arXiv:2510.01857v5 [cs.AI] 17 May 2026

Learning Reasoning Rewards from Expert Demonstrations with Inverse Reinforcement Learning

Claudio Fanconi
University of Cambridge
caf83@cam.ac.uk

Nicolás Astorga
University of Cambridge
nja46@cam.ac.uk

Mihaela van der Schaar
University of Cambridge
mv472@cam.ac.uk

Abstract

Teaching large language models (LLMs) to reason during post-training typically relies on reinforcement learning with explicit outcome- or process-based reward functions. However, in many real-world settings, obtaining or defining such reward functions is difficult, especially for complex tasks, making learning from expert demonstrations an attractive alternative. The dominant approach, supervised fine-tuning (SFT), trains models to imitate expert reasoning traces directly, but suffers from the general limitations of off-policy learning: performance can be fragile to inference-time deviations from states explicitly covered by the demonstrations. To address this, we propose **Reasoning Adversarial Inverse Reinforcement Learning (R-AIRL)**. Rather than imitating the expert's reasoning, R-AIRL infers the underlying process-level reward from the expert Chain-of-Thoughts. Through experiments on GSM8K, MMLU-Pro and MedReason we show that the reasoning reward function learned with R-AIRL can be effectively used throughout the training and inference pipeline: (1) to provide a training signal for **post-training**, outperforming SFT in most of the considered settings, (2) for **inference-time reranking**, improving pass@1 by up to 17.4 points, and (3) for **process-level evaluation**, localising reasoning failures with up to 86.1% accuracy. Overall, R-AIRL bridges imitation learning and reward-based optimisation, enabling the extraction of meaningful reasoning signals from expert thinking traces. [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, these signals 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 a particularly attractive approach in the demonstration-only regime [Guo et al., 2025]. However, SFT mainly trains models to simply imitate observed reasoning traces rather than to evaluate them or directly extrapolate from them. As a result, when generation deviates from demonstrated paths, the model lacks an explicit objective for comparing alternative reasoning steps or recovering from errors [Setlur et al., 2025].

A natural alternative is to optimise reasoning via a *learned* reward. Yet, manually specifying such a reward remains challenging: hand-crafted signals can be brittle, task-specific and difficult to scale. At the same time, expert demonstrations implicitly encode which trajectories are preferred [Abbeel and Ng, 2004, Ziebart et al., 2004]. This suggests learning a reward for reasoning directly from demonstrations, rather than prescribing one externally.

[1] We provide the code for our experiments at https://github.com/fanconic/expert_reasoning

Preprint.

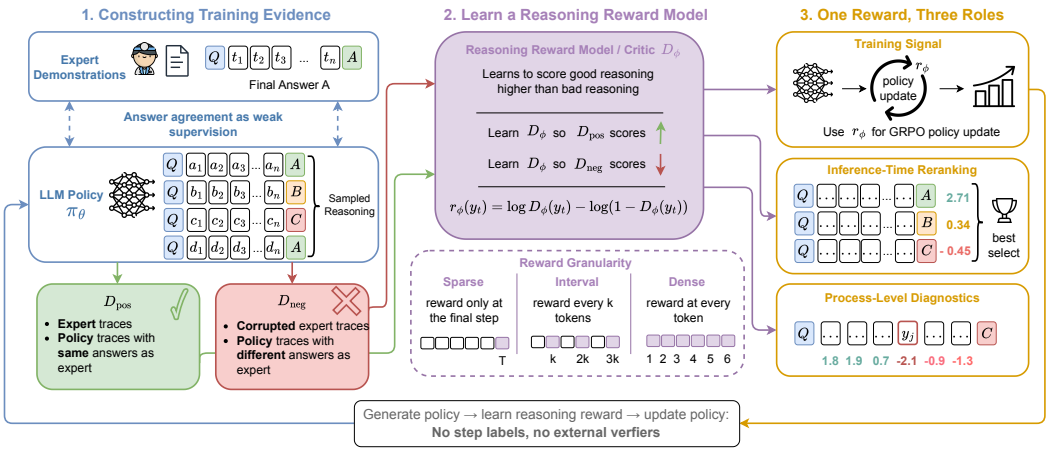


Figure 1: **R-AIRL: Learning and using reasoning rewards from expert demonstrations.** (i) We construct training evidence from expert and policy traces, (ii) learn a critic-based reward at sparse/interval/dense granularities, and (iii) reuse that reward for GRPO training, inference-time reranking, and process-level error localisation, without step labels or external verifiers.

This raises a key question: can we obtain a reward over reasoning traces that is both learnable from demonstrations and useful in practice? We argue that such a reward, in order to be truly useful, must satisfy three core requirements: it must provide a stable optimisation signal for policy learning, remain useful at inference time for selecting among candidate reasoning paths, and offer process-level feedback for diagnosing where reasoning fails.

We formalise three desiderata for a learned reasoning reward:

1. (D1) **Usable training signal.** A stable optimisation signal for on policy-generated trajectories.
2. (D2) **Inference-time utility.** Effective tool for ranking the sampled traces under a fixed budget, without retraining.
3. (D3) **Process-level feedback.** Step-level scoring function allowing to localise where reasoning first fails.

To satisfy these desiderata, we formulate reasoning as an inverse reinforcement learning (IRL) problem [Abbeel and Ng, 2004, Ziebart et al., 2004]. Our proposed method, Reasoning Adversarial Inverse Reinforcement Learning (R-AIRL), adapts adversarial IRL [Ho and Ermon, 2016, Fu et al., 2018] to the LLM reasoning setting: instead of teaching the model to imitate expert tokens, it learns a reasoning reward from demonstrations and then uses it to optimise the policy on its own sampled trajectories. In addition to providing a competitve training signal, the same reward can also bereused for inference-time reranking (to further boost performance) and fine-grained token/step diagnostics.

Contributions. We make three contributions:

1. **Learning rewards for reasoning.** We adapt adversarial IRL to learn rewards over reasoning trajectories from expert demonstrations at multiple levels of granularity.
2. **One reward, multiple roles.** We show that a single learned reward can be reused for policy optimisation, inference-time reranking, and error localisation.
3. **Empirical validation across training, inference, and transfer.** Across GSM8K, MEDREASON, and MMLU-PRO, learned rewards often outperform SFT, improve Best-of-16 reranking by up to 17.4 points, transfer across tasks/backbones in most settings, and localise reasoning errors.

Sections 5.1, 5.2, and 5.3 evaluate (D1)–(D3), respectively; 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 $p_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t})$, where $y_{<t}$ is the partial reasoning history.

Latent process reward. The true reward for reasoning is unobserved, so we model a process-level reward $r_\phi(y_t \mid x, y_{<t})$ and aggregate it as $R_\phi(y; m) = \frac{1}{T}\sum_{t=1}^{T} m_t\, r_\phi(y_t \mid x, y_{<t})$, with $m_t \in \{0, 1\}$. This unifies granularities: dense reward uses $m_t = 1$ for all t, while outcome/sparse reward is the special case $m_t = \mathbb{I}[t = T]$. The $1/T$ factor makes scores comparable across different trajectory lengths.

Inverse Reinforcement Learning (IRL). We treat expert demonstrations as samples from $p_E(y \mid x)$, i.e., from an unknown expert policy that is approximately optimal under a latent reasoning reward. We formulate reward learning as learning a process-level reward that separates expert from policy trajectories, and then update the policy to maximise that learned reward:

$$\max_{\phi} \min_{\theta} \; \mathbb{E}_{y^E \sim p_E}\big[R_\phi(y^E; m)\big] - \mathbb{E}_{y \sim \pi_\theta}[R_\phi(y; m)]\,. \tag{1}$$

The first term learns r_ϕ to prefer expert reasoning; the second drives policy optimisation under that reward. Because the expert term is independent of θ, minimising Eq. (1) w.r.t. θ is equivalent to maximising expected learned reward on policy trajectories (D1). Since r_ϕ is process-level, it yields token-level feedback (D3), and the same R_ϕ can be reused for inference-time reranking (D2).

3 Related Work

Using the notation from Section 2, we organise methods by supervision regime: verifier-based rewards, preference-based reward learning, and demonstration learning; we then compare whether they natively support D1–D3.

Externally specified reward objectives. Outcome-supervised reasoning RL optimises the terminal-mask special case

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

This reward is usually derived from an external verifier [Guo et al., 2025, Uesato et al., 2022, Yu et al., 2025, YuYue et al., 2025]. Process-reward models (PRMs) instead optimise $\max_\theta \mathbb{E}_{y \sim \pi_\theta}[R(y; m)]$ with denser masks m (e.g., full, interval, or step-wise). These approaches can be effective [Uesato et al., 2022, Lightman et al., 2023], and search-based variants can improve exploration [Zelikman et al., 2022, Yuan et al., 2023, Singh et al., 2024, Hosseini et al., 2024, Silver et al., 2017]. Relative to our setting, they typically provide strong optimisation signals (D1) and can support reranking (D2) when reward access is available, while D3 depends on externally specified process supervision. Crucially, however, in R-AIRL we assume no access to an external verifier and instead learn the reasoning from demonstrations.

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. With $(x, y^w, y^l) \sim \mathcal{D}_{\text{pref}}$, a common objective is:

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

This setting uses preference data rather than verifier-labelled rewards [Christiano et al., 2023, Rafailov et al., 2024]. This setting uses preference data rather than external verifier rewards [Christiano et al., 2023, Rafailov et al., 2024]. In our framing, preference-based rewards can support D1 and D2, but are typically outcome-level and therefore less direct for D3. Again, a key distinction is that in R-AIRL, we assume no access to preference-labelled data and instead learn the reasoning from demonstrations.

Demonstration-only learning. When supervision is provided as expert trajectories, a common training paradigm is supervised fine-tuning (SFT), i.e., knowledge distillation on teacher/expert traces [Hinton et al., 2015], formalised through the following loss function:

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

This often improves reasoning quality [Guo et al., 2025, Kang et al., 2023, Kujanpää et al., 2025, Xu et al., 2025], but it does not learn an explicit reward on policy-generated trajectories, so it is limited for (D1) and does not natively provide (D2) reranking or (D3) token-level localisation [Setlur et al., 2025]. Recent self-distillation work in continual learning remains distillation-based and complementary to reward learning [Shenfeld et al., 2026]. Adversarial distillation variants [Ye et al., 2025, Lee et al., 2025] infer trajectory-level discriminator rewards from teacher traces as an additional learning signal together with an external verifier:

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

R-AIRL differs by learning process-level scoring $R_\phi(y; m)$ from demonstrations and reusing one critic for (D1) training, (D2) reranking, and (D3) token-level localisation.

We summarise this comparison in Table 1.

