5) for the 4B and 7B models. To evaluate performance, we test our models on five diverse mathematical reasoning benchmarks: MATH500 (Hendrycks et al.), Minerva-Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AIME 2024, and AMC 2023 (Li et al., 2024a). Final answers are programmatically checked for correctness using the Math-Verify (Kydlíček, 2025) library. All results are reported as pass@1 using greedy decoding (temperature of 0).

Training Details. For each problem, we generate a group of 8 responses using sampling with a temperature of 1.0. We report the average results across three runs with different random seeds for all experiments. All models were trained on a single node with $8 \times \text{NVIDIA H}100 \ 80\text{GB GPUs}$. Further details on hyperparameters, such as learning rate and batch size, are provided in Appendix A.5.

7 RESULTS AND ABLATIONS

7.1 EXPERIMENTAL RESULT

Table 1: **Results on reasoning benchmarks.** We report pass@1 accuracy across five datasets. Both Sparse-PACR and Dense-PACR consistently outperform the strong Dr.GRPO baseline across all model sizes. † is marked for the score reproduced and other baseline scores are from Liu et al. (2025). The colored numbers indicate the absolute performance change relative to the Dr.GRPO baseline, with green for improvements and red for degradations.

Base model + Method	AIME24	AMC	MATH500	Minerva	OlympiadBench	Average
Qwen2.5-Math-1.5B	20.0	32.5	33.0	12.5	22.8	24.2
R1-Distill-Qwen-1.5B @ 3k	2.5	21.7	52.2	16.3	17.3	22.0
Qwen2.5-Math-1.5B-Instruct	10.0	48.2	74.2	26.5	40.2	39.8
Qwen2.5-Math-1.5B + Dr.GRPO †	13.3	47.0	76.8	32.3	39.0	41.7
Qwen2.5-Math-1.5B + Sparse-PACR	20.0 +6.7	48.4 +1.4	77.4 +0.6	29.4 -2.9	37.8 _1.2	42.6 +0.9
Qwen2.5-Math-1.5B + Dense-PACR	23.3 +10.0	49.4 +2.4	77.4 +0.6	31.7 -0.6	39.0 0.0	44.2 +2.5
Qwen2.5-Math-7B	16.7	38.6	50.6	9.9	16.6	26.5
SimpleRL-Zero-7B	26.7	60.2	78.2	27.6	40.3	46.6
PRIME-Zero-7B	16.7	62.7	83.8	36.0	40.9	48.0
OpenReasoner-Zero-7B @ 3k	13.3	47.0	79.2	31.6	44.0	43.0
R1-Distill-Qwen-7B @ 3k	10.0	26.2	60.1	23.0	23.1	28.5
Qwen2.5-Math-7B-Instruct	16.7	53.0	83.6	29.8	42.7	45.1
Qwen2.5-Math-7B + Dr.GRPO †	30.0	56.6	81.8	34.6	45.2	49.6
Qwen2.5-Math-7B + Sparse-PACR	36.7 +6.7	55.4 _1.2	82.6 +0.8	34.6 0.0	45.6 +0.4	51.0 +1.4
Qwen2.5-Math-7B + Dense-PACR	43.3 +13.3	56.1 -0.5	81.9 +0.1	35.6 +1.0	46.1 +0.9	52.6 +3.0
Qwen3-4B	13.3	32.5	40.2	9.19	39.4	26.9
Qwen3-4B + Dr.GRPO †	40.0	63.8	88.4	33.8	46.8	54.6
Qwen3-4B + Sparse-PACR	33.3-6.7	67.5+3.7	86.2-2.2	35.3+1.5	54.4+7.6	55.3+0.7
Qwen3-4B + Dense-PACR	46.7+6.7	63.4-0.4	86.8-1.6	36.0+2.2	55.0+8.2	57.6+3.0

Table 1 presents the quantitative results on various math benchmarks. For the Qwen2.5-series, we also include the instruct models at the sample scale and R1-Distill models for comparison by following (Liu et al., 2025). Our proposed reward, PACR, demonstrates significant improvements over the outcome-based reward baseline (+Dr.GRPO) in both its Sparse and Dense setting. This shows that our core method provides a positive inductive bias for improving the reasoning skills of language models.

