

LEARNING REASONING REWARD MODELS FROM EXPERT DEMONSTRATION VIA 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

Current approaches to improving reasoning in large language models (LLMs) primarily rely on either supervised fine-tuning (SFT) over expert traces or reinforcement learning (RL) with outcome-level rewards. However, SFT is fundamentally imitative, while outcome-based RL assumes access to a well-specified verifier. To address this gap, we propose an adversarial inverse reinforcement learning (AIRL) framework that learns reasoning rewards directly from expert demonstrations. We evaluate this framework across reward granularities (sparse, interval, and dense). Granularity controls the resolution of credit assignment: sparse rewards emphasise global trajectory quality and training stability, while denser rewards provide higher-resolution step-level supervision for error localisation but are harder to optimise stably. We show that the learned reasoning rewards are useful in three complementary ways. First, as a training signal, they often outperform SFT, with the best variant improving over SFT on medical reasoning (MedReason), mathematics (GSM8K), and challenging scientific question-answering (MMLU-Pro). Second, as an inference-time reranker, they gain up to 17.4 percentage points under a fixed sampling budget. Third, the learned reward transfers across tasks and backbones, suggesting that part of the signal is reusable beyond a single domain or model, and that finer-grained rewards identify the first step at which a trajectory deviates from a correct path. This supports the diagnosis of reasoning failures and the improvement of test-time selection. Together, these results show that AIRL can recover a reusable intermediate reasoning step from demonstrations alone, bridging the gap between pure imitation and reward-driven optimisation for LLM reasoning.¹

1 INTRODUCTION

Recent advancements in large language models (LLMs) have driven rapid progress on multi-step reasoning tasks. A dominant method is to transfer reasoning behaviours from human experts or stronger models via supervised fine-tuning (SFT) on their reasoning traces (DeepSeek-AI et al., 2025). While effective, this strategy mainly optimises imitation of observed teacher trajectories. This is valuable when test-time reasoning remains close to demonstrated traces, but it provides limited guidance once the model reaches off-demonstration states, because it does not explicitly learn a step-level objective for scoring alternative intermediate reasoning (Setlur et al., 2025).

Defining a faithful reward for reasoning is non-trivial: hand-crafted signals are task-specific and often incentivise shortcuts. By contrast, expert reasoning traces are comparatively easier to collect at scale than carefully engineered step-level reward functions, and implicitly contain information about which intermediate steps matter.

This motivates three requirements for a learned reasoning reward model: it should improve policy learning, remain useful at inference time, and provide helpful process feedback.

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

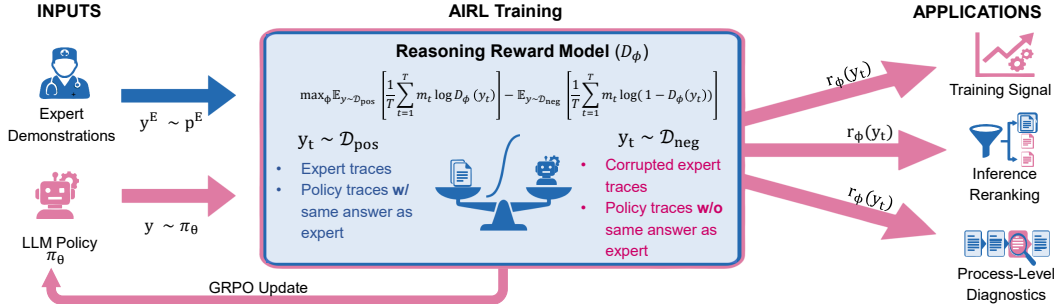


Figure 1: **Eliciting expert reasoning via adversarial inverse reinforcement learning.** The model learns a reasoning reward function from expert demonstrations using adversarial IRL.

We organise the paper around three desiderata for a learned reasoning reward:

1. **(D1) Reasoning training signal.** The reward should provide a usable optimisation signal for policy learning on policy-generated trajectories.
2. **(D2) Inference-time assistance.** The reward should help rank candidate traces under a fixed sampling budget, without requiring policy retraining.
3. **(D3) Error localisation.** The reward should identify where a reasoning trajectory first diverges from a correct path, potentially enabling diagnosis of failure modes.

To meet these desiderata, we formulate reasoning as an inverse reinforcement learning (IRL) problem (Abbeel and Ng, 2004; Ziebart et al., 2004). We still use expert demonstrations, but differently from behaviour cloning: rather than treating expert tokens as direct behaviour targets, we infer a reasoning reward from expert traces and optimise the policy on its own sampled trajectories. The same learned reward is then reused for inference-time reranking, yielding step-level diagnostics at finer granularity.

Contributions. In this work, we make three contributions:

1. We adapt adversarial IRL (Fu et al., 2018; Ho and Ermon, 2016) for large language models to learn reasoning rewards from expert demonstrations at multiple granularities.
2. We show that a single learned reward can be reused across training and inference: as a policy-learning signal and as a reranker under fixed sampling budgets.
3. Across GSM8K, MEDREASON, and MMLU-PRO, we show that learned rewards often outperform SFT as a training signal, improve Best-of-16 reranking by up to 17.4 percentage points, transfer across tasks and backbones in most settings, and provide examples of localisation of reasoning error.

The experimental sections 5.1, 5.2, and 5.3 evaluate empirically the desiderata (D1)–(D3), respectively. A systematic overview of our method for learning a dense reasoning reward model is shown in Figure 1.

2 PROBLEM FORMALISM

We model reasoning as an autoregressive generation process. Given a prompt x drawn from a distribution \mathcal{Q} , the LLM generates a response sequence $y = (y_1, \dots, y_T)$, where each token y_t is selected from a vocabulary \mathcal{V} . Generation is governed by a policy π_θ , which induces a probability distribution over complete reasoning traces:

$$p_\theta(y | x) = \prod_{t=1}^T \pi_\theta(y_t | x, y_{<t}),$$

where $y_{<t}$ denotes the history of tokens generated up to step t .

Reward Structure. Unlike standard Reinforcement Learning (RL) where a ground-truth reward is given, we assume the true evaluation of reasoning is latent. We distinguish between two formulations:

1. **Sparse output reward** $r(y_T | x, y_{<T})$, a scalar signal received only upon completion of the sequence at token T (e.g., correctness of the final answer).
2. **Dense process reward** $r(y_t | x, y_{<t})$, a step-wise signal evaluating the quality of the reasoning process at each token $t \in \{0, \dots, T\}$, or at selected intermediate tokens.

We focus on the latter, defining the total return of a trace as the sum of its dense rewards: $r(y) = \frac{1}{T} \sum_{t=1}^T r(y_t | x, y_{<t})$.

Inverse Reinforcement Learning (IRL). Let π_θ denote the current reasoning policy induced by the LLM parameters θ . We initialise π_θ from a base policy and update θ during Adversarial IRL training. We assume access to a dataset of expert demonstrations $\mathcal{D}_E = \{(x, y^E)\}$ sampled from an expert distribution $p_E(y | x)$. Our goal is to infer a dense reward model $r_\phi(y_t | x, y_{<t})$ (for notation simplicity also referred to as $r_\phi(y_t)$), parametrised with ϕ , that scores the expert’s reasoning, while jointly optimising π_θ against this learned reward. We formulate this as a minimax game:

$$\max_{\phi} \min_{\theta} \mathbb{E}_{y^E \sim p_E} \left[\frac{1}{T_E} \sum_{t=1}^{T_E} r_\phi(y_t^E) \right] - \mathbb{E}_{y \sim \pi_\theta} \left[\frac{1}{T} \sum_{t=1}^T r_\phi(y_t) \right]. \quad (1)$$

At each iteration, trajectories $y \sim \pi_\theta$ are sampled from the current LLM and contrasted with expert trajectories $y^E \sim p_E$. The outer maximisation updates r_ϕ to assign higher scores to expert trajectories, while the inner minimisation updates π_θ . Since the expert term is independent of θ , minimising Eq. (1) with respect to θ is equivalent to maximising $\mathbb{E}_{y \sim \pi_\theta} [\sum_t r_\phi(y_t)]$, i.e., improving the policy under the current learned reward. This notation is used in Section 3 to compare prior methods against **(D1)**–**(D3)**.

3 RELATED WORK

Using the notation from Section 2, prior work differs mainly by (i) which reward signal is available and (ii) whether it natively supports **(D1)**–**(D3)**. We therefore organise related work by objective class and then contrast limitations against our setting.

Imitation objectives. A common objective is to match behaviour to expert trajectories, such as performing supervised fine-tuning (SFT) as a form of knowledge distillation on existing chain-of-thoughts from a teacher model or expert demonstrations (Hinton et al., 2015). Formally, our objective is to maximise the log probability of the generated tokens of the LLM on the demonstrations:

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

This initially improves reasoning quality in many settings (DeepSeek-AI et al., 2025; Kang et al., 2023; Kujanpää et al., 2025; Xu et al., 2025), but it does not learn an explicit reward from policy-generated traces. As a result, it is limited for **(D1)** in our sense (reward-based optimisation on sampled trajectories), and does not directly provide **(D2)** reranking or **(D3)** token-level error localisation (Setlur et al., 2025).

Externally specified reward objectives. Using the notation from Section 2, outcome-supervised reasoning RL optimises

$$\max_{\theta} \mathbb{E}_{y \sim \pi_\theta} [r(y_T)], \quad (2)$$

so only the terminal step T contributes. This outcome reward is typically derived from an external verifier; this paradigm underpins several recent reasoning pipelines (DeepSeek-AI et al., 2025; Uesato et al., 2022; Yu et al., 2025; Yu Yue et al., 2025). On the other hand, process-reward models (PRMs) instead optimise

$$\max_{\theta} \mathbb{E}_{y \sim \pi_\theta} \left[\frac{1}{T} \sum_{t=1}^T r(y_t) \right],$$

with a reward assigned at all intermediate timesteps, or at selected intermediate timesteps for step-wise or interval PRMs. These approaches can be effective (Uesato et al., 2022; Lightman et al., 2023), and search-based variants (e.g., MCTS) can improve exploration (Zelikman et al., 2022; Yuan et al., 2023;

Singh et al., 2024; Hosseini et al., 2024; Silver et al., 2017). However, the main limitation of both these methods in our setting is their dependence on externally specified reward signals, which are often task-specific and costly to construct. In contrast, we learn $r_\phi(y_t | x, y_{<t})$ directly from expert demonstrations without a programmatic verifier, and reuse the same learned reward for training (D1), reranking (D2), and finer-grained diagnostics (D3).

Learned rewards from preferences or demonstrations. Preference-based RL (RLHF-style) learns a scalar reward function r_ϕ , parametrised by ϕ from pairwise preferences between a winning response y^w and a losing response y^l for the same prompt x . With $(x, y^w, y^l) \sim \mathcal{D}_{\text{pref}}$, the reward model is trained by

$$\max_{\phi} \mathbb{E}_{(y^w, y^l) \sim \mathcal{D}_{\text{pref}}} [\log \sigma(r_\phi(y_T^w) - r_\phi(y_T^l))].$$

The learned reward is then used to optimise the policy, similar in the same way as described in the previous paragraph in Equation 2:

$$\max_{\theta} \mathbb{E}_{y \sim \pi_\theta} [r_\phi(y_T)].$$

This setting uses preference data rather than external verifiers (Christiano et al., 2023; Rafailov et al., 2024). On the other hand, IRL-style methods infer reward from demonstrations, often with adversarial objectives:

$$\max_{\phi} \min_{\theta} \mathbb{E}_{y^E \sim p_E} [r_\phi(y_T^E)] - \mathbb{E}_{y \sim \pi_\theta} [r_\phi(y_T)].$$

This line is closest to our setup, with works from traditional RL theory (Ho and Ermon, 2016; Fu et al., 2018; Lin and Zhang, 2018; Li et al., 2017), and two more recent advances when using LLMs (Ye et al., 2025; Lee et al., 2025). However, these formulations operate at the trajectory level, and none have applications in LLM reasoning. Our method instantiates Eq. (1) with process-level $1/T \sum_{t=0}^T r_\phi(y_t)$, enabling a single learned critic to support (D1) training, (D2) inference-time reranking, and (D3) process-level localisation.

We summarise this desiderata-centric comparison in Table 1.

Method Class	Representative Works	Optimisation Objective	Reward Type	External Verifier	(D1)	(D2)	(D3)
SFT / Distillation	Hinton et al. (2015) DeepSeek-AI et al. (2025) Kang et al. (2023) Kujanpää et al. (2025) Xu et al. (2025)	$\max_{\theta} \mathbb{E}_{y^E \sim \mathcal{D}_E} [\frac{1}{T} \sum_t \log \pi_\theta(y_t^E)]$	Implicit (e.g. Logits)	✗	✗	✗	✗
Outcome-supervised RL	DeepSeek-AI et al. (2025) Yu et al. (2025) YuYue et al. (2025)	$\max_{\theta} \mathbb{E}_{y \sim \pi_\theta} [r(y_T)]$	Outcome	✓	(✓)	(✓)	✗
Process-supervised RL	Uesato et al. (2022) Lightman et al. (2023)	$\max_{\theta} \mathbb{E}_{y \sim \pi_\theta} [\frac{1}{T} \sum_{t=1}^T r(y_t)]$	Process	✓	(✓)	(✓)	(✓)
Preference-based RL (RLHF-style)	Christiano et al. (2023) Rafailov et al. (2024)	$\max_{\phi} \mathbb{E}_{(y^w, y^l) \sim \mathcal{D}_{\text{pref}}} [\log \sigma(r_\phi(y_T^w) - r_\phi(y_T^l))]$ then $\max_{\theta} \mathbb{E}_{y \sim \pi_\theta} [r_\phi(y_T)]$	Outcome (learned)	✗	✓	✓	✗
Adversarial imitation IRL (trajectory-level)	Ye et al. (2025) Lee et al. (2025)	$\max_{\phi} \min_{\theta} (\mathbb{E}_{y^E \sim p_E} [r_\phi(y_T^E)] - \mathbb{E}_{y \sim \pi_\theta} [r_\phi(y_T)])$	Outcome (learned)	✗	✓	✓	✗
Expert Reasoning (ours)	this work	$\max_{\phi} \min_{\theta} (\mathbb{E}_{y^E \sim p_E} [\frac{1}{T} \sum_t r_\phi(y_t)] - \mathbb{E}_{y \sim \pi_\theta} [\frac{1}{T} \sum_t r_\phi(y_t)])$	Process (learned)	✗	✓	✓	✓

Table 1: **Related work comparison.** Objectives are written to match the equations in Section 3. (D1): usable reward-based training signal on policy-generated traces. (D2): same reward can assist inference-time reranking. (D3): token-level process diagnostics for error localisation. Parenthesised checkmarks indicate partial support.

4 METHOD

We adapt adversarial IRL (Ho and Ermon, 2016; Fu et al., 2018) to large language modelling. Directly solving Eq. (1) from Section 2 induces nested optimisation loops: an outer maximisation over reward-model parameters ϕ , and, within each outer step, an inner minimisation over policy parameters θ that requires repeated policy updates and fresh rollouts under the current reward. This bilevel procedure is computationally too expensive when using LLMs, so we instead use an alternating iterative approach where a discriminator LLM provides implicit rewards and the reasoning policy is updated via Group Relative Policy Optimisation (GRPO).

4.1 REASONING REWARD MODEL

We train a reasoning reward model, D_ϕ , as a discriminator to distinguish expert reasoning traces from those generated by the policy.

Discriminator Training. A naive adversarial discriminator that treats all expert traces as positive and all policy traces as negative can separate classes using source/style artifacts rather than reasoning quality. To reduce this shortcut, we define labels using answer agreement with the paired expert demonstration. For each prompt, we compare the final answer produced by a policy trace to the final answer in the corresponding expert trace. Importantly, this does not require an external verifier or task-specific checker. Let $\mathcal{O}(y)$ denote the final answer extracted from trace y . We define the positive set \mathcal{D}_{pos} as expert traces plus policy traces that match the expert answer:

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

The base-negative set contains policy traces whose final answer does not match the corresponding expert answer. To improve robustness to subtle errors, we optionally augment negatives with synthetic corruptions $\mathcal{C}(y)$ derived from expert traces (e.g., flipped numerical quantities in mathematics or plausible but incorrect medical reasoning):

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

Contrasting with prior work, outcome- and process-supervised RL typically rely on externally specified reward signals or verifiers (DeepSeek-AI et al., 2025; Uesato et al., 2022; Lightman et al., 2023), and preference-based methods rely on pairwise preference labels (Christiano et al., 2023; Rafailov et al., 2024). Trajectory-level adversarial imitation often separates expert and policy traces by source (Ye et al., 2025; Lee et al., 2025), which can encourage style discrimination. Our construction instead uses demonstration-only answer consistency and includes answer-consistent policy traces as positives, encouraging the discriminator to score reasoning quality rather than provenance/style while avoiding external verifier dependence.