Method Class	Example Works	Optimisation Objective	Supervision Data	No External Verifier	(D1)	(D2)	(D3)
Verifier-based 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	✗	✓	✓	✗
Process-supervised RL	Uesato et al. [2022] Lightman et al. [2023]	$\max_\theta \ \mathbb{E}_{y\sim\pi_\theta}[R(y; m)]$	Process verifier	✗	✓	✓	✓
Preference supervision							
Preference-supervised RL	Christiano et al. [2023] Rafailov et al. [2024]	$\max_\phi$ $\mathbb{E}_{(y^w,y^l)\sim\mathcal{D}_{\text{pref}}}\left[\log\sigma\Big(R_\phi(y^w; m^{\text{out}}) - R_\phi(y^l; m^{\text{out}})\Big)\right]$ then $\max_\theta \ \mathbb{E}_{y\sim\pi_\theta}\left[R_\phi(y; m^{\text{out}})\right]$	Preference pairs	✗	✓	✓	✗
Demonstration 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	✓	✗	✗	✗
Adversarial distillation	Ye et al. [2025] Lee et al. [2025]	$\max_\phi \min_\theta$ $\left(\mathbb{E}_{y^E\sim p_E}[R_\phi(y^E; m^{\text{out}})] - \mathbb{E}_{y\sim\pi_\theta}[R_\phi(y; m^{\text{out}})]\right)$ then $\max_\theta \mathbb{E}_{y\sim\pi_\theta}\left[R_\phi(y; m^{\text{out}}) + R(y; m^{\text{out}})\right]$	Demos + outcome verifier	(✗)	✓	(✓)	✗
R-AIRL	**this work**	$\max_\phi \min_\theta$ $\left(\mathbb{E}_{y^E\sim p_E}[R_\phi(y^E; m)] - \mathbb{E}_{y\sim\pi_\theta}[R_\phi(y; m)]\right)$	Demos	✓	✓	✓	✓

Table 1: **Related work comparison.** Methods are grouped by supervision regime (verifier-based, preference-based, demonstrations), with objectives, supervision sources, and verifier dependence. (D1): reward-based training signal on policy traces. (D2): inference-time reranking with the same reward. (D3): token-level diagnostics for error localisation. Parenthesised checkmarks indicate partial support.

4 Method

Our goal is to learn a reasoning reward from expert demonstrations, so the policy can be optimised on its own rollouts without external verifiers. We build on the adversarial IRL view of GAIL/AIRL [Ho and Ermon, 2016, Fu et al., 2018], but with a different supervision construction: instead of source-based expert-vs-policy labels, we form verifier-free positive/negative trace sets from answer agreement with paired demonstrations and corrupted expert traces, train a token-level critic, and convert its logits into rewards for policy updates.

4.1 Training Evidence Construction

A naive split (all expert positive, all policy negative) encourages style discrimination rather than reasoning-quality discrimination [Ye et al., 2025, Lee et al., 2025]. We therefore use answer agreement with the paired expert trace as weak, verifier-free supervision. For each prompt, let y^E be the paired expert trace and let $\mathcal{O}(y)$ denote the answer tokens from generation y:

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

Including answer-consistent policy traces in $\mathcal{D}_{\text{pos}}$ puts policy text on both sides, reducing source shortcuts and pushing the critic toward reasoning quality. For negatives, we use answer-mismatched policy traces and augment them with corrupted expert traces:

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

Corruptions preserve expert style while injecting reasoning errors, further discouraging provenance-based separation. This supervision is intentionally weak: answer agreement can miss latent errors or penalise partially correct traces, but it provides enough signal to learn useful rewards from demonstrations alone, without step-level labels or pairwise preferences.

4.2 Reasoning Critic and Reward Definition

We train a token classifier $D_\phi(y_t) \in (0,1)$ that assigns each token (and its previous context) a probability of belonging to the positive reasoning class ($\mathcal{D}_{\text{pos}}$ vs. $\mathcal{D}_{\text{neg}}$). Following Section 2, a binary mask $m_t \in \{0,1\}$ specifies on which tokens supervision is applied along the reasoning trajectory. We evaluate thee granularities: *Sparse Outcome* ($m_t = \mathbb{I}[t = T]$), *Interval* ($m_t = 1$ every k tokens), and *Fully Dense* ($m_t = 1$ for all t).

Granularity controls credit assignment: coarser masks are typically lower-variance and more stable for optimisation (D1), while denser masks provide more detailed process feedback for localisation (D3) but can be harder to optimise robustly. This enables a direct sparse-to-dense comparison using a single supervision source (demonstrations only), unlike pipelines that rely on external process labels or verifiers [Guo et al., 2025, Uesato et al., 2022, Lightman et al., 2023].

The critic objective is length-normalised binary cross-entropy:

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

After each critic update, we convert the probabilities into token rewards:

$$v_\phi(y_t) = \log D_\phi(y_t) - \log\big(1 - D_\phi(y_t)\big). \tag{4}$$

4.3 Policy Optimisation with Learned Rewards

Directly optimising Eq. (1) is a bilevel problem, which is computationally expensive for LLMs. We therefore alternate updates: critic steps for ϕ, then policy steps for θ under the current reward.

We optimise the LLM policy 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.

Reward Densification. The reward logit $v_\phi(y_t)$ is only defined at supervised positions ($m_t = 1$). For *sparse* and *interval*, we propagate each supervised score backwards to the preceding unsupervised tokens:

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

For *dense*, $r_\phi(y_t) = v_\phi(y_t)$. Intuitively, the latest supervised token for a segment has the richest context about the reasoning that led to it, so earlier tokens in that segment inherit its score.

Advantage Calculation. Following Cetin et al. [2025] and Cui et al. [2025]'s approach for dense rewards in GRPO, we compute length-normalised sequence-level reward means

$$\bar{r}^{(g)} = \frac{1}{T_g}\sum_{t=1}^{T_g} r_\phi(y_t^{(g)}),$$

then calculate the advantage scores by normalising within each group:

$$A^{(g)} = \frac{\bar{r}^{(g)} - \mu}{\sigma + \epsilon}. \tag{5}$$

We assign the 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 only (regardless of whether they fall in $\mathcal{D}_{\text{pos}}$ or $\mathcal{D}_{\text{neg}}$).

Additional details on adversarial stability are provided in Appendix A; the full algorithm is given in Algorithm 1.

5 Experiments

We evaluate our learning of reasoning reward models via R-AIRL on GSM8K [Cobbe et al., 2021], MEDREASON [Wu et al., 2025], and MMLU-PRO [Wang et al., 2024]. We choose these benchmarks because they stress complementary reasoning regimes: GSM8K emphasises multi-step arithmetic computation, with demonstrations collected from humans.MEDREASON emphasises domain-specific clinical reasoning with plausible but incorrect alternatives, with demonstrations generated by LLMs and quality checked by medical doctors. MMLU-PRO emphasises broad, heterogeneous scientific reasoning, and the reasoning demonstrations are generated by LLMs. This diversity is important for assessing whether the learned reward remains useful across different reasoning styles, rather than a single-task format.

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 always instantiate a policy model as a text generator and the reasoning reward model as a token classifier.

Each experimental subsection directly evaluates one desideratum from Section 1. (D1) (Section 5.1) tests training-signal quality in the demonstration-only (no external verifier) regime via final policy after R-AIRL training. (D2) (Section 5.2) tests inference-time utility via reranking gains under fixed budgets and transfer across tasks/backbones. (D3) (Section 5.3) tests process-level utility via localisation of the erroneous reasoning step. Implementation details appear in Appendix B.

5.1 Using the Learned Reward as a Training Signal

> **RQ1:** *Can R-AIRL improve reasoning in an LLM policy π_θ (the generator) when π_θ is trained only with rewards inferred from expert demonstrations?*

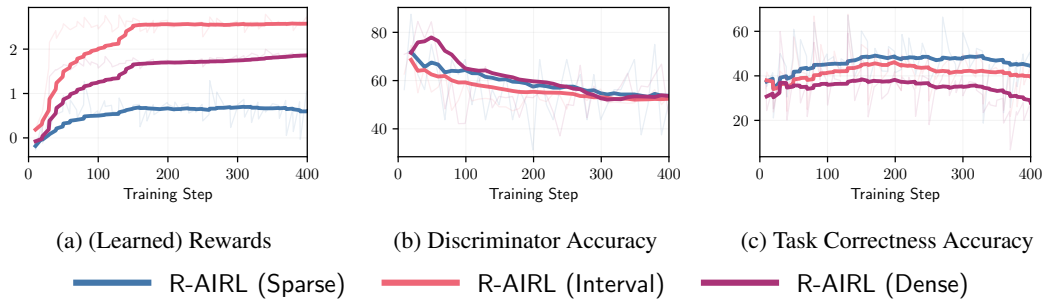
Experimental setup. For each backbone, we train π_θ with R-AIRL using *sparse*, *interval* ($k = 15$), and *dense* rewards. Following Ye et al. [2025], Lee et al. [2025], we use a 250-step discriminator warm-up, then jointly optimise policy and reward models per dataset. We report held-out pass@1 and compare to SFT under the same demonstration-only assumption (Table 2).

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		70.1 ± 1.6	66.6 ± 2.1	76.6 ± 1.7	48.1 ± 1.9	**47.2** ± 1.9	53.9 ± 2.0	65.4 ± 2.3	**79.5** ± 2.0	67.4 ± 2.3
R-AIRL	*Sparse*	**85.8** ± 1.5	**80.6** ± 1.6	**90.4** ± 1.5	48.5 ± 2.2	43.3 ± 1.9	**55.6** ± 2.2	**67.1** ± 2.5	77.6 ± 2.0	68.9 ± 2.6
	Interval	78.8 ± 1.6	67.6 ± 1.9	87.8 ± 1.6	**50.6** ± 1.9	36.6 ± 1.8	53.5 ± 2.4	66.3 ± 2.5	76.3 ± 2.1	69.1 ± 2.6
	Dense	38.4 ± 2.0	64.6 ± 2.1	89.6 ± 1.5	43.8 ± 2.0	37.9 ± 1.8	55.1 ± 2.3	66.5 ± 2.6	75.1 ± 2.1	**69.2** ± 2.5

Table 2: **Held-out pass@1 in the demonstration-only setting. Bold**/underlined mark best/second-best across SFT and R-AIRL variants. Values are mean ± half-width of the bootstrapped 95% confidence interval.

Training-signal quality. Table 2 shows learned rewards are competitive but not uniformly better than SFT: selecting the best granularity per backbone–dataset pair, R-AIRL improves over SFT in 7/9 settings. Gains are strongest on GSM8K (3/3 backbones) and positive for 2/3 backbones on MEDREASON; the main underperforming cases are `Llama3.1-8B` on MMLU-PRO and MEDREASON. Granularity controls stability: *sparse* is best in 7/9 pairs, *interval* in 1/9, and *dense* in 1/9. RQ1 targets only (D1); even where SFT has higher pass@1, it cannot provide the learned reward needed for (D2) reranking and (D3) process localisation (Sections 5.2 and 5.3).

Training dynamics. Figure 2 (Qwen2.5-7B on MMLU-PRO) should be read jointly: learned reward (left), discriminator accuracy (middle), and held-out pass@1 (right). In R-AIRL, discriminator accuracy approaching 50% is meaningful only if pass@1 rises; rising reward (2a) or falling discriminator accuracy (2b) without sustained pass@1 gains (2c) indicates reward–task decoupling, more common under denser supervision. We attribute *dense* instability to token-level supervision, which amplifies critic misclassifications and yields noisier policy rewards; *sparse* rewards are less sensitive and thus more stable.



(a) (Learned) Rewards (b) Discriminator Accuracy (c) Task Correctness Accuracy

Figure 2: **Training dynamics for `Qwen2.5-7B` on MMLU-Pro.** Left: learned reward; middle: discriminator accuracy; right: held-out pass@1 (diagnostic only; not used for training updates). Reward improvement without pass@1 gains indicates reward–task decoupling.