While the sparse trajectory-level reward, Sparse-PACR, is effective on its own, we observe that Dense-PACR, which provides a more fine-grained reward, consistently achieves better performance. This highlights that enriching the training process with a dense reward signal allows the model to learn from more detailed feedback, leading to further gains in its reasoning capabilities.

¹For the Qwen3-4B model, we set 'enable_thinking=False' to disable its built-in chain-of-thought capabilities, allowing for a direct comparison of how our method versus standard GRPO teaches this capability.

7.2 Training Curve: PACR Accelerates Exploration and Improves Convergence

Figure 7 illustrates the training dynamics, plotting the average pass@1 accuracy over training steps (left) and the corresponding rate of accuracy improvement (right). The right plot highlights that both PACR variants have a significantly higher rate of improvement compared to the Dr.GRPO baseline, especially during the critical early exploration phase of RL training. As shown on the left, this accelerated learning ultimately allows the PACR methods to converge to a higher final accuracy.

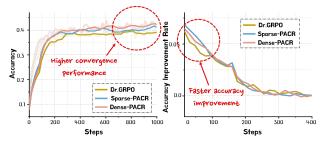


Figure 7: **Training dynamics for Qwen2.5-Math-1.5B.** Average pass@1 accuracy (left) and rate of accuracy improvement (right) during training. PACR methods show a faster initial rate of improvement, accelerating exploration and converging to a higher final performance.

7.3 ANALYSIS ON ADVANTAGE FORMULATION: IMPACT OF PENALIZING INTERMEDIATE STEPS

In this section, we analyze the impact of the advantage formulation in our Dense-PACR setting. A crucial design choice is how to normalize the raw discounted returns $(G_{k,\mathrm{dense}}^{(i)})$ into a stable advantage signal. We compare our Min-Max normalization against a widely used Leave-One-Out (LOO) baseline (Ahmadian et al., 2024; Cui et al., 2025).

The key difference is that the LOO baseline centers the returns, which can assign **negative advantages** that penalize steps with below-average confidence gains:

$$A_{k, \rm loo}^{(i)} = G_{k, \rm dense}^{(i)} - {\rm mean}(\{G_{k, \rm dense}^{(j)}\}_{j=1, j \neq i}^N). \eqno(13)$$

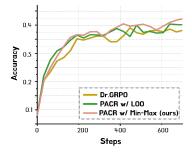


Figure 8: **Advantage Normalization.** Comparison of Min-Max normalization and a Leave-One-Out (LOO) baseline for Dense-PACR.

In contrast, our Min-Max normalization (Eq. 11) scales returns to a [0, 1] range, creating a **purely positive signal** for the reasoning process that only rewards confidence growth.

Figure 8 shows this design choice has a clear impact on the training dynamics. The penalizing nature of the LOO baseline initially accelerates learning by aggressively pruning suboptimal steps, but this leads to premature convergence and a performance plateau. Conversely, our non-penalizing Min-Max approach encourages more sustained exploration, ultimately converging to a higher final accuracy. With our method, process-level penalization is avoided; a negative training signal is only applied by the main GRPO reward when the model produces a definitively incorrect final answer.

8 CONCLUSION

In this work, we addressed the limitations of sparse, outcome-based rewards in RLVR by introducing the Progressively Ascending Confidence Reward (PACR), a dense, model-intrinsic signal derived from the model's evolving belief in the ground-truth answer. Through a series of empirical observations and a formal theoretical proof, we validated that confidence growth serves as a powerful inductive bias, effectively constraining the search space to regions richer in logically sound and faithful reasoning paths. Our experiments demonstrated that augmenting GRPO with PACR not only accelerates training but also converges to a higher final performance across multiple reasoning benchmarks, with the fine-grained Dense-PACR variant proving most effective. Ultimately, our work shows that informative, dense rewards for complex reasoning can be effectively extracted from the internal dynamics of the learning policy itself, suggesting a promising direction for creating more effective and reliable methods for fine-tuning the reasoning capabilities of large language models.