Granularities. The discriminator $D_\phi(y_t)$ is trained as a token-level classifier. We introduce a binary mask $m_t \in \{0, 1\}$ to control reward granularity, i.e., when supervision is applied along the trajectory. We vary this granularity because it directly sets the credit-assignment scale: coarser masks provide lower-variance, trajectory-level signals that are typically easier to optimise (D1), while finer masks provide higher-resolution process feedback for localising errors (D3) but are harder to optimise robustly in adversarial training.

This also makes the contrast to prior reasoning RL explicit. Outcome-supervised methods correspond to terminal-only rewards (DeepSeek-AI et al., 2025), whereas process-reward methods typically assume externally specified step-level labels or verifiers (Uesato et al., 2022; Lightman et al., 2023). In our setting, the reward is learned from demonstrations and the same AIRL pipeline is evaluated across granularities via m_t , enabling a direct sparse-to-dense comparison without changing the supervision source.

We evaluate four levels: (1) *Sparse Outcome* ($m_t = 1$ only at $t = T$); (2) *Interval* ($m_t = 1$ every k tokens); (3) *Fully Dense* ($m_t = 1$ for all t); and (4) *Step-wise* ($m_t = 1$ at sentence / reasoning delimiters; reported in the Appendix). The objective is to minimise the masked binary cross-entropy:

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

Implicit Reward. After updating the discriminator, we derive the implicit reward from its logits:

$$v_\phi(y_t) = \log D_\phi(y_t) - \log(1 - D_\phi(y_t)). \quad (4)$$

4.2 POLICY LEARNING WITH DENSE REASONING REWARDS

We optimise the policy π_θ using GRPO (Shao et al., 2024), adapted here for dense process rewards. At each policy-update iteration, we first freeze a snapshot of the current policy parameters and denote it by θ_{old} (the rollout/reference policy for that iteration). For a given prompt x , we then sample a group of G outputs $\{y^{(g)}\}_{g=1}^G$ from $\pi_{\theta_{\text{old}}}$. We subsequently update θ while keeping θ_{old} fixed.

Reward Densification (Backfilling). The raw signal $v_\phi(y_t)$ is only defined at positions where $m_t = 1$. For *Sparse*, *Step-wise*, and *Interval* masks, we therefore densify by backfilling: each unmasked token inherits the reward of its next supervised checkpoint. Formally, with

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

For the fully *dense* case, $r_\phi(y_t) = v_\phi(y_t)$.

We use this piecewise-constant densification for two reasons: (i) it keeps supervision demonstration-only, and (ii) it allows a controlled sparse-to-dense comparison under the same AIRL objective. This differs from prior reasoning pipelines that typically obtain dense supervision from externally specified process labels or verifiers (Uesato et al., 2022; Lightman et al., 2023).

Subsequently, to avoid exploding positive and negative reward updates and avoid reward hacking (see Section 5.1), we clip dense rewards to a symmetric bound:

$$r_\phi(y_t) \leftarrow \text{clip}(r_\phi(y_t), -\beta, \beta), \quad (5)$$

where $\beta > 0$ is a scalar threshold. This is crucial because v_ϕ is a logit and therefore unbounded: as $D_\phi \rightarrow 0$ or 1 , $|v_\phi| \rightarrow \infty$, which can cause numerical instability and over-optimisation.

Group Standardisation. To reduce the variance of the policy-gradient estimator in the policy objective (Eq. 7), we standardise sequence-level rewards within each sampled group, following Cetin et al. (2025) and Cui et al. (2025). Here, a sequence means one sampled model output (reasoning trace) for the current prompt, denoted $(y_1^{(g)}, \dots, y_{T_g}^{(g)})$, with $g \in \{1, \dots, G\}$. For each such sequence g , we compute its mean dense reward:

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

We then compute the group mean μ and standard deviation σ over $\{\bar{r}^{(g)}\}_{g=1}^G$, and define the normalised sequence-level advantage:

$$A^{(g)} = \frac{\bar{r}^{(g)} - \mu}{\sigma + \epsilon}. \quad (6)$$

This normalised value is broadcast to tokens in the same sequence, i.e., $A_t^{(g)} := A^{(g)}$ for all t . Finally, the policy is updated using the PPO clip loss on the policy samples $\mathcal{D}_P = \{y^{(g)}\}$:

$$\mathcal{L}_\pi(\theta) = \mathbb{E}_{y \sim \mathcal{D}_P} \left[\frac{1}{T} \sum_{t=1}^T \min \left(w_t A_t, \text{clip}(w_t, 1 - \epsilon, 1 + \epsilon) A_t \right) \right], \quad (7)$$

where

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

with θ_{old} fixed to the pre-update policy parameters used to generate \mathcal{D}_P in the current iteration.

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 introduce two complementary stabilisation mechanisms. First, we maintain replay buffers \mathcal{B}_{pos} and \mathcal{B}_{neg} that accumulate positive and negative traces across iterations; each discriminator update samples mini-batches from these buffers rather than solely from the current policy rollouts, broadening the distribution of training examples 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. Together, these modifications keep the discriminator well-calibrated throughout adversarial training, thereby reducing the frequency of mode collapse observed in adversarial learning.

The training algorithm for adversarial inverse reinforcement learning for reasoning reward models is summarised in Appendix Algorithm 1.

5 EXPERIMENTS

We evaluate our learning of reasoning reward models via 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. All three provide expert reasoning traces that align with our demonstration-driven IRL setting.

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). Moreover, we report results for weaker models (Llama3.2-3B and Qwen2.5-3B) in the Appendix. We always instantiate a policy model as a text generator and the reasoning reward model as a token classifier. Based on the different densities of reasoning reward (*sparse*, *interval*, and *dense*), we backfill token rewards as described in Section 4.2.

Each experimental subsection directly evaluates one desideratum from Section 1. **(D1)** (Section 5.1) tests training-signal quality via final policy after AIRL training, compared to SFT and outcome-reward RL. **(D2)** (Section 5.2) tests inference-time assistance via reranking gains under a fixed sample budget. **(D3)** (Section 5.3) tests process-level utility via cross-task/backbone transfer and token-level localisation of the erroneous reasoning step. Implementation details appear in Appendix A.

5.1 USING THE LEARNED REWARD AS A TRAINING SIGNAL

RQ1: *Can 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 LLM, we treat the generator as a policy π_θ and train it using learned reasoning rewards from our IRL procedure, with *sparse*, *interval* ($k = 15$), and *dense* reward variants. Following Ye et al. (2025) and Lee et al. (2025), we perform a 250-step warm-up. Subsequently, we jointly optimise the policy and reward models for each dataset and evaluate the resulting policies using pass@1. We compare against two baselines: GRPO with outcome rewards from a ground-truth checker, which serves as an empirical upper bound because it directly optimises the same task-level correctness signal used for evaluation; and supervised fine-tuning (SFT) on expert traces, which provides an imitation-learning baseline. This distinction is important in IRL: our method must infer the reward from demonstrations, whereas evaluation still uses the underlying ground-truth correctness signal. Results are shown in Table 2.

Method	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
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	65.0 ± 2.2	73.4 ± 2.1	56.3 ± 2.6
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	53.0 ± 1.9	63.5 ± 1.9	<u>57.6</u> ± 1.8
Ours (<i>Sparse</i>)	85.8 ± 1.5	80.6 ± 1.6	90.4 ± 1.5	<u>48.5</u> ± 2.2	<u>43.3</u> ± 1.9	55.6 ± 2.2	25.2 ± 1.8	73.1 ± 1.9	66.1 ± 2.1
Ours (<i>Interval</i>)	<u>78.8</u> ± 1.6	<u>67.6</u> ± 1.9	87.8 ± 1.6	50.6 ± 1.9	36.6 ± 1.8	53.5 ± 2.4	63.8 ± 2.0	<u>71.4</u> ± 1.8	52.1 ± 2.5
Ours (<i>Dense</i>)	38.4 ± 2.0	64.6 ± 2.1	89.6 ± 1.5	43.8 ± 2.0	37.9 ± 1.8	55.1 ± 2.3	58.3 ± 2.2	69.8 ± 2.0	53.7 ± 2.1

Table 2: **Pass@1 Performance (%)**. **Bold** indicates the best performance, and underlined the second best, compared between SFT and our methods. Outcome reward is provided as an empirical upper bound. Values are reported as mean ± half-width of the 95% confidence interval.

Analysis. Table 2 shows that learned rewards are effective overall: taking the best reward variant per backbone–dataset pair, AIRL outperforms SFT in 8/9 settings. However, performance is backbone- and granularity-sensitive. A key failure case is Qwen2.5-7B on MEDREASON: *Sparse* drops to 25.2% versus 53.0% for SFT, while the same backbone recovers with finer rewards (*Interval*: 63.8%, *Dense*: 58.3%). We attribute this drop to unsuccessful adversarial training in that specific run rather than a persistent backbone limitation; Appendix Figure 8 (MedReason row) shows unstable training dynamics and no sustained improvement in task correctness for this setting. A second difficult

case is Llama3.1-8B on MMLU-PRO, where all learned-reward variants remain below SFT (43.3/36.6/37.9 vs 47.2). Notably, the outcome-reward upper bound is also close (48.4), suggesting limited headroom and a harder reward-inference regime on this heterogeneous benchmark. Sensitivity to model choice is therefore substantial and interacts with granularity. For example, on GSM8K, Qwen2.5-7B ranges from 85.8 (*Sparse*) to 38.4 (*Dense*), whereas Qwen3-4B stays consistently high (87.8-90.4). This motivates reporting multiple backbones rather than a single-model result.

Effect of reward granularity. Across datasets, *Sparse* is the most reliable variant and is often the strongest performer, especially on GSM8K. *Interval* is also competitive, and is the best variant for Qwen2.5-7B on both MMLU-PRO (50.6%) and MEDREASON (63.8%). By contrast, *Dense* is less stable: although it remains competitive in some settings, it also shows the largest drops, such as 38.4% on GSM8K for Qwen2.5-7B. This likely reflects the fact that discriminating experts from policy trajectories is easier when the full trajectory is available, as in the *Sparse* setting, than when only local segments are scored.

Training dynamics. To better understand optimisation, Figure 2 shows Qwen2.5-7B on MMLU-PRO. The three panels should be interpreted jointly: the left panel shows the surrogate signal produced by the learned critic, the middle panel shows discriminator accuracy (how separable expert and policy traces are), and the right panel shows the true task metric (pass@1 correctness).

In adversarial training, discriminator accuracy moving toward 0.5 is desirable only when the right panel improves at the same time: this indicates policy traces are becoming expert-like rather than merely confusing the critic. If reward rises (2a) or discriminator accuracy drops (2b) without a sustained gain in task correctness (2c), this indicates reward-task decoupling.

This also clarifies granularity differences: *Sparse* rewards are typically more stable because supervision is tied to full-trajectory outcomes, giving lower-variance updates; *Dense* rewards provide finer local feedback but are more sensitive to optimisation noise and can overfit local cues. In this figure, *Sparse* is strongest because it yields the most stable increase in task correctness while the discriminator settles near the balanced regime.

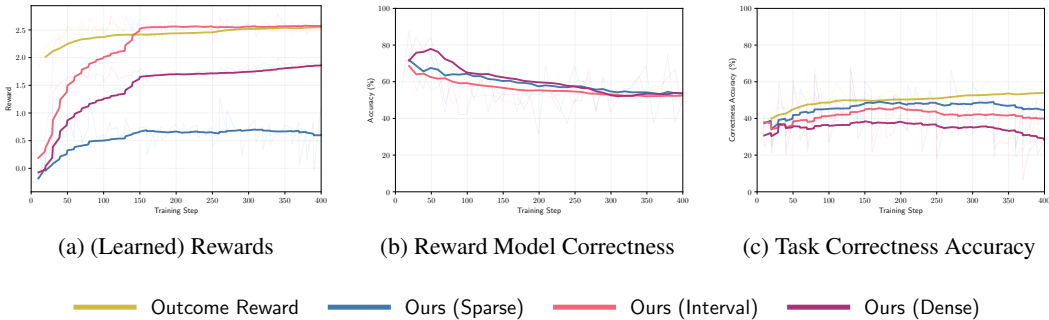


Figure 2: **Training behaviour of reward, reward-model correctness, and task correctness** for Qwen2.5-7B on MMLU-PRO. Subfigure 2a: learned reward. Subfigure 2b: reward-model correctness (discriminator accuracy). Subfigure 2c: task correctness accuracy.

For completeness, Appendix Table 5 reports pass@1 for smaller models and *step-wise* rewards, confirming the same overall trends. Additional training curves for all backbones are shown in Appendix Figures 6-10. Moreover, we perform additional ablation experiments in AppendixC, where we inspect the effect of expert corruptions, group size, clipping, and the effect of keeping the IRL objective by purely separating policy from expert demonstrations.

Takeaway. RQ1 is answered positively, with caveats: learned AIRL rewards improve over SFT in most settings, but gains are sensitive to backbone and reward granularity. Sparse rewards are usually the most stable, while denser rewards can be more brittle.

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. To study this, we use the learned reasoning reward at inference time, after training it with AIRL. For each prompt, we sample $N = 16$ candidate traces from the policy, score each trace by its mean learned reward, and select the top-ranked candidate (Best-of-16). We compare this against the standard pass@1 baseline, i.e. random selection from the same sample set. Results across GSM8K, MEDREASON, and MMLU-PRO are reported in Table 3. In the Appendix, we report all scores in Table 6 and the AUROC and ECE scores in Table 7.

Backbone	Method	GSM8K			MEDREASON			MMLU-PRO		
		Random	Reward	Δ (pp)	Random	Reward	Δ (pp)	Random	Reward	Δ (pp)
Qwen2.5-7B	Sparse	85.8 ± 1.5	88.8 ± 1.7	(↑ +3.0)	25.2 ± 1.8	32.3 ± 2.3	(↑ +7.0)	48.5 ± 2.2	51.5 ± 2.5	(↑ +3.0)
	Interval	78.8 ± 1.6	82.5 ± 2.0	(↑ +3.7)	63.8 ± 2.0	63.4 ± 2.4	(↓ -0.4)	50.6 ± 1.9	53.1 ± 2.5	(↑ +2.5)
	Dense	38.4 ± 2.0	55.7 ± 2.6	(↑ +17.4)	58.3 ± 2.2	57.7 ± 2.5	(↓ -0.6)	43.8 ± 2.0	46.0 ± 2.5	(↑ +2.2)
Llama3.1-8B	Sparse	80.6 ± 1.6	82.1 ± 2.1	(↑ +1.5)	73.1 ± 1.9	74.6 ± 2.3	(↑ +1.5)	43.3 ± 1.9	42.3 ± 2.4	(↓ -1.0)
	Interval	67.6 ± 1.9	78.2 ± 2.2	(↑ +10.7)	71.4 ± 1.8	75.3 ± 2.2	(↑ +3.9)	36.6 ± 1.8	38.1 ± 2.4	(↑ +1.6)
	Dense	64.6 ± 2.1	71.1 ± 2.4	(↑ +6.5)	69.8 ± 2.0	73.0 ± 2.2	(↑ +3.2)	37.9 ± 1.8	39.7 ± 2.5	(↑ +1.9)
Qwen3-4B	Sparse	90.4 ± 1.5	93.0 ± 1.4	(↑ +2.5)	66.1 ± 2.1	70.5 ± 2.3	(↑ +4.4)	55.6 ± 2.2	62.1 ± 2.4	(↑ +6.4)
	Interval	87.8 ± 1.6	91.2 ± 1.5	(↑ +3.5)	52.1 ± 2.5	51.6 ± 2.6	(↓ -0.5)	53.5 ± 2.4	62.2 ± 2.4	(↑ +8.7)
	Dense	89.6 ± 1.5	91.4 ± 1.5	(↑ +1.8)	53.7 ± 2.1	61.6 ± 2.4	(↑ +7.9)	55.1 ± 2.3	59.2 ± 2.4	(↑ +4.1)

Table 3: **Best-of-N Reranking Performance (%)**. Comparison of reward model reranking vs. random reranking baseline (in expectation same as pass@1). Values are percentages. Δ indicates percentage-point improvement over Pass@1. Blue is positive, purple is negative.