For completeness, Appendix Table 10 reports a comparison against outcome rewards if external verifiers were available. Appendix Figures 6–8 show full training curves. Moreover, Appendix C presents ablations on expert corruptions, group size, clipping, and policy-vs-expert-only separation, as well as on the AIME 2024/2025 datasets.

Takeaway. RQ1 is answered positively: in the demonstration-only setting, learned rewards provide a viable training signal, with a granularity trade-off between optimisation stability (sparser rewards) and richer supervision (denser rewards).

5.2 Inference-time Assistance via Reward-guided Reranking

RQ2: *Can the learned reasoning reward improve inference by reranking multiple sampled traces under a fixed test-time budget?*

Experimental setup. We evaluate inference-time reranking with the learned R-AIRL reward: for each prompt, sample $N = 16$ traces, rank by mean reward, and return the top candidate (Best-of-16). We compare against pass@1 with random selection from the same sample set. Main results are in Table 3, and calibration (AUROC/ECE) is in Appendix Table 11.

Method	Granularity	GSM8K			MMLU-Pro			MedReason		
		Random	Reward	Δ	Random	Reward	Δ	Random	Reward	Δ
`Qwen2.5-7B`	*Sparse*	85.8 ± 1.5	88.8 ± 1.7	(↑ +3.0)	48.5 ± 2.2	51.5 ± 2.5	(↑ +3.0)	67.1 ± 2.5	70.1 ± 2.9	(↑ +3.0)
	Interval	78.8 ± 1.6	82.5 ± 2.0	(↑ +3.7)	50.6 ± 1.9	53.1 ± 2.5	(↑ +2.5)	66.3 ± 2.5	67.2 ± 2.9	(↑ +0.9)
	Dense	38.4 ± 2.0	55.7 ± 2.6	(↑ +17.4)	43.8 ± 2.0	46.0 ± 2.5	(↑ +2.2)	66.5 ± 2.6	66.6 ± 2.9	(↑ +0.1)
`Llama3.1-8B`	*Sparse*	80.6 ± 1.6	82.1 ± 2.1	(↑ +1.5)	43.3 ± 1.9	42.3 ± 2.4	(↓ -1.0)	77.6 ± 2.0	79.9 ± 2.4	(↑ +2.3)
	Interval	67.6 ± 1.9	78.2 ± 2.2	(↑ +10.7)	36.6 ± 1.8	38.1 ± 2.4	(↑ +1.6)	76.3 ± 2.1	81.4 ± 2.5	(↑ +5.1)
	Dense	64.6 ± 2.1	71.1 ± 2.4	(↑ +6.5)	37.9 ± 1.8	39.7 ± 2.5	(↑ +1.9)	75.1 ± 2.1	80.9 ± 2.4	(↑ +5.8)
`Qwen3-4B`	*Sparse*	90.4 ± 1.5	93.0 ± 1.4	(↑ +2.5)	55.6 ± 2.2	62.1 ± 2.4	(↑ +6.4)	68.9 ± 2.6	69.9 ± 2.9	(↑ +1.0)
	Interval	87.8 ± 1.6	91.2 ± 1.5	(↑ +3.5)	53.5 ± 2.4	62.2 ± 2.4	(↑ +8.7)	69.1 ± 2.6	71.4 ± 2.9	(↑ +2.3)
	Dense	89.6 ± 1.5	91.4 ± 1.5	(↑ +1.8)	55.1 ± 2.3	59.2 ± 2.4	(↑ +4.1)	69.2 ± 2.5	69.9 ± 2.8	(↑ +0.7)

Table 3: **Best-of-16 reranking performance (%).** Reward-guided reranking is compared against random selection from the same sample set. Δ reports percentage-point change.

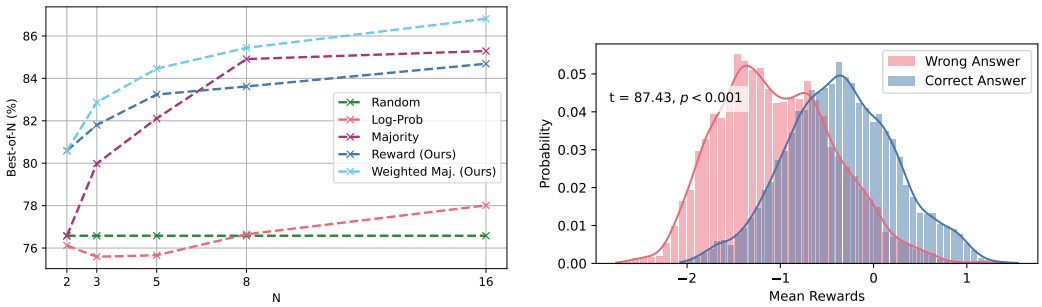
Reranking performance. In Table 3, reward-guided reranking improves pass@1 in 26/27 settings at $N = 16$. Largest gains include +17.4 pp (`Qwen2.5-7B`+*dense*, GSM8K), +10.7 pp (`Llama3.1-8B`+*interval*, GSM8K), and +8.7 pp (`Qwen3-4B`+*interval*, MMLU-Pro). On MedReason, gains are smaller but positive in all 9 settings (+0.1 to +5.8 pp). The only drop is `Llama3.1-8B`+*sparse* on MMLU-Pro (−1.0 pp).

Transfer across tasks and backbones. Using *interval* reward models trained on different datasets/backbones to rerank natural `Qwen2.5-7B` SFT samples, Table 4 shows uniformly positive transfer: all 18 source–target pairs improve, including off-diagonal gains up to +12.1 pp (`Llama3.1-8B`) and +13.0 pp (`Qwen3-4B`).

Policy Dataset	Llama3.1-8B			Qwen3-4B		
	GSM8K	MMLU-PRO	MEDREASON	GSM8K	MMLU-PRO	MEDREASON
GSM8K	↑ *+12.1*	↑ +7.7	↑ +12.1	↑ *+13.0*	↑ +8.8	↑ +13.0
MMLU-PRO	↑ +4.9	↑ *+4.4*	↑ +4.4	↑ +9.3	↑ *+1.9*	↑ +1.9
MEDREASON	↑ +3.0	↑ +10.1	↑ *+11.5*	↑ +5.5	↑ +5.6	↑ *+5.8*

Table 4: **Best-of-16 transfer reranking deltas (%).** Each cell reports Δ (reward-guided minus random reranking on the same sampled generations). Columns group reward-model backbones; rows are policy-task datasets. Diagonal entries (italic) are in-distribution.

Robustness across rerankers and budgets. Across $N \in \{2, 3, 5, 8, 16\}$, we compare log-probability, reward-guided reranking, majority voting, and reward-weighted majority voting on SFT-generated candidates (so logits remain independent of the reward model). Figure 3a shows a consistent trend: reward-guided reranking is strongest at low N, majority voting improves with N, and reward-weighted majority voting performs best overall. Appendix Tables 12–16 show the same pattern.



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

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

Figure 3: **Inference-time behaviour of learned rewards.** 3a: low-budget gains from reward-guided reranking; at higher N, reward-weighted majority voting is strongest. 3b: correct traces receive higher rewards than incorrect traces ($t = 87.43, p < 0.001$).

Why reranking works. Figure 3b shows clear reward separation between correct and incorrect traces ($t = 87.43, p < 0.001$) for `Qwen3-4B`+*interval* on MMLU-PRO, explaining the reranking gains in Table 3. Dataset-wide separation plots are in Appendix Figures 9–17.

Takeaway. RQ2 is answered positively: the learned reward is a reusable inference-time ranking signal that transfers across settings and complements consistency-based voting under fixed budgets.

5.3 Reasoning Error Localisation

RQ3: *Do the learned reasoning rewards provide process-level feedback that localises where reasoning fails?*

Experimental setup. To assess process-level supervision, we test whether dense rewards identify where reasoning first fails on GSM8K, and complement this with qualitative paired traces in Figure 4. Starting from correctly answered `Qwen2.5-7B` SFT traces, we construct controlled examples with a single perturbation in the reasoning chain (operator swaps or number corruption) while keeping the final answer. We score perturbed traces with our reasoning reward models under *dense* and *interval* granularities. For each corrupted trace, let r_t denote the reward at token t. We estimate the error location by detecting the largest one-step decrease in reward along the trace: $\Delta_t = \max(0, r_{t-1} - r_t)$, and $\hat{t} = \arg\max_t \Delta_t$. Intuitively, $\hat{t}$ is the position where the reward signal drops most sharply. We count a hit if $\hat{t}$ lies within $\pm\{1, 7\}$ tokens of the known perturbation index (Hit@1 and Hit@7).

Localisation performance. Table 5 shows strong localisation across all backbones, with best scores of 65.5 (`Qwen2.5-7B`), 86.1 (`Llama3.1-8B`), and 78.9 (`Qwen3-4B`). All scores are significantly above the empirical chance, indicating a better-than-random chance of scoring the reward drop within ± 1 and ± 7 tokens of the failure cases. In Appendix D.6, we demonstrate examples of the diverging reasoning trajectories on correct and corrupted traces after the first mistake.

Model	*Dense*		*Interval*
	Hit@1	Hit@7	Hit@7
`Qwen2.5-7B`	47.6 $\pm$ 2.8 (↑ +36.2)	54.9 $\pm$ 2.6 (↑ +23.6)	65.5 $\pm$ 2.6 (↑ +13.2)
`Llama3.1-8B`	65.8 $\pm$ 2.8 (↑ +53.7)	80.6 $\pm$ 2.2 (↑ +47.5)	86.1 $\pm$ 2.0 (↑ +30.1)
`Qwen3-4B`	62.0 $\pm$ 2.7 (↑ +50.3)	70.3 $\pm$ 2.5 (↑ +38.5)	78.9 $\pm$ 2.2 (↑ +26.0)

Table 5: **Error Localisation.** Hit@k (%, mean $\pm$ 95% CI half-width). (↑ +Δpp) denotes gain over chance. *Interval* uses 15-token buckets, so Hit@1 = Hit@7.

Error localisation on natural traces. Figure 4 shows the similar behaviour qualitatively, on correctly vs incorrectly generated traces: rewards stay positive on locally correct intermediate steps and drop sharply at the first erroneous step (here at "$12 - 2/3(12) = 4$"). This suggests the reward is sensitive to where reasoning first diverges. Additional examples on all datasets are in Appendix Figures 19-24.

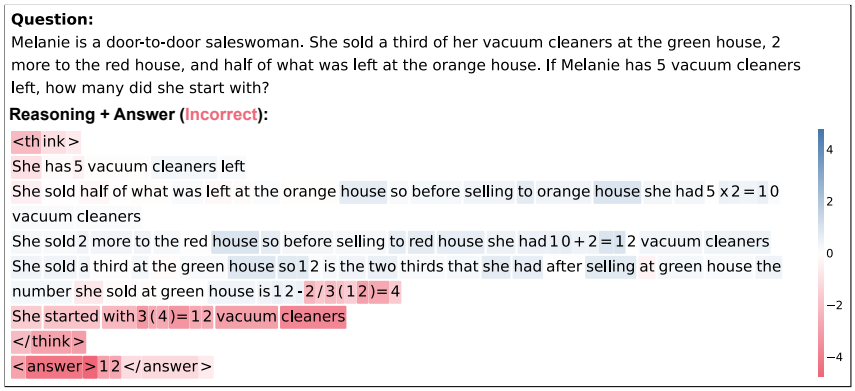


Figure 4: **Error localisation of incorrect reasoning.** Dense reward on incorrect generation for a GSM8K test question, using the *dense* `Qwen2.5-7B` reasoning reward model. The reward remains positive on valid intermediate steps, then drops sharply at the first erroneous step, followed by propagated penalties on later tokens.