REFERENCES

- Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- Meng Cao, Lei Shu, Lei Yu, Yun Zhu, Nevan Wichers, Yinxiao Liu, and Lei Meng. Drlc: Reinforcement learning with dense rewards from llm critic, 2024.
- Alex J Chan, Hao Sun, Samuel Holt, and Mihaela Van Der Schaar. Dense reward for free in reinforcement learning from human feedback. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 6136–6154, 2024.
- Jie Cheng, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Gang Xiong, Yisheng Lv, and Fei-Yue Wang. Stop summation: Min-form credit assignment is all process reward model needs for reasoning. arXiv preprint arXiv:2504.15275, 2025.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261, 2025.
- Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv* preprint arXiv:2502.01456, 2025.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- John Hale. A probabilistic earley parser as a psycholinguistic model. In *North American Chapter of the Association for Computational Linguistics*, 2001. URL https://api.semanticscholar.org/CorpusID:5490143.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *Sort*, 2(4): 0–6.
- Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. arXiv preprint arXiv:2503.24290, 2025.
- Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. In *The Thirteenth International Conference on Learning Representations*.
- Hynek Kydlíček. Math-Verify: Math Verification Library, 2025. URL https://github.com/huggingface/math-verify.
- Roger Levy. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126-1177, 2008. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2007.05.006. URL https://www.sciencedirect.com/science/article/pii/S0010027707001436.

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

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

- Wendi Li and Yixuan Li. Process reward model with q-value rankings. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=wOEdh2cgEk.
- Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: a simple, effective, and efficient reinforcement learning method for aligning large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 29128–29163, 2024b.
 - Aiwei Liu, Haoping Bai, Zhiyun Lu, Yanchao Sun, Xiang Kong, Xiaoming Simon Wang, Jiulong Shan, Albin Madappally Jose, Xiaojiang Liu, Lijie Wen, et al. Tis-dpo: Token-level importance sampling for direct preference optimization with estimated weights. In *The Thirteenth International Conference on Learning Representations*.
- Zichen Liu, Changyu Chen, Xinyi Wan, Chao Du, Wee Sun Lee, and Min Lin. Oat: A research-friendly framework for llm online alignment, 2024.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.
- OpenAI. gpt-oss-120b gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to $\hat{q*}$: Your language model is secretly a q-function. In *First Conference on Language Modeling*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024a.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b.
- Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL https://qwenlm.github.io/blog/qwq-32b/.

- Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith,
 Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for
 language model training. In *Thirty-seventh Conference on Neural Information Processing Systems*,
 2023. URL https://openreview.net/forum?id=CSbGXyCswu.
 - An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.
 - An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.
 - Zhaohui Yang, Chenghua He, Xiaowen Shi, Linjing Li, Qiyue Yin, Shihong Deng, and Daxin Jiang. Beyond the first error: Process reward models for reflective mathematical reasoning. *arXiv* preprint *arXiv*:2505.14391, 2025b.
 - Eunseop Yoon, Hee Suk Yoon, Soo Hwan Eom, Gunsoo Han, Daniel Wontae Nam, Daejin Jo, Kyoung Woon On, Mark Hasegawa-Johnson, Sungwoong Kim, and Chang D Yoo. Tlcr: Token-level continuous reward for fine-grained reinforcement learning from human feedback. In *Findings of the 62nd Annual Meeting of the Association for Computational Linguistics*, *ACL 2024*, pp. 14969–14981. Association for Computational Linguistics (ACL), 2024.
 - Hee Suk Yoon, Eunseop Yoon, Mark A Hasegawa-Johnson, Sungwoong Kim, and Chang D Yoo. Confpo: Exploiting policy model confidence for critical token selection in preference optimization. In *Forty-second International Conference on Machine Learning*.
 - Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
 - Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*, 2024.
 - Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerlzoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv* preprint arXiv:2503.18892, 2025.
 - Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level direct preference optimization. In *Forty-first International Conference on Machine Learning*.
 - Danyang Zhang, Situo Zhang, Ziyue Yang, Zichen Zhu, Zihan Zhao, Ruisheng Cao, Lu Chen, and Kai Yu. Progrm: Build better gui agents with progress rewards. *arXiv preprint arXiv:2505.18121*, 2025.
 - Han Zhong, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. Dpo meets ppo: Reinforced token optimization for rlhf. In *ICML 2024 Workshop on Models of Human Feedback for AI Alignment*.
 - Mingkang Zhu, Xi Chen, Zhongdao Wang, Bei Yu, Hengshuang Zhao, and Jiaya Jia. Tgdpo: Harnessing token-level reward guidance for enhancing direct preference optimization. In *Forty-second International Conference on Machine Learning*.