Analysis. Table 3 shows that the learned reward is an effective reranker overall. Reward-guided reranking improves pass@1 in 23 out of 27 settings, often by a substantial margin. The gains are particularly strong on GSM8K and MMLU-PRO: for instance, Qwen2.5-7B with the *Dense* reward improves from 38.4% to 55.7% on GSM8K (+17.4 pp), Llama3.1-8B with the *Interval* reward improves from 67.6% to 78.2% (+10.7 pp), and Qwen3-4B with the *Interval* reward improves from 53.5% to 62.2% on MMLU-PRO (+8.7 pp). There are, however, a small number of negative cases, mostly on MEDREASON, showing that reranking is highly effective but not uniformly beneficial. The full table is can be found in Appendix Table 6.

Additional reranking baselines. We compare against log-probability, majority voting, and reward-weighted majority voting across sample budgets $N \in \{2, 3, 5, 8, 16\}$, using candidates generated from the SFT model so that logits remain independent of the reward model. Figure 3 shows the results for Qwen3-4B on GSM8K. Three trends emerge. First, log-probability offers little improvement over the baseline, reflecting the limitations of the underlying SFT model. Secondly, at small sample budgets, reward-based reranking already performs strongly, both on its own and when combined with majority voting. Thirdly, at larger budgets, reward-weighted majority voting performs best overall, suggesting that the learned reward complements answer consistency. Appendix Tables B.5-B.5 show the same pattern.

Reward separation. To understand why reranking works, Figure 4 analyses the reward distribution assigned to correct and incorrect traces for Qwen3-4B with the *Interval* reward on MMLU-PRO. The two distributions are clearly separated, with correct answers receiving consistently higher scores ($t = 87.43, p < 0.001$). This separation translates directly into better reranking performance: Figure 4b shows that reward-guided reranking

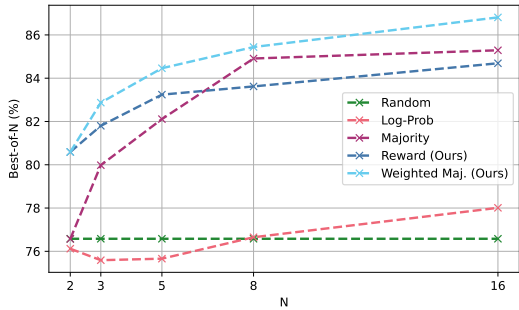


Figure 3: **Comparison of reranking baselines** for Qwen3-4B with *interval* rewards on GSM8K. Combining learned rewards with majority ranking yields the greatest improvement.

improves $\text{pass}@k \mid 16$ over random ranking across all $k \in \{1, 3, 5, 10\}$, with the largest gains at low k . In the Appendix, we report the complete stratified reward distributions for GSM8K (Figures 11–14), MEDREASON (Figures 15–18), and MMLU-PRO (Figures 19–22).

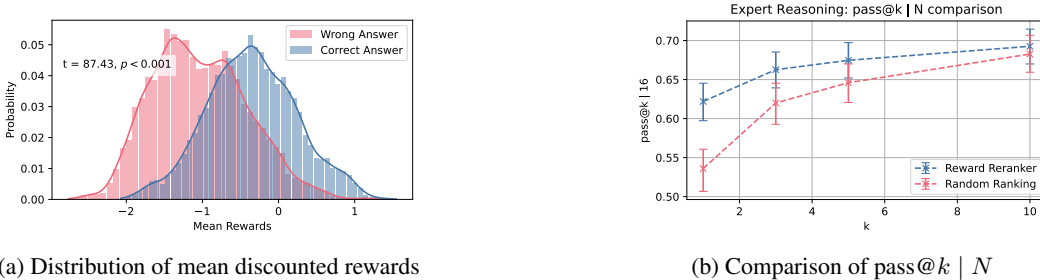


Figure 4: **Benefit of the reasoning reward at inference** for Qwen3–4B with *interval* rewards. Left (4a): reward distributions for correct versus incorrect answers. Right (4b): $\text{pass}@k \mid 16$ using reward-guided reranking versus random ranking on MMLU-PRO.

Takeaway. RQ2 is answered positively: the learned reward is a strong inference-time ranker, improving $\text{pass}@1$ in most settings and often by a large margin. Reward-only reranking is already effective at small budgets, while combining reward with majority voting performs best as the sample budget grows.

5.3 GENERALISATION OF REASONING REWARD MODELS AND ERROR LOCALISATION

RQ3: *Do the learned reasoning rewards generalise beyond the backbone and task on which they were trained, and do they provide process-level feedback that localises where reasoning fails?*

Experimental setup. We study transfer by taking natural generations from Qwen2 . 5–7B SFT models and reranking them with *interval* reward models trained on different task distributions (GSM8K, MEDREASON, and MMLU-PRO) and different backbones (Llama3 . 1–8B and Qwen3–4B). Table 4 reports Best-of-16 gains over the corresponding SFT $\text{pass}@1$ baseline. To assess process-level supervision, we then inspect token-level dense rewards on paired correct and incorrect traces from the same GSM8K question, shown in Figure 5.

Transfer across tasks and backbones. Table 4 shows that transfer is strong overall: 16 out of 18 source–target combinations yield positive reranking gains, including many off-diagonal improvements across both tasks and backbones. For example, the Llama3 . 1–8B reward trained on MEDREASON improves reranking not only on MEDREASON itself (+7.8 pp), but also if trained on GSM8K (+4.6 pp) and especially if trained on MMLU-PRO (+16.4 pp). Likewise, the Qwen3–4B reward trained on GSM8K transfers positively to MMLU-PRO (+9.3 pp). A clear negative case is the Qwen3–4B reward trained on MEDREASON and evaluated on GSM8K (−2.8 pp), suggesting that the learned signal is broadly reusable, but not fully invariant across domains and backbones.

Task	Llama3 . 1–8B			Qwen3–4B		
	GSM8K	MEDREASON	MMLU-PRO	GSM8K	MEDREASON	MMLU-PRO
GSM8K	<i>↑12.1</i>	↑7.6	↑11.3	↑13.0	↓2.8	↑8.8
MEDREASON	↑4.6	↑7.8	↑16.4	↑5.1	↑5.7	↑9.5
MMLU-PRO	↑3.0	↑3.7	↑4.4	↑9.3	↓0.2	↑2.0

Table 4: **Reward Model Transferability (Best-of-16 Δ over random reranking).** Generations are sourced from Qwen2 . 5–7B SFT models. We score them using reward models (*interval* granularity) trained on different task distributions. Diagonal entries (italic) represent in-distribution performance.

Error localisation on natural traces. Figure 5 shows that the dense reward remains positive on locally correct intermediate steps, even within an overall incorrect trajectory, and turns sharply negative at the first erroneous step. In the incorrect GSM8K trace, the early steps correctly reconstruct

the number of vacuum cleaners before each sale and receive positive reward, but the signal drops exactly when the model makes the faulty computation " $12 - 2/3(12) = 4$ ". This behaviour suggests that the reward is not merely reacting to the final answer, but is sensitive to where the reasoning process first diverges. We provide more qualitative examples in Appendix Figures 23a-28d. We especially appreciate Figure 27b showing the reward becoming negative when proposing "being married" as a reduced risk for Alzheimer's.

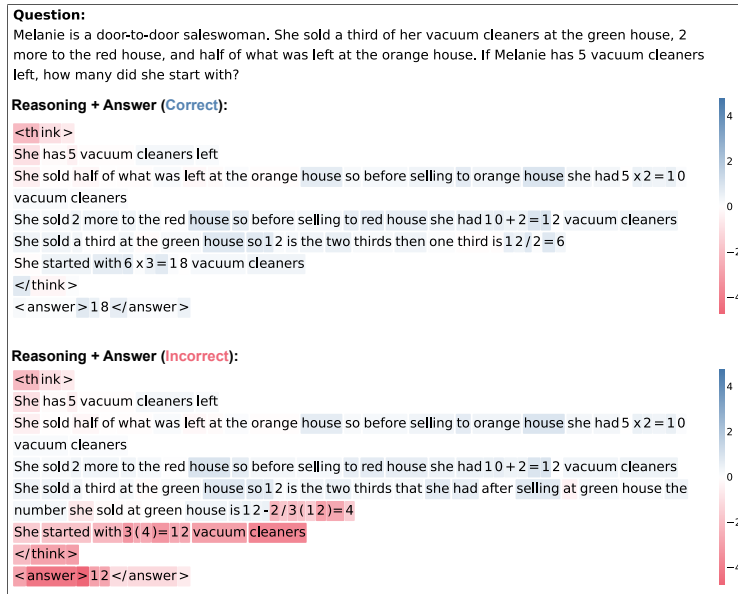


Figure 5: **Error localisation of incorrect reasoning.** Dense reward on a correct (top) and incorrect (bottom) generation for the same GSM8K test question, using the *dense* Qwen2.5-7B reasoning reward model. The reward remains positive on valid intermediate steps, then drops sharply at the first erroneous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

Takeaway. RQ3 is answered positively: the learned reward transfers well across tasks and backbones, and it can localise the step at which a reasoning trace fails. However, transfer is not uniform, indicating that some domain- and backbone-dependent effects remain.

6 LIMITATIONS

Our approach has several limitations. First, we observe a trade-off between reward density and optimisation stability: while sparse, step-wise signals are generally robust, dense signals are more prone to adversarial mode collapse and non-stationarity, and can occasionally lead to a decoupling of reward and correctness during training. Moreover, this AIRL approach incurs higher computational costs due to alternating adversarial updates between the two models. Finally, our evaluation is restricted to arithmetic, medical reasoning, and scientific question answering, leaving open-ended generation and long-context domains for future work.

7 CONCLUSION

In this work, we introduced an inverse reinforcement learning framework that distils (partially) dense reasoning rewards directly from expert demonstrations. The learned reasoning critic serves a dual purpose: it acts as a supervisor that trains policies to outperform supervised fine-tuning baselines, and as an inference-time discriminator that significantly improves performance via reranking. Beyond accuracy, the dense reward provides interpretable process supervision, capable of distinguishing valid reasoning steps and localising the precise moment of error. While balancing reward granularity with optimisation stability remains a challenge, our results demonstrate that learned reasoning reward models effectively support training, verification, and diagnosis within a single, unified framework.

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 Kobalczuk, Paulius Rauba, Byung-Kwan Lee, 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

- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, 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, 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, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanbiao Zhao, 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, Yanhong Xu, 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: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, January 2025. URL <http://arxiv.org/abs/2501.12948>. arXiv:2501.12948 [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.
- 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].
- Jonathan Ho and Stefano Ermon. Generative Adversarial Imitation Learning, June 2016. URL <http://arxiv.org/abs/1606.03476>. arXiv:1606.03476 [cs].
- 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].
- 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].
- Qiyang 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, Jiase 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, Qiyang Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiase 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.
- 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].
- 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=1NAyUngGFK>.
- 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.
- Jiahao Lin and Zongzhang Zhang. ACGAIL: Imitation Learning About Multiple Intentions with Auxiliary Classifier GANs. In Xin Geng and Byeong-Ho Kang, editors, *PRICAI 2018: Trends in Artificial Intelligence*, pages 321–334, Cham, 2018. Springer International Publishing. ISBN 978-3-319-97304-3. doi: 10.1007/978-3-319-97304-3_25.
- Yunzhu Li, Jiaming Song, and Stefano Ermon. InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations, November 2017. URL <http://arxiv.org/abs/1703.08840>. arXiv:1703.08840 [cs].
- 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].
- 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].
- 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, Ziyang 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, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.
- Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann, editors, *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR, 07–08 Apr 2022. URL <https://proceedings.mlr.press/v174/pal22a.html>.
- Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace’s Transformers: State-of-the-art Natural Language Processing, July 2020. URL <http://arxiv.org/abs/1910.03771>. arXiv:1910.03771 [cs].
- Michael Han Daniel Han and Unsloth team. Unsloth, 2023. URL <http://github.com/unslothai/unsloth>.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL <https://arxiv.org/abs/2106.09685>.

A 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 1’500 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.

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} . 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. We follow the standard GSM8K and the filtered MEDREASON train-val-test split. 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. Tokenisation uses the native tokeniser of each backbone. For evaluation, we decode with temperature $T = 1.0$ and `top_p = 0.95` unless otherwise noted. We validate every run after 100 steps, save the model with the best validation accuracy, and use these weights to evaluate on the test set.

System Prompt

```
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>
```

Inference time scoring. At inference time, we draw $N = 16$ samples per prompt, compute the mean discounted reward over the answer tokens for each sample, and rerank by this score. We report `pass@k | N`, the fraction of prompts for which at least one of the top k ranked samples is correct when N samples are available. Unless otherwise noted $N = 16$ and $k \in \{1, 3, 5, 10\}$.

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 artificially create wrong reasoning traces that lead to the wrong results on the purpose with ChatGPT-5.2, by providing in-context the correct reasoning trace and the right answer, and instruct it to corrupt it.

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.

A.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.

A.2 ALGORITHM

Algorithm 1 Adversarial inverse reasoning with (partially) dense rewards

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

- 1: Initialise policy π_θ , discriminator D_ϕ , and replay buffers $\mathcal{B}_{\text{pos}} \leftarrow \emptyset$, $\mathcal{B}_{\text{neg}} \leftarrow \emptyset$
- 2: **Warm-up Phase:**
- 3: **for** $j \leftarrow 1$ to N_{warm} **do**
- 4: Sample prompts $x \sim \mathcal{Q}$ and generate $\mathcal{D}_P \sim \pi_{\theta_{\text{mit}}}(\cdot | x)$
- 5: $\mathcal{D}_{\text{pos}} \leftarrow \mathcal{D}_E \cup \{y \in \mathcal{D}_P \mid \mathcal{O}(y) = \mathcal{O}(y^E)\}$
- 6: $\mathcal{D}_{\text{neg}} \leftarrow \{y \in \mathcal{D}_P \mid \mathcal{O}(y) \neq \mathcal{O}(y^E)\} \cup \{\mathcal{C}(y^E) \mid y^E \in \mathcal{D}_E\}$
- 7: Update D_ϕ to minimise Eq. (3) using \mathcal{D}_{pos} and \mathcal{D}_{neg}
- 8: **end for**
- 9: **Adversarial Training Phase:**
- 10: **for** $i \leftarrow 1$ to N_{step} **do**
- 11: Sample prompts $x \sim \mathcal{Q}$
- 12: Generate group $\mathcal{D}_P \leftarrow \{y^{(g)}\}_{g=1}^G \sim \pi_\theta(\cdot | x)$
- 13: Construct training sets based on answer correctness:
- 14: $\mathcal{D}_{\text{pos}} \leftarrow \mathcal{D}_E \cup \{y \in \mathcal{D}_P \mid \mathcal{O}(y) = \mathcal{O}(y^E)\}$
- 15: $\mathcal{D}_{\text{neg}} \leftarrow \{y \in \mathcal{D}_P \mid \mathcal{O}(y) \neq \mathcal{O}(y^E)\} \cup \{\mathcal{C}(y^E) \mid y^E \in \mathcal{D}_E\}$
- 16: $\mathcal{B}_{\text{pos}} \leftarrow \mathcal{B}_{\text{pos}} \cup \mathcal{D}_{\text{pos}}$; $\mathcal{B}_{\text{neg}} \leftarrow \mathcal{B}_{\text{neg}} \cup \mathcal{D}_{\text{neg}}$ ▷ Update replay buffers
- 17: **Reward Model Update:**
- 18: **for** $k \leftarrow 1$ to N_{disc} **do**
- 19: Sample mini-batches $\tilde{\mathcal{D}}_{\text{pos}} \sim \mathcal{B}_{\text{pos}}$ and $\tilde{\mathcal{D}}_{\text{neg}} \sim \mathcal{B}_{\text{neg}}$
- 20: Update D_ϕ to minimise Eq. (3) using $\tilde{\mathcal{D}}_{\text{pos}}$ and $\tilde{\mathcal{D}}_{\text{neg}}$
- 21: **end for**
- 22: **Policy Update:**
- 23: Get dense rewards $r_\phi(y_t^{(g)})$ via backfilling logits (Eq. 4 and Eq. 5)
- 24: Compute advantages $A_t^{(g)}$ via group standardisation (Eq. 6)
- 25: Optimise π_θ using GRPO loss on \mathcal{D}_P (Eq. 7)
- 26: **end for**