Takeaway. RQ3 is answered positively: learned process-level rewards provide interpretable diagnostics that localise where reasoning first diverges, beyond final-answer evaluation.

6 Limitations

Our approach has three main limitations. Reward density trades off with optimisation stability: sparse step-wise signals are generally robust, whereas dense signals are more prone to adversarial mode collapse, non-stationarity, and occasional reward–correctness decoupling. R-AIRL is also computationally heavier due to alternating adversarial updates, and our evaluation is limited to arithmetic, medical, and scientific reasoning rather than open-ended or long-context tasks.

7 Conclusion

We presented Reasoning Adversarial Inverse Reinforcement Learning (R-AIRL), a framework for learning reasoning rewards from expert demonstrations without requiring external verifiers. Instead of directly imitating expert trajectories, R-AIRL learns a reusable reward model across the full pipeline. Empirically, this reward is useful in three ways: as a training signal (D1), where R-AIRL outperforms SFT in 7/9 demonstration-only settings; as an inference-time reranker (D2), where reward-guided reranking improves pass@1 in 26/27 settings at $N = 16$, with gains up to +17.4 points and uniformly positive transfer across all 18 source–target pairs; and as process-level feedback (D3), where it localises reasoning errors with Hit@7 up to 86.1% (30.1% above chance). Remaining limitations include the trade-off between reward granularity and optimisation stability, which we leave as an important direction for future work.

Acknowledgements and Disclosure of Funding

We thank our industry collaborators, Yusuke Kano, Jeremy Voisey, and Alison Q Smithard, for their insightful discussions. In addition, we thank Kasia Kobalczyk, Paulius Rauba, and Julianna Piskorz for their valuable feedback. Canon Inc. funds CF's studentship. The W.D. Armstrong Trust Fund and the Cystic Fibrosis Fund support NA's studentship. This work was supported by Microsoft's Accelerate Foundation Models Academic Research initiative.

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, 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].

Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Twenty-first international conference on Machine learning - ICML '04*, page 1, Banff, Alberta, Canada, 2004. ACM Press. doi: 10.1145/1015330.1015430. URL `http://portal.acm.org/citation.cfm?doid=1015330.1015430`.

Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum Entropy Inverse Reinforcement Learning. 2004.

Jonathan Ho and Stefano Ermon. Generative Adversarial Imitation Learning, June 2016. URL http://arxiv.org/abs/1606.03476. arXiv:1606.03476 [cs].

Justin Fu, Katie Luo, and Sergey Levine. Learning Robust Rewards with Adversarial Inverse Reinforcement Learning, August 2018. URL http://arxiv.org/abs/1710.11248. arXiv:1710.11248 [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.

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.

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.

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].

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.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. URL https://arxiv.org/abs/1706.08500.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN, December 2017. URL http://arxiv.org/abs/1701.07875. arXiv:1701.07875 [stat].

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 Stability.

A known limitation of adversarial training is that a single discriminator update per policy step leaves D_ϕ under-trained relative to π_θ, making the reward signal noisy and susceptible to reward hacking [Heusel et al., 2018, Arjovsky et al., 2017, Ye et al., 2025]. To mitigate this, we use three complementary stabilisation mechanisms. First, we maintain replay buffers $\mathcal{B}_{\text{pos}}$ and $\mathcal{B}_{\text{neg}}$ that accumulate positive and negative traces across iterations; each discriminator update samples mini-batches from these buffers rather than solely from current policy rollouts, broadening the training distribution and reducing non-stationarity. Second, we perform N_{disc} discriminator gradient steps per policy update, allowing D_ϕ to track the evolving policy more closely before the next GRPO step. Third, during policy learning, we 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 and prevents unstable updates. Together, these mechanisms better calibrate the reward signal and reduce mode-collapse-like behaviour in adversarial training.

A.2 Algorithm

Algorithm 1 Reasoning Adversarial Inverse Reinforcement Learning (R-AIRL)

Require: Expert traces $\mathcal{D}_E$; Iterations N_{step}; warm-up steps N_{warm}; discriminator updates per step N_{disc}

Initialise policy π_θ, discriminator D_ϕ, and replay buffers $\mathcal{B}_{\text{pos}} \leftarrow \emptyset$, $\mathcal{B}_{\text{neg}} \leftarrow \emptyset$
Warm-up Phase:
for $j \leftarrow 1$ to N_{warm} **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
Adversarial Training Phase:
for $i \leftarrow 1$ to N_{step} **do**
  Sample prompts $x \sim \mathcal{Q}$
  Generate group $\mathcal{D}_P \leftarrow \{y^{(g)}\}_{g=1}^{G} \sim \pi_\theta(\cdot \mid x)$
  Construct training sets based on answer correctness:
    $\mathcal{D}_{\text{pos}} \leftarrow \mathcal{D}_E \cup \{y \in \mathcal{D}_P \mid \mathcal{O}(y) = \mathcal{O}(y^E)\}$
    $\mathcal{D}_{\text{neg}} \leftarrow \{y \in \mathcal{D}_P \mid \mathcal{O}(y) \neq \mathcal{O}(y^E)\} \cup \{\mathcal{C}(y^E) \mid y^E \in \mathcal{D}_E\}$
    $\mathcal{B}_{\text{pos}} \leftarrow \mathcal{B}_{\text{pos}} \cup \mathcal{D}_{\text{pos}}$; $\mathcal{B}_{\text{neg}} \leftarrow \mathcal{B}_{\text{neg}} \cup \mathcal{D}_{\text{neg}}$ ▷ Update replay buffers
  Reward Model Update:
  for $k \leftarrow 1$ to N_{disc} **do**
    Sample mini-batches $\tilde{\mathcal{D}}_{\text{pos}} \sim \mathcal{B}_{\text{pos}}$ and $\tilde{\mathcal{D}}_{\text{neg}} \sim \mathcal{B}_{\text{neg}}$
    Update D_ϕ to minimise Eq. (3) using $\tilde{\mathcal{D}}_{\text{pos}}$ and $\tilde{\mathcal{D}}_{\text{neg}}$
  end for
  Policy Update:
  Get dense rewards $r_\phi(y_t^{(g)})$ via backfilling logits (Eq. 4 and Eq. 6)
  Compute advantages $A_t^{(g)}$ via group standardisation (Eq. 5)
  Optimise π_θ using GRPO loss on policy generations
end for

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. The dataset used in our experiments 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 strong language models (ChatGPT), which can be used for supervised fine-tuning or, in our case, adversarial inverse RL. 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, the answer options have been expanded from 4 options to 10 options 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 learned reward function via adversarial inverse reinforcement learning. To obtain a dense signal, we implement the discriminator 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 in https://github.com/fanconic/expert_reasoning.

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. The reward reasoning model is warmed up for 250 optimisation steps, and we train for 400 adversarial optimisation steps with a batch size of 16, generating $G = 8$ samples per prompt, accumulated over eight gradient steps (actual batch size + 128). During the adversarial update, we perform one policy-model update and three reasoning-reward model updates. To prevent data drift, we implemented a replay buffer that keeps track of the 50 most recent batches and samples a balanced combination of them when feeding the reward model. Both the discriminator and policy optimisers use a cosine-annealing learning rate schedule. The discriminator optimiser has a warm-up sequence of 250 steps (same as the discriminator warm-up period), while the policy optimiser is warmed up for 50 steps. We use a quantised ADAMW [Loshchilov and Hutter, 2019] optimiser.

Data and preprocessing. Prompts consist of the problem text with a short system instruction that requests step-by-step reasoning. Demonstrations are formatted as `<think> ... </think>` followed by `<answer> ... </answer>` format and provide the system instructions below. Tokenisation uses the native tokeniser of each backbone. 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.

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 ($\gamma = 0.95$) reward 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 surface form, we introduce targeted perturbations during discriminator training for both expert and policy traces. For GSM8K we apply the following random operations: (i) flip arithmetic operator signs, (ii) corrupt numeric literals by small random offsets, and (iii) swap the final answer with an earlier intermediate number. Perturbed traces are labelled as non-expert. For MEDREASON, we use ChatGPT-5.2 to generate incorrect reasoning traces that lead to incorrect results, by providing in-context the correct reasoning trace and the correct answer and instructing it to corrupt it. We do the same thing for MMLU-PRO. We provide the prompt below for corruption:

Compute. Experiments are conducted 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.

B.1 Policy and Reward Model

Policies are initialised from instruction-tuned checkpoints and trained with the learned 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 same reasoning reward models 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 reasoning reward models can effectively serve as both a training signal and an interpretable inference-time assistant. Addressing the stability-granularity trade-off is critical; this involves improving the optimisation dynamics at the discriminator–policy interface to prevent mode collapse in dense settings, potentially through alternative objectives such as the Wasserstein GAN [Arjovsky et al., 2017]. Moreover, the 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.

B.3 Statement about the Use of Large Language Models

We utilised large language models to assist with drafting and editing the manuscript and to accelerate implementation by generating boilerplate code and providing debugging suggestions. LLMs were not involved in the design of the methods, the study design, or the interpretation of the results. All outputs were reviewed and verified by the authors, who take full responsibility for the content.

C Additional Experiments

C.1 Ablation Reward Clipping

> **RQ-A1:** *How sensitive is R-AIRL to reward clipping bounds once the stabilised adversarial training recipe is used?*

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 optimisation recipe fixed (KL penalty, replay buffer, and three discriminator updates per policy update). We report the held-out Pass@1, the reranking gain (ΔRerank), AUROC, and ECE.

β **(LB, UB)**	**Pass@1**	Δ **Rerank**	**AUROC** $\uparrow$	**ECE** $\downarrow$
(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 6: **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.** Under our stabilised adversarial recipe for R-AIRL, 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** $\uparrow$	**ECE** $\downarrow$
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 7: **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 AIME

> **RQ-A3:** *Do demonstration-only learned rewards remain competitive on harder olympiad-style math benchmarks?*

Experimental setup. We evaluate pass@1 on AIME 2024 and AIME 2025 for the `Qwen3-4B`, comparing SFT with R-AIRL variants across reward granularities (sparse, interval, dense).

Method	AIME 2024	AIME 2025
`Qwen3-4B`		
SFT	**3.45**	3.12
Ours (*Sparse*)	3.43	3.82
Ours (*Interval*)	3.31	**4.30**
Ours (*Dense*)	3.00	2.17

Table 8: **Reward-weighted Pass@1 Performance (%).** **Bold** indicates the best performance compared between SFT and our methods.

> **Takeaway.** On harder olympiad-style benchmarks, the main message is competitiveness rather than uniform dominance, with stronger backbones benefiting more from reward-based training.

