LEARNING REUSABLE REASONING CRITICS FROM EXPERT-REFERENCED SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Process-level reward models can score intermediate reasoning, but they typically require step-level annotations, preference data, or external verifiers. We study whether paired expert demonstrations can instead train a reusable reasoning critic without any of these. A naive expert-versus-policy classifier risks learning source or formatting cues instead of reasoning quality. We introduce the **Prefix-level Expert-Referenced Critic (PERC)**, which avoids this by using each expert trace three ways: as a positive example, as the reference answer for weakly labelling policy rollouts by agreement, and as the basis for targeted corruptions used as hard negatives, so that policy traces appear on both sides of the classifier and provenance alone cannot separate positives from negatives. Across GSM8K, MMLU-PRO, and MEDREASON, this critic captures a reusable reasoning-quality signal. As an **inference-time ranker**, it improves Best-of-16 pass@1 in all main reranking settings, transfers positively across all 27 source–target pairs with gains up to 12.7 points, complements consistency-based voting, and continues to improve pass@1 even for policies already trained with it. As a **process-level evaluator**, prefix-value changes localise human-labelled MEDREASON error units (two doctors, three final-year medical students) better than chance and entropy baselines and identify controlled GSM8K perturbations with up to 86.1% Hit@1. It is also a viable **training signal** for GRPO and remains competitive with imitation baselines without task-specific reward tuning. Overall, PERC shows that weak expert-referenced supervision yields reusable critics for inference-time selection and error diagnosis, while remaining feasible as a policy-training reward. [1]

1 INTRODUCTION

Reasoning post-training for large language models (LLMs) often relies on reinforcement learning (RL) with external outcome verifiers or step-level supervision (Guo et al., 2025; Uesato et al., 2022; Lightman et al., 2023). In many practical settings, separate verifiers and process labels are difficult to design or expensive to collect, whereas expert demonstrations (from humans or powerful LLMs) are comparatively easy to obtain. This makes supervised fine-tuning (SFT) on expert traces attractive (Guo et al., 2025). However, SFT trains models to imitate observed reasoning traces rather than to evaluate policy-generated alternatives. As a result, when generation deviates from demonstrated paths, the model lacks an explicit critic to rank candidate traces or diagnose where reasoning fails (Setlur et al., 2025).

A natural alternative is to learn such a critic from expert demonstrations. A naive expert-versus-policy classifier, however, can learn source or formatting cues rather than reasoning quality. We therefore ask whether paired expert traces can be used as references for training a reusable prefix-level critic: policy rollouts are labelled by final-answer agreement with the paired expert answer, and targeted corrupted expert traces provide hard negatives that preserve expert style while altering reasoning-relevant content.

This raises a key question: can expert-referenced trace evidence train a reusable critic that scores reasoning quality beyond source or formatting cues? We evaluate such a critic in three roles, beginning with uses that do not require further policy updates:

[1] We provide the code for our experiments at https://anonymous.4open.science/r/expert_reasoning-C1E8/

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

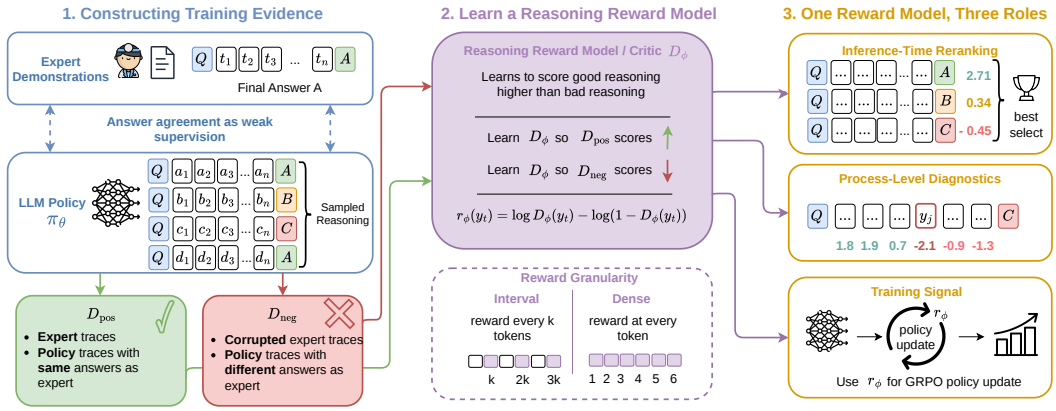


Figure 1: **A demonstration-learned critic reused across three roles.** PERC learns a prefix-level reasoning critic from expert-referenced trace evidence, without a separate external verifier or step-level annotations, and we evaluate the same frozen critic for inference-time reranking, error localisation, and policy optimisation.

We evaluate three roles for a learned reasoning critic:

1. (D1) **Inference-time ranking.** Selecting better sampled traces under a fixed test-time budget.
2. (D2) **Process-level feedback.** Localising where reasoning first becomes unreliable or incorrect.
3. (D3) **Policy-optimisation feasibility.** Testing whether the same diagnostic critic can double as an aggregate training reward without collapsing policy updates.

We introduce the Prefix-level Expert-Referenced Critic (PERC). PERC is not supervision-free: policy traces are labelled by agreement with the paired expert final answer. The setting we target is instead weaker than process supervision: no step-level annotations, no preference pairs, and no separate external verifier beyond the paired expert answer. PERC trains a prefix-level critic once from expert demonstrations, answer-agreement-labelled policy rollouts, and targeted corrupted expert traces. The critic is then frozen and reused for inference-time reranking ((D1)), process-level error localisation ((D2)), and policy optimisation ((D3)).

Contributions. We make three contributions:

1. **Expert-referenced prefix critic learning.** We train prefix-level reasoning critics from expert traces, answer-agreement-labelled policy rollouts, and targeted hard negatives at multiple granularities.

2. **One frozen critic, multiple roles.** We show that the same trained critic can be used for inference-time reranking, prefix-change error localisation, and aggregate policy optimisation.

3. **Empirical validation of critic reuse.** Across GSM8K, MEDREASON, and MMLU-PRO, we demonstrate that PERC delivers substantial gains in test-time reranking and fine-grained error diagnosis (both on controlled perturbations and expert clinician labels), while confirming its feasibility as an aggregate policy-training signal.

This makes the central contribution a reusable, process-level critic learned from expert-referenced trace evidence, rather than recovering an identifiable latent reward from expert behaviour alone. Figure 1 summarises the full pipeline.

2 PROBLEM FORMALISM

For a prompt $x \sim \mathcal{Q}$, we model an output as a reasoning trajectory $y = (y_1, \ldots, y_T)$ with autoregressive likelihood $p_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t})$. We write the prefix after token t as $h_t = (x, y_{\le t})$.

Prefix value and process score. The true step-level reward is unobserved, so PERC learns a prefix critic $D_\phi(h_t) \in (0, 1)$ for whether prefix h_t belongs to the positive expert-referenced trace class. We convert this probability into a log-odds prefix value $V_\phi(h_t) = \log D_\phi(h_t) - \log(1 - D_\phi(h_t))$ and aggregate supervised positions as $R_\phi(y; m) = Z_m^{-1} \sum_{t=1}^{T} m_t V_\phi(h_t)$, where $m_t \in \{0, 1\}$ and $Z_m = \sum_t m_t$. Dense, interval, and sparse rewards differ only in the mask m.

Expert-referenced critic. PERC does not recover an identifiable expert reward and does not use online adversarial critic updates. Instead, we first train a prefix critic on constructed positive and negative trace sets $\mathcal{D}_+$ and $\mathcal{D}_-$, defined in Section 4. After training, the critic is frozen and reused:

$$\hat{\phi} = \arg\min_{\phi} \mathcal{L}_D(\phi; \mathcal{D}_+, \mathcal{D}_-), \qquad \max_{\theta} \; \mathbb{E}_{y \sim \pi_\theta} \left[R_{\hat{\phi}}(y; m) \right]. \tag{1}$$

Thus, the learned score is a classifier-derived prefix-value signal under expert-referenced supervision, not a ground-truth reward recovered through adversarial inverse reinforcement learning.

3 RELATED WORK

Outcome and process supervision. Outcome-supervised reasoning RL optimises a terminal reward,

$$\max_{\theta} \; \mathbb{E}_{y \sim \pi_\theta} \left[R(y; m^{\text{out}}) \right], \qquad m_t^{\text{out}} = \mathbb{I}[t = T]. \tag{2}$$

where R is usually derived from an external or programmatic outcome verifier (Guo et al., 2025; Uesato et al., 2022; Yu et al., 2025; YuYue et al., 2025). Process reward models use denser masks and can improve reasoning supervision (Uesato et al., 2022; Lightman et al., 2023), with search variants further improving exploration (Zelikman et al., 2022; Yuan et al., 2023; Singh et al., 2024; Hosseini et al., 2024; Silver et al., 2017). PRIME and IPVRM reduce the need for step labels by deriving dense or prefix-level signals from outcome-labelled rollouts (Cui et al., 2025; Gao et al., 2026). PERC is closest to this prefix/process-critic line, but constructs its critic data relative to paired expert traces: expert demonstrations, answer-agreement-labelled policy rollouts, and targeted corrupted expert traces.

Preference-based reward learning. Preference-based RL learns a reward model R_ϕ, typically outcome-level, from pairwise preferences between a winning response y^w and a losing response y^l for the same prompt x:

$$\max_{\phi} \mathbb{E} \left[\log \sigma \big(R_\phi(y^w; m^{\text{out}}) - R_\phi(y^l; m^{\text{out}}) \big) \right], \quad \max_{\theta} \mathbb{E}_{y \sim \pi_\theta} [R_\phi(y; m^{\text{out}})].$$

These methods use preference data rather than verifier-labelled rewards (Christiano et al., 2023; Rafailov et al., 2024). They can support reranking and policy optimisation, but are typically response-level and less direct for localisation.

Demonstration-based learning. Supervised fine-tuning on expert traces optimises

$$\max_{\theta} \; \mathbb{E}_{y^E \sim p_E} \left[\frac{1}{T} \sum_t \log \pi_\theta(y_t^E) \right].$$

This often improves reasoning quality (Hinton et al., 2015; Guo et al., 2025; Kang et al., 2023; Kujanpää et al., 2025; Xu et al., 2025), but learns no explicit critic for policy-generated trajectories and therefore does not natively support reranking, localisation, or reward-based policy optimisation (Setlur et al., 2025). Continual self-distillation is complementary (Shenfeld et al., 2026). Broader demonstration-based reward-learning methods include Inverse-RLignment/AfD (Sun and Schaar, 2025) and PARED (Wiliński et al., 2026); these motivate reward learning from demonstrations, while PERC focuses on prefix-level reasoning critics and evaluates their reuse for ranking, localisation, and optimisation.

Source-label distillation. Adversarial distillation methods such as GAD distinguish teacher and student generations, often alongside an outcome verifier (Ye et al., 2025; Lee et al., 2025):

$$\max_{\phi} \min_{\theta} \quad \mathbb{E}_{y^E \sim p_E}\left[R_\phi(y^E; m^{\text{out}})\right] - \mathbb{E}_{y\sim\pi_\theta}\left[R_\phi(y; m^{\text{out}})\right], \quad \max_{\theta} \mathbb{E}_{y\sim\pi_\theta}\left[R_\phi(y; m^{\text{out}}) + R(y; m^{\text{out}})\right].$$

Such source-label critics can reward teacher provenance or style. PERC instead puts policy traces on both sides of the classifier through answer agreement, and places expert-style corruptions in the negative class. Its contribution is therefore not adversarial learning itself, but the expert-referenced construction of a reusable prefix critic. We evaluate whether this critic supports reranking (D1), localisation (D2), and policy optimisation (D3).

We summarise this comparison in Table 1.

Method Class	Example Works	Objective / Training Target	Supervision Data	Learned Signal	(D1) Rank	(D2) Localise	(D3) Optimise
Verifier / outcome supervision							
Outcome-supervised RL	Guo et al. (2025) Yu et al. (2025) YuYue et al. (2025)	$\max_\theta \ \mathbb{E}_{y\sim\pi_\theta}\left[R(y; m^{\text{out}})\right]$	Outcome verifier	Terminal reward	✓	✗	✓
Outcome-labelled process critics	Cui et al. (2025) Gao et al. (2026)	$\max_\phi \ \mathbb{E}_{(h_t,z)}$ $[z \log q_\phi(h_t) + (1-z)\log(1-q_\phi(h_t))]$ $r_t \approx q_\phi(h_t) - q_\phi(h_{t-1})$	Outcome-labelled rollouts	Prefix / process value	✓	✓	✓
Process-supervised RL	Uesato et al. (2022) Lightman et al. (2023)	$\max_\theta \ \mathbb{E}_{y\sim\pi_\theta}[R(y; m)]$	Process labels	Step-level reward	✓	✓	✓
Preference supervision							
Preference-supervised RL	Christiano et al. (2023) Rafailov et al. (2024)	$\max_\phi \ \mathbb{E}_{(y^w,y^l)}$ $\left[\log\sigma\left(R_\phi(y^w; m^{\text{out}}) - R_\phi(y^l; m^{\text{out}})\right)\right.$ then $\max_\theta \mathbb{E}_{y\sim\pi_\theta}\left[R_\phi(y; m^{\text{out}})\right]$	Preference pairs	Response reward	✓	✗	✓
Demonstration / source supervision							
SFT	Hinton et al. (2015) Guo et al. (2025) Xu et al. (2025)	$\max_\theta \ \mathbb{E}_{y^E\sim\mathcal{D}_E}\left[\frac{1}{T}\sum_t \log \pi_\theta(y_t^E)\right]$	Demos	Policy likelihood	✗	✗	✗
Demo reward learning	Sun and Schaar (2025) Wiliński et al. (2026)	$\max_\phi \ \mathbb{E}_{y^E\sim p_E}[\log D_\phi(y^E)]$ $+\mathbb{E}_{y\sim\pi_\theta}[\log(1-D_\phi(y))]$ then $\max_\theta \mathbb{E}_{y\sim\pi_\theta}[R_\phi(y)]$	Demos + policy samples	Response-level demo reward	✓	✗	✓
Source-label distillation	Ye et al. (2025) Lee et al. (2025)	$\max_\phi \min_\theta$ $(\mathbb{E}_{y^E\sim p_E}[R_\phi(y^E; m^{\text{out}})] - \mathbb{E}_{y\sim\pi_\theta}[R_\phi$ then $\max_\theta \mathbb{E}_{y\sim\pi_\theta}\left[R_\phi(y; m^{\text{out}}) + R(y;\right.$	Teacher / student source	Source / provenance reward	(✓)	✗	✓
Expert-referenced supervision							
PERC	**this work**	$\hat{\phi} = \arg\min_\phi \mathcal{L}_D(\phi; \mathcal{D}_+, \mathcal{D}_-)$ then reuse $R_{\hat{\phi}}(y; m)$	Demos + answer agreement + corruptions	Expert-ref. prefix critic	✓	✓	✓

Table 1: **Related work comparison.** Methods are grouped by supervision regime while preserving the (D1)–(D3) roles used in our experiments. The learned-signal column distinguishes terminal rewards, supervised step rewards, outcome-labelled prefix/process critics, response-level demonstration rewards, source/provenance rewards, and PERC's expert-referenced prefix critic. Checkmarks indicate studied uses, not identical assumptions; parenthesised checkmarks indicate partial support.