B ADDITIONAL RESULTS

B.1 COMPLETE PERFORMANCE TABLE

Method	GSM8K Pass@1	MMLU-PRO Pass@1	MEDREASON Pass@1
Qwen2.5-3B			
Verifiable Reward	83.6 [81.8, 85.4]	43.4 [41.5, 45.3]	55.7 [53.5, 57.7]
SFT	63.8 [62.0, 65.6]	40.0 [38.2, 41.8]	46.4 [44.6, 48.2]
Ours (<i>Sparse</i>)	78.5 [76.8, 80.1]	40.3 [38.5, 42.1]	49.4 [47.5, 51.4]
Ours (<i>Step-wise</i>)	<u>65.2</u> [63.3, 67.0]	36.4 [34.5, 38.3]	35.2 [33.3, 37.1]
Ours (<i>Dense</i>)	43.9 [42.0, 46.0]	28.8 [26.9, 30.8]	<u>48.4</u> [46.2, 50.6]
Llama3.2-3B			
Verifiable Reward	79.5 [77.6, 81.3]	37.1 [35.3, 38.9]	71.5 [69.3, 73.7]
SFT	62.2 [60.3, 64.2]	36.5 [34.8, 38.3]	57.6 [55.6, 59.4]
Ours (<i>Sparse</i>)	71.3 [69.5, 73.2]	30.1 [28.0, 32.2]	<u>63.8</u> [61.8, 65.8]
Ours (<i>Step-wise</i>)	<u>66.9</u> [64.9, 69.0]	<u>31.6</u> [29.8, 33.3]	66.8 [64.5, 69.2]
Ours (<i>Dense</i>)	50.9 [48.8, 53.0]	0.0* [0.0, 0.0]	57.9 [55.8, 59.9]
Qwen2.5-7B			
Verifiable Reward	89.0 [87.5, 90.5]	53.5 [51.4, 55.5]	65.0 [62.8, 67.1]
SFT	<u>70.1</u> [68.5, 71.7]	<u>48.1</u> [46.4, 50.1]	<u>53.0</u> [51.1, 54.9]
Ours (<i>Sparse</i>)	85.8 [84.4, 87.3]	48.5 [46.4, 50.7]	25.2 [23.5, 27.1]
Ours (<i>Step-wise</i>)	64.5 [62.4, 66.6]	45.6 [43.6, 47.6]	40.9 [39.3, 42.6]
Ours (<i>Dense</i>)	38.4 [36.4, 40.4]	43.8 [41.7, 45.8]	58.3 [56.2, 60.6]
Llama3.1-8B			
Verifiable Reward	83.3 [81.6, 84.9]	48.4 [46.6, 50.5]	73.4 [71.4, 75.5]
SFT	66.6 [64.5, 68.7]	47.2 [45.3, 49.2]	63.5 [61.6, 65.4]
Ours (<i>Sparse</i>)	<u>80.6</u> [78.9, 82.2]	<u>43.3</u> [41.4, 45.3]	73.1 [71.2, 74.9]
Ours (<i>Step-wise</i>)	80.8 [79.2, 82.5]	42.6 [40.7, 44.5]	69.3 [67.4, 71.3]
Ours (<i>Dense</i>)	64.6 [62.5, 66.6]	37.9 [36.2, 39.8]	<u>69.8</u> [67.7, 71.7]
Qwen3-4B			
Verifiable Reward	91.6 [90.2, 92.9]	57.1 [54.8, 59.3]	56.3 [53.8, 59.0]
SFT	76.6 [74.9, 78.3]	53.9 [51.9, 55.9]	57.6 [55.8, 59.5]
Ours (<i>Sparse</i>)	90.4 [89.0, 91.9]	55.6 [53.4, 57.8]	66.1 [64.1, 68.3]
Ours (<i>Step-wise</i>)	87.4 [85.8, 88.9]	54.8 [52.5, 57.0]	<u>60.1</u> [57.7, 62.6]
Ours (<i>Dense</i>)	<u>89.6</u> [88.1, 91.1]	<u>55.1</u> [52.7, 57.3]	53.7 [51.5, 55.8]

Table 5: **Pass@1 Performance (%)**. **Bold** indicates the best performance compared between SFT and our methods. Verifiable reward is provided as a reference upper bound. * symbolises an adversarial mode collapse (results grayed out). The values inside brackets indicate the 95% confidence interval.

B.2 TRAINING BEHAVIOUR

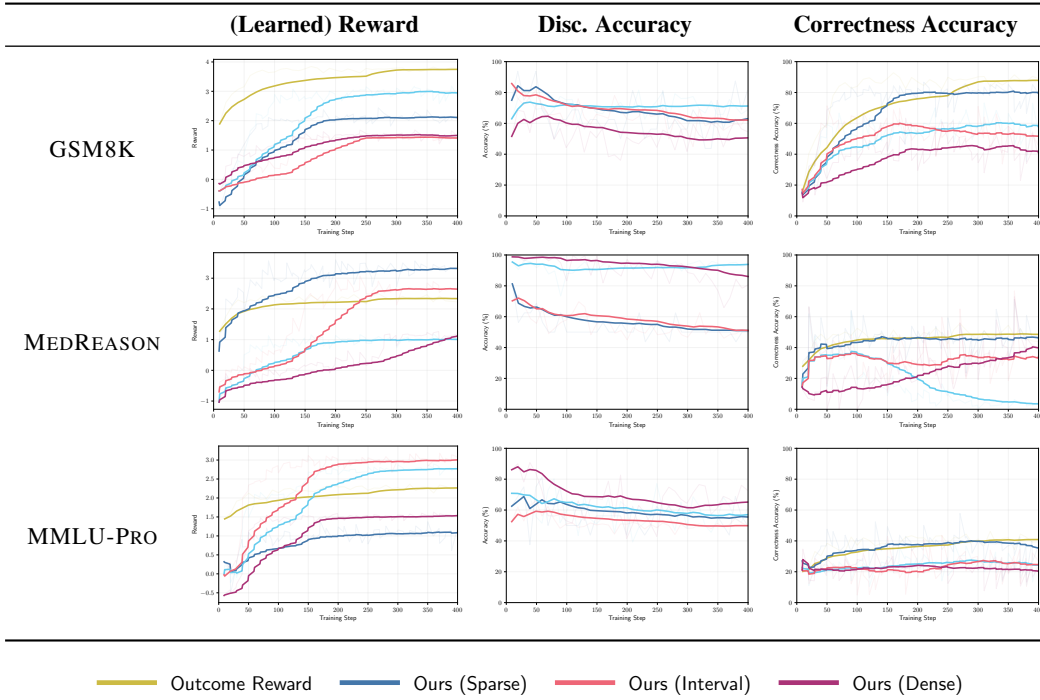


Figure 6: Training dynamics for Qwen2.5-3B.

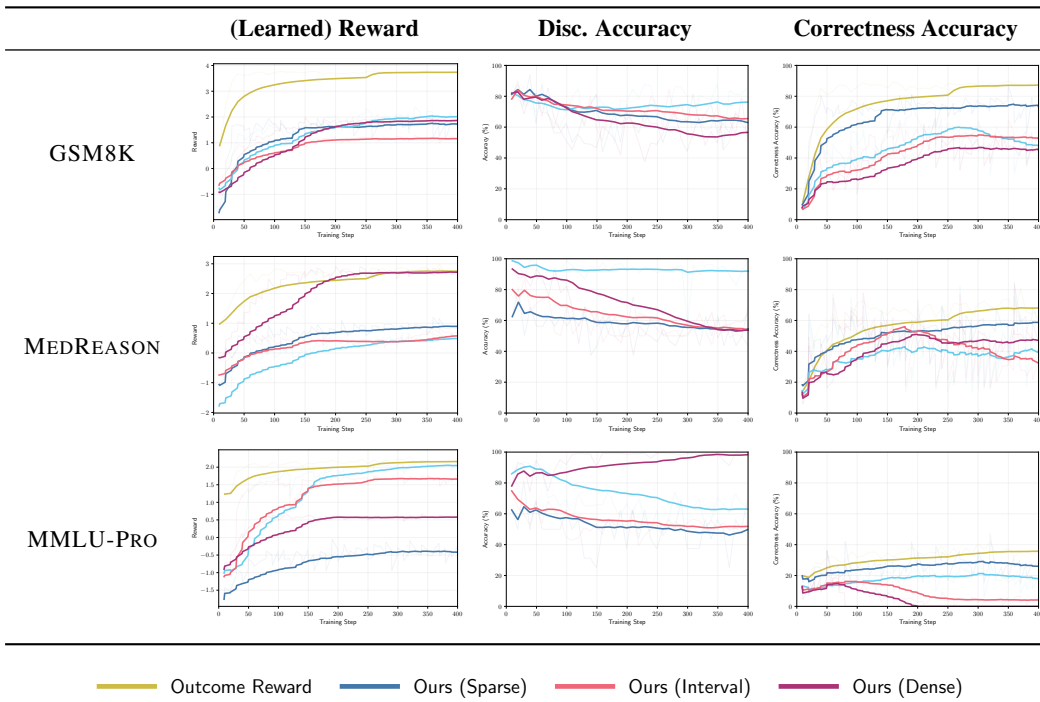


Figure 7: Training dynamics for Llama3.2-3B.

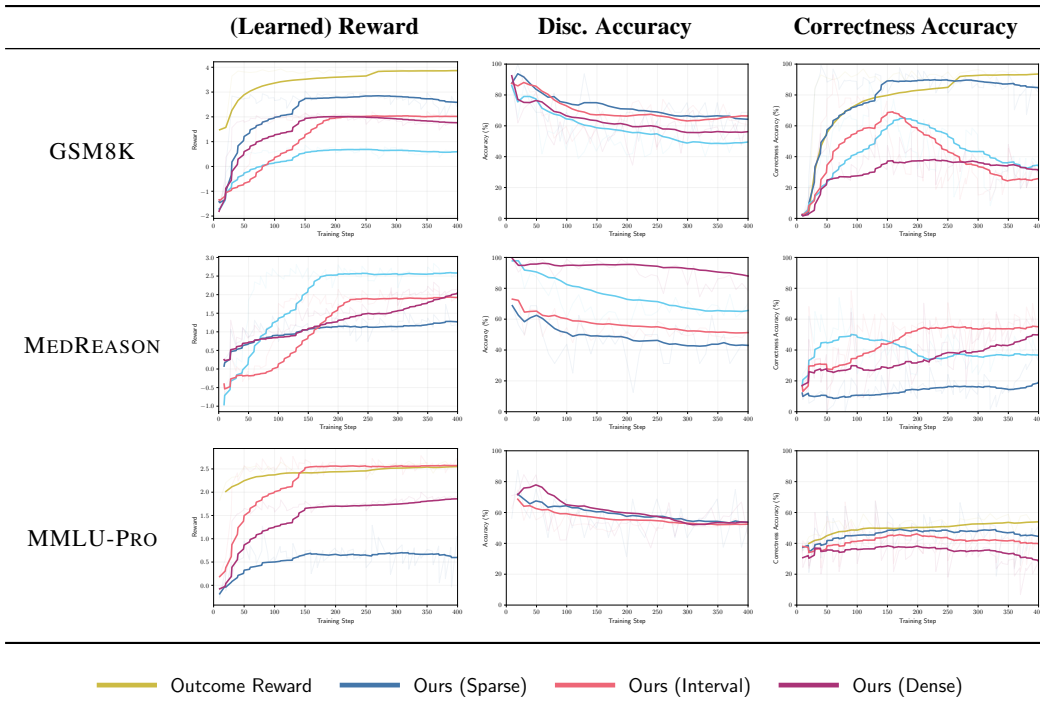


Figure 8: Training dynamics for Qwen2.5-7B.

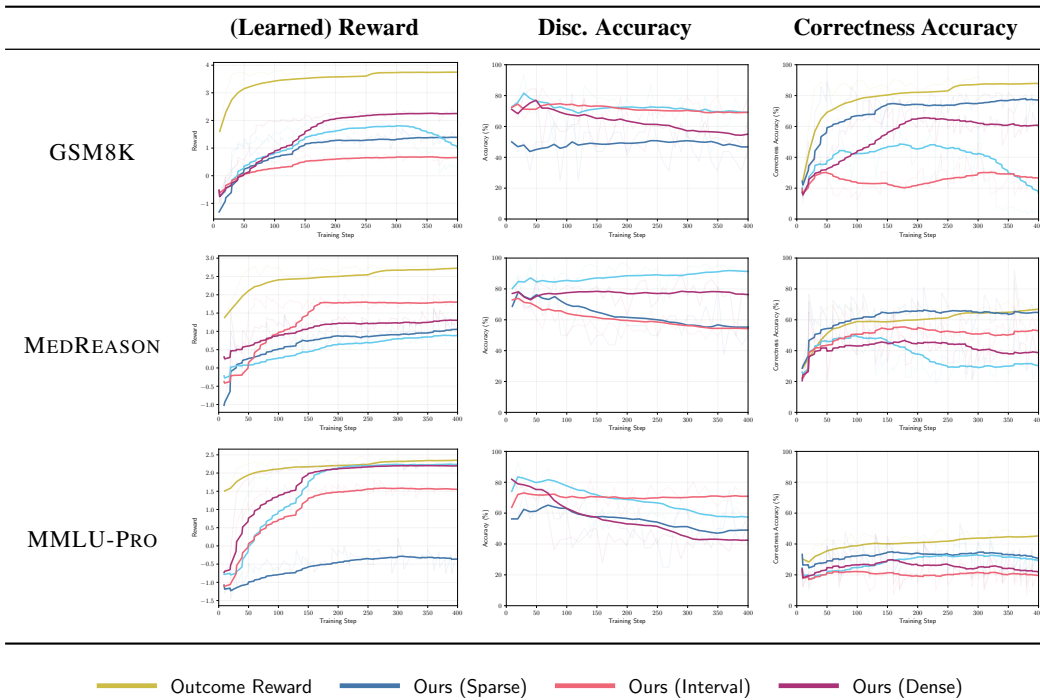


Figure 9: Training dynamics for Llama3.1-8B.

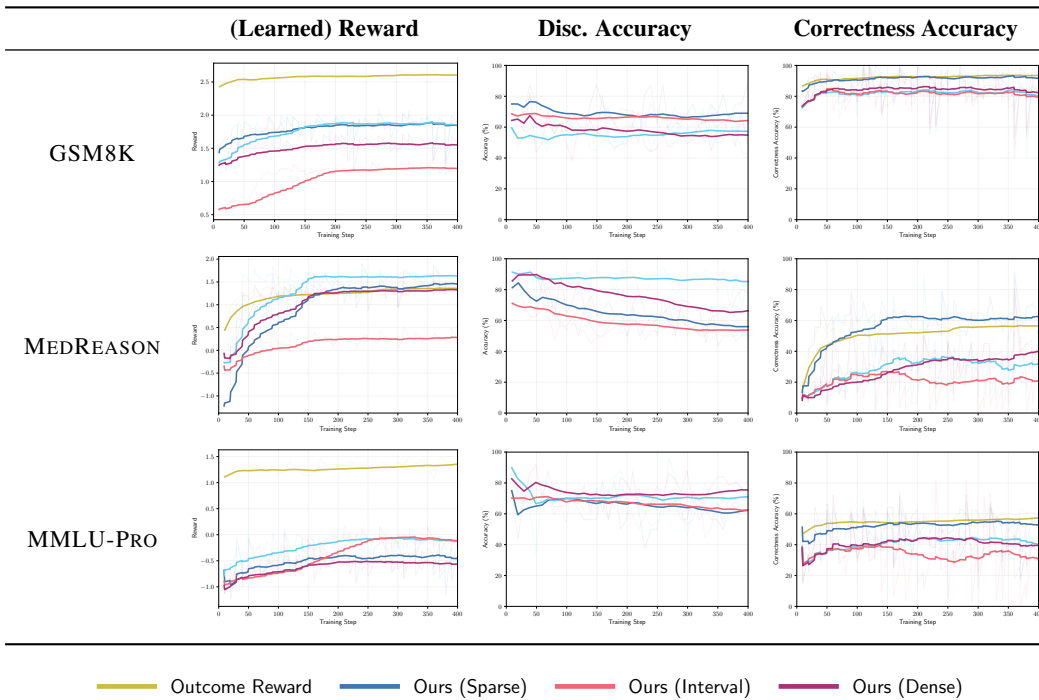


Figure 10: Training dynamics for Qwen3-4B.