A APPENDIX

A.1 LIMITATIONS AND FUTURE WORK

While our study demonstrates that Progressively Ascending Confidence Reward (PACR) provides a powerful inductive bias for mathematical reasoning, a limitation is that our work is primarily confined to language models. Therefore, a promising direction for future work is to investigate the efficacy of the proposed PACR framework in multimodal reasoning tasks, such as visual math problems, using Vision Language Models (VLMs).

A.2 Broader Impact

This work introduces a new inductive bias designed to improve the reasoning capabilities of Large Language Models. By leveraging the model's intrinsic confidence dynamics, our method provides fine-grained, step-level supervision without the significant overhead of training separate reward models or requiring manual data annotation. By eliminating the need for external process-reward models or human-annotated datasets, this research significantly lowers the computational and financial barriers to entry for training sophisticated reasoning agents.

A.3 THE USE OF LLMS

We used LLMs solely for light editing such as correcting grammatical errors and polishing some words. They did not contribute to research ideation, experiments, analysis, or substantive writing. We have reviewed all AI-assisted edits and take full responsibility for the final content of this paper.

A.4 ETHIC STATEMENT

This research adheres to the highest standards of academic integrity. All existing work is appropriately cited, and this paper does not violate the use of others' work without reference. The experiments conducted do not introduce new datasets or utilize any sensitive data related to demographic or identity characteristics.

A.5 TRAINING DETAILS

We present the details of our training configuration as follows. We use a total batch size of 128 and perform one PPO epoch per rollout. The per-device batch size is set to 4 for Qwen2.5-Math-1.5B, and 2 for both Qwen2.5-Math-7B and Qwen3-4B. During rollouts, we use a sampling temperature of 1.0 and generate 8 rollouts per prompt. For optimization, we use the AdamW optimizer (Loshchilov & Hutter, 2017) with a constant learning rate of 1e-6, without warmup or scheduler. The maximum prompt and generation lengths are set to 1024 and 3000 tokens, respectively. For the KL penalty, we set the coefficient $\beta=0$, effectively deactivating it during training. For the λ_1 , and λ_2 , we search in the range of [1, 0.99, 0.9, 0.8, 0.5] and [0.01, 0.1, 0.2, 0.5], and for the both sparse and dense setting, λ_1 and λ_2 are set to 0.9, and 0.1, respectively

A.6 PROMPT USED FOR OBSERVATION

To analyze the coherence of the reasoning paths (Observation 2) and the correlation between the large stepwise confidence gain and the pivotal reasoning step (Observation 3) in Section 4.1, we utilize GPT-5 as an evaluator. The prompts used to evaluate the reasoning steps for these respective observations are shown in Figures 9 and 10.

A.7 Examples for Observation 3

This section provides additional qualitative examples that support the central claim of Observation 3. As illustrated by the reasoning trajectories from Qwen2.5-Math-7B (Figure 11) and GPT-OSS-20B (Figure 12, 13 and 14), large positive spikes in the stepwise confidence gain C_k consistently align with pivotal problem-solving steps, such as applying a key formula or executing a critical calculation.

735 736

738

739

740

741

742 743

744

745

746

747

748 749

750

```
704
705
706
                         USER
707
                        You are a strict verifier. Given a question, and a proposed thinking process,
                             assign a LOGIC score from 0-5 for how logically valid the thinking is.
708
709
                              Scoring rubric (integers only):
                             5 = Fully sound: steps follow logically from the question; no gaps; math/symbol use correct.
710
                             4 = Mostly sound: one minor gap/assumption or small imprecision; overall valid.
                              3 = Mixed: at least one non-trivial gap or unjustified step; partially correct reasoning.
711
                             2 = Largely flawed: major gaps, speculative leaps, or misuse of evidence; little support.
1 = Almost entirely illogical: mostly wrong or incoherent reasoning.
712
                             0 = Nonsensical/contradictory or unrelated to the question.
713
714
                              - Evaluate the thinking itself, not whether the final option is correct.
715
                              - Assume minor grammar issues are irrelevant.
716
                              - Do not penalize brevity if logically sufficient.
                             - Refer to visible evidence briefly, but DO NOT reveal your own chain-of-thought. - Output JSON ONLY with fields: {{"justification": "<<=40 words>", "score": <0-5>}}.
717
718
                             Question: {question}
719
                              Proposed thinking:
                              {rationale}
720
721
                             Output a LOGIC score from 0-5 for how logically valid the thinking is.
722
723
```