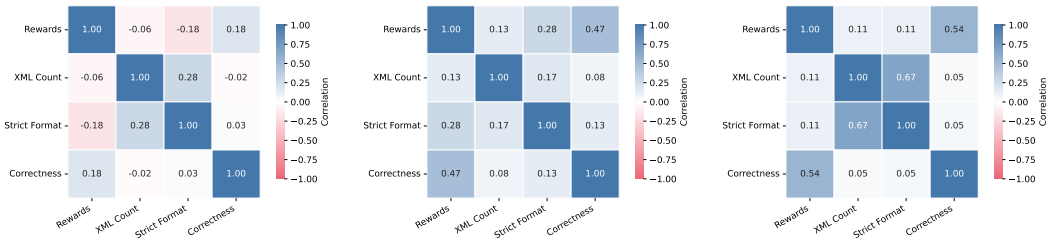
C.4 Ablation Corruptions

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

Experimental setup. We run a progressive ablation on `Qwen2.5-3B` (interval) on GSM8K: no answer agreement, answer agreement as soft supervision only, then answer agreement plus number corruption, number swapping, or symbol corruption. We report Pass@1, Δ Rerank, AUROC, and ECE (Table 9), and analyse reward 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 9: **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 rewards 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 rewards that track reasoning quality rather than superficial style cues. Corruptions can make this further robust.

D Additional Results

D.1 Complete Performance Table

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
External verifier										
Outcome Reward	–	89.0 ± 1.5	83.3 ± 1.6	91.6 ± 1.3	53.5 ± 2.0	48.4 ± 2.0	57.1 ± 2.2	69.2 ± 2.4	82.8 ± 2.1	71.8 ± 2.5
Demonstrations only										
SFT	–	70.1 ± 1.6	66.6 ± 2.1	76.6 ± 1.7	48.1 ± 1.9	**47.2** ± 1.9	53.9 ± 2.0	65.4 ± 2.3	**79.5** ± 2.0	67.4 ± 2.3
R-AIRL	*Sparse*	**85.8** ± 1.5	**80.6** ± 1.6	**90.4** ± 1.5	48.5 ± 2.2	43.3 ± 1.9	**55.6** ± 2.2	67.1 ± 2.5	77.6 ± 2.0	68.9 ± 2.6
	Interval	78.8 ± 1.6	67.6 ± 1.9	87.8 ± 1.6	**50.6** ± 1.9	36.6 ± 1.8	53.5 ± 2.4	66.3 ± 2.5	76.3 ± 2.1	69.1 ± 2.6
	Dense	38.4 ± 2.0	64.6 ± 2.1	89.6 ± 1.5	43.8 ± 2.0	37.9 ± 1.8	55.1 ± 2.3	**67.6** ± 2.4	76.7 ± 2.0	**69.3** ± 2.6

Table 10: **Held-out pass@1 by supervision regime.** Methods are separated into *External verifier* (Outcome Reward) and *Demonstrations only* (SFT and R-AIRL variants). **Bold** indicates the best performance and underlined the second best among demonstration-only methods. Values are reported as mean ± half-width of the 95% confidence interval bootstrapped over the test set.

D.2 Training Behaviour

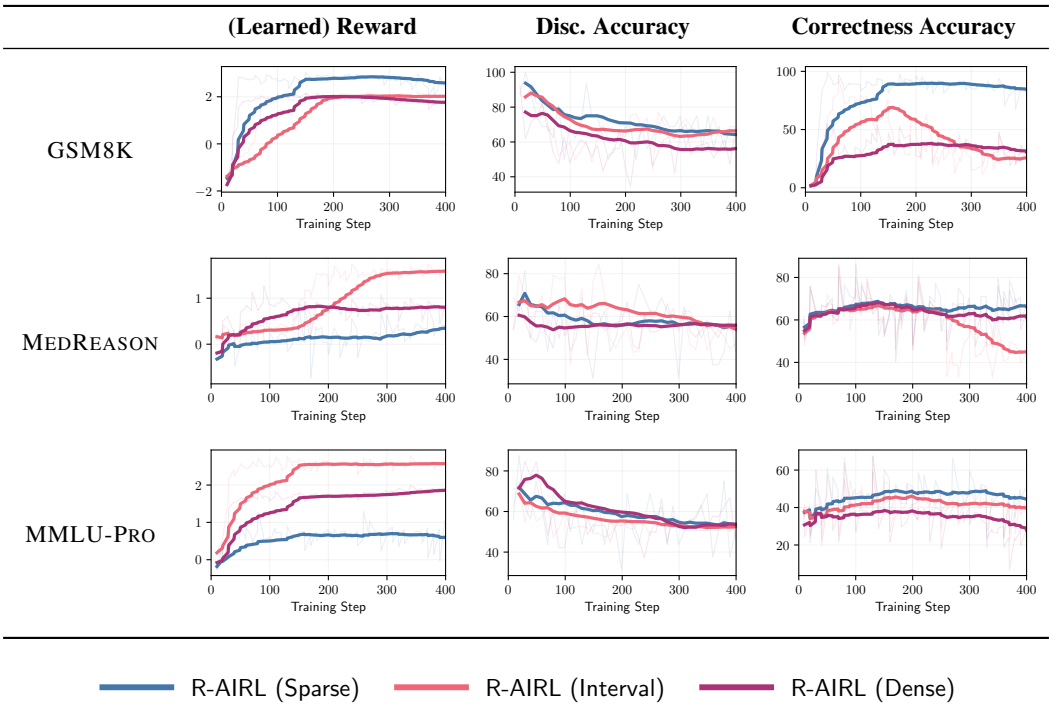


Figure 6: **Training dynamics for** `Qwen2.5-7B`**.**

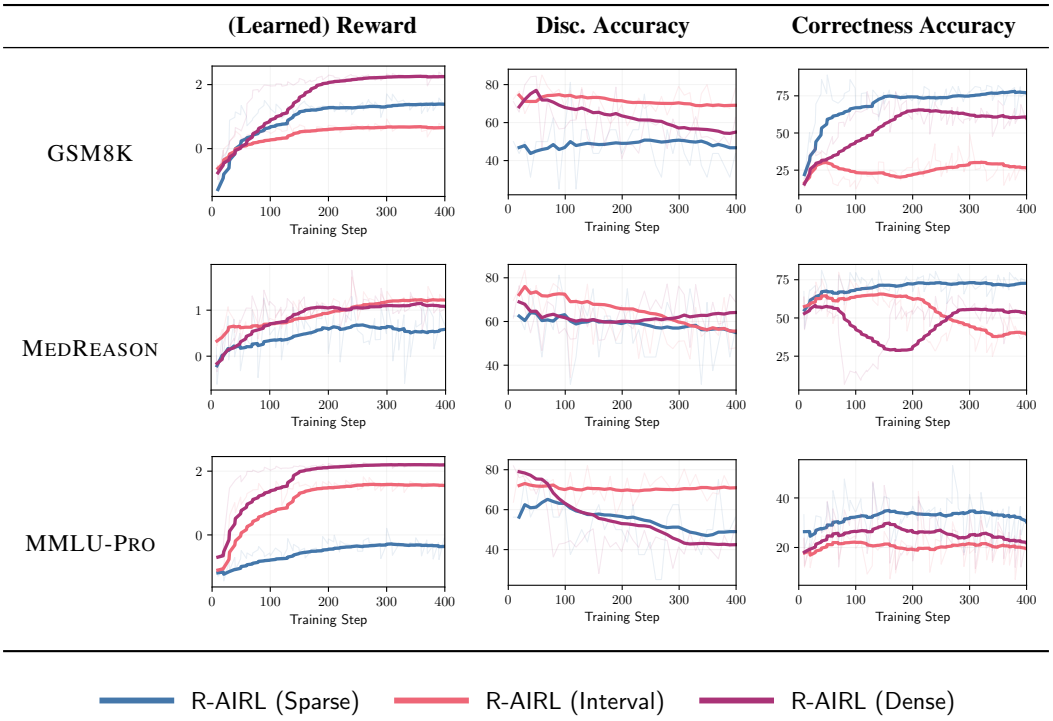


Figure 7: **Training dynamics for** `Llama3.1-8B`**.**

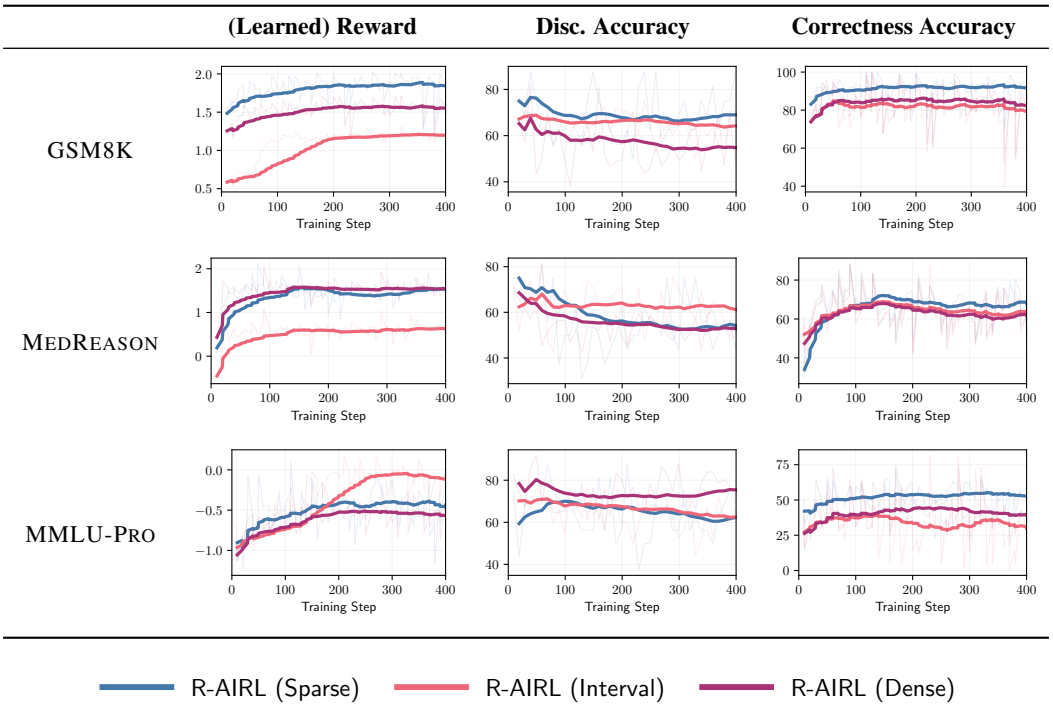


Figure 8: **Training dynamics for** `Qwen3-4B`**.**

D.3 Reranking Calibration

Backbone	Method	GSM8K		MEDREASON		MMLU-PRO	
		AUROC (%) ↑	ECE (%) ↓	AUROC (%) ↑	ECE (%) ↓	AUROC (%) ↑	ECE (%) ↓
Qwen2.5-7B	*Sparse*	69.9 [68.8, 71.0]	8.3 [7.9, 8.8]	76.8 [76.0, 77.6]	15.8 [15.1, 16.4]	70.9 [70.3, 71.5]	17.2 [16.6, 17.7]
	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	*Sparse*	75.0 [74.1, 75.9]	2.4 [1.9, 2.9]	82.2 [81.5, 83.0]	14.1 [13.5, 14.6]	68.7 [68.0, 69.3]	9.8 [9.2, 10.3]
	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	*Sparse*	84.1 [83.0, 85.1]	6.8 [6.5, 7.1]	73.9 [73.1, 74.7]	11.0 [10.4, 11.7]	86.8 [86.3, 87.2]	9.9 [9.4, 10.3]
	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.4 Reranking Performance Ablation

Backbone	Method	GSM8K					MMLU-PRO				
		Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.
Qwen2.5-4B	*Sparse*	76.6%	(↓ -0.5)	(0.0)	**(↑ +3.2)**	**(↑ +3.2)**	53.9%	(↑ +2.7)	(↑ +0.5)	**(↑ +4.5)**	**(↑ +4.5)**
	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	*Sparse*	70.1%	(↑ +0.4)	(↑ +0.3)	**(↑ +6.0)**	**(↑ +6.0)**	48.1%	(↑ +0.9)	(↓ -0.6)	**(↑ +3.5)**	**(↑ +3.5)**
	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	*Sparse*	66.6%	(↑ +1.4)	(↑ +0.2)	**(↑ +2.3)**	**(↑ +2.3)**	47.2%	(↑ +1.7)	(↑ +0.6)	**(↑ +2.8)**	**(↑ +2.8)**
	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.