B.3 COMPLETE RERANKING RESULTS

Backbone	Method	GSM8K			MEDREASON			MMLU-Pro		
		Pass@1	Reward	Δ (pp)	Pass@1	Reward	Δ (pp)	Pass@1	Reward	Δ (pp)
Qwen2.5-3B	<i>Sparse</i>	78.5 [76.8, 80.1]	81.7 [79.8, 83.9]	(\uparrow +3.3)	49.4 [47.5, 51.4]	50.9 [48.4, 53.2]	(\uparrow +1.5)	40.3 [38.5, 42.1]	42.2 [39.7, 44.6]	(\uparrow +1.9)
	<i>Step-wise</i>	65.2 [63.3, 67.0]	77.2 [74.9, 79.5]	(\uparrow +12.0)	35.2 [33.3, 37.1]	32.7 [30.3, 35.1]	(\downarrow -2.5)	36.4 [34.5, 38.3]	39.5 [36.9, 42.1]	(\uparrow +3.1)
	<i>Interval</i>	68.9 [67.1, 70.7]	77.5 [75.3, 79.8]	(\uparrow +8.6)	47.8 [45.2, 50.3]	47.9 [45.5, 50.6]	(\uparrow +0.1)	33.4 [31.4, 35.7]	32.1 [29.7, 34.5]	(\downarrow -1.4)
	<i>Dense</i>	43.9 [42.0, 46.0]	62.3 [59.7, 65.0]	(\uparrow +18.4)	48.4 [46.2, 50.6]	47.9 [45.4, 50.4]	(\downarrow -0.5)	28.8 [26.9, 30.8]	28.7 [26.5, 31.1]	(\downarrow -0.1)
Llama3.2-3B	<i>Sparse</i>	71.3 [69.5, 73.2]	75.9 [73.6, 78.2]	(\uparrow +4.6)	63.8 [61.8, 65.8]	65.7 [63.4, 68.1]	(\uparrow +1.8)	30.1 [28.0, 32.2]	32.4 [29.9, 34.7]	(\uparrow +2.3)
	<i>Step-wise</i>	66.9 [64.9, 69.0]	71.4 [69.1, 73.8]	(\uparrow +4.5)	66.8 [64.5, 69.2]	66.5 [64.0, 68.9]	(\downarrow -0.3)	31.6 [29.8, 33.3]	35.4 [32.8, 37.9]	(\uparrow +3.8)
	<i>Interval</i>	58.3 [56.2, 60.2]	65.1 [62.7, 67.5]	(\uparrow +6.9)	62.9 [60.5, 65.2]	63.2 [60.9, 65.5]	(\uparrow +0.3)	20.9 [19.5, 22.4]	20.4 [18.5, 22.4]	(\downarrow -0.5)
	<i>Dense</i>	50.9 [48.8, 53.0]	56.4 [53.9, 59.0]	(\uparrow +5.5)	57.9 [55.8, 59.9]	53.7 [51.2, 56.4]	(\downarrow -4.1)	0.0* [0.0, 0.0]	0.0* [0.0, 0.0]	(+0.0)
Qwen2.5-7B	<i>Sparse</i>	85.8 [84.4, 87.3]	88.8 [87.0, 90.5]	(\uparrow +3.0)	25.2 [23.5, 27.1]	32.3 [29.9, 34.5]	(\uparrow +7.0)	48.5 [46.4, 50.7]	51.5 [49.1, 54.1]	(\uparrow +3.0)
	<i>Step-wise</i>	64.5 [62.4, 66.6]	80.3 [78.2, 82.5]	(\uparrow +15.8)	40.9 [39.3, 42.6]	42.7 [40.2, 45.2]	(\uparrow +1.8)	45.6 [43.6, 47.6]	47.9 [45.4, 50.2]	(\uparrow +2.3)
	<i>Interval</i>	78.8 [77.1, 80.4]	82.5 [80.5, 84.6]	(\uparrow +3.7)	63.8 [61.8, 65.7]	63.4 [60.9, 65.7]	(\downarrow -0.4)	50.6 [48.6, 52.5]	53.1 [50.5, 55.5]	(\uparrow +2.5)
	<i>Dense</i>	38.4 [36.4, 40.4]	55.7 [53.1, 58.2]	(\uparrow +17.4)	58.3 [56.2, 60.6]	57.7 [55.3, 60.3]	(\downarrow -0.6)	43.8 [41.7, 45.8]	46.0 [43.5, 48.5]	(\uparrow +2.2)
Llama3.1-8B	<i>Sparse</i>	80.6 [78.9, 82.2]	82.1 [79.9, 84.2]	(\uparrow +1.5)	73.1 [71.2, 74.9]	74.6 [72.2, 76.8]	(\uparrow +1.5)	43.3 [41.4, 45.3]	42.3 [40.0, 44.9]	(\downarrow -1.0)
	<i>Step-wise</i>	80.8 [79.2, 82.5]	83.3 [81.3, 85.3]	(\uparrow +2.5)	69.3 [67.4, 71.3]	71.9 [69.5, 74.0]	(\uparrow +2.5)	42.6 [40.7, 44.5]	49.1 [46.7, 51.6]	(\uparrow +6.5)
	<i>Interval</i>	67.6 [65.7, 69.5]	78.2 [76.0, 80.4]	(\uparrow +10.7)	71.4 [69.5, 73.2]	75.3 [73.1, 77.4]	(\uparrow +3.9)	36.6 [34.9, 38.6]	38.1 [35.9, 40.7]	(\uparrow +1.6)
	<i>Dense</i>	64.6 [62.5, 66.6]	71.1 [68.7, 73.5]	(\uparrow +6.5)	69.8 [67.7, 71.7]	73.0 [70.9, 75.3]	(\uparrow +3.2)	37.9 [36.2, 39.8]	39.7 [37.3, 42.4]	(\uparrow +1.9)
Qwen3-4B	<i>Sparse</i>	90.4 [89.0, 91.9]	93.0 [91.5, 94.3]	(\uparrow +2.5)	66.1 [64.1, 68.3]	70.5 [68.2, 72.8]	(\uparrow +4.4)	55.6 [53.4, 57.8]	62.1 [59.6, 64.4]	(\uparrow +6.4)
	<i>Step-wise</i>	87.4 [85.8, 88.9]	90.1 [88.4, 91.7]	(\uparrow +2.7)	60.1 [57.7, 62.6]	59.9 [57.4, 62.3]	(\downarrow -0.2)	54.8 [52.5, 57.0]	60.9 [58.3, 63.4]	(\uparrow +6.1)
	<i>Interval</i>	87.8 [86.1, 89.3]	91.2 [89.7, 92.7]	(\uparrow +3.5)	52.1 [49.6, 54.6]	51.6 [48.9, 54.2]	(\downarrow -0.5)	53.5 [51.0, 55.8]	62.2 [59.7, 64.5]	(\uparrow +8.7)
	<i>Dense</i>	89.6 [88.1, 91.1]	91.4 [89.9, 92.9]	(\uparrow +1.8)	53.7 [51.5, 55.8]	61.6 [59.2, 64.0]	(\uparrow +7.9)	55.1 [52.7, 57.3]	59.2 [56.7, 61.5]	(\uparrow +4.1)

Table 6: **Best-of-N Reranking Performance (%)**. Comparison of standard Pass@1 baseline vs. Reward Model reranking. Values are percentages. **Bold** is best, underline is second best. Δ indicates percentage-point improvement over Pass@1. **Blue** is positive, **purple** is negative. * symbolises an adversarial mode collapse (results grayed out).

B.4 CALIBRATION OF REASONING REWARD MODELS

Backbone	Method	GSM8K		MEDREASON		MMLU-Pro	
		AUROC (%) \uparrow	ECE (%) \downarrow	AUROC (%) \uparrow	ECE (%) \downarrow	AUROC (%) \uparrow	ECE (%) \downarrow
Qwen2.5-3B	<i>Sparse</i>	78.1 [77.3, 79.0]	8.2 [7.7, 8.7]	52.5 [51.8, 53.2]	45.5 [44.9, 46.1]	67.0 [66.3, 67.7]	31.3 [30.7, 31.9]
	<i>Step-wise</i>	82.7 [82.2, 83.3]	25.9 [25.4, 26.6]	55.5 [54.7, 56.2]	39.8 [39.2, 40.3]	62.0 [61.3, 62.7]	57.6 [57.0, 58.2]
	<i>Interval</i>	76.6 [76.0, 77.4]	6.0 [5.5, 6.6]	49.7 [48.9, 50.4]	46.4 [45.8, 47.1]	52.2 [51.5, 53.0]	61.9 [61.3, 62.4]
	<i>Dense</i>	82.0 [81.5, 82.5]	35.3 [34.7, 35.9]	53.2 [52.4, 53.9]	30.1 [29.5, 30.7]	53.8 [53.0, 54.6]	54.7 [54.1, 55.3]
Llama3.2-3B	<i>Sparse</i>	81.1 [80.5, 81.7]	13.7 [13.2, 14.2]	60.7 [60.0, 61.4]	10.7 [10.1, 11.3]	66.5 [65.7, 67.2]	15.8 [15.2, 16.3]
	<i>Step-wise</i>	75.5 [74.8, 76.2]	20.3 [19.7, 20.8]	71.2 [70.5, 71.8]	4.9 [4.4, 5.5]	72.1 [71.4, 72.8]	49.8 [49.2, 50.3]
	<i>Interval</i>	75.1 [74.4, 75.8]	16.3 [15.7, 17.0]	58.4 [57.6, 59.1]	2.9 [2.4, 3.5]	64.8 [64.0, 65.7]	59.9 [59.4, 60.4]
	<i>Dense</i>	73.8 [73.0, 74.4]	34.6 [34.0, 35.3]	51.3 [50.5, 52.1]	36.0 [35.4, 36.7]	-	-
Qwen2.5-7B	<i>Sparse</i>	69.9 [68.8, 71.0]	8.3 [7.9, 8.8]	63.0 [62.2, 63.9]	50.7 [50.1, 51.2]	70.9 [70.3, 71.5]	17.2 [16.6, 17.7]
	<i>Step-wise</i>	87.9 [87.4, 88.4]	21.6 [21.0, 22.1]	78.8 [78.2, 79.4]	48.4 [47.8, 49.0]	66.3 [65.6, 67.0]	44.0 [43.3, 44.5]
	<i>Interval</i>	70.1 [69.1, 71.0]	7.5 [7.0, 8.0]	61.5 [60.8, 62.3]	21.7 [21.1, 22.3]	68.0 [67.3, 68.7]	35.6 [35.0, 36.2]
	<i>Dense</i>	71.4 [70.7, 72.1]	45.2 [44.6, 45.9]	50.8 [50.1, 51.5]	32.5 [31.9, 33.1]	60.2 [59.5, 60.9]	41.3 [40.7, 41.9]
Llama3.1-8B	<i>Sparse</i>	75.0 [74.1, 75.9]	2.4 [1.9, 2.9]	78.0 [77.3, 78.6]	5.5 [5.0, 6.0]	68.7 [68.0, 69.3]	9.8 [9.2, 10.3]
	<i>Step-wise</i>	75.8 [74.9, 76.7]	6.3 [5.9, 6.9]	78.1 [77.5, 78.7]	2.5 [2.1, 3.0]	73.5 [72.8, 74.1]	46.9 [46.4, 47.5]
	<i>Interval</i>	81.8 [81.2, 82.5]	8.7 [8.2, 9.2]	73.1 [72.4, 73.8]	15.7 [15.1, 16.2]	70.8 [70.1, 71.5]	39.7 [39.2, 40.2]
	<i>Dense</i>	75.9 [75.1, 76.6]	27.2 [26.6, 27.8]	67.9 [67.2, 68.7]	10.9 [10.3, 11.5]	60.2 [59.4, 60.9]	50.9 [50.4, 51.5]
Qwen3-4B	<i>Sparse</i>	84.1 [83.0, 85.1]	6.8 [6.5, 7.1]	73.5 [72.8, 74.1]	15.6 [15.1, 16.1]	86.8 [86.3, 87.2]	9.9 [9.4, 10.3]
	<i>Step-wise</i>	80.1 [79.1, 81.1]	5.1 [4.7, 5.4]	52.9 [52.1, 53.7]	25.1 [24.5, 25.7]	75.8 [75.2, 76.5]	8.9 [8.3, 9.4]
	<i>Interval</i>	78.3 [77.1, 79.5]	13.5 [13.2, 13.9]	52.0 [51.3, 52.8]	6.9 [6.3, 7.5]	84.2 [83.7, 84.7]	21.5 [21.0, 22.0]
	<i>Dense</i>	78.3 [77.2, 79.4]	7.3 [7.0, 7.7]	70.5 [69.8, 71.1]	25.2 [24.6, 25.8]	75.0 [74.4, 75.7]	18.6 [18.0, 19.2]

Table 7: **Critic Calibration Metrics**. All values are reported as percentages (%). AUROC indicates ranking ability; ECE measures calibration error (lower is better).