Figure 9: Prompt used to evaluate reasoning quality for Observation 2.

```
You are a PAIRWISE key-step judge.
     Goal: Given a question, the full reasoning trajectory, and its predicted answer, decide which candidate step
(i or j) more causally increases the likelihood of arriving at the given predicted answer.
     VALID_IDS = {valid_indices}
     Inputs:
     - Ouestion: {auestion}
     - Full reasoning trajectory (JSON array of objects {{id, text}}): {steps_json}
     Predicted answer: {predicted_answer}
     - Compare: i = \{i\}, j = \{j\}
     Decision rules:
     - Judge only by causal contribution toward the *predicted* answer within the provided steps
     - Prefer the step that adds or applies a decisive constraint, calculation, elimination, or conclusion that moves
the reasoning closer to (or locks in) the predicted answer
     - Do not infer facts or use outside knowledge. Use VALID_IDS only.
     Output (STRICT JSON; no extra text):
     {{"winner": "i" | "j" | "TIE"}}
```

Figure 10: Prompt used to evaulate pairwise comparison for the impactful reasoning steps in Observation 3.

```
Let x^8+3x^4-4=p_1(x)p_2(x)\dots p_k(x), where each non-constant polynomial p_l(x) is monic with integer coefficients, and cannot be factored further over the integers. Compute p_1(1)+p_2(1)+\cdots+p_k(1).
Reasoning:
                    To solver the problem, we need to factor the polynomial (x^8 + 3x^4 - 4) into irreducible factors
                    over the integers and then evaluate the sum of the values of these factors at (x = 1).
                    Let's start by factoring the polynomial
                    we can use a substitution to make the polynomial easier to factor
                    Let (v = x^4)
                     Then the polynomial becomes (y^2 + 3y - 4).
                     we can factor this quadratic polynomial as follows: y^2+3y-4=(y-1)(y+4).
                     Substituting back (y = x^4), we get: x^8 + 3x^4 - 4 = (x^4 - 1)(x^4 + 4).
                     vext, we need to factor (x^4-1) and (x^4+4) further
                     The polynomial (x^4-1) can be factored as: x^4-1=(x^2-1)(x^2+1)=(x-1)(x+1)(x^2+1)
                     The polynomial (x^4+4) can be factored using the Sophie Germain identity: x^4+4=(x^2+2x+2)(x^2-2x+2)
                     These factors are all irreducible over the integers.
                    Now, we need to evaluate each factor at (x=1) and sum the results: p_1(1)=1-1=0, p_2(1)=1+1=2,
       C_8: 0.00
                    p_3(1) = 1^2 + 1 = 2, p_4(1) = 1^2 + 2 \cdot 1 + 2 = 5, p_5(1) = 1^2 - 2 \cdot 1 + 2 = 1
                    So, the sum is: 0+2+2+5+1=10
```

Figure 11: Example trajectory from Qwen2.5-Math-7B for a polynomial factorization problem. The model tackles the 8th-degree polynomial by reducing it to a quadratic form, then applying systematic factorization to break it down into irreducible integer-coefficient components. The pivotal moment occurs at step h_6 , where the model executes the core factorization. By successfully factoring the simplified quadratic expression $y^2 + 3y - 4$ and subsequently substituting x^4 back in for y, the model achieves the central algebraic decomposition of the original polynomial into two more tractable factors ($x^4 - 1$ and $x^4 + 4$). This step represents the main breakthrough required to solve the problem, and the corresponding confidence gain ($C_6 = +1.08$) indicates the model's recognition that the most significant hurdle in the factorization process has been overcome.

```
Question: If f(x) = ax^4 - bx^2 + x + 5 and f(-3) = 2, then what is the value of f(3)?

Reasoning:

(a) C<sub>1</sub>:-2.25 we have polynomial f(x) = ax^4 - bx^2 + x + 5.

(b) C<sub>2</sub>:+2.71 we know f(-3) = 2.

(c) C<sub>3</sub>:-2.78 we need f(3).

(d) C<sub>4</sub>:+0.12 we don't know a, b.

(e) Subt maybe symmetry? Let's compute f(-3) expression: a \cdot (-3)^4 - b \cdot (-3)^2 + (-3) + 5 = 81a - 9b - 3 + 5 = 81a - 9b + 2.

(e) C<sub>6</sub>:+3.96 Set equal to 2: 81a - 9b + 2 = 2 \Rightarrow 81a - 9b = 0 \Rightarrow b = 9a.

(e) C<sub>6</sub>:+3.96 Set equal to 2: 81a - 9b + 2 = 2 \Rightarrow 81a - 9b = 0 \Rightarrow b = 9a.

(e) C<sub>7</sub>: 0.00 Now f(3) = a \cdot 3^4 - b \cdot 3^2 + 3 + 5 = 81a - 9b + 8.

(a) C<sub>8</sub>: 0.00 Substitue b = 9a: 81a - 9 \cdot (9a) + 8 = 81a - 81a + 8 = 8.

(b) C<sub>9</sub>: 0.00 So answer 8.
```