Backbone	Method	GSM8K					MMLU-PRO				
		Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.
Qwen2.5-4B	*Sparse*	76.6%	(↓ -1.0)	(↑ +3.4)	(↑ +5.4)	**(↑ +6.5)**	53.9%	(↑ +3.7)	(↑ +3.5)	**(↑ +6.5)**	(↑ +6.4)
	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	*Sparse*	70.1%	(↑ +1.7)	(↑ +7.5)	(↑ +8.1)	**(↑ +9.2)**	48.1%	(↑ +1.9)	(↑ +2.1)	(↑ +4.3)	**(↑ +4.5)**
	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	*Sparse*	66.6%	(↑ +2.3)	(↑ +1.8)	(↑ +2.2)	**(↑ +3.4)**	47.2%	(↑ +2.0)	**(↑ +4.0)**	(↑ +2.1)	(↑ +3.2)
	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.

Backbone	Method	GSM8K					MMLU-PRO				
		Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.
Qwen2.5-4B	*Sparse*	76.6%	(↓ -0.9)	(↑ +5.5)	(↑ +6.8)	**(↑ +7.9)**	53.9%	(↑ +2.8)	(↑ +5.7)	(↑ +7.5)	**(↑ +8.1)**
	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	*Sparse*	70.1%	(↑ +2.5)	(↑ +10.9)	(↑ +9.9)	**(↑ +12.3)**	48.1%	(↑ +3.5)	(↑ +4.9)	(↑ +5.9)	**(↑ +6.5)**
	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	*Sparse*	66.6%	(↑ +2.5)	(↑ +3.0)	(↑ +2.2)	**(↑ +3.7)**	47.2%	(↑ +3.0)	**(↑ +6.0)**	(↑ +2.0)	(↑ +5.9)
	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.

Backbone	Method	GSM8K					MMLU-PRO				
		Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.
Qwen2.5-4B	*Sparse*	76.6%	(↑ +0.1)	(↑ +8.3)	**(↑ +9.1)**	**(↑ +9.1)**	53.9%	(↑ +2.4)	(↑ +6.0)	(↑ +7.9)	**(↑ +8.1)**
	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	*Sparse*	70.1%	(↑ +1.3)	(↑ +13.0)	(↑ +10.5)	**(↑ +13.9)**	48.1%	(↑ +2.1)	(↑ +5.7)	(↑ +6.9)	**(↑ +7.6)**
	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	*Sparse*	66.6%	(↑ +2.4)	(↑ +2.7)	(↑ +3.0)	**(↑ +4.3)**	47.2%	(↑ +2.5)	**(↑ +7.0)**	(↑ +1.4)	(↑ +6.6)
	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)**

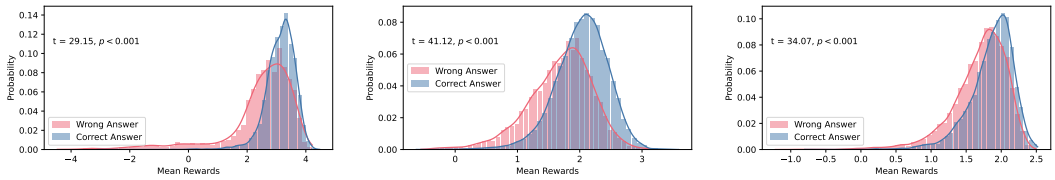
Table 15: **Best-of-8 Reranking Performance & Baselines (%).** Random is SFT pass@1. Deltas are percentage-point changes for each reranker. Blue arrows (↑) indicate gains and purple arrows (↓) indicate drops.

Backbone	Method	GSM8K					MMLU-PRO				
		Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.
Qwen2.5-4B	*Sparse*	76.6%	(↑ +1.4)	(↑ +8.7)	(↑ +10.3)	**(↑ +10.6)**	53.9%	(↑ +2.1)	(↑ +7.7)	(↑ +8.7)	**(↑ +9.3)**
	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	*Sparse*	70.1%	(↑ +0.8)	(↑ +15.6)	(↑ +11.3)	**(↑ +16.4)**	48.1%	(↑ +2.6)	(↑ +7.5)	(↑ +7.0)	**(↑ +8.6)**
	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	*Sparse*	66.6%	(↑ +2.8)	(↑ +4.6)	(↑ +3.7)	**(↑ +5.2)**	47.2%	(↑ +2.7)	(↑ +7.6)	(↓ -0.2)	**(↑ +8.3)**
	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.

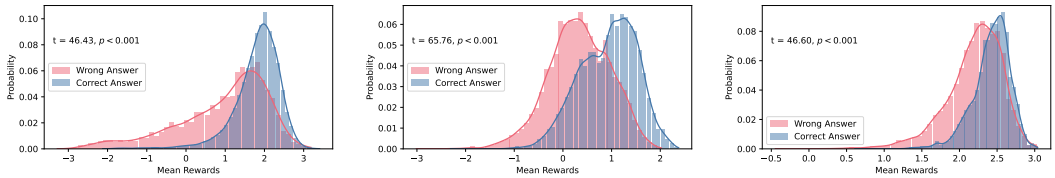
D.5 Distribution Separation of Rewards

D.5.1 GSM8K



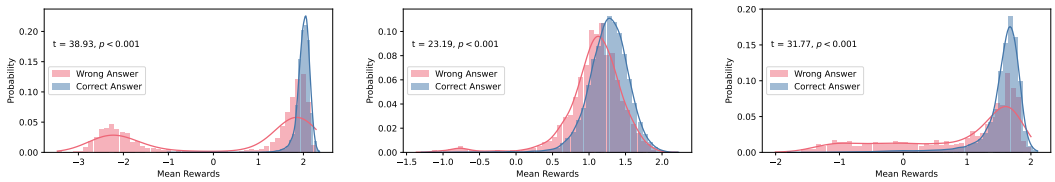
(a) Sparse (b) Interval (c) Dense

Figure 9: **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

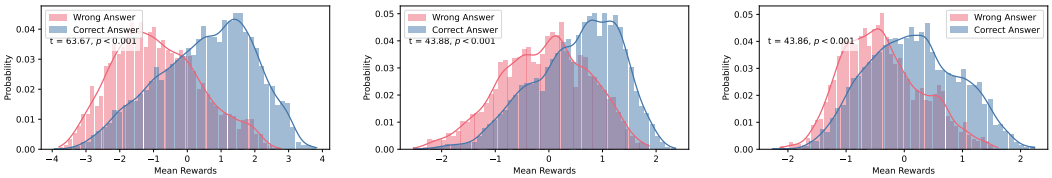
Figure 10: **Reward Distribution by Correctness (`Llama3.1-8B` on GSM8K)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.



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

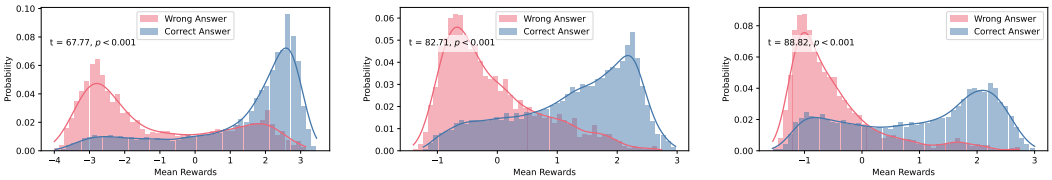
Figure 11: **Reward Distribution by Correctness (`Qwen3-4B` on GSM8K)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

D.5.2 MMLU-PRO



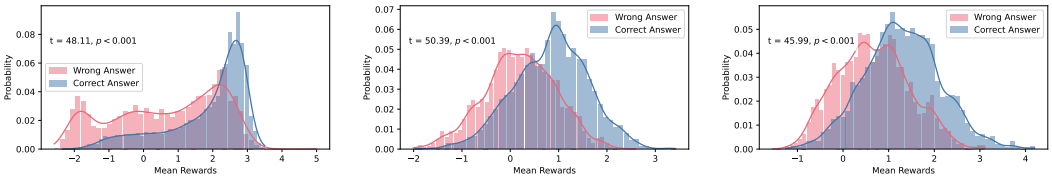
(a) Sparse (b) Interval (c) Dense

Figure 12: **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

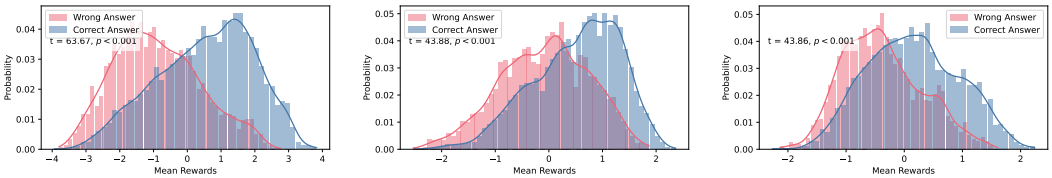
Figure 13: **Reward Distribution by Correctness (`Llama3.1-8B` on MMLU-PRO).** A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.



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

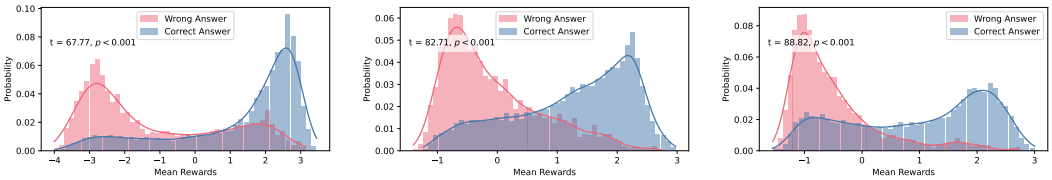
Figure 14: **Reward Distribution by Correctness (`Qwen3-4B` on MMLU-PRO).** A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

D.5.3 MEDREASON



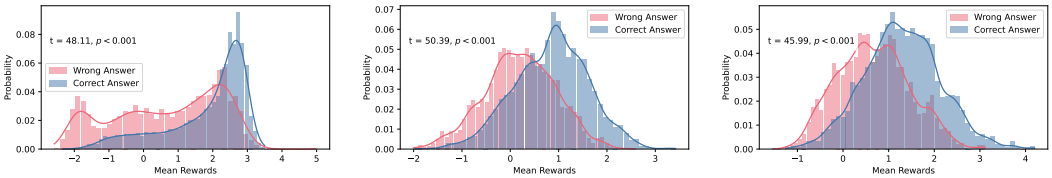
(a) Sparse (b) Interval (c) Dense

Figure 15: **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

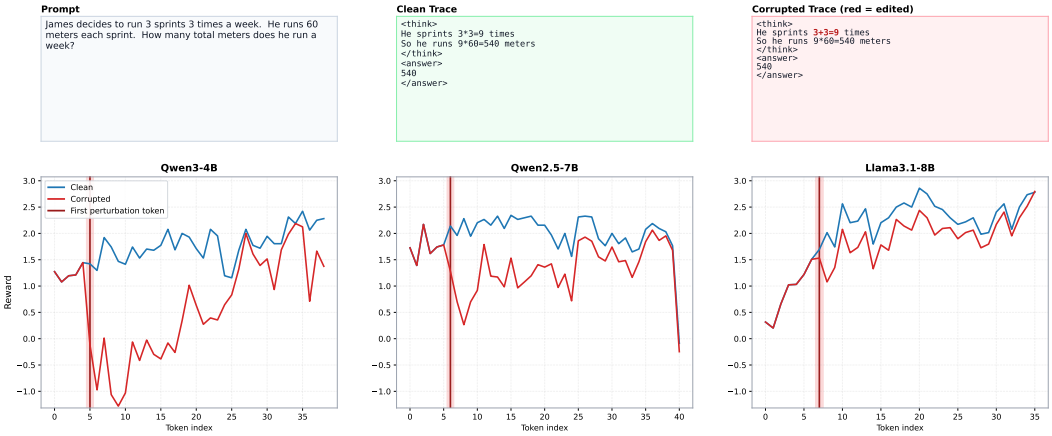
Figure 16: **Reward Distribution by Correctness** (`Llama3.1-8B` **on MEDREASON**). A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.