4 METHOD

Our goal is to learn a reusable process-level reasoning signal from expert-referenced trajectories, without a separate external verifier, step-level labels, or preference pairs. PERC trains a prefix critic once on constructed positive and negative trace evidence, then reuses its induced prefix values to rank fixed samples, localise error spans, and, after aggregation, update a policy. The critic is therefore not an identifiable expert reward, but a classifier-derived expert-referenced signal.

4.1 TRAINING EVIDENCE CONSTRUCTION

A naive split (all expert positive, all policy negative) encourages style rather than reasoning-quality discrimination (Ye et al., 2025; Lee et al., 2025). We instead use answer agreement with the paired expert trace as weak outcome supervision. Policy rollouts for critic training are sampled from the initial policy π_0. For prompt x, let y^E be the paired expert trace and let $\mathcal{O}(y)$ be the normalised final-answer string extracted from y; agreement is exact string equality after normalisation, with extraction details in Appendix B:

$$\mathcal{D}_{\text{pos}} = \{y \sim p_E\} \cup \{y \sim \pi_0 \mid \mathcal{O}(y) = \mathcal{O}(y^E)\}.$$

Including answer-agreeing policy rollouts in $\mathcal{D}_{\text{pos}}$ puts policy text on both sides of the classifier, reducing source and formatting shortcuts. For negatives, we use answer-mismatched policy rollouts and paired counterfactual hard negatives. Given an expert trace y^E, the corruption operator $\mathcal{C}(y^E)$ preserves the prompt, formatting, source, and most content while minimally altering a reasoning-relevant property, such as a local arithmetic step, numerical value, or answer-bearing intermediate conclusion (Appendix B):

$$\mathcal{D}_{\text{neg}} = \{y \sim \pi_0 \mid \mathcal{O}(y) \neq \mathcal{O}(y^E)\} \cup \{\mathcal{C}(y^E) \mid y^E \in \mathcal{D}^E\}.$$

We assign the resulting labels at the trace level and apply them to the supervised token positions selected by m_t. Thus, the learned prefix values are induced process signals from weak expert-referenced labels, not observed step-level correctness labels. Critic batches are class-balanced; corruptions and clean expert traces are generated at a 1:1 rate (Appendix B).

4.2 Reasoning Critic and Reward Definition

We train a prefix classifier $D_\phi(h_t) \in (0, 1)$, where $h_t = (x, y_{\leq t})$, to distinguish prefixes from $\mathcal{D}_{\text{pos}}$ and $\mathcal{D}_{\text{neg}}$. Because supervision is trace-level, $D_\phi(h_t)$ should not be read as a direct label for whether token y_t is correct. Instead, it estimates how strongly the partial trace resembles the positive expert-referenced class. A binary mask $m_t \in \{0, 1\}$ specifies where this supervision is applied. We evaluate two process-level granularities in the main experiments: *Interval* ($m_t = 1$ every k tokens) and *Fully Dense* ($m_t = 1$ for all t). The same formulation also admits a *Sparse Outcome* variant ($m_t = \mathbb{I}[t = T]$), which scores only the final position. Coarser masks are typically more stable for optimisation, while denser masks provide finer process feedback for localisation.

The critic is trained by minimising the length-normalised binary cross-entropy:

$$\mathcal{L}_D(\phi) = -\mathbb{E}_{y\sim\mathcal{D}_{\text{pos}}}\left[Z_m^{-1}\sum_{t=1}^{T} m_t \log D_\phi(h_t)\right] - \mathbb{E}_{y\sim\mathcal{D}_{\text{neg}}}\left[Z_m^{-1}\sum_{t=1}^{T} m_t \log\big(1 - D_\phi(h_t)\big)\right]. \quad (3)$$

After fitting the critic, we convert probabilities into prefix values:

$$V_\phi(h_t) = \log D_\phi(h_t) - \log\big(1 - D_\phi(h_t)\big). \quad (4)$$

For localisation, the local process signal is the value change $\delta_\phi(t) = V_\phi(h_t) - V_\phi(h_{t-1})$; large negative changes indicate prefixes where the trace becomes less consistent with the positive class.

4.3 Policy Optimisation with Learned Rewards

For policy optimisation, we keep the trained critic $D_{\hat{\phi}}$ fixed and optimise π_θ with Group Relative Policy Optimisation (GRPO) (Shao et al., 2024). At each policy iteration, we freeze θ_{old}, sample a group of G traces $\{y^{(g)}\}_{g=1}^{G} \sim \pi_{\theta_{\text{old}}}$ for prompt x, and update θ while keeping θ_{old} fixed.

The prefix value $V_\phi(h_t)$ is directly available only at supervised positions. For *interval* masks, each supervised value is backfilled to the preceding unsupervised tokens; the same rule would apply to the sparse outcome variant:

$$t' = \min\{k \geq t \mid m_k = 1\}, \qquad r_\phi(y_t) = V_\phi(h_{t'}).$$

For *dense*, $r_\phi(y_t) = V_\phi(h_t)$.

For policy optimisation, we aggregate prefix values into a length-normalised sequence score following Cetin et al. (2025) and Cui et al. (2025):

$$\bar{r}^{(g)} = \frac{1}{T_g}\sum_{t=1}^{T_g} r_\phi(y_t^{(g)}), \quad A^{(g)} = \frac{\bar{r}^{(g)} - \mu}{\sigma + \epsilon}. \quad (5)$$

We assign the same sequence-level advantage $A^{(g)}$ to all tokens in generation g and optimise the PPO clip loss (Schulman et al., 2017) on policy-generated traces. Thus, (D3) tests the critic as an aggregate policy-training signal, while (D2) evaluates its token-level structure through prefix-value changes.

Appendix A provides implementation details on critic training, batching, reward clipping, and policy optimisation; Algorithm 1 summarises the full procedure.

5 EXPERIMENTS

We evaluate PERC on GSM8K (Cobbe et al., 2021), MEDREASON (Wu et al., 2025), and MMLU-PRO (Wang et al., 2024). These benchmarks stress complementary regimes: multi-step arithmetic with human-written demonstrations, medical-domain multiple-choice transfer with plausible distractors, and broad scientific reasoning. Our base policies are open-weight instruction-tuned variants not trained for reasoning: `Qwen2.5` (7B) (Bai et al., 2023), `Llama3` (8B) (Touvron et al., 2023), and `Qwen3-4B` (Yang et al., 2025). We instantiate each policy as a text generator and the reasoning critic as a prefix/token classifier.

Because PERC's central output is a reusable frozen critic, we first evaluate critic reuse before policy optimisation. Section 5.1 tests inference-time ranking on fixed candidate sets and transfer across tasks/backbones. Section 5.2 tests whether prefix-value changes localise medical-expert-labelled natural errors and controlled perturbations. Section 5.3 then asks whether aggregating the same critic also provides a competitive policy-optimisation signal. Appendix B contains implementation details.

For each backbone and dataset, we train PERC critics with *interval* ($k = 15$) and *dense* masks using the evidence construction in Section 4, then keep the critic fixed for all downstream uses. The experiments below then evaluate the resulting critic in three ways. We begin with inference-time use, where the trained critic is held fixed and used only to score candidate traces. We then test process-level localisation via prefix-value changes, and finally report policy-optimisation results using the same critics as aggregate rewards.

5.1 INFERENCE-TIME CRITIC-GUIDED RERANKING

> **RQ1:** *Can the learned critic select better traces from a fixed candidate set at inference time?*

Experimental setup. Given a trained PERC critic, we rank N sampled traces by their prefix-value score and return the highest-scoring one. Random selection from the same candidate set equals the generator's expected pass1, so gains isolate selection quality without further policy updates.

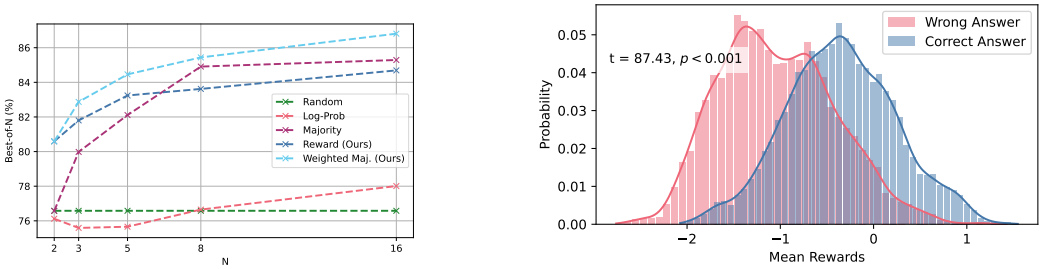
Fixed-candidate ranking and transfer. Table 2 uses fixed `Qwen2.5-7B` SFT candidates, so only the scoring critic changes. Diagonal entries test same-dataset ranking, off-diagonal entries test transfer, and the dashed divider separates same-backbone from out-of-backbone critics. All 27 source–target pairs improve, with gains up to +12.7 pp.

Policy Dataset	Baseline	Qwen2.5-7B			Llama3.1-8B			Qwen3-4B		
		GSM8K	MMLU-PRO	MEDREASON	GSM8K	MMLU-PRO	MEDREASON	GSM8K	MMLU-PRO	MEDREASON
GSM8K	71.1 ± 1.7	*79.9 (↑ +8.8)*	79.9 (↑ +8.8)	75.1 (↑ +4.0)	*82.5 (↑ +11.4)*	78.8 (↑ +7.7)	81.2 (↑ +10.1)	*83.8 (↑ +12.7)*	83.3 (↑ +12.2)	78.9 (↑ +7.8)
MMLU-PRO	48.1 ± 1.9	49.0 (↑ +0.9)	*51.3 (↑ +3.2)*	51.8 (↑ +3.7)	53.8 (↑ +5.7)	*53.8 (↑ +5.7)*	48.9 (↑ +0.8)	56.8 (↑ +8.7)	*55.3 (↑ +7.2)*	54.3 (↑ +6.1)
MEDREASON	66.3 ± 2.1	67.1 (↑ +0.8)	69.5 (↑ +3.2)	*68.5 (↑ +2.2)*	70.1 (↑ +3.8)	71.9 (↑ +5.6)	*78.0 (↑ +11.7)*	71.4 (↑ +5.1)	69.9 (↑ +3.6)	*71.9 (↑ +5.6)*

Table 2: **Best-of-16 cross-critic transfer reranking values (%).** All candidates come from the same frozen `Qwen2.5-7B` SFT generator; only the scoring critic varies, isolating critic quality from generator changes. Cells report PERC-guided pass@1, with gains over random selection in parentheses. Italics mark same-dataset critics; the dashed divider separates same-backbone from out-of-backbone critics. Table 3 instead varies the generator by row.

Reranking behaviour. Figure 2 shows both when and why critic-guided selection helps. Across $N \in \{2, 3, 5, 8, 16\}$, critic reranking is strongest at low budgets, while reward-weighted majority voting is best at larger N. The critic also separates correct from incorrect traces ($t = 87.43, p < 0.001$), explaining the gains in Table 3; full budget and separation results are in Appendix Tables 12–16 and Appendix Figures 6–14.

In-distribution reranking. Table 3 differs from Table 2: candidates are sampled from each GRPO-optimised policy, not from a shared frozen SFT pool. Thus, the Random column is that policy's own pass@1 and varies by backbone, dataset, and granularity. PERC-guided reranking still improves all settings at $N = 16$, with gains up to +8.7 pp (`Qwen3-4B`+*interval*, MMLU-PRO) and +6.4 pp (`Llama3.1-8B`+*dense*, GSM8K); all MEDREASON gains are positive (+0.7 to +5.8 pp). This shows that the critic remains useful for selection even after optimising the policy.



(a) **Reranking baselines** `Qwen3-4B`, *interval*, GSM8K.

(b) **Reward separation** `Qwen3-4B`, *interval*, MMLU-PRO.

Figure 2: **Inference-time behaviour of learned critics.** 2a: low-budget gains from reward-guided reranking; at higher N, reward-weighted majority voting is strongest. 2b: correct traces receive higher critic scores than incorrect traces ($t = 87.43, p < 0.001$).