B.5 RERANKING PERFORMANCE ABLATION

Backbone	Method	GSM8K					MEDREASON					MLLU-Pro				
		Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.
Qwen2.5-3B	Sparse	63.8%	(\uparrow +1.4)	(\uparrow +0.3)	(\uparrow +4.4)	(\uparrow +4.4)	46.4%	(\uparrow +5.5)	(\uparrow +1.4)	(\uparrow +5.4)	(\uparrow +5.4)	40.0%	(\uparrow +2.1)	(\downarrow -0.1)	(\uparrow +2.6)	(\uparrow +2.6)
	Step-wise	63.8%	(\uparrow +1.4)	(\uparrow +0.3)	(\uparrow +4.6)	(\uparrow +4.6)	46.4%	(\uparrow +4.5)	(\uparrow +1.4)	(\uparrow +3.2)	(\uparrow +3.2)	40.0%	(\uparrow +2.1)	(\downarrow -0.1)	(\uparrow +2.8)	(\uparrow +2.8)
	Interval	63.8%	(\uparrow +1.4)	(\uparrow +0.3)	(\uparrow +3.5)	(\uparrow +3.5)	46.4%	(\uparrow +4.5)	(\uparrow +1.4)	(\uparrow +3.4)	(\uparrow +3.4)	40.0%	(\uparrow +2.1)	(\downarrow -0.1)	(\uparrow +0.2)	(\uparrow +0.2)
	Dense	63.8%	(\uparrow +1.4)	(\uparrow +0.3)	(\uparrow +4.7)	(\uparrow +4.7)	46.4%	(\uparrow +4.5)	(\uparrow +1.4)	(\uparrow +2.0)	(\uparrow +2.0)	40.0%	(\uparrow +2.1)	(\downarrow -0.1)	(\uparrow +1.2)	(\uparrow +1.2)
Llama3.2-3B	Sparse	62.2%	(\uparrow +3.3)	(\uparrow +1.4)	(\uparrow +5.3)	(\uparrow +5.3)	57.6%	(\uparrow +6.6)	(\uparrow +0.2)	(\uparrow +6.9)	(\uparrow +6.9)	36.5%	(\uparrow +2.0)	(\downarrow -0.3)	(\uparrow +1.7)	(\uparrow +1.7)
	Step-wise	62.2%	(\uparrow +3.3)	(\uparrow +1.4)	(\uparrow +5.3)	(\uparrow +5.3)	57.6%	(\uparrow +6.6)	(\uparrow +0.2)	(\uparrow +7.3)	(\uparrow +7.3)	36.5%	(\uparrow +2.0)	(\downarrow -0.3)	(\uparrow +2.4)	(\uparrow +2.4)
	Interval	62.2%	(\uparrow +3.3)	(\uparrow +1.4)	(\uparrow +3.9)	(\uparrow +3.9)	57.6%	(\uparrow +6.6)	(\uparrow +0.2)	(\uparrow +4.8)	(\uparrow +4.8)	36.5%	(\uparrow +2.0)	(\downarrow -0.3)	(\uparrow +1.7)	(\uparrow +1.7)
	Dense	62.2%	(\uparrow +3.3)	(\uparrow +1.4)	(\uparrow +4.6)	(\uparrow +4.6)	57.6%	(\uparrow +6.6)	(\uparrow +0.2)	(\uparrow +4.0)	(\uparrow +4.0)	36.5%	(\uparrow +2.0)	(\downarrow -0.3)	(\downarrow -0.6)	(\downarrow -0.6)
Qwen2.5-4B	Sparse	76.6%	(\downarrow -0.5)	(0.0)	(\uparrow +3.2)	(\uparrow +3.2)	57.6%	(\uparrow +0.3)	(\downarrow -0.3)	(\uparrow +3.3)	(\uparrow +3.3)	53.9%	(\uparrow +2.7)	(\uparrow +0.5)	(\uparrow +4.5)	(\uparrow +4.5)
	Step-wise	76.6%	(\downarrow -0.5)	(0.0)	(\uparrow +3.9)	(\uparrow +3.9)	57.6%	(\uparrow +0.3)	(\downarrow -0.3)	(\uparrow +1.3)	(\uparrow +1.3)	53.9%	(\uparrow +2.7)	(\uparrow +0.5)	(\uparrow +2.0)	(\uparrow +2.0)
	Interval	76.6%	(\downarrow -0.5)	(0.0)	(\uparrow +4.0)	(\uparrow +4.0)	57.6%	(\uparrow +0.3)	(\downarrow -0.3)	(\uparrow +1.0)	(\uparrow +1.0)	53.9%	(\uparrow +2.7)	(\uparrow +0.5)	(\uparrow +0.9)	(\uparrow +0.9)
	Dense	76.6%	(\downarrow -0.5)	(0.0)	(\uparrow +3.6)	(\uparrow +3.6)	57.6%	(\uparrow +0.3)	(\downarrow -0.3)	(\uparrow +1.9)	(\uparrow +1.9)	53.9%	(\uparrow +2.7)	(\uparrow +0.5)	(\uparrow +1.9)	(\uparrow +1.9)
Qwen2.5-7B	Sparse	70.1%	(\uparrow +0.4)	(\uparrow +0.3)	(\uparrow +6.0)	(\uparrow +6.0)	53.0%	(\uparrow +3.6)	(\uparrow +0.7)	(\uparrow +0.1)	(\uparrow +0.1)	48.1%	(\uparrow +0.9)	(\downarrow -0.6)	(\uparrow +3.5)	(\uparrow +3.5)
	Step-wise	70.1%	(\uparrow +0.4)	(\uparrow +0.3)	(\uparrow +5.5)	(\uparrow +5.5)	53.0%	(\uparrow +3.6)	(\uparrow +0.7)	(\uparrow +4.7)	(\uparrow +4.7)	48.1%	(\uparrow +0.9)	(\downarrow -0.6)	(\uparrow +1.8)	(\uparrow +1.8)
	Interval	70.1%	(\uparrow +0.4)	(\uparrow +0.3)	(\uparrow +5.4)	(\uparrow +5.4)	53.0%	(\uparrow +3.6)	(\uparrow +0.7)	(\uparrow +1.4)	(\uparrow +1.4)	48.1%	(\uparrow +0.9)	(\downarrow -0.6)	(\uparrow +1.1)	(\uparrow +1.1)
	Dense	70.1%	(\uparrow +0.4)	(\uparrow +0.3)	(\uparrow +6.4)	(\uparrow +6.4)	53.0%	(\uparrow +3.6)	(\uparrow +0.7)	(\uparrow +3.8)	(\uparrow +3.8)	48.1%	(\uparrow +0.9)	(\downarrow -0.6)	(\downarrow -0.1)	(\downarrow -0.1)
Llama3.1-8B	Sparse	66.6%	(\uparrow +1.4)	(\uparrow +0.2)	(\uparrow +2.3)	(\uparrow +2.3)	63.5%	(\uparrow +7.3)	(\uparrow +0.6)	(\uparrow +8.6)	(\uparrow +8.6)	47.2%	(\uparrow +1.7)	(\uparrow +0.6)	(\uparrow +2.8)	(\uparrow +2.8)
	Step-wise	66.6%	(\uparrow +1.4)	(\uparrow +0.2)	(\uparrow +2.8)	(\uparrow +2.8)	63.5%	(\uparrow +7.3)	(\uparrow +0.6)	(\uparrow +8.4)	(\uparrow +8.4)	47.2%	(\uparrow +1.7)	(\uparrow +0.6)	(\uparrow +3.7)	(\uparrow +3.7)
	Interval	66.6%	(\uparrow +1.4)	(\uparrow +0.2)	(\uparrow +3.1)	(\uparrow +3.1)	63.5%	(\uparrow +7.3)	(\uparrow +0.6)	(\uparrow +7.5)	(\uparrow +7.5)	47.2%	(\uparrow +1.7)	(\uparrow +0.6)	(\uparrow +3.7)	(\uparrow +3.7)
	Dense	66.6%	(\uparrow +1.4)	(\uparrow +0.2)	(\uparrow +3.0)	(\uparrow +3.0)	63.5%	(\uparrow +7.3)	(\uparrow +0.6)	(\uparrow +3.2)	(\uparrow +3.2)	47.2%	(\uparrow +1.7)	(\uparrow +0.6)	(\uparrow +2.0)	(\uparrow +2.0)

Table 8: **Best-of-2 Reranking Performance & Baselines (%)**. Random is SFT pass@1. Deltas are percentage-point changes for each reranker. **Blue arrows (\uparrow)** indicate gains and **purple arrows (\downarrow)** indicate drops.

Backbone	Method	GSM8K					MEDREASON					MLLU-Pro				
		Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.
Qwen2.5-3B	Sparse	63.8%	(\uparrow +0.8)	(\uparrow +4.2)	(\uparrow +5.4)	(\uparrow +6.8)	46.4%	(\uparrow +6.0)	(\uparrow +5.4)	(\uparrow +6.6)	(\uparrow +8.4)	40.0%	(\uparrow +2.0)	(\uparrow +2.3)	(\uparrow +3.9)	(\uparrow +5.0)
	Step-wise	63.8%	(\uparrow +0.8)	(\uparrow +4.2)	(\uparrow +6.4)	(\uparrow +7.2)	46.4%	(\uparrow +6.0)	(\uparrow +5.4)	(\uparrow +4.4)	(\uparrow +8.0)	40.0%	(\uparrow +2.0)	(\uparrow +2.3)	(\uparrow +3.4)	(\uparrow +4.0)
	Interval	63.8%	(\uparrow +0.8)	(\uparrow +4.2)	(\uparrow +4.0)	(\uparrow +5.8)	46.4%	(\uparrow +6.0)	(\uparrow +5.4)	(\uparrow +2.7)	(\uparrow +7.0)	40.0%	(\uparrow +2.0)	(\uparrow +2.3)	(\downarrow -0.5)	(\uparrow +2.2)
	Dense	63.8%	(\uparrow +0.8)	(\uparrow +4.2)	(\uparrow +6.4)	(\uparrow +7.4)	46.4%	(\uparrow +6.0)	(\uparrow +5.4)	(\uparrow +0.8)	(\uparrow +5.3)	40.0%	(\uparrow +2.0)	(\uparrow +2.3)	(\uparrow +0.4)	(\uparrow +2.7)
Llama3.2-3B	Sparse	62.2%	(\uparrow +4.6)	(\uparrow +5.1)	(\uparrow +6.5)	(\uparrow +7.8)	57.6%	(\uparrow +8.6)	(\uparrow +5.3)	(\uparrow +8.0)	(\uparrow +9.1)	36.5%	(\uparrow +2.6)	(\uparrow +0.5)	(\uparrow +2.0)	(\uparrow +2.1)
	Step-wise	62.2%	(\uparrow +4.6)	(\uparrow +5.1)	(\uparrow +5.8)	(\uparrow +7.6)	57.6%	(\uparrow +8.6)	(\uparrow +5.3)	(\uparrow +8.9)	(\uparrow +9.1)	36.5%	(\uparrow +2.6)	(\uparrow +0.5)	(\uparrow +2.9)	(\uparrow +3.2)
	Interval	62.2%	(\uparrow +4.6)	(\uparrow +5.1)	(\uparrow +5.3)	(\uparrow +7.1)	57.6%	(\uparrow +8.6)	(\uparrow +5.3)	(\uparrow +6.1)	(\uparrow +8.9)	36.5%	(\uparrow +2.6)	(\uparrow +0.5)	(\uparrow +1.5)	(\uparrow +2.8)
	Dense	62.2%	(\uparrow +4.6)	(\uparrow +5.1)	(\uparrow +5.3)	(\uparrow +6.6)	57.6%	(\uparrow +8.6)	(\uparrow +5.3)	(\uparrow +5.6)	(\uparrow +8.5)	36.5%	(\uparrow +2.6)	(\uparrow +0.5)	(\downarrow -1.5)	(\uparrow +2.0)
Qwen2.5-4B	Sparse	76.6%	(\downarrow -1.0)	(\uparrow +3.4)	(\uparrow +5.4)	(\uparrow +6.5)	57.6%	(\downarrow -1.0)	(\uparrow +2.9)	(\uparrow +3.6)	(\uparrow +4.7)	53.9%	(\uparrow +3.7)	(\uparrow +3.5)	(\uparrow +6.5)	(\uparrow +6.4)
	Step-wise	76.6%	(\downarrow -1.0)	(\uparrow +3.4)	(\uparrow +5.7)	(\uparrow +6.7)	57.6%	(\downarrow -1.0)	(\uparrow +2.9)	(\uparrow +2.1)	(\uparrow +3.5)	53.9%	(\uparrow +3.7)	(\uparrow +3.5)	(\uparrow +4.6)	(\uparrow +4.1)
	Interval	76.6%	(\downarrow -1.0)	(\uparrow +3.4)	(\uparrow +5.2)	(\uparrow +6.3)	57.6%	(\downarrow -1.0)	(\uparrow +2.9)	(\uparrow +1.4)	(\uparrow +2.7)	53.9%	(\uparrow +3.7)	(\uparrow +3.5)	(\uparrow +2.1)	(\uparrow +3.3)
	Dense	76.6%	(\downarrow -1.0)	(\uparrow +3.4)	(\uparrow +5.2)	(\uparrow +6.1)	57.6%	(\downarrow -1.0)	(\uparrow +2.9)	(\uparrow +3.3)	(\uparrow +4.3)	53.9%	(\uparrow +3.7)	(\uparrow +3.5)	(\uparrow +3.7)	(\uparrow +4.3)
Qwen2.5-7B	Sparse	70.1%	(\uparrow +1.7)	(\uparrow +7.5)	(\uparrow +8.1)	(\uparrow +9.2)	53.0%	(\uparrow +4.0)	(\uparrow +4.6)	(\downarrow -2.4)	(\downarrow -1.1)	48.1%	(\uparrow +1.9)	(\uparrow +2.1)	(\uparrow +4.3)	(\uparrow +4.5)
	Step-wise	70.1%	(\uparrow +1.7)	(\uparrow +7.5)	(\uparrow +6.8)	(\uparrow +8.6)	53.0%	(\uparrow +4.0)	(\uparrow +4.6)	(\uparrow +4.5)	(\uparrow +6.1)	48.1%	(\uparrow +1.9)	(\uparrow +2.1)	(\uparrow +4.1)	(\uparrow +3.7)
	Interval	70.1%	(\uparrow +1.7)	(\uparrow +7.5)	(\uparrow +6.4)	(\uparrow +8.5)	53.0%	(\uparrow +4.0)	(\uparrow +4.6)	(\uparrow +0.8)	(\uparrow +2.9)	48.1%	(\uparrow +1.9)	(\uparrow +2.1)	(\uparrow +2.4)	(\uparrow +3.4)
	Dense	70.1%	(\uparrow +1.7)	(\uparrow +7.5)	(\uparrow +9.4)	(\uparrow +10.1)	53.0%	(\uparrow +4.0)	(\uparrow +4.6)	(\uparrow +3.8)	(\uparrow +6.6)	48.1%	(\uparrow +1.9)	(\uparrow +2.1)	(\uparrow +0.2)	(\uparrow +2.5)
Llama3.1-8B	Sparse	66.6%	(\uparrow +2.3)	(\uparrow +1.8)	(\uparrow +2.2)	(\uparrow +3.4)	63.5%	(\uparrow +8.2)	(\uparrow +4.2)	(\uparrow +10.0)	(\uparrow +11.1)	47.2%	(\uparrow +2.0)	(\uparrow +4.0)	(\uparrow +2.1)	(\uparrow +3.2)
	Step-wise	66.6%	(\uparrow +2.3)	(\uparrow +1.8)	(\uparrow +4.3)	(\uparrow +4.3)	63.5%	(\uparrow +8.2)	(\uparrow +4.2)	(\uparrow +9.0)	(\uparrow +9.8)	47.2%	(\uparrow +2.0)	(\uparrow +4.0)	(\uparrow +4.5)	(\uparrow +4.8)
	Interval	66.6%	(\uparrow +2.3)	(\uparrow +1.8)	(\uparrow +3.9)	(\uparrow +4.4)	63.5%	(\uparrow +8.2)	(\uparrow +4.2)	(\uparrow +8.6)	(\uparrow +9.5)	47.2%	(\uparrow +2.0)	(\uparrow +4.0)	(\uparrow +4.0)	(\uparrow +4.8)
	Dense	66.6%	(\uparrow +2.3)	(\uparrow +1.8)	(\uparrow +3.7)	(\uparrow +3.6)	63.5%	(\uparrow +8.2)	(\uparrow +4.2)	(\uparrow +3.2)	(\uparrow +6.4)	47.2%	(\uparrow +2.0)	(\uparrow +4.0)	(\uparrow +2.8)	(\uparrow +4.4)

Table 9: **Best-of-3 Reranking Performance & Baselines (%)**. Random is SFT pass@1. Deltas are percentage-point changes for each reranker. **Blue arrows (\uparrow)** indicate gains and **purple arrows (\downarrow)** indicate drops.