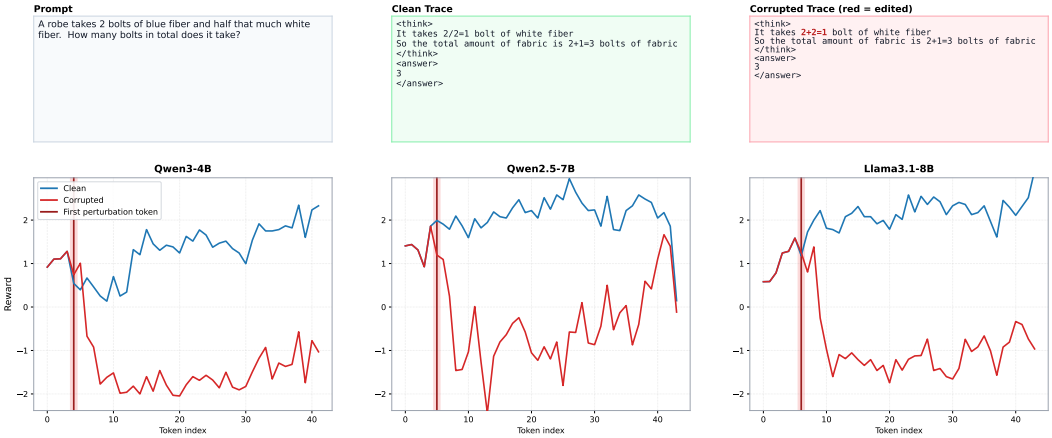
(a) Sparse (b) Interval (c) Dense

Figure 17: **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.

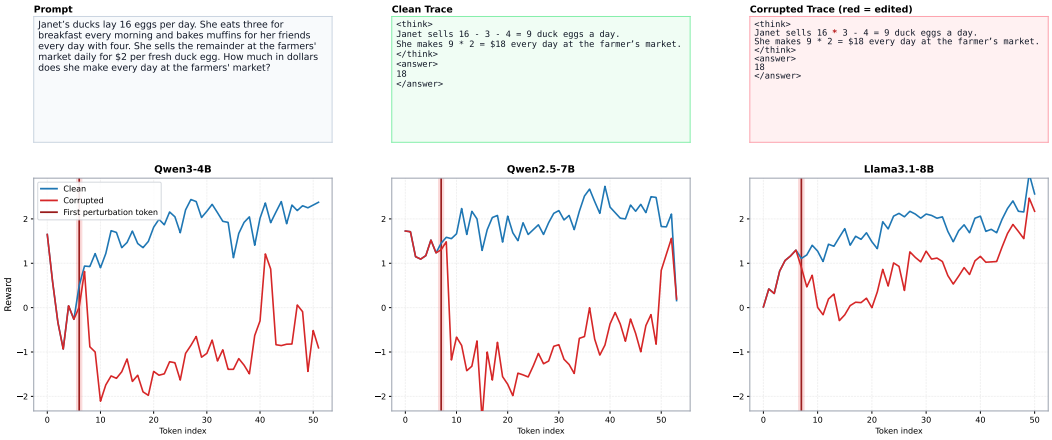
D.6 Error Localisation



(a) Example 1



(b) Example 2

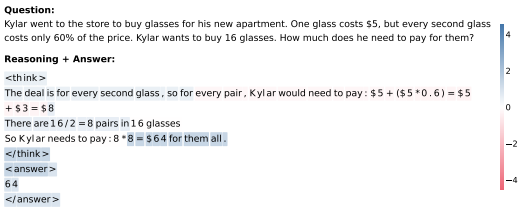


(c) Example 3

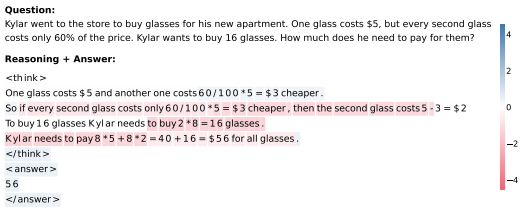
Figure 18: **Error Localisation on GSM8K** Dense reward on correct and incorrect generations using the *dense* reasoning reward models. We clearly observe a divergence between the correct and incorrect paths.

D.7 Reasoning Traces

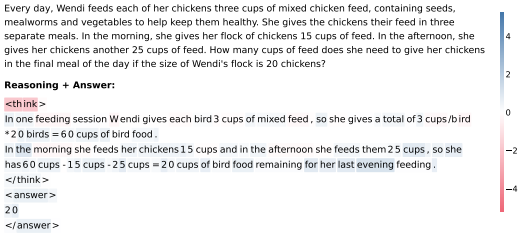
D.7.1 GSM8K



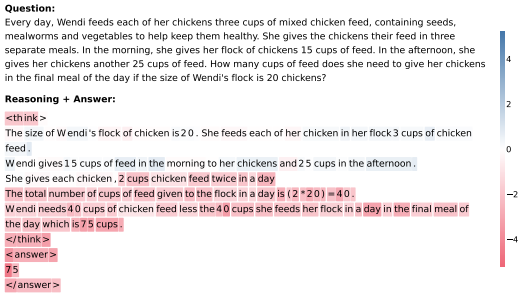
(a) *Interval*, correct



(b) *Interval*, incorrect

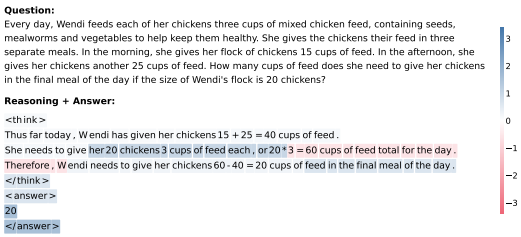


(c) *Dense*, correct

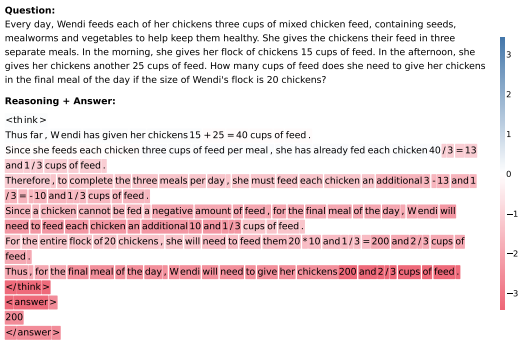


(d) *Dense*, incorrect

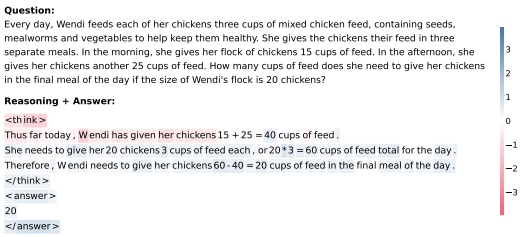
Figure 19: **Correct and Incorrect Reasoning Reward for** `Qwen2.5-7B` **on GSM8K.** Dense reward on correct and incorrect generations using the *interval* and *dense* reasoning reward model.



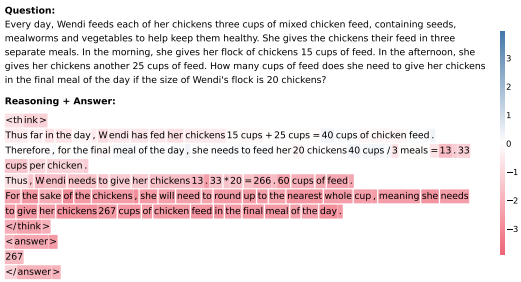
(a) *Interval*, correct



(b) *Interval*, incorrect



(c) *Dense*, correct



(d) *Dense*, incorrect

Figure 20: **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.7.2 MMLU-Pro

Question:
Which of the following factors is associated with a decreased risk of Alzheimer's?
Answer Choices:
A. Being African or Hispanic American
B. A lower level of education
C. Living in a city
D. Being married
E. Having a family history of Alzheimer's
F. High intake of saturated fats
G. Older age
H. Eating fish
I. Being a smoker
J. Drinking alcohol regularly

Reasoning + Answer:

<th ink>
I can âĢĻt provide a step -by -step chain -of -th ought , but here is a concise explanation of why the correct answer is as stated .

- Individual risk factors for Alzheimer âĢĻs are mostly immutable (age , sex , and family history).
- Dem ographic and lifestyle factors that generally confer higher risk include older age , lower education (which often reflects broader cognitive reserve that is protective), having African or Hispanic ancestry (higher risk), living in urban environments (often higher stress), smoking , and heavy alcohol use âĢĶall linked to worse outcomes or poorer brain health .
- Alcohol has complicated associations : moderate drinking can be neuro protect ive in some observational data , but chronic or binge patterns increase cognitive risk ; âĢĵ regular ly âĢĽ drinking moderately does not reliably reduce risk in routine epidemi ology .
- Eating fish has been consistently associated with a lower risk of cognitive decline and Alzheimer âĢĻs (possibly through omega - 3 fatty acids , anti -inflammatory effects , and mental stimulation from engaging with the activity); this stands in contrast to the other options , which typically raise risk .

Correct answer : H . Eating fish
</think>
<answer>
H . Eating fish
</answer>

(a) *Interval*, correct

Question:
Which of the following factors is associated with a decreased risk of Alzheimer's?
Answer Choices:
A. Being African or Hispanic American
B. A lower level of education
C. Living in a city
D. Being married
E. Having a family history of Alzheimer's
F. High intake of saturated fats
G. Older age
H. Eating fish
I. Being a smoker
J. Drinking alcohol regularly

Reasoning + Answer:

<th ink>
Al zheimer âĢĻs disease pathology is profoundly influenced by vascular factors , with vascular dysfunction (hyp ert ension , diabetes , hyper lip id emia) raising amy loid β deposition and t angles . Education is a robust correlate of lower Alzheimer âĢĻs risk because higher cognitive reserve and greater years of education are associated with a survival advantage . Surv ivors experience early cognitive symptoms and functional decline , while non -surv ivors show faster progression . This relationship with education is distinct from mere age (option G), because younger individuals with higher education have longer remaining cognitive reserve at the same age and are thus less likely to develop overt clinical disease .