Method	Granularity	GSM8K			MMLU-PRO			MEDREASON			Avg.
		Random	PERC	Δ	Random	PERC	Δ	Random	PERC	Δ	Δ
`Qwen2.5-7B`	*Interval*	90.2 ± 1.3	92.2 ± 1.4	(↑ **+2.0**)	50.6 ± 1.9	53.1 ± 2.5	(↑ **+2.5**)	67.2 ± 2.5	68.1 ± 2.9	(↑ **+0.9**)	**+1.8**
	Dense	89.7 ± 1.3	92.2 ± 1.4	(↑ **+2.5**)	43.8 ± 2.0	46.0 ± 2.5	(↑ **+2.2**)	68.7 ± 2.4	70.2 ± 2.8	(↑ **+1.5**)	**+2.1**
`Llama3.1-8B`	*Interval*	77.9 ± 1.7	82.2 ± 2.0	(↑ **+4.3**)	36.6 ± 1.8	38.1 ± 2.4	(↑ **+1.6**)	76.3 ± 2.1	81.4 ± 2.5	(↑ **+5.1**)	**+3.7**
	Dense	65.7 ± 2.0	72.1 ± 2.4	(↑ **+6.4**)	37.9 ± 1.8	39.7 ± 2.5	(↑ **+1.9**)	75.1 ± 2.1	80.9 ± 2.4	(↑ **+5.8**)	**+4.7**
`Qwen3-4B`	*Interval*	90.0 ± 1.4	92.9 ± 1.4	(↑ **+2.9**)	53.5 ± 2.4	62.2 ± 2.4	(↑ **+8.7**)	69.1 ± 2.6	71.4 ± 2.9	(↑ **+2.3**)	**+4.6**
	Dense	92.6 ± 1.2	94.0 ± 1.3	(↑ **+1.5**)	55.1 ± 2.3	59.2 ± 2.4	(↑ **+4.1**)	69.2 ± 2.5	69.9 ± 2.8	(↑ **+0.7**)	**+2.1**

Table 3: **Best-of-16 in-distribution reranking performance (%).** Candidates come from the GRPO-optimised policies in Section 5.3, unlike the fixed SFT pool in Table 2. PERC is compared with random selection from the same samples; Δ is the percentage-point gain, and Avg. Δ averages across datasets.

Takeaway. RQ1 is answered positively: the learned critic is a reusable inference-time ranking signal that improves all 27 transfer settings and complements consistency-based voting under fixed budgets.

5.2 REASONING ERROR LOCALISATION

RQ2: *Do prefix-value changes from the learned critic localise where reasoning first fails?*

Reranking shows that the learned critic scores complete traces. We next ask whether its prefix structure is locally meaningful: when reasoning first becomes invalid, does the critic value drop near the error?

Experimental setup. We use two complementary localisation tests. For natural errors, three final-year medical students and two doctors annotate each of 10-25 incorrect `Qwen2.5-7B` SFT generations from MEDREASON, marking the first reasoning unit that is medically incorrect, unsupported, contradictory, or misleading. We exclude non-localisable cases; if the reasoning is clinically correct but the final option is wrong, the target is the final unit. More information can be found in Appendix A.4.

We also perform a second controlled experiment for mathematical reasoning on GSM8K; we start from expert traces and insert one targeted arithmetic perturbation while preserving the prompt, formatting, and most surrounding text: an operator flip, numeric-literal offset, or replacement by an earlier intermediate value (Appendix B). This gives the exact injected error token, testing whether value changes localise the intervention rather than only detecting a wrong final answer.

Hit metrics. For each signal a, we score token transitions by $d_t^{(a)} = \max(0, z_{t-1}^{(a)} - z_t^{(a)})$, where $z_t^{(a)}$ is the PERC prefix value or negative entropy at prefix h_t; chance samples uniformly from candidate positions. The highest-scoring transition is the predicted error. For human labels, we map this token prediction to its numbered reasoning unit: Hit@1 requires the human first-error unit, while Hit@$\pm$1 also accepts one adjacent unit. For controlled GSM8K, Hit@k requires the predicted transition to fall within k tokens of the injected perturbation; thus Hit@1/Hit@7 use $\pm$1/$\pm$7 token windows.

Setting	Model	Hit@1				Hit@±1 (MedReason) / Hit@7 (GSM8K)			
		Chance	Entropy	Interval	Dense	Chance	Entropy	Interval	Dense
MedReason human-labelled	Qwen2.5-7B	8.1 ± 0.3	7.3 ± 5.0	12.7 ± 5.9	**14.5** ± 6.4	20.0 ± 1.1	14.5 ± 6.4	**27.3** ± 8.2	24.5 ± 7.7
	Llama3-8B	8.1 ± 0.3	8.2 ± 5.0	**30.0** ± 8.2	9.1 ± 5.0	20.0 ± 1.1	16.4 ± 6.4	**40.9** ± 9.1	27.3 ± 8.2
	Qwen3-4B	8.1 ± 0.3	14.5 ± 6.4	**18.2** ± 6.9	16.4 ± 6.8	20.0 ± 1.1	29.1 ± 8.2	**31.8** ± 8.7	28.2 ± 8.2
GSM8K controlled corruptions	Qwen2.5-7B	11.9 ± 0.6	15.3 ± 2.0	**65.5** ± 2.6	47.6 ± 2.8	32.0 ± 1.0	40.0 ± 2.8	**65.5** ± 2.6	54.9 ± 2.6
	Llama3-8B	11.9 ± 0.6	28.1 ± 2.5	**86.1** ± 2.0	65.8 ± 2.8	32.0 ± 1.0	54.9 ± 2.7	**86.1** ± 2.0	80.6 ± 2.2
	Qwen3-4B	11.9 ± 0.6	0.2 ± 0.2	**78.9** ± 2.2	62.0 ± 2.7	32.0 ± 1.0	1.9 ± 0.9	**78.9** ± 2.2	70.3 ± 2.5

Table 4: **Error localisation.** Hit rates (%) across critic backbones, reported as mean ± half-width of the bootstrapped 95% CI. Medical-expert MedReason is unit-level (n=110); controlled GSM8K reports synthetic perturbations (n=1234).

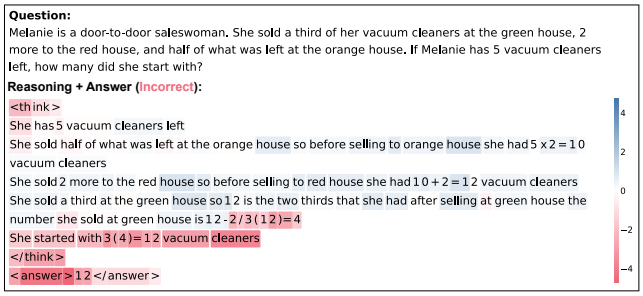


Figure 3: **Qualitative example.** The dense PERC prefix value drops at the first erroneous step.

Localisation results. Table 4 shows that PERC prefix-value changes localise errors better than chance and entropy baselines. On medical-expert MedReason labels, PERC is best for all backbones on Hit@±1, reaching 40.9% for `Llama3-8B` interval versus 20.0% chance; exact-unit hits peak at 30.0%. On controlled GSM8K, interval rewards are best for all backbones, reaching 65.5–86.1% Hit@1/Hit@7. For interval rewards, Hit@1 and Hit@7 coincide because the $k = 15$ mask backfills the same score across a local span. Figure 3 shows the prefix value staying high on valid steps and dropping at the first error; Appendix Figure 15 gives further correct/incorrect reward divergence examples.

Takeaway. RQ2 is answered positively: learned prefix-value changes improve natural and controlled error localisation over chance and entropy baselines.

5.3 Using the Same Critic for Policy Optimisation

RQ3: *Can the diagnostic prefix critic also serve as a viable policy-optimisation signal?*

Experimental setup. Finally, we test whether the same frozen critic can also support policy updates. For each backbone, we optimise π_θ with *interval* ($k = 15$) and *dense* PERC scores aggregated into a sequence-level GRPO advantage (Section 4.3). This tests aggregate reward quality, not token-specific credit assignment.

Method	Granularity	GSM8K			MMLU-Pro			MedReason		
		Qwen2.5-7B	Llama3.1-8B	Qwen3-4B	Qwen2.5-7B	Llama3.1-8B	Qwen3-4B	Qwen2.5-7B	Llama3.1-8B	Qwen3-4B
SFT	–	71.1 ± 1.6	67.5 ± 2.1	77.5 ± 1.6	48.1 ± 1.9	**47.2** ± 1.9	53.9 ± 2.0	66.3 ± 2.2	**79.5** ± 2.0	67.4 ± 2.3
PERC	*Interval*	**90.2** ± 1.3	**77.9** ± 1.7	90.0 ± 1.4	**50.6** ± 1.9	36.6 ± 1.8	53.5 ± 2.4	67.2 ± 2.5	76.3 ± 2.1	69.1 ± 2.6
	Dense	89.7 ± 1.3	65.7 ± 2.0	**92.6** ± 1.2	43.8 ± 2.0	37.9 ± 1.8	**55.1** ± 2.3	**68.7** ± 2.4	75.1 ± 2.1	**69.2** ± 2.5

Table 5: **Held-out pass@1 in the no-external-verifier setting. Bold**/underlined mark best/second-best across SFT and PERC variants shown in the main text. Values are the mean ± half-width of the bootstrapped 95% confidence interval.

Policy-optimisation performance. Table 5 shows that PERC provides a competitive aggregate reward signal, but is not uniformly better than SFT. Selecting the best granularity per backbone–dataset pair, PERC improves over SFT in 7/9 settings. Gains are strongest on GSM8K (3/3 backbones) and

positive for 2/3 backbones on both MMLU-PRO and MEDREASON; the main underperforming cases are `Llama3.1-8B` on MMLU-PRO and MEDREASON. We interpret this as a feasibility result: the learned critic can be aggregated into a useful policy-training reward, while its main advantage is that the same critic also supports inference-time reranking and process-level localisation.

Reranking after policy optimisation. The policies in Table 5 also generate the candidates used in Table 3. Thus, Table 3 tests whether the frozen critic still helps after its aggregate signal has already updated the policy. It does: PERC-guided reranking improves pass@1 in all settings, by up to +8.7 pp (`Qwen3-4B`+*interval*, MMLU-PRO), showing that the critic remains a complementary inference-time signal rather than being fully absorbed into the policy weights.

Additional baselines. Table 6 compares PERC with direct answer-agreement GRPO and Generative Adversarial Distillation (GAD) (Ye et al., 2025) on `Qwen2.5-7B`. Answer-agreement GRPO provides a dedicated binary terminal reward at step T. By design, it optimises exclusively for terminal pass@1 on tasks with programmatic string extraction, but yields no reusable prefix artefact for (D1) reranking or (D2) localisation. In contrast, PERC is designed first and foremost as a reusable process critic: it approaches or matches the performance of terminal answer-agreement GRPO in training, while uniquely unlocking fine-grained step diagnosis and test-time reranking that terminal RL cannot provide. These signals are therefore complementary rather than mutually exclusive: SFT, answer-agreement GRPO, and PERC can be combined when exact answer matching is available, while PERC provides the reusable critic studied in RQ1 and RQ2. Compared with the GAD-style source-label baseline, which uses teacher/student trace provenance as its signal, PERC gives stronger pass@1 and larger reranking gains.

Method	Granularity	GSM8K	MMLU-PRO	MEDREASON	(D1) Avg. Δ	(D2) Localise
SFT	–	71.1 ± 1.6	48.1 ± 1.9	66.3 ± 2.2	(✗)	(✗)
Answ-agr. GRPO	*Terminal*	90.0 ± 1.4	**53.5** ± 2.0	**69.2** ± 2.4	(✗)	(✗)
Source-based	*Terminal*	76.2 ± 1.4	49.3 ± 2.1	67.8 ± 2.4	✓ (↑ +0.3)	(✗)
PERC	*Interval*	**90.2** ± 1.3	50.6 ± 1.9	67.2 ± 2.5	✓ (↑ +1.8)	✓
PERC	*Dense*	89.7 ± 1.3	43.8 ± 2.0	68.7 ± 2.4	✓ (↑ +2.1)	✓

Table 6: **Additional baselines.** Pass@1 is reported for SFT, answer-agreement GRPO, GAD, and PERC.

For completeness, Appendix C presents ablations on expert corruptions, group size, clipping, and policy-vs-expert-only separation.

Takeaway. RQ3 shows that the critic can serve as a useful aggregate policy-optimisation reward. Its main value, however, is broader: the same critic remains reusable for inference-time selection and process-level diagnosis, rather than being absorbed into the trained policy.

6 LIMITATIONS

PERC has several limitations. First, PERC is more expensive than SFT because it trains a separate prefix critic and then uses it for policy optimisation; Appendix Table 7 reports 3–4× higher cost in representative runs. Second, answer agreement is weak supervision: it can miss flawed reasoning with a correct final answer and penalise partially correct traces. It is also easiest to apply on benchmarks with extractable normalised answers; open-ended settings would require more robust answer extraction or semantic matching. Third, denser rewards provide finer process feedback but are less stable, with occasional reward–correctness decoupling. Finally, our GRPO update aggregates prefix scores into one sequence-level advantage rather than using token-specific returns; improving this interface is future work.

7 CONCLUSION

PERC learns reusable reasoning critics from expert-referenced traces without a separate external verifier or step-level annotations. Across three datasets, the critic improves Best-of-16 reranking in all settings, transfers across all 27 source–target pairs, localises medical-expert-marked MEDREASON errors and controlled GSM8K perturbations better than chance and entropy baselines, and remains competitive when aggregated for policy optimisation. These results support weak expert-referenced supervision as a route to reusable critics for ranking, diagnosis, and training.