Backbone	Method	GSM8K					MEDREASON					MLLU-Pro				
		Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.
Qwen2.5-3B	Sparse	63.8%	(\uparrow +2.4)	(\uparrow +9.8)	(\uparrow +6.6)	(\uparrow +10.6)	46.4%	(\uparrow +5.6)	(\uparrow +8.2)	(\uparrow +6.0)	(\uparrow +8.4)	40.0%	(\uparrow +3.1)	(\uparrow +4.6)	(\uparrow +3.4)	(\uparrow +6.4)
	Step-wise	63.8%	(\uparrow +2.4)	(\uparrow +9.8)	(\uparrow +8.6)	(\uparrow +11.9)	46.4%	(\uparrow +5.6)	(\uparrow +8.2)	(\uparrow +6.2)	(\uparrow +10.0)	40.0%	(\uparrow +3.1)	(\uparrow +4.6)	(\uparrow +4.0)	(\uparrow +5.2)
	Interval	63.8%	(\uparrow +2.4)	(\uparrow +9.8)	(\uparrow +7.2)	(\uparrow +11.7)	46.4%	(\uparrow +5.6)	(\uparrow +8.2)	(\uparrow +3.8)	(\uparrow +9.2)	40.0%	(\uparrow +3.1)	(\uparrow +4.6)	(\downarrow -0.6)	(\uparrow +5.2)
	Dense	63.8%	(\uparrow +2.4)	(\uparrow +9.8)	(\uparrow +9.0)	(\uparrow +12.0)	46.4%	(\uparrow +5.6)	(\uparrow +8.2)	(\uparrow +0.6)	(\uparrow +8.4)	40.0%	(\uparrow +3.1)	(\uparrow +4.6)	(\uparrow +0.6)	(\uparrow +4.7)
Llama3.2-3B	Sparse	62.2%	(

Backbone	Method	GSM8K					MEDREASON					MLLU-Pro				
		Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.
Qwen2.5-3B	<i>Sparse</i>	63.8%	(\uparrow +2.0)	(\uparrow +11.8)	(\uparrow +5.8)	(\uparrow +11.8)	46.4%	(\uparrow +6.0)	(\uparrow +9.9)	(\uparrow +5.6)	(\uparrow +9.8)	40.0%	(\uparrow +3.4)	(\uparrow +6.1)	(\uparrow +3.4)	(\uparrow +8.3)
	<i>Step-wise</i>	63.8%	(\uparrow +2.0)	(\uparrow +11.8)	(\uparrow +9.2)	(\uparrow +12.4)	46.4%	(\uparrow +6.0)	(\uparrow +9.9)	(\uparrow +5.8)	(\uparrow +11.1)	40.0%	(\uparrow +3.4)	(\uparrow +6.1)	(\uparrow +4.4)	(\uparrow +6.4)
	<i>Interval</i>	63.8%	(\uparrow +2.0)	(\uparrow +11.8)	(\uparrow +6.7)	(\uparrow +13.3)	46.4%	(\uparrow +6.0)	(\uparrow +9.9)	(\uparrow +4.6)	(\uparrow +10.1)	40.0%	(\uparrow +3.4)	(\uparrow +6.1)	(\uparrow +0.1)	(\uparrow +5.8)
	<i>Dense</i>	63.8%	(\uparrow +2.0)	(\uparrow +11.8)	(\uparrow +9.6)	(\uparrow +13.5)	46.4%	(\uparrow +6.0)	(\uparrow +9.9)	(\downarrow -0.6)	(\uparrow +9.9)	40.0%	(\uparrow +3.4)	(\uparrow +6.1)	(\uparrow +0.3)	(\uparrow +6.2)
Llama3.2-3B	<i>Sparse</i>	62.2%	(\uparrow +4.5)	(\uparrow +10.8)	(\uparrow +7.5)	(\uparrow +12.6)	57.6%	(\uparrow +11.4)	(\uparrow +10.9)	(\uparrow +8.9)	(\uparrow +12.0)	36.5%	(\uparrow +5.3)	(\uparrow +7.5)	(\uparrow +1.5)	(\uparrow +7.9)
	<i>Step-wise</i>	62.2%	(\uparrow +4.5)	(\uparrow +10.8)	(\uparrow +7.5)	(\uparrow +11.7)	57.6%	(\uparrow +11.4)	(\uparrow +10.9)	(\uparrow +10.0)	(\uparrow +12.4)	36.5%	(\uparrow +5.3)	(\uparrow +7.5)	(\uparrow +5.1)	(\uparrow +8.1)
	<i>Interval</i>	62.2%	(\uparrow +4.5)	(\uparrow +10.8)	(\uparrow +6.0)	(\uparrow +11.1)	57.6%	(\uparrow +11.4)	(\uparrow +10.9)	(\uparrow +8.1)	(\uparrow +11.8)	36.5%	(\uparrow +5.3)	(\uparrow +7.5)	(\uparrow +3.3)	(\uparrow +8.7)
	<i>Dense</i>	62.2%	(\uparrow +4.5)	(\uparrow +10.8)	(\uparrow +5.6)	(\uparrow +11.5)	57.6%	(\uparrow +11.4)	(\uparrow +10.9)	(\uparrow +6.9)	(\uparrow +11.8)	36.5%	(\uparrow +5.3)	(\uparrow +7.5)	(\uparrow +0.7)	(\uparrow +7.2)
Qwen2.5-4B	<i>Sparse</i>	76.6%	(\uparrow +0.1)	(\uparrow +8.3)	(\uparrow +9.1)	(\uparrow +9.1)	57.6%	(\downarrow -0.3)	(\uparrow +7.7)	(\uparrow +5.7)	(\uparrow +8.7)	53.9%	(\uparrow +2.4)	(\uparrow +6.0)	(\uparrow +7.9)	(\uparrow +8.1)
	<i>Step-wise</i>	76.6%	(\uparrow +0.1)	(\uparrow +8.3)	(\uparrow +6.9)	(\uparrow +8.9)	57.6%	(\downarrow -0.3)	(\uparrow +7.7)	(\uparrow +4.4)	(\uparrow +8.8)	53.9%	(\uparrow +2.4)	(\uparrow +6.0)	(\uparrow +4.7)	(\uparrow +6.7)
	<i>Interval</i>	76.6%	(\uparrow +0.1)	(\uparrow +8.3)	(\uparrow +7.0)	(\uparrow +8.9)	57.6%	(\downarrow -0.3)	(\uparrow +7.7)	(\uparrow +2.5)	(\uparrow +8.1)	53.9%	(\uparrow +2.4)	(\uparrow +6.0)	(\uparrow +1.9)	(\uparrow +5.8)
	<i>Dense</i>	76.6%	(\uparrow +0.1)	(\uparrow +8.3)	(\uparrow +7.7)	(\uparrow +9.1)	57.6%	(\downarrow -0.3)	(\uparrow +7.7)	(\uparrow +4.1)	(\uparrow +8.9)	53.9%	(\uparrow +2.4)	(\uparrow +6.0)	(\uparrow +4.6)	(\uparrow +6.5)
Qwen2.5-7B	<i>Sparse</i>	70.1%	(\uparrow +1.3)	(\uparrow +13.0)	(\uparrow +10.5)	(\uparrow +13.9)	53.0%	(\uparrow +4.9)	(\uparrow +10.3)	(\downarrow -7.7)	(\downarrow -2.6)	48.1%	(\uparrow +2.1)	(\uparrow +5.7)	(\uparrow +6.9)	(\uparrow +7.6)
	<i>Step-wise</i>	70.1%	(\uparrow +1.3)	(\uparrow +13.0)	(\uparrow +10.0)	(\uparrow +13.5)	53.0%	(\uparrow +4.9)	(\uparrow +10.3)	(\uparrow +3.6)	(\uparrow +10.1)	48.1%	(\uparrow +2.1)	(\uparrow +5.7)	(\uparrow +3.9)	(\uparrow +6.7)
	<i>Interval</i>	70.1%	(\uparrow +1.3)	(\uparrow +13.0)	(\uparrow +8.7)	(\uparrow +13.7)	53.0%	(\uparrow +4.9)	(\uparrow +10.3)	(\downarrow -3.2)	(\uparrow +9.2)	48.1%	(\uparrow +2.1)	(\uparrow +5.7)	(\uparrow +3.2)	(\uparrow +6.2)
	<i>Dense</i>	70.1%	(\uparrow +1.3)	(\uparrow +13.0)	(\uparrow +11.0)	(\uparrow +14.4)	53.0%	(\uparrow +4.9)	(\uparrow +10.3)	(\uparrow +6.2)	(\uparrow +10.9)	48.1%	(\uparrow +2.1)	(\uparrow +5.7)	(\uparrow +0.9)	(\uparrow +6.2)
Llama3.1-8B	<i>Sparse</i>	66.6%	(\uparrow +2.4)	(\uparrow +2.7)	(\uparrow +3.0)	(\uparrow +4.3)	63.5%	(\uparrow +9.6)	(\uparrow +10.7)	(\uparrow +9.8)	(\uparrow +13.4)	47.2%	(\uparrow +2.5)	(\uparrow +7.0)	(\uparrow +1.4)	(\uparrow +6.6)
	<i>Step-wise</i>	66.6%	(\uparrow +2.4)	(\uparrow +2.7)	(\uparrow +5.3)	(\uparrow +4.6)	63.5%	(\uparrow +9.6)	(\uparrow +10.7)	(\uparrow +9.0)	(\uparrow +12.4)	47.2%	(\uparrow +2.5)	(\uparrow +7.0)	(\uparrow +4.9)	(\uparrow +7.8)
	<i>Interval</i>	66.6%	(\uparrow +2.4)	(\uparrow +2.7)	(\uparrow +5.3)	(\uparrow +4.8)	63.5%	(\uparrow +9.6)	(\uparrow +10.7)	(\uparrow +10.8)	(\uparrow +13.1)	47.2%	(\uparrow +2.5)	(\uparrow +7.0)	(\uparrow +4.1)	(\uparrow +7.8)
	<i>Dense</i>	66.6%	(\uparrow +2.4)	(\uparrow +2.7)	(\uparrow +5.8)	(\uparrow +4.1)	63.5%	(\uparrow +9.6)	(\uparrow +10.7)	(\uparrow +1.2)	(\uparrow +11.6)	47.2%	(\uparrow +2.5)	(\uparrow +7.0)	(\uparrow +3.7)	(\uparrow +7.4)

Table 11: **Best-of-8 Reranking Performance & Baselines (%)**. Random is SFT pass@1. Deltas are percentage-point changes for each reranker. **Blue arrows (\uparrow)** indicate gains and **purple arrows (\downarrow)** indicate drops.

Backbone	Method	GSM8K					MEDREASON					MLLU-Pro				
		Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.	Random	Δ Logp.	Δ Maj.	Δ Rew.	Δ W.Maj.
Qwen2.5-3B	<i>Sparse</i>	63.8%	(\uparrow +1.8)	(\uparrow +14.0)	(\uparrow +5.7)	(\uparrow +14.4)	46.4%	(\uparrow +5.5)	(\uparrow +10.4)	(\uparrow +1.8)	(\uparrow +9.4)	40.0%	(\uparrow +3.8)	(\uparrow +7.2)	(\uparrow +3.2)	(\uparrow +9.5)
	<i>Step-wise</i>	63.8%	(\uparrow +1.8)	(\uparrow +14.0)	(\uparrow +10.4)	(\uparrow +15.5)	46.4%	(\uparrow +5.5)	(\uparrow +10.4)	(\uparrow +4.0)	(\uparrow +10.7)	40.0%	(\uparrow +3.8)	(\uparrow +7.2)	(\uparrow +3.7)	(\uparrow +7.8)
	<i>Interval</i>	63.8%	(\uparrow +1.8)	(\uparrow +14.0)	(\uparrow +7.1)	(\uparrow +15.2)	46.4%	(\uparrow +5.5)	(\uparrow +10.4)	(\uparrow +2.0)	(\uparrow +10.4)	40.0%	(\uparrow +3.8)	(\uparrow +7.2)	(\uparrow +1.0)	(\uparrow +7.4)
	<i>Dense</i>	63.8%	(\uparrow +1.8)	(\uparrow +14.0)	(\uparrow +11.3)	(\uparrow +15.8)	46.4%	(\uparrow +5.5)	(\uparrow +10.4)	(\downarrow -3.3)	(\uparrow +10.6)	40.0%	(\uparrow +3.8)	(\uparrow +7.2)	(\downarrow -0.5)	(\uparrow +7.2)
Llama3.2-3B	<i>Sparse</i>	62.2%	(\uparrow +5.5)	(\uparrow +12.0)	(\uparrow +8.9)	(\uparrow +14.6)	57.6%	(\uparrow +11.5)	(\uparrow +13.0)	(\uparrow +9.4)	(\uparrow +13.8)	36.5%	(\uparrow +4.3)	(\uparrow +8.2)	(\uparrow +0.1)	(\uparrow +8.9)
	<i>Step-wise</i>	62.2%	(\uparrow +5.5)	(\uparrow +12.0)	(\uparrow +8.3)	(\uparrow +12.5)	57.6%	(\uparrow +11.5)	(\uparrow +13.0)	(\uparrow +8.6)	(\uparrow +13.6)	36.5%	(\uparrow +4.3)	(\uparrow +8.2)	(\uparrow +4.5)	(\uparrow +8.2)
	<i>Interval</i>	62.2%	(\uparrow +5.5)	(\uparrow +12.0)	(\uparrow +6.3)	(\uparrow +12.7)	57.6%	(\uparrow +11.5)	(\uparrow +13.0)	(\uparrow +8.6)	(\uparrow +13.3)	36.5%	(\uparrow +4.3)	(\uparrow +8.2)	(\uparrow +4.4)	(\uparrow +8.3)
	<i>Dense</i>	62.2%	(\uparrow +5.5)	(\uparrow +12.0)	(\uparrow +8.3)	(\uparrow +12.1)	57.6%	(\uparrow +11.5)	(\uparrow +13.0)	(\uparrow +6.4)	(\uparrow +13.6)	36.5%	(\uparrow +4.3)	(\uparrow +8.2)	(\uparrow +1.3)	(\uparrow +8.4)
Qwen2.5-4B	<i>Sparse</i>	76.6%	(\uparrow +1.4)	(\uparrow +8.7)	(\uparrow +10.3)	(\uparrow +10.6)	57.6%	(\uparrow +1.1)	(\uparrow +9.6)	(\uparrow +6.5)	(\uparrow +9.5)	53.9%	(\uparrow +2.1)	(\uparrow +7.7)	(\uparrow +8.7)	(\uparrow +9.3)
	<i>Step-wise</i>	76.6%	(\uparrow +1.4)	(\uparrow +8.7)	(\uparrow +8.3)	(\uparrow +10.3)	57.6%	(\uparrow +1.1)	(\uparrow +9.6)	(\uparrow +4.1)	(\uparrow +9.8)	53.9%	(\uparrow +2.1)	(\uparrow +7.7)	(\uparrow +5.5)	(\uparrow +8.3)
	<i>Interval</i>	76.6%	(\uparrow +1.4)	(\uparrow +8.7)	(\uparrow +8.1)	(\uparrow +10.2)	57.6%	(\uparrow +1.1)	(\uparrow +9.6)	(\uparrow +0.9)	(\uparrow +9.1)	53.9%	(\uparrow +2.1)	(\uparrow +7.7)	(\uparrow +2.2)	(\uparrow +7.5)
	<i>Dense</i>	76.6%	(\uparrow +1.4)	(\uparrow +8.7)	(\uparrow +9.2)	(\uparrow +10.6)	57.6%	(\uparrow +1.1)	(\uparrow +9.6)	(\uparrow +3.4)	(\uparrow +9.0)	53.9%	(\uparrow +2.1)	(\uparrow +7.7)	(\uparrow +4.7)	(\uparrow +7.5)
Qwen2.5-7B	<i>Sparse</i>	70.1%	(\uparrow +0.8)	(\uparrow +15.6)	(\uparrow +11.3)	(\uparrow +16.4)	53.0%	(\uparrow +5.9)	(\uparrow +11.3)	(\downarrow -13.6)	(\downarrow -5.8)	48.1%	(\uparrow +2.6)	(\uparrow +7.5)	(\uparrow +7.0)	(\uparrow +8.6)
	<i>Step-wise</i>	70.1%	(\uparrow +0.8)	(\uparrow +15.6)	(\uparrow +9.0)	(\uparrow +15.7)	53.0%	(\uparrow +5.9)	(\uparrow +11.3)	(\uparrow +1.4)	(\uparrow +11.6)	48.1%	(\uparrow +2.6)	(\uparrow +7.5)	(\uparrow +3.5)	(\uparrow +8.3)
	<i>Interval</i>	70.1%	(\uparrow +0.8)	(\uparrow +15.6)	(\uparrow +8.3)	(\uparrow +15.3)	53.0%	(\uparrow +5.9)	(\uparrow +11.3)	(\downarrow -5.9)	(\uparrow +10.8)	48.1%	(\uparrow +2.6)	(\uparrow +7.5)	(\uparrow +4.1)	(\uparrow +8.1)
	<i>Dense</i>	70.1%	(\uparrow +0.8)	(\uparrow +15.6)	(\uparrow +11.1)	(\uparrow +15.7)	53.0%	(\uparrow +5.9)	(\uparrow +11.3)	(\uparrow +5.2)	(\uparrow +12.0)	48.1%	(\uparrow +2.6)	(\uparrow +7.5)	(\uparrow +0.5)	(\uparrow +7.7)
Llama3.1-8B	<i>Sparse</i>	66.6%	(\uparrow +2.8)	(\uparrow +4.6)	(\uparrow +3.7)	(\uparrow +5.2)	63.5%	(\uparrow +11.8)	(\uparrow +12.8)	(\uparrow +9.5)	(\uparrow +15.6)	47.2%	(\uparrow +2.7)	(\uparrow +7.6)	(\downarrow -0.2)	(\uparrow +8.3)
	<i>Step-wise</i>	66.6%	(\uparrow +2.8)	(\uparrow +4.6)	(\uparrow +5.5)	(\uparrow +6.1)	63.5%	(\uparrow +11.8)	(\uparrow +12.8)	(\uparrow +7.2)	(\uparrow +14.5)	47.2%	(\uparrow +2.7)	(\uparrow +7.6)	(\uparrow +4.4)	(\uparrow +8.0)
	<i>Interval</i>	66.6%	(\uparrow +2.8)	(\uparrow +4.6)	(\uparrow +5.9)	(\uparrow +6.8)	63.5%	(\uparrow +11.8)	(\uparrow +12.8)	(\uparrow +12.0)	(\uparrow +14.9)	47.2%	(\uparrow +2.7)	(\uparrow +7.6)	(\uparrow +4.4)	(\uparrow +8.4)
	<i>Dense</i>	66.6%	(\uparrow +2.8)	(\uparrow +4.6)	(\uparrow +6.5)	(\uparrow +5.8)	63.5%	(\uparrow +11.8)	(\uparrow +12.8)	(\downarrow -0.9)	(\uparrow +13.4)	47.2%	(\uparrow +2.7)	(\uparrow +7.6)	(\uparrow +3.2)	(\uparrow +7.9)

Table 12: **Best-of-16 Reranking Performance & Baselines (%)**. Random is SFT pass@1. Deltas are percentage-point changes for each reranker. **Blue arrows (\uparrow)** indicate gains and **purple arrows (\downarrow)** indicate drops.