Figure 12: Example trajectory from GPT-OSS-20B for a polynomial symmetry problem.

The problem appears ostensibly unsolvable due to the unknown coefficients a and b. The critical insight emerges across steps h_5 and h_6 , where the model leverages the inherent symmetry of the function's even-powered terms $(ax^4, -bx^2)$ and utilizes the given condition f(-3) = 2. By evaluating the expression for f(-3) and equating it to 2, the model uncovers the essential relationship between the unknown coefficients (b = 9a). This discovery is the pivotal moment that unlocks the entire problem, as it enables the cancellation of the unknown terms in the subsequent calculation of f(3).

```
Question: The sum of 27 consecutive positive integers is 37. What is their median?
Reasoning:
                     we need to find 27 consecutive positive integers whose sum is 3^7
      C_2: +0.12 3^7 = 2187.
                     Sum of 27 consecutive integers: if they are a, a+1, \ldots, a+26
                     Sum = 27a + (0 + 1 + \dots + 26) = 27a + \frac{(26 \cdot 27)}{2} = 27a + 351
                     Because sum of o to 26 = \frac{26.27}{1} = 351.
                     5 \circ 27a + 351 = 2187
       C_7: +6.28 Solve: 27a = 2187 - 351 = 1826.
           +5.18 1836/27 = ?27 \cdot 68 = 1836
                     So a = 68.
      C_{10}: +0. 33
                     So the numbers are 68 to 94 inclusive
       \mathcal{C}_{11}: 0.00 Median of 27 numbers is the 14<sup>th</sup> number (since 27 odd).
       C_{12}: 0.00 Starting at 68, 14th number is 68 + 13 = 81.
       C_{13}: 0.00 So median is 81.
```

Figure 13: Example trajectory from GPT-OSS-20B for a consecutive integer median problem. The initial phase of this problem involves correctly establishing an algebraic equation for the sum of the integers: 27a + 351 = 2187. The pivotal moment is a two-step process spanning h_7 and h_8 . In h_7 , the model transitions from algebraic formulation to the decisive computation to solve for the first integer, a, by successfully isolating the term 27a. This is immediately followed by the critical execution in h_8 , where the actual division is performed to find the explicit value of a. Securing the value of a is the primary breakthrough, as it provides the key to determining all numbers in the sequence and thus the median.

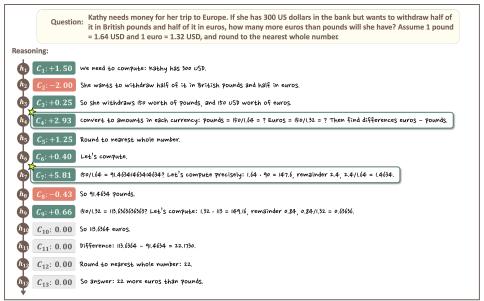


Figure 14: Example trajectory from GPT-OSS-20B for a currency exchange problem.

This reasoning trajectory features two pivotal moments. First, step h_4 serves as a critical planning phase, where the model correctly formulates the computational roadmap required for the solution: two currency conversions via division, followed by a subtraction. This demonstrates a comprehensive understanding of the problem's logic. The second, more significant pivotal moment occurs at the execution phase in step h_7 , where the model accurately performs the first of the two required divisions. Successfully clearing this key computational hurdle provides the model with high confidence ($C_7 = +5.81$) that its strategy is effective and the path to the final answer is now clear.