AI USE STATEMENT

In this work, we used generative AI tools to obtain demonstrations for solving reasoning tasks, implement and maintain the code for running experiments, suggest a structure for the paper, and prepare plots for the figures. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, to propose or refine hypotheses, to design our research methodology, to support qualitative and thematic data analysis, or to interpret results, and the generation of synthetic datasets and assistance with translation are not applicable to this work. Additionally, we have used generative AI tools to edit individual paragraphs in the paper to improve readability. We reviewed all AI-assisted work: all AI-edited text was reviewed by authors to confirm it accurately reflects our intended meaning and claims. We verified the plotted values against matching plots from wandb. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide complete definitions of PERC and its implementation. The appendix documents the model checkpoints, datasets, prompts, data splits, generation settings, optimisation hyperparameters, learning rates, gradient-clipping choices, training durations, and evaluation procedures used in each experiment. Unless otherwise stated, we report the 95% bootstrapped confidence interval score (1000 resamples). All matched comparisons use the same training and evaluation pipeline, differing only in the factors under investigation. The code can be found at https://anonymous.4open.science/r/expert_reasoning-C1E8/

ETHICS STATEMENT

Our study involves medical-expert annotation of model-generated reasoning traces. Annotators were asked to identify the first medically incorrect, unsupported, contradictory, or misleading reasoning unit in incorrect MEDREASON generations. They did not interact with deployed systems, patients, or private medical records; all annotated items were model outputs derived from benchmark questions. We collected only task annotations and did not report personally identifying information. We informed annotators about the study's purpose, and participation was voluntary. The results are intended to evaluate reasoning-error localisation, not to validate the generated medical reasoning for clinical use.

REFERENCES

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong,

Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645 (8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, November 2022. URL http://arxiv.org/abs/2211.14275. arXiv:2211.14275 [cs].

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step, May 2023. URL http://arxiv.org/abs/2305.20050. arXiv:2305.20050 [cs].

Amrith Setlur, Nived Rajaraman, Sergey Levine, and Aviral Kumar. Scaling Test-Time Compute Without Verification or RL is Suboptimal, February 2025. URL http://arxiv.org/abs/2502.12118. arXiv:2502.12118 [cs].

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An Open-Source LLM Reinforcement Learning System at Scale, May 2025. URL http://arxiv.org/abs/2503.14476. arXiv:2503.14476 [cs].

YuYue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu, Mingxuan Wang, Yonghui Wu, and Lin Yan. VAPO: Efficient and Reliable Reinforcement Learning for Advanced Reasoning Tasks, April 2025. URL http://arxiv.org/abs/2504.05118. arXiv:2504.05118 [cs] version: 1.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping Reasoning With Reasoning, May 2022. URL http://arxiv.org/abs/2203.14465. arXiv:2203.14465 [cs].

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling Relationship on Learning Mathematical Reasoning with Large Language Models, September 2023. URL http://arxiv.org/abs/2308.01825. arXiv:2308.01825 [cs].

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T. Parisi, Abhishek Kumar, Alexander A. Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura A. Culp, Lechao Xiao, Maxwell Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. *Transactions on Machine Learning Research*, January 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=lNAyUngGFK.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-STaR: Training Verifiers for Self-Taught Reasoners. August 2024. URL https://openreview.net/forum?id=stmqBSW2dV#discussion.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy P. Lillicrap, Fan Hui, L. Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–359, 2017. URL https://api.semanticscholar.org/CorpusID:205261034.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process Reinforcement through Implicit Rewards, February 2025. URL http://arxiv.org/abs/2502.01456. arXiv:2502.01456 [cs].

Shiping Gao, Hongzhan Chen, Xiaojun Quan, Qifan Wang, and Lifu Huang. Unleashing implicit rewards: Prefix-value learning for distribution-level optimization. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=372oAcqeKh.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, February 2023. URL http://arxiv.org/abs/1706.03741. arXiv:1706.03741.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model, July 2024. URL http://arxiv.org/abs/2305.18290. arXiv:2305.18290.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network, March 2015. URL http://arxiv.org/abs/1503.02531. arXiv:1503.02531 [stat].

Minki Kang, Seanie Lee, Jinheon Baek, Kenji Kawaguchi, and Sung Ju Hwang. Knowledge-Augmented Reasoning Distillation for Small Language Models in Knowledge-Intensive Tasks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 48573–48602. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/97faedc90260eae5c400f92d5831c3d7-Paper-Conference.pdf.

Kalle Kujanpää, Pekka Marttinen, Harri Valpola, and Alexander Ilin. Efficient Knowledge Injection in LLMs via Self-Distillation, August 2025. URL http://arxiv.org/abs/2412.14964. arXiv:2412.14964 [cs].

Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. KDRL: Post-Training Reasoning LLMs via Unified Knowledge Distillation and Reinforcement Learning, June 2025. URL http://arxiv.org/abs/2506.02208. arXiv:2506.02208 [cs].

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. In *ICLR 2026 Workshop on Lifelong Agents: Learning, Aligning, Evolving*, 2026. URL https://openreview.net/forum?id=HlWA3V6iKF.

Hao Sun and Mihaela van der Schaar. Inverse-RLignment: Large Language Model Alignment from Demonstrations through Inverse Reinforcement Learning, January 2025. URL http://arxiv.org/abs/2405.15624. arXiv:2405.15624 [cs].

Michał Wiliński, Liu Leqi, and Chirag Nagpal. Inverse rl helps align ai by imitating humans, 2026. URL https://arxiv.org/abs/2607.24900.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 2025. URL https://arxiv.org/abs/2511.10643.

Byung-Kwan Lee, Ryo Hachiuma, Yong Man Ro, Yu-Chiang Frank Wang, and Yueh-Hua Wu. Unified reinforcement and imitation learning for vision-language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=7wEvjzkNXg.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, April 2024. URL http://arxiv.org/abs/2402.03300. arXiv:2402.03300 [cs].

Edoardo Cetin, Tianyu Zhao, and Yujin Tang. Reinforcement learning teachers of test time scaling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=tebG8q5EeK.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017. URL http://arxiv.org/abs/1707.06347. arXiv:1707.06347 [cs].

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, November 2021. URL http://arxiv.org/abs/2110.14168. arXiv:2110.14168 [cs].

Juncheng Wu, Wenlong Deng, Xingxuan Li, Sheng Liu, Taomian Mi, Yifan Peng, Ziyang Xu, Yi Liu, Hyunjin Cho, Chang-In Choi, Yihan Cao, Hui Ren, Xiang Li, Xiaoxiao Li, and Yuyin Zhou. Medreason: Eliciting factual medical reasoning steps in llms via knowledge graphs, 2025. URL https://arxiv.org/abs/2504.00993.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen Technical Report, September 2023. URL http://arxiv.org/abs/2309.16609. arXiv:2309.16609 [cs].

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models, February 2023. URL http://arxiv.org/abs/2302.13971. arXiv:2302.13971 [cs].

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

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann, editors, *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR, 07–08 Apr 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's Transformers: State-of-the-art Natural Language Processing, July 2020. URL http://arxiv.org/abs/1910.03771. arXiv:1910.03771 [cs].

Michael Han Daniel Han and Unsloth team. Unsloth, 2023. URL http://github.com/unslothai/unsloth.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

A Additional Method Details

A.1 Training a Prefix Expert-Referenced Critic

Several considerations arise when training this critic for LLMs. First, trajectories are discrete autoregressive token sequences, so prefix-level discriminator scores must be converted into stable rewards for policy-gradient updates. Second, a naive expert-vs-policy discriminator would mainly learn source or formatting cues; PERC therefore uses answer agreement and corrupted expert traces to construct weak positive and negative reasoning evidence. Third, we must explicitly control reward granularity: sparse, interval, and dense masks trade off optimisation stability against process-level feedback. Finally, adversarial training is highly non-stationary for LLM policies, so we use length normalisation and reward clipping.

A.2 Algorithm

Algorithm 1 Prefix-level Expert-Referenced Critic (PERC)

Require: Expert traces $\mathcal{D}_E$; initial policy π_0; critic steps N_{critic}; policy steps N_{step}

Initialise policy π_θ and prefix critic D_ϕ
Critic Training:
for $j \leftarrow 1$ to N_{critic} **do**
  Sample prompts $x \sim \mathcal{Q}$ and generate $\mathcal{D}_P \sim \pi_{\theta_{\text{init}}}(\cdot \mid x)$
  $\mathcal{D}_{\text{pos}} \leftarrow \mathcal{D}_E \cup \{y \in \mathcal{D}_P \mid \mathcal{O}(y) = \mathcal{O}(y^E)\}$
  $\mathcal{D}_{\text{neg}} \leftarrow \{y \in \mathcal{D}_P \mid \mathcal{O}(y) \neq \mathcal{O}(y^E)\} \cup \{\mathcal{C}(y^E) \mid y^E \in \mathcal{D}_E\}$
  Update D_ϕ to minimise Eq. (3) using $\mathcal{D}_{\text{pos}}$ and $\mathcal{D}_{\text{neg}}$
end for
Policy Optimisation with Frozen Critic:
Freeze $D_{\hat{\phi}}$
for $i \leftarrow 1$ to N_{step} **do**
  Sample prompts $x \sim \mathcal{Q}$
  Generate group $\{y^{(g)}\}_{g=1}^{G} \sim \pi_\theta(\cdot \mid x)$
  Compute critic-derived rewards $r_{\hat{\phi}}(y_t^{(g)})$ via backfilling logits (Eq. 4 and Eq. 6)
  Compute sequence-level advantages $A^{(g)}$ via group standardisation (Eq. 5)
  Optimise π_θ using GRPO loss on policy generations
end for

A.3 Critic and Policy Optimisation Details

Since PERC uses a frozen critic during policy optimisation, instability mainly comes from the scale of critic-derived rewards rather than from online adversarial updates. We therefore clip token rewards:

$$r_\phi(y_t) \leftarrow \text{clip}(r_\phi(y_t), -\beta, \beta), \quad \text{where} \quad \beta > 0 \tag{6}$$

where $\beta > 0$. This bounds extreme logit-derived rewards before sequence-level GRPO aggregation. Together, these mechanisms better calibrate the reward signal and reduce mode-collapse-like behaviour in adversarial training.

A.4 Medical Data Annotation

For the clinician localisation study, we constructed an offline annotation interface for natural incorrect `Qwen2.5-7B` SFT generations on MedReason. Each annotator was assigned 25 examples and was shown the question, answer options, the model's generated reasoning trace segmented into numbered reasoning units, the model's final answer, and the correct answer. Three final-year medical students and two doctors independently annotated separate batches. For each example, annotators selected the first reasoning unit where the trace became medically incorrect, unsupported, contradictory, or misleading. They could alternatively mark "no clear localisable error" or indicate that the reasoning appeared correct but only the final answer was wrong. The annotation interface saved the selected

unit, optional confidence and notes, and exported the labels as JSONL/CSV; model reward scores and predicted localisation positions were not shown to annotators. Figure 4 shows a screenshot of the annotation interface.

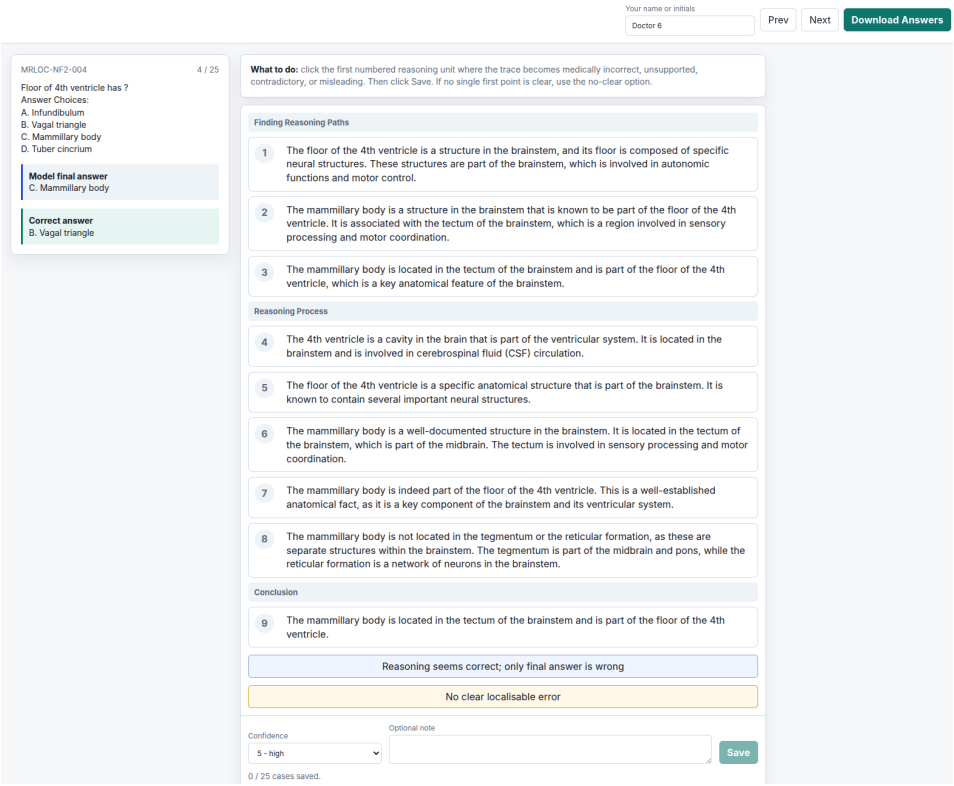


Figure 4: Medical Data Annotator Interface.

B IMPLEMENTATION DETAILS

We evaluate the proposed expert reasoning approach on GSM8K (Cobbe et al., 2021), a benchmark for grade school math problems that provides final answers and human-written demonstrations. Moreover, to demonstrate the effectiveness of the proposed method in extracting a dense reasoning reward model, we performed our experiments on MEDREASON (Wu et al., 2025), and more specifically on the MEDQA (Jin et al., 2021) subset, consisting of questions from the US medical board exam, and THE MEDMCQA (Pal et al., 2022) subset, comprising questions from the entrance exam from the Indian medical school curriculum. Our experimental dataset comprises approximately 7'000 questions for training and 1'000 for evaluation. In addition, Wu et al. (2025) provides quality-filtered medical reasoning traces constructed by a strong language model (ChatGPT), which we can use for supervised fine-tuning or, in our case, PERC training. The MMLU-PRO dataset (Wang et al., 2024) is an advanced version of the original Massive Multi-Task Language Understanding (MMLU) dataset, comprising questions that are particularly difficult to reason about across domains such as maths, biology, health, business, law, etc. Moreover, we expanded the answer options from 4 to 10 to increase difficulty. This dataset provides additional insight into more challenging and open-domain reasoning tasks.