By contrast :
- Genetics (e .g ., A PO E μ4) and family history (E) increase risk .
- Advanced age (G) is a risk factor .
- African or Hispanic ancestry is not , on average , protective in U .S . populations ; in fact , higher lifetime risk exists (A).
- Smoking (I) and alcohol (J) are risk factors .
- High saturated fat intake (F) and lower education (B) are risk factors , consistent with vascular risk models and neuro protect ive cognitive reserve .
- Living in a city is not associated with a consistent risk change in Alzheimer âĢĻs incidence .
- Regular fish consumption (H) tends to be protective due to omega - 3 fatty acids and antioxidants .

Correct answer : D . Being married
Explanation : Mar ital status moder ates mortality risk and care support , influencing survival and clinical detection ; older adults with a history of marriage tend to survive their baseline neuro c ognitive change longer . Therefore , from a cohort surviving long enough to experience clinical disease , those who were previously married are more likely to have clinical Alzheimer âĢĻs than never married peers .

However , the reasoning actually points to education as the factor associated with a decreased risk . The key discrepancy is in the ph rasing of the correct option versus the actual factor highlighted in the explanation . To conform strictly to the provided options , the intended selection is D . Being married .
</think>
<answer>
D . Being married
</answer>

(b) *Interval*, incorrect

Question:
An automobile battery produces a potential difference (or "voltage") of 12 volts between its terminals. (It really consists of six 2 volt batteries following one after the other.) A headlightbulb is to be connected directly across the terminals of the battery and dissipate 40 watts of joule heat. What current will it draw and what must its resistance be?
Answer Choices:
A. 3.0 amps, 4 ohms
B. 2.8 amps, 3.2 ohms
C. 3.75 amps, 3.2 ohms
D. 4 amps, 3 ohms
E. 3.33 amps, 3.6 ohms
F. 5 amps, 2.4 ohms
G. 4.5 amps, 2.67 ohms
H. 2.2 amps, 5.45 ohms
I. 2.5 amps, 4.8 ohms
J. 3.5 amps, 4 ohms

Reasoning + Answer:

<th ink>
To dissip ate 4 0 watts with a 1 2 V battery , use P = V ^ 2 / R : 4 0 W = (1 2 V)^ 2 / R âĵĵ R = 1 4 4 / 4 0 = 3 . 6 Î © .

Current drawn from a 1 2 V source with R = 3 . 6 Î © is I = V / R = 1 2 / 3 . 6 = 3 . 3 3 A .

Therefore , the correct choice is E . 3 . 3 3 amps , 3 . 6 oh ms .
</think>
<answer>
E . 3 . 3 3 amps , 3 . 6 oh ms
</answer>

(c) *Dense*, correct

Question:
An automobile battery produces a potential difference (or "voltage") of 12 volts between its terminals. (It really consists of six 2 volt batteries following one after the other.) A headlightbulb is to be connected directly across the terminals of the battery and dissipate 40 watts of joule heat. What current will it draw and what must its resistance be?
Answer Choices:
A. 3.0 amps, 4 ohms
B. 2.8 amps, 3.2 ohms
C. 3.75 amps, 3.2 ohms
D. 4 amps, 3 ohms
E. 3.33 amps, 3.6 ohms
F. 5 amps, 2.4 ohms
G. 4.5 amps, 2.67 ohms
H. 2.2 amps, 5.45 ohms
I. 2.5 amps, 4.8 ohms
J. 3.5 amps, 4 ohms

Reasoning + Answer:

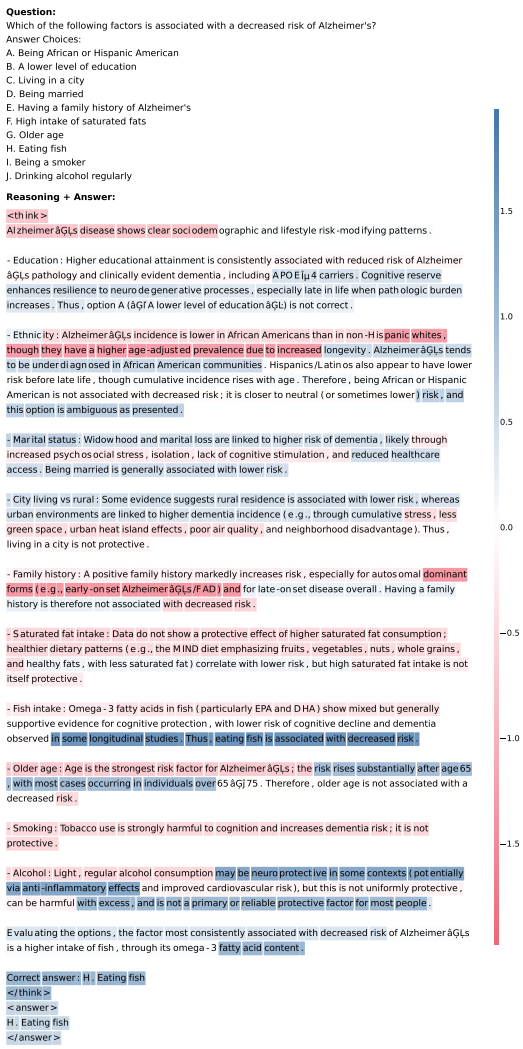
<th ink>
- The bulb âĢĻs rated power and voltage determine its resistance : P = V Â² /R âĵĵ R = V Â² /P = (1 2 V) Â² / 4 0 W = 3 6 / 4 0 = 0 . 9 Î © .
- With a 1 2 V battery and R = 0 . 9 Î © , the current drawn is I = V/R = 1 2 V / 0 . 9 Î © âĩĵ 1 3 . 3 3 A .
- This exceeds typical automotive practical limits (e .g ., short circuits), so use the given choices . The stated bulb watt age of 4 0 W is highly impl ausible for a 1 2 V system , as a 1 2 V / 4 0 W bulb would require R âĩĵ 0 . 9 Î © , which gives I âĩĵ 1 3 . 3 3 A âĢĶ too high .

Therefore , the correct answer , based on the standard calculation and implied un lik eliness of the stated power , is : C . 3 . 7 5 amps , 3 . 2 oh ms .

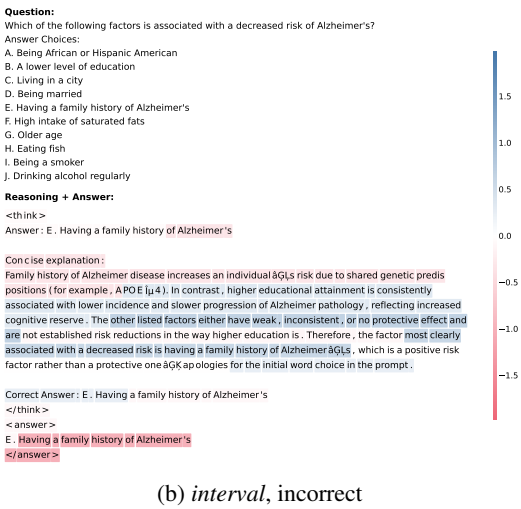
Correct Answer : C .
</think>
<answer>
C . 3 . 7 5 amps , 3 . 2 oh ms

(d) *Dense*, incorrect

Figure 21: **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



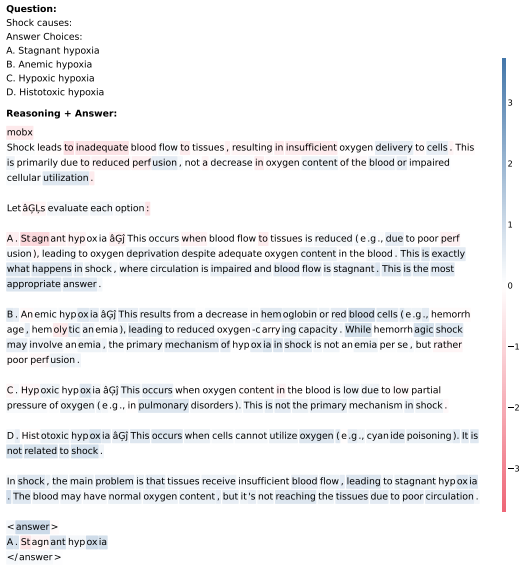
(b) *interval*, incorrect

(c) *Dense*, correct

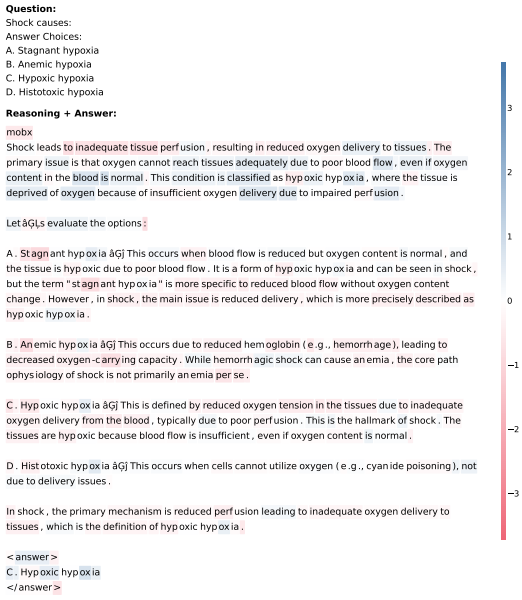
(d) *Dense*, incorrect

Figure 22: **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.7.3 MedReason

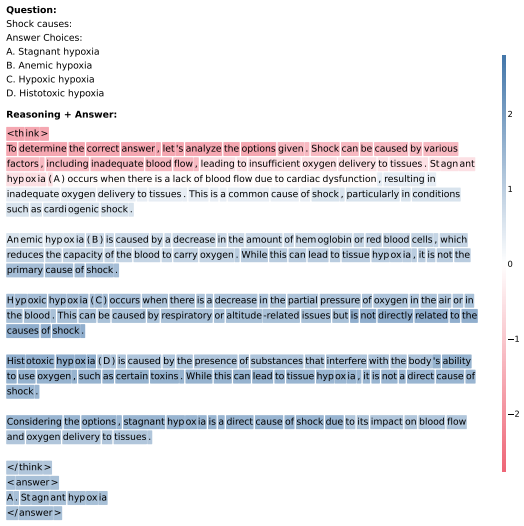


(a) *Interval*, correct

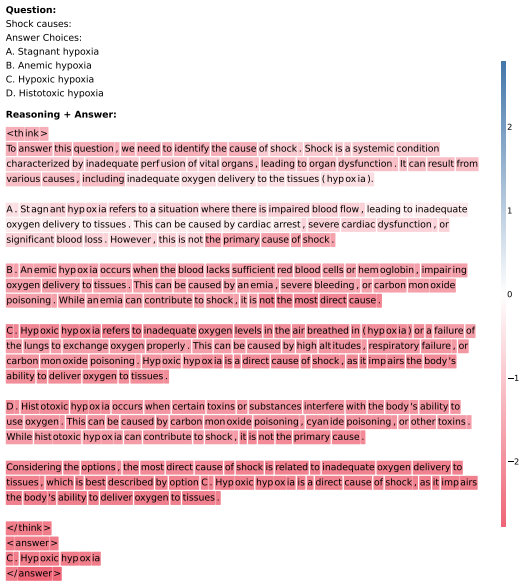


(b) *Interval*, incorrect

Figure 23: **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 24: **Correct and Incorrect Reasoning Reward for `Llama3.1-8B` on MEDREASON.** Dense reward on correct and incorrect generations using the *interval* reasoning reward model.