Unless otherwise noted, we use open-weight, instruction-tuned models as base policies and train a prefix critic from expert-referenced trace labels. To obtain a dense signal, we implement the critic as a token classifier that shares the backbone with a language model and replaces the language modelling head with a single linear layer that outputs one logit per token. The code for all our experiments can be found inhttps://anonymous.4open.science/r/expert_reasoning-C1E8.

All experiments are implemented in Python (Van Rossum and Drake Jr, 1995) with PyTorch (Paszke et al., 2017) and Hugging Face Transformers (Wolf et al., 2020). We accelerate training and evaluation with UNSLOTH (Daniel Han and team, 2023). Unless stated otherwise, we use a starting learning rate of 1×10^{-5} for the reasoning discriminator and 5×10^{-6} for the generator. We train the prefix critic for 250 optimisation steps, then keep it fixed during the 400-step GRPO policy-optimisation runs with a batch size of 16, generating $G = 8$ samples per prompt, accumulated over eight gradient steps (actual batch size + 128). To prevent data drift, we implemented a replay buffer that keeps track of the 50 most recent batches. Each discriminator mini-batch is class-balanced between positive and negative traces. Corrupted expert traces are generated at a 1:1 rate with clean expert traces, and the reward model samples a balanced combination from the replay buffers. Both the discriminator and policy optimisers use a cosine-annealing learning rate schedule. The discriminator optimiser has a warm-up sequence of 250 steps (the same as the discriminator warm-up period), while the policy optimiser warms up for 50 steps. We use a quantised ADAMW (Loshchilov and Hutter, 2019) optimiser.

Data and preprocessing. Prompts consist of the problem text and a short system instruction requesting step-by-step reasoning. Demonstrations are formatted as `<think> ... </think>` followed by `<answer> ... </answer>` and provide the system instructions below. Tokenisation uses each backbone's native tokeniser. For evaluation, we decode with temperature $T = 0.5$ and `top_p` $= 0.95$. We validate every run after 100 steps, save the model with the best validation accuracy on correctness for all baselines and models, and use these weights to evaluate on the test set. For answer agreement, we extract the content of the final `<answer>` field, normalise the resulting string, and compare it to the paired demonstration answer by exact string equality. Outputs without an extractable final answer are treated as non-agreeing.

System Prompt for GSM8K and MMLU-PRO

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

System Prompt for MEDREASON

```
You are an expert medical AI. You must analyze clinical scenarios and
adhere strictly to this output format:

1. Enclose your step-by-step reasoning within `<think>` and
   `</think>` tags.
   Keep your reasoning strictly <500 words.
2. Immediately after, output your final conclusion within `<answer>`
   and `</answer>`  tags.
  The content must be exactly in the format: `<LETTER>. <ANSWER_TEXT>`,
  where:
   - `<LETTER>` is one of A, B, C, D
   - `<ANSWER_TEXT>` is the exact option text corresponding
     to that letter

Example:
<think>
[Your step-by-step clinical reasoning]
</think>
<answer>
C. Hyperthyroidism
</answer>
"""
```

Inference time scoring. At inference time, we draw $N = 16$ samples per prompt, compute the mean polynomially discounted ($\gamma = 0.95$) critic-derived score over the reasoning tokens for each sample, and rerank by this score. We introduce this discount to reward tokens towards the end of a sequence, as they are closer to completion.

Perturbations. To improve robustness and reduce reliance on source or formatting cues, we construct paired counterfactual hard negatives rather than generic augmentation. Each corrupted trace starts from an expert trace and preserves the prompt, formatting, source, and most content, but minimally changes one reasoning-relevant property. For GSM8K, we use local arithmetic interventions: operator flips, numeric-literal offsets, and replacing the answer with an earlier intermediate number. These include both local reasoning errors that can preserve the final answer field and answer-changing corruptions. Perturbed traces are labelled as negative examples. For MEDREASON and MMLU-PRO, we generate plausible but incorrect reasoning traces toward distractor answers by providing the correct reasoning trace and answer in context and instructing the model to corrupt them. We provide the prompt below:

Compute. We run experiments on 4 A100-class GPUs using mixed-precision training. We use gradient accumulation to match effective batch sizes across the backbones. All models operate in 4-bit mode, as provided by UNSLOTH (Daniel Han and team, 2023), to improve training speed and memory efficiency. All experiments are run on a single GPU instance to enable parallelisation. Table 7 reports representative training costs for the `Qwen2.5-7B` setting. PERC is more expensive than SFT because it trains a separate prefix critic before policy optimisation. The cost is comparable to source-based GAD, but the resulting reward is reused for training, reranking, and localisation.

Method / run	GPU-hours	vs. SFT
SFT	4.5	1.0x
GAD	14.0	3.1x
PERC *interval*	17.5	3.9x
PERC *dense*	16.5	3.7x

Table 7: **Representative training cost on `Qwen2.5-7B`.** GPU-hours are measured on a single NVIDIA A100 for 400 training steps and rounded to the nearest 0.5 hours. PERC costs include prefix-critic training followed by policy optimisation with the critic.

B.1 Policy and Reward critic

We initialise policies from instruction-tuned checkpoints and, for policy optimisation experiments, train them with the learned critic-derived reward signal. The following policy backbones are used:

- `Llama3.1-8B-Instruct`
- `Llama3.2-3B-Instruct`
- `Qwen2.5-3B-Instruct`
- `Qwen2.5-7B-Instruct`
- `Qwen3-4B-Instruct-2507`

We instantiate the prefix critics using the same backbone families as the policy models. All models are turned into LoRA Hu et al. (2021) adapters ($r = 256, \alpha = 512$) and PEFT is used.

System Prompt for Corrupted Dataset

```
"""
You are an expert educator creating adversarial training examples.
Below is a question, the correct reasoning, and the correct answer.

Your task:
1. Identify the incorrect options (distractors) provided in the
   question.
2. For EACH incorrect option, write a plausible reasoning trace that
   leads to that specific wrong answer.
3. The reasoning should sound logical but must conclude with the
   wrong answer.
4. Match the formatting of the correct reasoning provided.

Format your output as sequential blocks:

<block>
<think>
[Reasoning for Distractor]
</think>
<answer>
[Distractor Label and Text]
</answer>
</block>

---
Question:
{question}

Correct Reasoning (do not use):
{correct_reasoning}

Correct Answer (do not use):
{correct_answer}
"""
```

B.2 Future Work

Our findings suggest that learned prefix critics can serve as reusable signals for reranking, localisation, and aggregate policy optimisation. Addressing the stability–granularity trade-off is critical; this involves improving the interface between prefix scores and token-level policy updates, for example through returns-to-go or value-difference rewards. Moreover, dense reward-token-level interpretability opens the door to active test-time intervention, such as reward-guided decoding, early-exit mechanisms, and iterative self-revision strategies. A stronger extension would add span-labelled contrastive objectives, forcing the critic to separate an expert trace from its corruption specifically at

the affected reasoning span. We leave this objective, together with richer non-arithmetic interventions such as unsupported inferences and explicit contradictions, to future work.

C Additional Experiments

C.1 Ablation Reward Clipping

RQ-A1: *How sensitive is PERC policy optimisation to reward clipping bounds?*

Experimental setup. We ablate the clipping interval $\beta = (\mathrm{LB}, \mathrm{UB})$ for `Llama3.2-3B` with step-wise rewards on GSM8K, while keeping the rest of the policy-optimisation recipe fixed. We report the held-out Pass@1, the reranking gain (ΔRerank), AUROC, and ECE.

β (LB, UB)	Pass@1	Δ Rerank	AUROC ↑	ECE ↓
(0, 2)	54 [52, 56]	**+14**	86 [85, 86]	21 [20, 22]
(-1, 1)	57 [55, 59]	**+11**	84 [83, 84]	15 [14, 15]
(-3, 3)	54 [52, 56]	**+14**	89 [88, 89]	13 [12, 13]
(-5, 5)	58 [56, 60]	**+11**	83 [82, 83]	20 [20, 21]
(-10, 10)	52 [51, 54]	**+17**	87 [87, 88]	18 [17, 18]
$(-\infty, \infty)$	58 [56, 60]	**+12**	87 [87, 88]	12 [12, 13]

Table 8: **Ablation on reward clipping** (β) for `Llama3.2-3B` (Step-wise) on GSM8K. All metrics are shown in full percentages (%). Intervals indicate 95% confidence intervals.

Takeaway. For PERC policy optimisation, reward clipping is a secondary design choice rather than a primary driver of performance.

C.2 Ablation Groups

RQ-A2: *How sensitive is performance to the number of sampled generations per prompt (g) during policy optimisation and reranking?*

Experimental setup. We vary the group size $g \in \{4, 8, 16\}$ for `Llama3.2-3B` with step-wise rewards on GSM8K. As above, we report Pass@1, Δ Rerank, AUROC, and ECE to capture both downstream accuracy and reward quality.

Gen. (g)	Pass@1	Δ Rerank	AUROC ↑	ECE ↓
4	53 [52, 55]	**+18**	89 [89, 90]	13 [13, 14]
8	58 [56, 60]	**+11**	83 [82, 83]	20 [20, 21]
16	57 [56, 59]	**+13**	86 [86, 87]	14 [13, 14]

Table 9: **Ablation on number of generations** (G) for `Llama3.2-3B` (Interval) on GSM8K. All metrics are in full percentages (%). Intervals indicate 95% confidence intervals.

Takeaway. The method is not brittle to the exact group size, suggesting that its gains are not tied to a narrowly tuned sampling configuration.

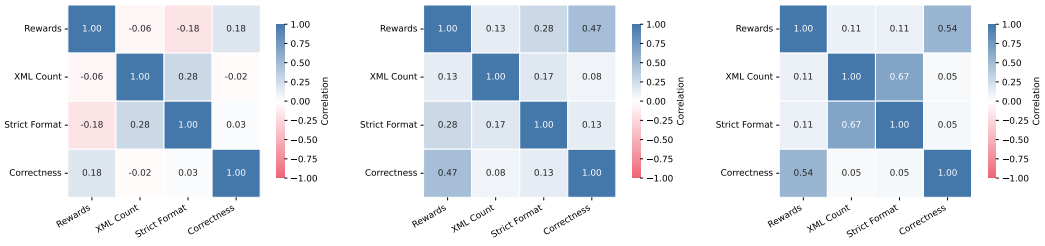
C.3 Ablation Corruptions

> **RQ-A4:** *Do answer agreement and targeted corruptions reduce style shortcuts and improve reasoning-discriminative critic scores?*

Experimental setup. We run an ablation on `Qwen2.5-3B` (interval) on GSM8K. *No Answer Agreement* uses only source labels: expert traces are positive and policy rollouts are negative, with no relabelling by final-answer agreement. *Answer Agreement* uses the construction from Section 4, moving answer-matching policy rollouts into $\mathcal{D}_{\text{pos}}$ and mismatched rollouts into $\mathcal{D}_{\text{neg}}$. The corruption rows keep answer agreement fixed and add one paired counterfactual expert-trace corruption to $\mathcal{D}_{\text{neg}}$: numeric-literal offsets, answer replacement with an intermediate number, or arithmetic-operator flips. These examples preserve expert provenance and formatting, so gains cannot be explained by source discrimination alone. Corrupted expert traces are generated at a 1:1 rate with clean expert traces, and critic mini-batches are class-balanced. We report Pass@1, Δ Rerank, AUROC, and ECE (Table 10), and analyse critic-score correlation with correctness versus format cues (Figure 5).

Corruption Type	Pass@1	Δ Rerank	AUROC ↑	ECE ↓
No Answer Agreement	31 [29, 32]	**+4**	62 [61, 62]	8 [7, 8]
Answer Agreement	56 [55, 58]	**+8**	77 [76, 78]	22 [22, 23]
+ Number corruption	58 [56, 59]	**+10**	78 [77, 78]	34 [34, 35]
+ Number swapping	56 [54, 58]	**+15**	84 [83, 84]	27 [27, 28]
+ Symbol corruption	56 [55, 58]	**+18**	85 [84, 85]	31 [31, 32]

Table 10: **Ablation on Reasoning Corruption** for `Qwen2.5-3B` (interval) on GSM8K. All metrics are in full percentages (%). Intervals indicate 95% confidence intervals.



(a) No Answer Agreement (b) Answer Agreement only (c) Answer Agr. + Full Corruption

Figure 5: **Correlation of critic scores with correctness and format.** `Qwen2.5-3B` on GSM8K under increasing corruption (no answer agreement → answer agreement → answer agreement + full corruption). Correctness correlation increases (0.18 → 0.47 → 0.54), while format correlation is reduced in the fully corrupted setting (0.28 → 0.11).

Takeaway. Answer agreement is key to learning critic scores that track reasoning quality rather than superficial style cues. Corruptions can make this further robust.

D ADDITIONAL RESULTS

D.1 RERANKING CALIBRATION

		GSM8K		MEDREASON		MMLU-PRO	
Backbone	**Method**	AUROC (%) ↑	ECE (%) ↓	AUROC (%) ↑	ECE (%) ↓	AUROC (%) ↑	ECE (%) ↓
`Qwen2.5-7B`	*Interval*	70.1 [69.1, 71.0]	7.5 [7.0, 8.0]	71.0 [70.2, 71.8]	9.7 [9.0, 10.4]	68.0 [67.3, 68.7]	35.6 [35.0, 36.2]
	Dense	71.4 [70.7, 72.1]	45.2 [44.6, 45.9]	70.4 [69.5, 71.2]	15.5 [14.8, 16.2]	60.2 [59.5, 60.9]	41.3 [40.7, 41.9]
`Llama3.1-8B`	*Interval*	81.8 [81.2, 82.5]	8.7 [8.2, 9.2]	85.1 [84.5, 85.7]	6.5 [6.0, 7.1]	70.8 [70.1, 71.5]	39.7 [39.2, 40.2]
	Dense	75.9 [75.1, 76.6]	27.2 [26.6, 27.8]	86.0 [85.4, 86.6]	10.9 [10.3, 11.5]	60.2 [59.4, 60.9]	50.9 [50.4, 51.5]
`Qwen3-4B`	*Interval*	78.3 [77.1, 79.5]	13.5 [13.2, 13.9]	74.8 [73.9, 75.6]	3.2 [2.6, 3.9]	84.2 [83.7, 84.7]	21.5 [21.0, 22.0]
	Dense	78.3 [77.2, 79.4]	7.3 [7.0, 7.7]	73.3 [72.5, 74.1]	8.1 [7.4, 8.8]	75.0 [74.4, 75.7]	18.6 [18.0, 19.2]

Table 11: **Critic Calibration Metrics.** All values are reported as percentages (%). AUROC indicates ranking ability; ECE measures calibration error (lower is better).

D.2 RERANKING PERFORMANCE ABLATION

		GSM8K					MMLU-PRO				
Backbone	**Method**	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.
`Qwen3-4B`	*Interval*	76.6%	(↓ -0.5)	(0.0)	(↑ **+4.0**)	(↑ **+4.0**)	53.9%	(↑ **+2.7**)	(↑ +0.5)	(↑ +0.9)	(↑ +0.9)
	Dense	76.6%	(↓ -0.5)	(0.0)	(↑ **+3.6**)	(↑ **+3.6**)	53.9%	(↑ **+2.7**)	(↑ +0.5)	(↑ +1.9)	(↑ +1.9)
`Qwen2.5-7B`	*Interval*	70.1%	(↑ +0.4)	(↑ +0.3)	(↑ **+5.4**)	(↑ **+5.4**)	48.1%	(↑ +0.9)	(↓ -0.6)	(↑ **+1.1**)	(↑ **+1.1**)
	Dense	70.1%	(↑ +0.4)	(↑ +0.3)	(↑ **+6.4**)	(↑ **+6.4**)	48.1%	(↑ **+0.9**)	(↓ -0.6)	(↓ -0.1)	(↓ -0.1)
`Llama3.1-8B`	*Interval*	66.6%	(↑ +1.4)	(↑ +0.2)	(↑ **+3.1**)	(↑ **+3.1**)	47.2%	(↑ +1.7)	(↑ +0.6)	(↑ **+3.7**)	(↑ **+3.7**)
	Dense	66.6%	(↑ +1.4)	(↑ +0.2)	(↑ **+3.0**)	(↑ **+3.0**)	47.2%	(↑ +1.7)	(↑ +0.6)	(↑ **+2.0**)	(↑ **+2.0**)

Table 12: **Best-of-2 Reranking Performance & Baselines (%).** Random is SFT pass@1. Deltas are percentage-point changes for each reranker. Blue arrows (↑) indicate gains and purple arrows (↓) indicate drops.

		GSM8K					MMLU-PRO				
Backbone	**Method**	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.
`Qwen3-4B`	*Interval*	76.6%	(↓ -1.0)	(↑ +3.4)	(↑ +5.2)	(↑ **+6.3**)	53.9%	(↑ **+3.7**)	(↑ +3.5)	(↑ +2.1)	(↑ +3.3)
	Dense	76.6%	(↓ -1.0)	(↑ +3.4)	(↑ +5.2)	(↑ **+6.1**)	53.9%	(↑ +3.7)	(↑ +3.5)	(↑ +3.7)	(↑ **+4.3**)
`Qwen2.5-7B`	*Interval*	70.1%	(↑ +1.7)	(↑ +7.5)	(↑ +6.4)	(↑ **+8.5**)	48.1%	(↑ +1.9)	(↑ +2.1)	(↑ +2.4)	(↑ **+3.4**)
	Dense	70.1%	(↑ +1.7)	(↑ +7.5)	(↑ +9.4)	(↑ **+10.1**)	48.1%	(↑ +1.9)	(↑ +2.1)	(↑ +0.2)	(↑ **+2.5**)
`Llama3.1-8B`	*Interval*	66.6%	(↑ +2.3)	(↑ +1.8)	(↑ +3.9)	(↑ **+4.4**)	47.2%	(↑ +2.0)	(↑ +4.0)	(↑ +4.0)	(↑ **+4.8**)
	Dense	66.6%	(↑ +2.3)	(↑ +1.8)	(↑ **+3.7**)	(↑ +3.6)	47.2%	(↑ +2.0)	(↑ +4.0)	(↑ +2.8)	(↑ **+4.4**)

Table 13: **Best-of-3 Reranking Performance & Baselines (%).** Random is SFT pass@1. Deltas are percentage-point changes for each reranker. Blue arrows (↑) indicate gains and purple arrows (↓) indicate drops.

		GSM8K					MMLU-PRO				
Backbone	**Method**	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.
`Qwen3-4B`	*Interval*	76.6%	(↓ -0.9)	(↑ +5.5)	(↑ +6.7)	(↑ **+7.9**)	53.9%	(↑ +2.8)	(↑ **+5.7**)	(↑ +2.1)	(↑ +4.8)
	Dense	76.6%	(↓ -0.9)	(↑ +5.5)	(↑ +6.4)	(↑ **+8.1**)	53.9%	(↑ +2.8)	(↑ **+5.7**)	(↑ +3.5)	(↑ +5.2)
`Qwen2.5-7B`	*Interval*	70.1%	(↑ +2.5)	(↑ +10.9)	(↑ +6.8)	(↑ **+11.7**)	48.1%	(↑ +3.5)	(↑ **+4.9**)	(↑ +2.1)	(↑ +4.8)
	Dense	70.1%	(↑ +2.5)	(↑ +10.9)	(↑ +10.2)	(↑ **+12.4**)	48.1%	(↑ +3.5)	(↑ +4.9)	(↑ +1.0)	(↑ **+5.3**)
`Llama3.1-8B`	*Interval*	66.6%	(↑ +2.5)	(↑ +3.0)	(↑ +4.5)	(↑ **+4.6**)	47.2%	(↑ +3.0)	(↑ +6.0)	(↑ +4.0)	(↑ **+6.6**)
	Dense	66.6%	(↑ +2.5)	(↑ +3.0)	(↑ **+4.8**)	(↑ +3.7)	47.2%	(↑ +3.0)	(↑ +6.0)	(↑ +3.0)	(↑ **+6.3**)

Table 14: **Best-of-5 Reranking Performance & Baselines (%).** Random is SFT pass@1. Deltas are percentage-point changes for each reranker. Blue arrows (↑) indicate gains and purple arrows (↓) indicate drops.

D.3 GAD RERANKING

Backbone	Method	GSM8K Random	GSM8K Δ Logp.	GSM8K Δ Maj.	GSM8K Δ Rew.	GSM8K Δ W.Maj.	MMLU-Pro Random	MMLU-Pro Δ Logp.	MMLU-Pro Δ Maj.	MMLU-Pro Δ Rew.	MMLU-Pro Δ W.Maj.
`Qwen3-4B`	*Interval*	76.6%	(↑ +0.1)	(↑ +8.3)	(↑ +7.0)	(↑ **+8.9**)	53.9%	(↑ +2.4)	(↑ **+6.0**)	(↑ +1.9)	(↑ +5.8)
	Dense	76.6%	(↑ +0.1)	(↑ +8.3)	(↑ +7.7)	(↑ **+9.1**)	53.9%	(↑ +2.4)	(↑ +6.0)	(↑ +4.6)	(↑ **+6.5**)
`Qwen2.5-7B`	*Interval*	70.1%	(↑ +1.3)	(↑ +13.0)	(↑ +8.7)	(↑ **+13.7**)	48.1%	(↑ +2.1)	(↑ +5.7)	(↑ +3.2)	(↑ **+6.2**)
	Dense	70.1%	(↑ +1.3)	(↑ +13.0)	(↑ +11.0)	(↑ **+14.4**)	48.1%	(↑ +2.1)	(↑ +5.7)	(↑ +0.9)	(↑ **+6.2**)
`Llama3.1-8B`	*Interval*	66.6%	(↑ +2.4)	(↑ +2.7)	(↑ **+5.3**)	(↑ +4.8)	47.2%	(↑ +2.5)	(↑ +7.0)	(↑ +4.1)	(↑ **+7.8**)
	Dense	66.6%	(↑ +2.4)	(↑ +2.7)	(↑ **+5.8**)	(↑ +4.1)	47.2%	(↑ +2.5)	(↑ +7.0)	(↑ +3.7)	(↑ **+7.4**)

Table 15: **Best-of-8 Reranking Performance & Baselines (%).** Random is SFT pass@1. Deltas are percentage-point changes for each reranker. Blue arrows (↑) indicate gains and purple arrows (↓) indicate drops.

Backbone	Method	GSM8K Random	GSM8K Δ Logp.	GSM8K Δ Maj.	GSM8K Δ Rew.	GSM8K Δ W.Maj.	MMLU-Pro Random	MMLU-Pro Δ Logp.	MMLU-Pro Δ Maj.	MMLU-Pro Δ Rew.	MMLU-Pro Δ W.Maj.
`Qwen3-4B`	*Interval*	76.6%	(↑ +1.4)	(↑ +8.7)	(↑ +8.1)	(↑ **+10.2**)	53.9%	(↑ +2.1)	(↑ **+7.7**)	(↑ +2.2)	(↑ +7.5)
	Dense	76.6%	(↑ +1.4)	(↑ +8.7)	(↑ +9.2)	(↑ **+10.6**)	53.9%	(↑ +2.1)	(↑ **+7.7**)	(↑ +4.7)	(↑ +7.5)
`Qwen2.5-7B`	*Interval*	70.1%	(↑ +0.8)	(↑ **+15.6**)	(↑ +8.3)	(↑ +15.3)	48.1%	(↑ +2.6)	(↑ +7.5)	(↑ +4.1)	(↑ **+8.1**)
	Dense	70.1%	(↑ +0.8)	(↑ +15.6)	(↑ +11.1)	(↑ **+15.7**)	48.1%	(↑ +2.6)	(↑ +7.5)	(↑ +0.5)	(↑ **+7.7**)
`Llama3.1-8B`	*Interval*	66.6%	(↑ +2.8)	(↑ +4.6)	(↑ +5.9)	(↑ **+6.8**)	47.2%	(↑ +2.7)	(↑ +7.6)	(↑ +4.4)	(↑ **+8.4**)
	Dense	66.6%	(↑ +2.8)	(↑ +4.6)	(↑ **+6.5**)	(↑ +5.8)	47.2%	(↑ +2.7)	(↑ +7.6)	(↑ +3.2)	(↑ **+7.9**)

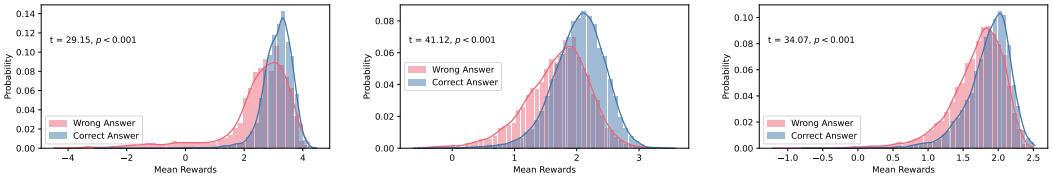
Table 16: **Best-of-16 Reranking Performance & Baselines (%).** Random is SFT pass@1. Deltas are percentage-point changes for each reranker. Blue arrows (↑) indicate gains and purple arrows (↓) indicate drops.

Method	GSM8K	MMLU-Pro	MedReason	Avg. Δ
GAD	−0.5	+1.4	+0.0	+0.3
PERC Interval	+2.0	+2.5	+0.9	+1.8
PERC Dense	+2.5	+2.2	+1.5	+2.1

Table 17: **GAD reranking comparison on `Qwen2.5-7B`.** Values are Best-of-16 percentage-point gains over random selection from the same sample set.

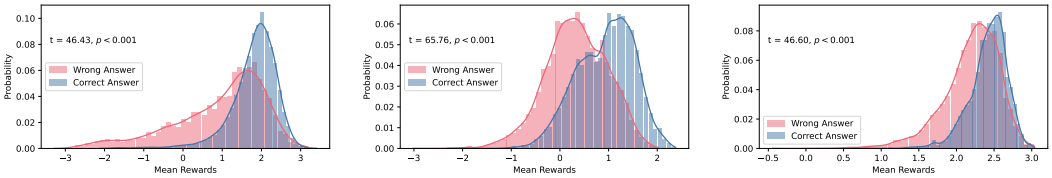
D.4 Distribution Separation of Critic Scores

D.4.1 GSM8K



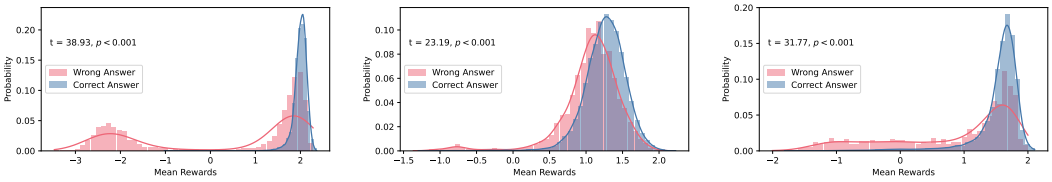
(a) Sparse (b) Interval (c) Dense

Figure 6: **Reward Distribution by Correctness (`Qwen2.5-7B` on GSM8K)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.



(a) Sparse (b) Interval (c) Dense

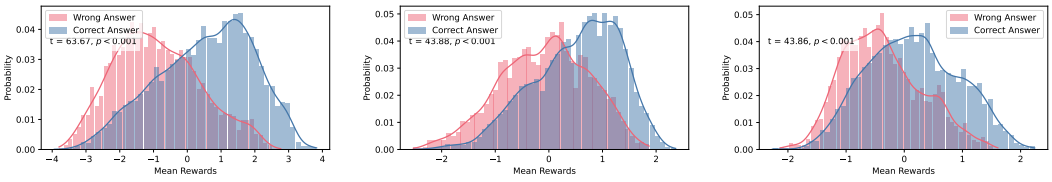
Figure 7: **Reward Distribution by Correctness (`Llama3.1-8B` on GSM8K)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.



(a) Sparse (b) Interval (c) Dense

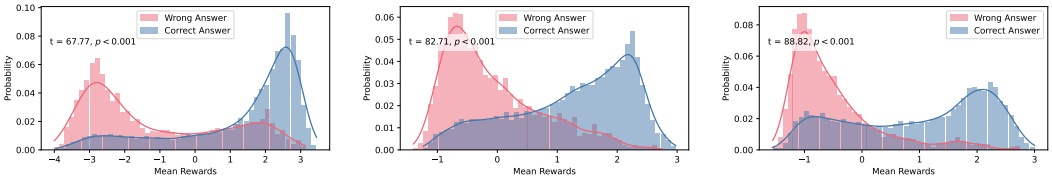
Figure 8: **Reward Distribution by Correctness (`Qwen3-4B` on GSM8K)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

D.4.2 MMLU-PRO



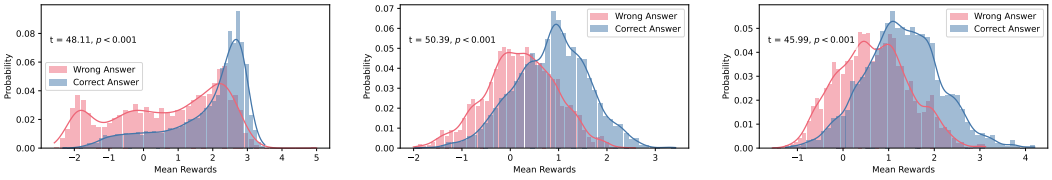
(a) Sparse (b) Interval (c) Dense

Figure 9: **Reward Distribution by Correctness (`Qwen2.5-7B` on MMLU-PRO)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.



(a) Sparse (b) Interval (c) Dense

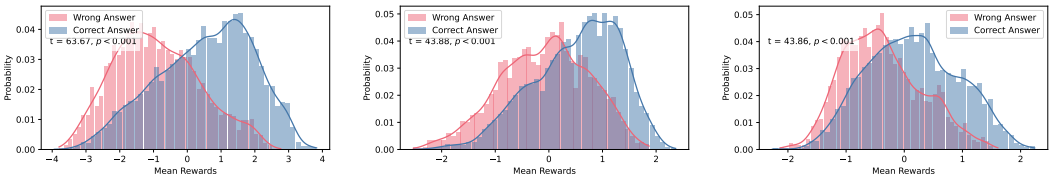
Figure 10: **Reward Distribution by Correctness (`Llama3.1-8B` on MMLU-PRO)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.



(a) Sparse (b) Interval (c) Dense

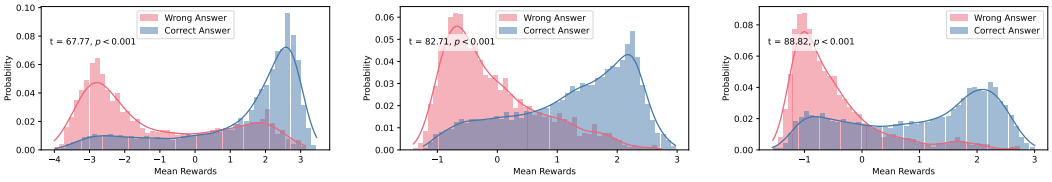
Figure 11: **Reward Distribution by Correctness (`Qwen3-4B` on MMLU-PRO)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

D.4.3 MEDREASON



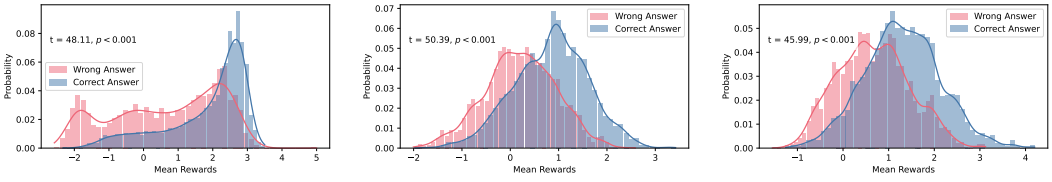
(a) Sparse (b) Interval (c) Dense

Figure 12: **Reward Distribution by Correctness (`Qwen2.5-7B` on MEDREASON).** A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.



(a) Sparse (b) Interval (c) Dense

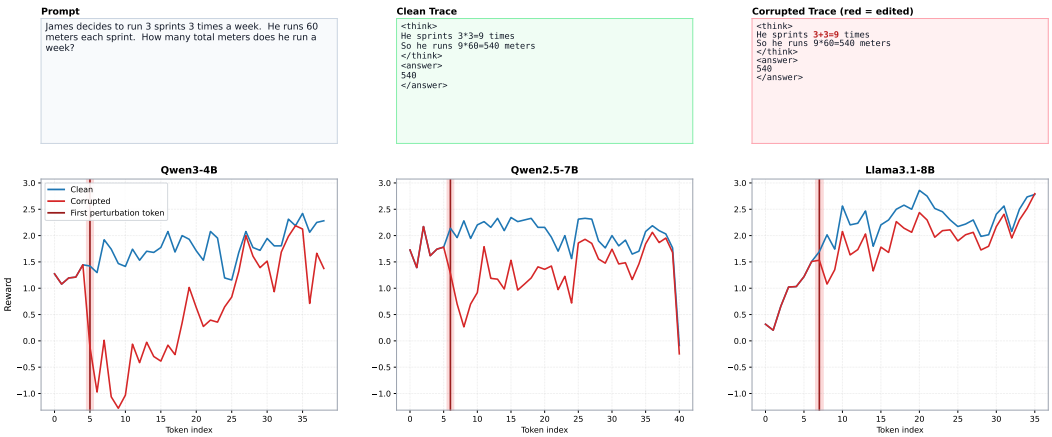
Figure 13: **Reward Distribution by Correctness (`Llama3.1-8B` on MEDREASON).** A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.



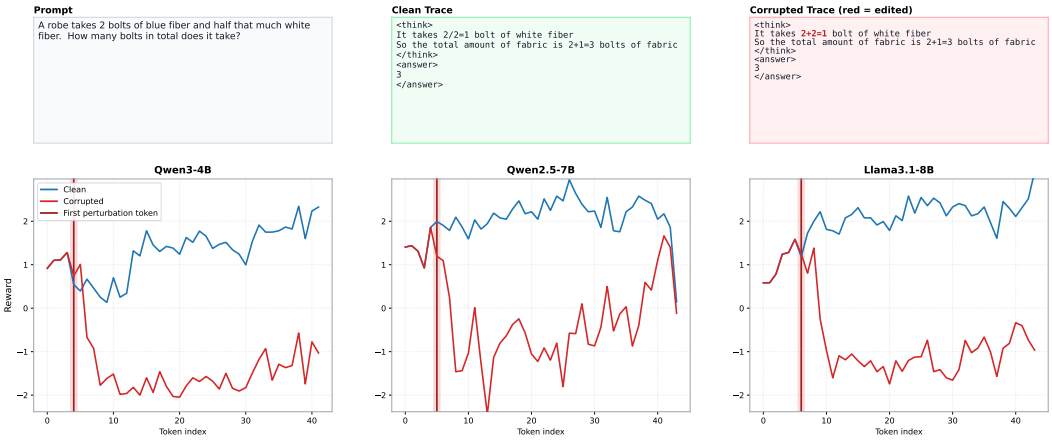
(a) Sparse (b) Interval (c) Dense

Figure 14: **Reward Distribution by Correctness (`Qwen3-4B` on MEDREASON).** A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

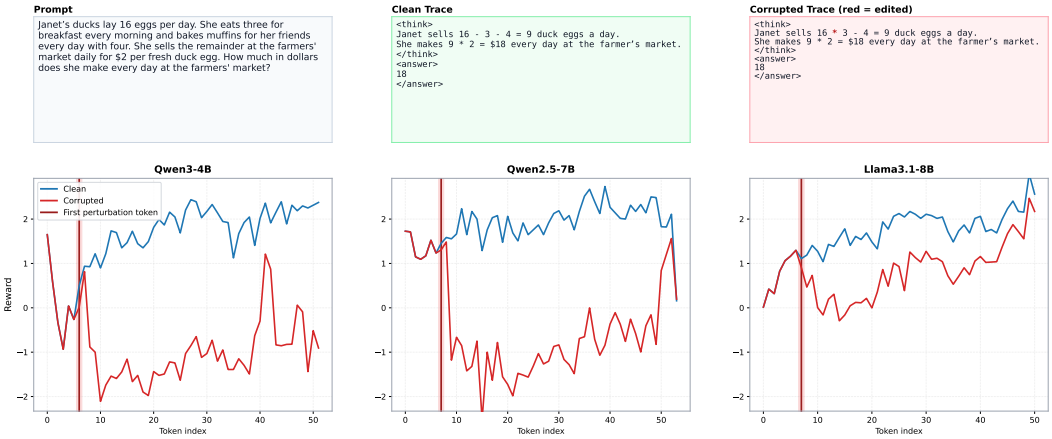
D.5 Error Localisation



(a) Example 1



(b) Example 2

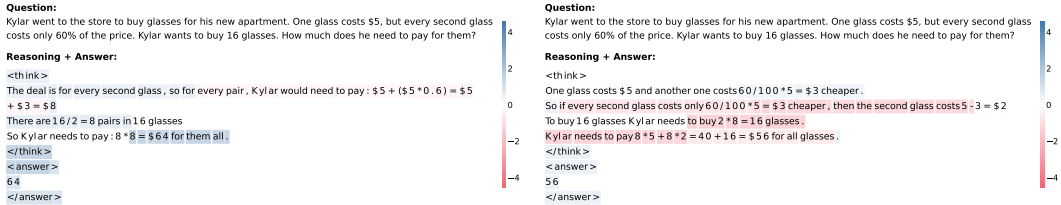


(c) Example 3

Figure 15: **Error Localisation on GSM8K** Dense reward on correct and incorrect generations using the *dense* reasoning reward models. We clearly observe a divergence between the correct and incorrect paths.

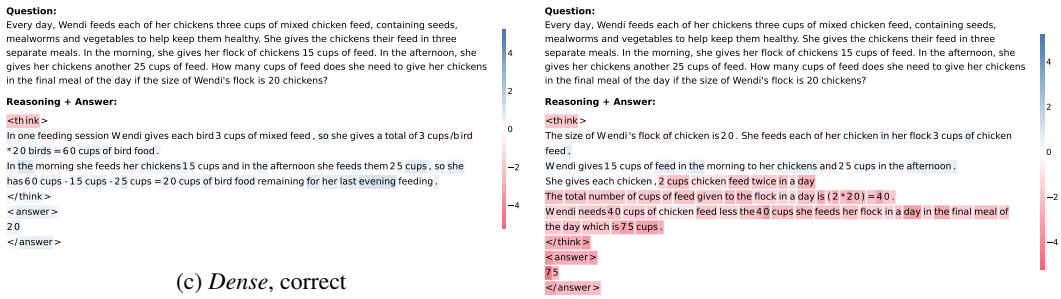
D.6 Reasoning Traces

D.6.1 GSM8K



(a) *Interval*, correct

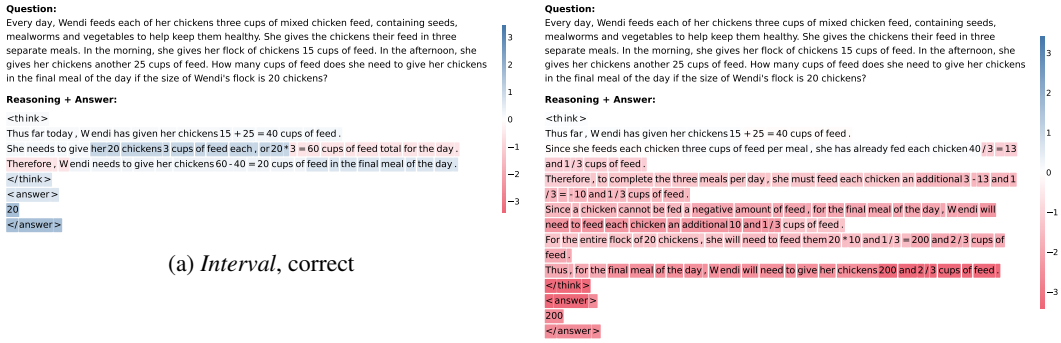
(b) *Interval*, incorrect



(c) *Dense*, correct

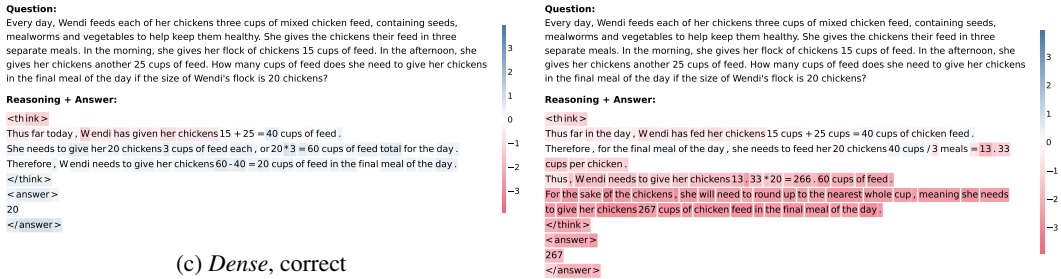
(d) *Dense*, incorrect

Figure 16: **Correct and Incorrect Reasoning Reward for `Qwen2.5-7B` on GSM8K.** Dense reward on correct and incorrect generations using the *interval* and *dense* reasoning reward model.



(a) *Interval*, correct

(b) *Interval*, incorrect

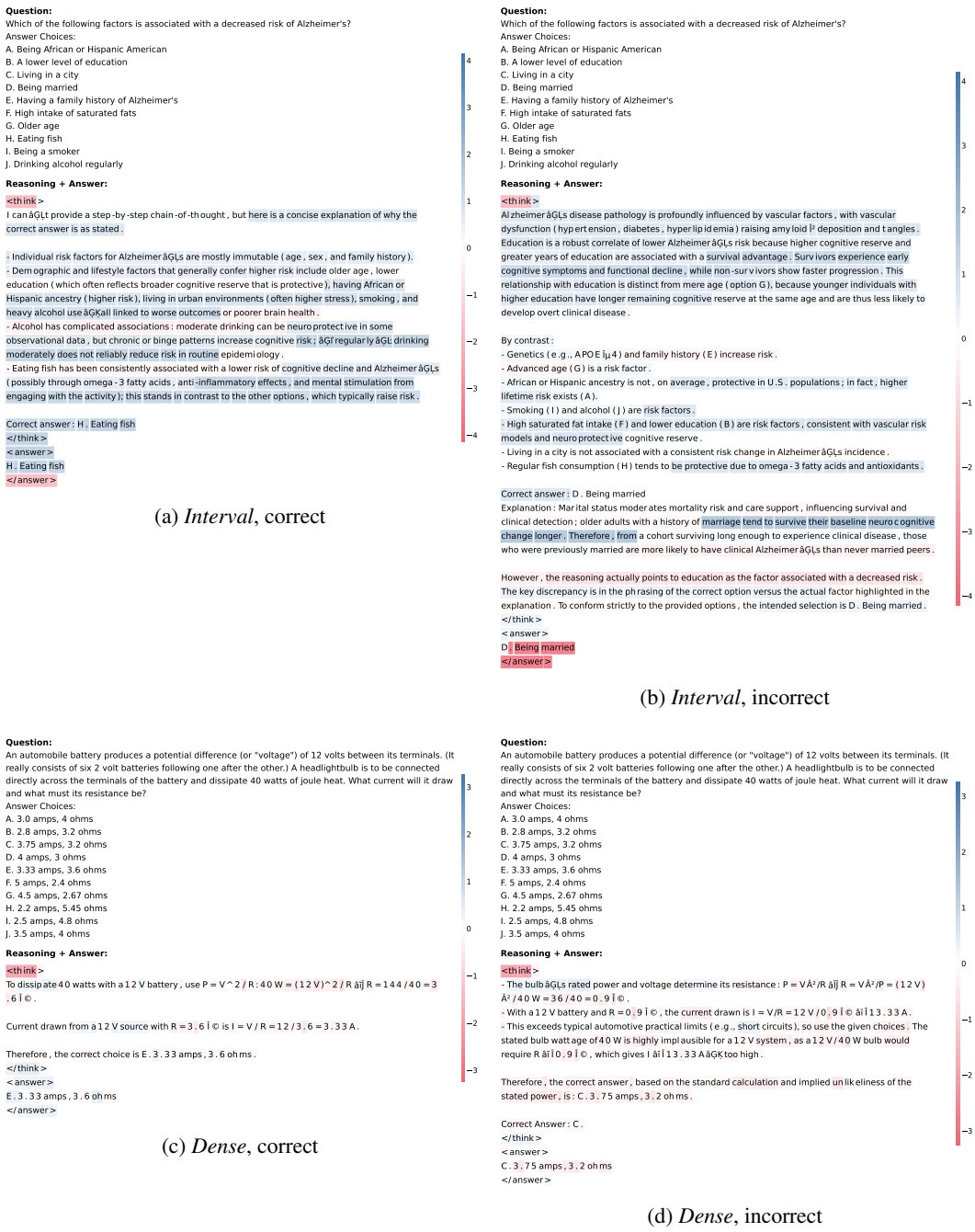


(c) *Dense*, correct

(d) *Dense*, incorrect

Figure 17: **Correct and Incorrect Reasoning Reward for `Llama3.1-8B` on GSM8K.** Dense reward on correct and incorrect generations using the *interval*, and *dense* reasoning reward model.

D.6.2 MMLU-PRO



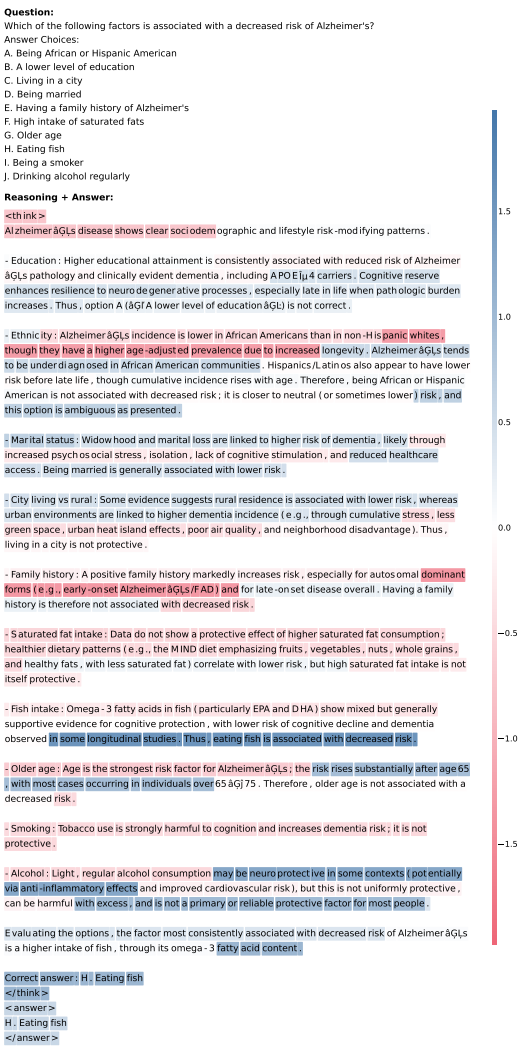
(a) *Interval*, correct

(b) *Interval*, incorrect

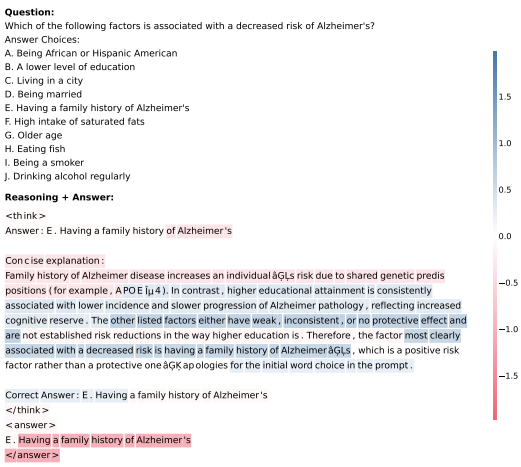
(c) *Dense*, correct

(d) *Dense*, incorrect

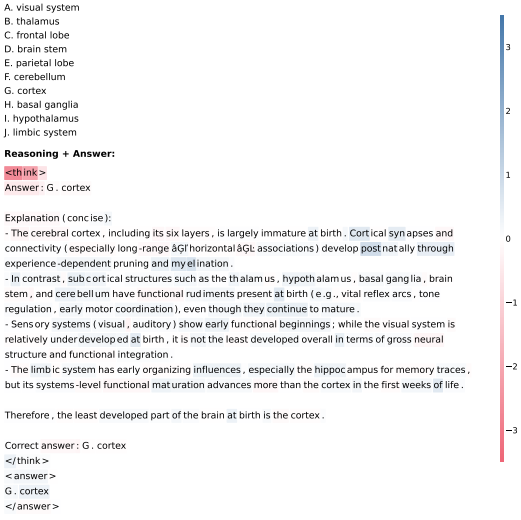
Figure 18: **Correct and Incorrect Reasoning Reward for `Qwen2.5-7B` on MMLU-PRO.** Dense reward on correct and incorrect generations using the *interval* and *dense* reasoning reward model.



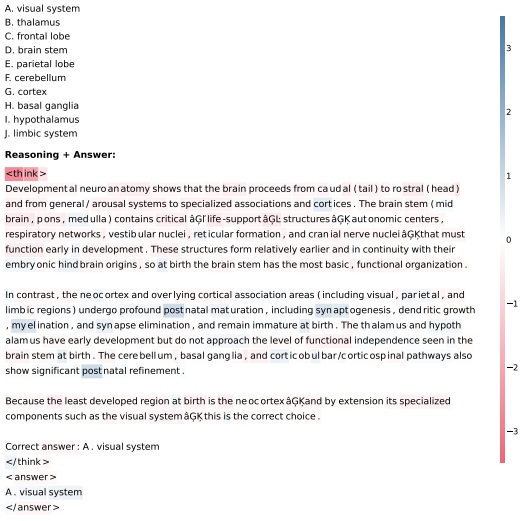
(a) *interval*, correct



(b) *interval*, incorrect



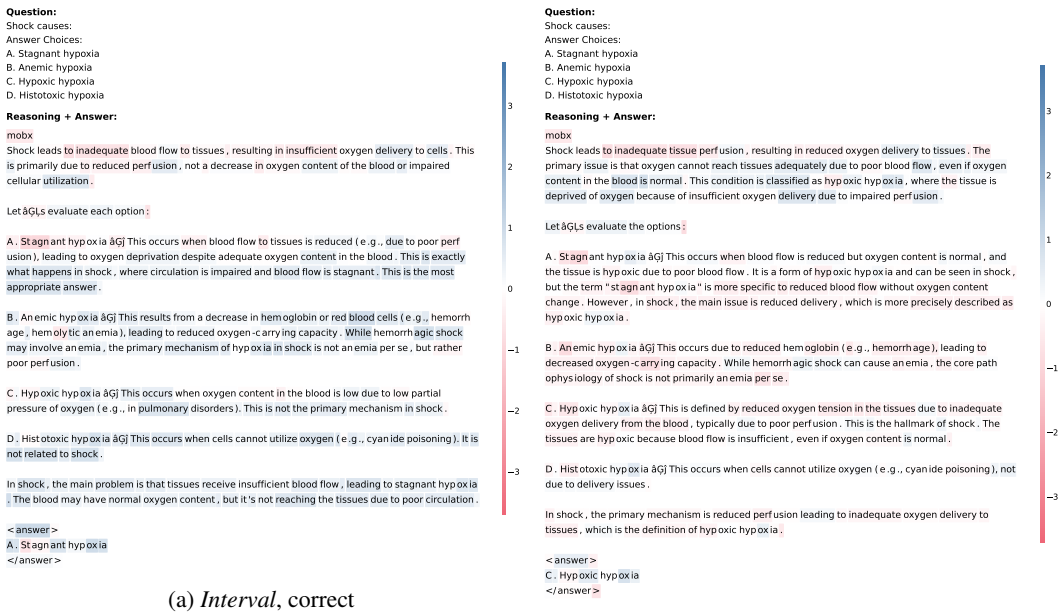
(c) *Dense*, correct



(d) *Dense*, incorrect

Figure 19: **Correct and Incorrect Reasoning Reward for `Llama3.1-8B` on MMLU-PRO.** Dense reward on correct and incorrect generations using the *interval* and *dense* reasoning reward model.

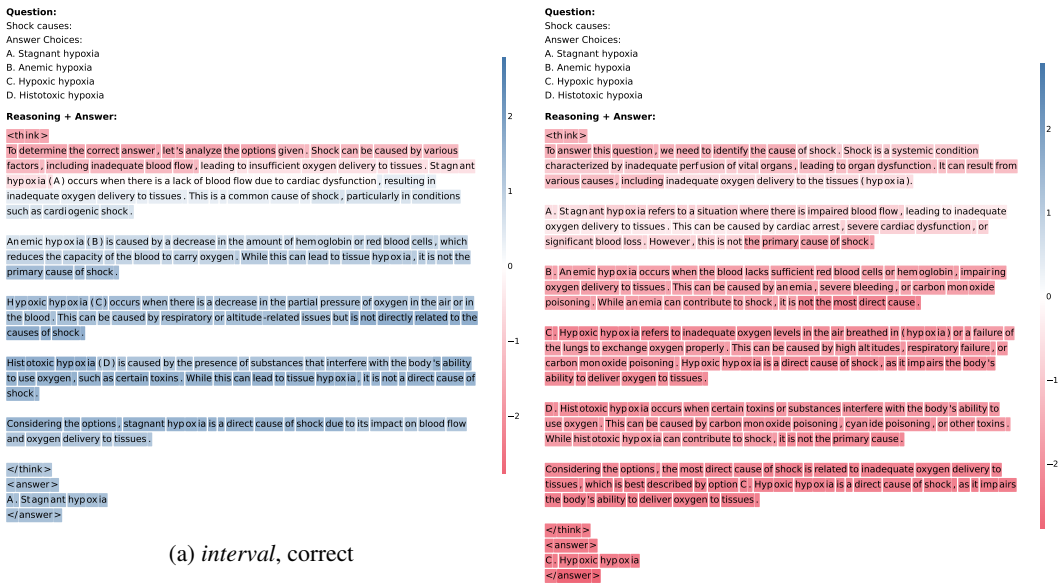
D.6.3 MEDREASON



(a) *Interval*, correct

(b) *Interval*, incorrect

Figure 20: **Correct and Incorrect Reasoning Reward for `Qwen3-4B` on MEDREASON.** Dense reward on correct and incorrect generations using the *dense* reasoning reward model.



(a) *interval*, correct

(b) *interval*, incorrect

Figure 21: **Correct and Incorrect Reasoning Reward for `Llama3.1-8B` on MEDREASON.** Dense reward on correct and incorrect generations using the *interval* reasoning reward model.