B.6 DISTRIBUTION SEPARATION OF REWARDS

B.6.1 GSM8K

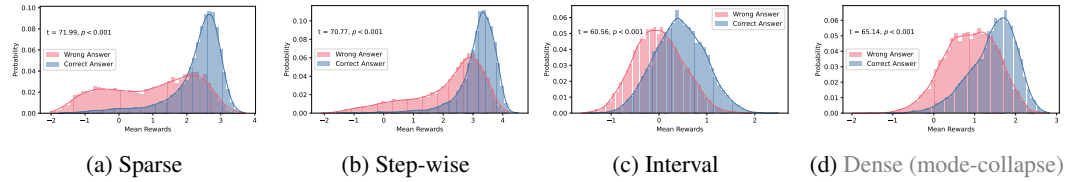


Figure 11: **Reward Distribution by Correctness (Qwen2.5-3B on GSM8K)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

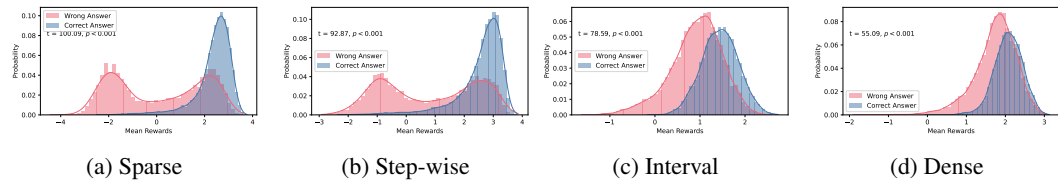


Figure 12: **Reward Distribution by Correctness (Llama3.2-3B on GSM8K)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

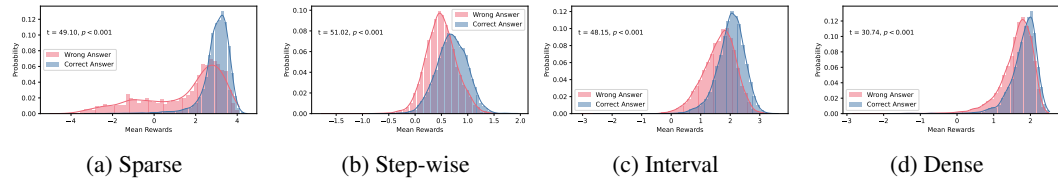


Figure 13: **Reward Distribution by Correctness (Qwen2.5-7B on GSM8K)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

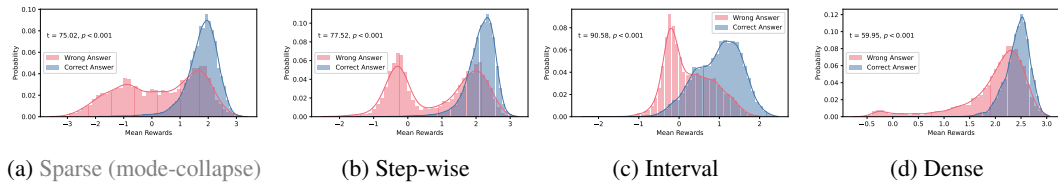


Figure 14: **Reward Distribution by Correctness (Llama3.1-8B on GSM8K)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

B.6.2 MEDREASON

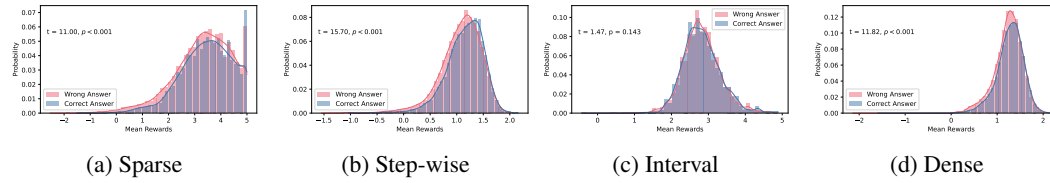


Figure 15: **Reward Distribution by Correctness (Qwen2.5-3B on MEDREASON)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

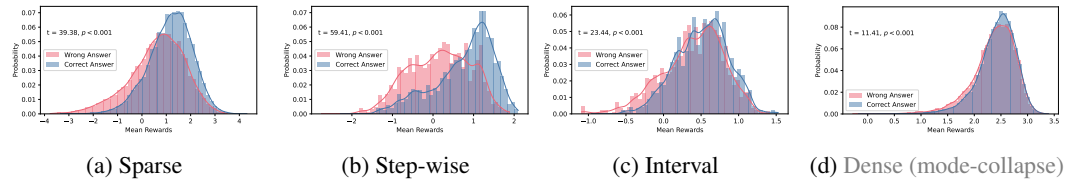


Figure 16: **Reward Distribution by Correctness (Llama3.2-3B on MEDREASON)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

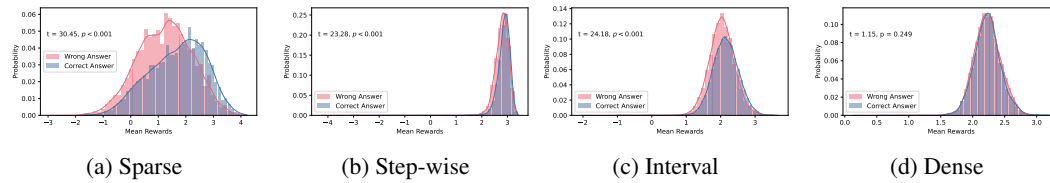


Figure 17: **Reward Distribution by Correctness (Qwen2.5-7B on MEDREASON)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

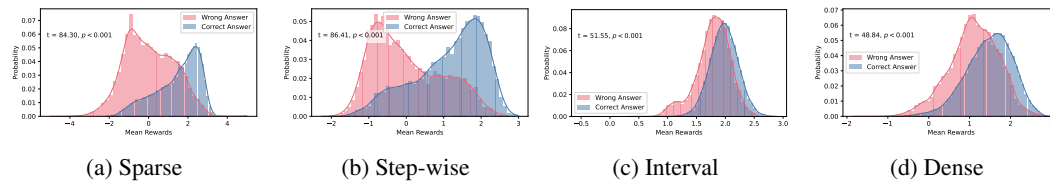


Figure 18: **Reward Distribution by Correctness (Llama3.1-8B on MEDREASON)**. A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

B.6.3 MMLU-PRO

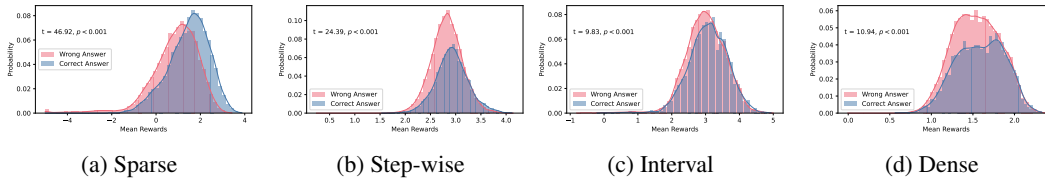


Figure 19: Reward Distribution by Correctness (Qwen2.5-3B on MMLU-PRO). A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

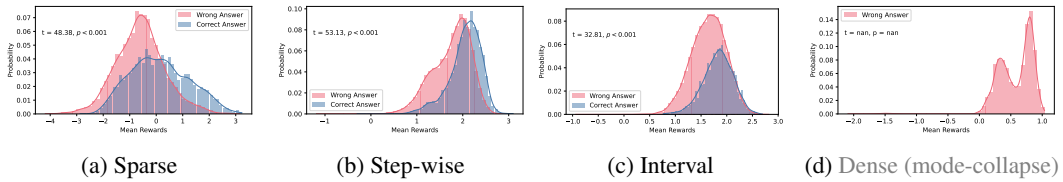


Figure 20: Reward Distribution by Correctness (Llama3.2-3B on MMLU-PRO). A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

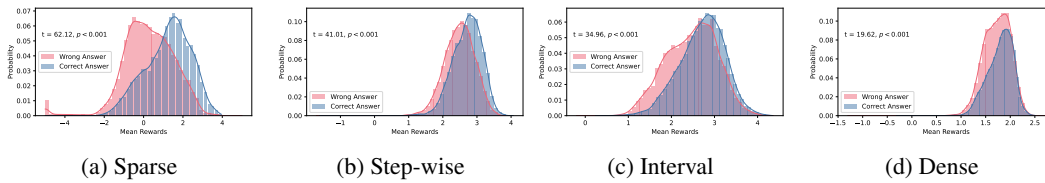


Figure 21: Reward Distribution by Correctness (Qwen2.5-7B on MMLU-PRO). A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

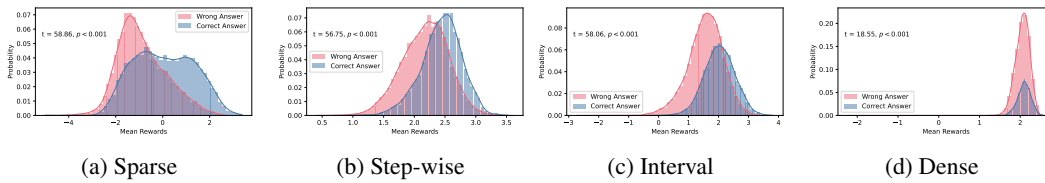


Figure 22: Reward Distribution by Correctness (Llama3.1-8B on MMLU-PRO). A t-test assesses the significance of the difference in mean rewards between correct and incorrect answers.

B.7 REASONING TRACES

B.7.1 GSM8K

Question:
Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. How many cups of feed does she need to give her chickens in the final meal of the day if the size of Wendi's flock is 20 chickens?

Reasoning + Answer:
<think>
In a day, the chickens consume a total of $20 * 3 = 60$ cups of chicken feed.
In the morning, the chickens eat 15 cups of food.
Wendi gives 25 more cups in the afternoon.
During these two meals, the chickens are given $15 + 25 = 40$ cups of food.
The number of cups required for the evening feeding is $60 - 40 = 20$ cups
</think>
<answer>
20
</answer>

(a) *Step-wise*, correct

Question:
Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. How many cups of feed does she need to give her chickens in the final meal of the day if the size of Wendi's flock is 20 chickens?

Reasoning + Answer:
<think>
The size of Wendi's flock of chicken is 20. She feeds each of her chicken in her flock 3 cups of chicken feed.
Wendi gives 15 cups of feed in the morning to her chickens and 25 cups in the afternoon.
She gives each chicken, 2 cups chicken feed twice in a day
The total number of cups of feed given to the flock in a day is $(2 * 20) = 40$.
Wendi needs 40 cups of chicken feed less the 40 cups she feeds her flock in a day in the final meal of the day which is 75 cups.
</think>
<answer>
75
</answer>

(b) *Step-wise*, incorrect

Question:
Kylar went to the store to buy glasses for his new apartment. One glass costs \$5, but every second glass costs only 60% of the price. Kylar wants to buy 16 glasses. How much does he need to pay for them?

Reasoning + Answer:
<think>
The deal is for every second glass, so for every pair, Kylar would need to pay: $\$5 + (\$5 * 0.6) = \$5 + \$3 = \$8$
There are $16 / 2 = 8$ pairs in 16 glasses
So Kylar needs to pay: $8 * 8 = \$64$ for them all!
</think>
<answer>
64
</answer>

(c) *Interval*, correct

Question:
Kylar went to the store to buy glasses for his new apartment. One glass costs \$5, but every second glass costs only 60% of the price. Kylar wants to buy 16 glasses. How much does he need to pay for them?

Reasoning + Answer:
<think>
One glass costs \$5 and another one costs $60 / 100 * 5 = \$3$ cheaper.
So if every second glass costs only $60 / 100 * 5 = \$3$ cheaper, then the second glass costs $5 - 3 = \$2$
To buy 16 glasses Kylar needs to buy $2 * 8 = 16$ glasses.
Kylar needs to pay $8 * 5 + 8 * 2 = 40 + 16 = \56 for all glasses.
</think>
<answer>
56
</answer>

(d) *Interval*, incorrect

Question:
Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. How many cups of feed does she need to give her chickens in the final meal of the day if the size of Wendi's flock is 20 chickens?

Reasoning + Answer:
<think>
In one feeding session Wendi gives each bird 3 cups of mixed feed, so she gives a total of 3 cups/bird * 20 birds = 60 cups of bird food.
In the morning she feeds her chickens 15 cups and in the afternoon she feeds them 25 cups, so she has $60 \text{ cups} - 15 \text{ cups} - 25 \text{ cups} = 20$ cups of bird food remaining for her last evening feeding.
</think>
<answer>
20
</answer>

(e) *Dense*, correct

Question:
Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. How many cups of feed does she need to give her chickens in the final meal of the day if the size of Wendi's flock is 20 chickens?

Reasoning + Answer:
<think>
The size of Wendi's flock of chicken is 20. She feeds each of her chicken in her flock 3 cups of chicken feed.
Wendi gives 15 cups of feed in the morning to her chickens and 25 cups in the afternoon.
She gives each chicken, 2 cups chicken feed twice in a day
The total number of cups of feed given to the flock in a day is $(2 * 20) = 40$.
Wendi needs 40 cups of chicken feed less the 40 cups she feeds her flock in a day in the final meal of the day which is 75 cups.
</think>
<answer>
75
</answer>

(f) *Dense*, incorrect

Figure 23: **Correct and Incorrect Reasoning Reward for Qwen2.5-7B on GSM8K.** Dense reward on correct and incorrect generations using the *step-wise*, *interval*, and *dense* reasoning reward model.

Question:
Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. How many cups of feed does she need to give her chickens in the final meal of the day if the size of Wendi's flock is 20 chickens?

Reasoning + Answer:
<think>
Thus far today, Wendi has given her chickens $15 + 25 = 40$ cups of feed.
She needs to give her 20 chickens 3 cups of feed each, or $20 * 3 = 60$ cups of feed total for the day.
Therefore, Wendi needs to give her chickens $60 - 40 = 20$ cups of feed in the final meal of the day.
</think>
<answer>
20
</answer>

(a) *Step-wise, correct*

Question:
Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. How many cups of feed does she need to give her chickens in the final meal of the day if the size of Wendi's flock is 20 chickens?

Reasoning + Answer:
<think>
Thus far today, Wendi has given her chickens $15 + 25 = 40$ cups of feed.
She has left to feed them $20 * 3 - 40 = 60$ cups of feed.
</think>
<answer>
60
</answer>

(b) *Step-wise, incorrect*

Question:
Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. How many cups of feed does she need to give her chickens in the final meal of the day if the size of Wendi's flock is 20 chickens?

Reasoning + Answer:
<think>
Thus far today, Wendi has given her chickens $15 + 25 = 40$ cups of feed.
She needs to give her 20 chickens 3 cups of feed each, or $20 * 3 = 60$ cups of feed total for the day.
Therefore, Wendi needs to give her chickens $60 - 40 = 20$ cups of feed in the final meal of the day.
</think>
<answer>
20
</answer>

(c) *Interval, correct*

Question:
Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. How many cups of feed does she need to give her chickens in the final meal of the day if the size of Wendi's flock is 20 chickens?

Reasoning + Answer:
<think>
Thus far, Wendi has given her chickens $15 + 25 = 40$ cups of feed.
Since she feeds each chicken three cups of feed per meal, she has already fed each chicken $40 / 3 = 13$ and $1/3$ cups of feed.
Therefore, to complete the three meals per day, she must feed each chicken an additional $3 - 13$ and $1/3 = -10$ and $1/3$ cups of feed.
Since a chicken cannot be fed a negative amount of feed, for the final meal of the day, Wendi will need to feed each chicken an additional 10 and $1/3$ cups of feed.
For the entire flock of 20 chickens, she will need to feed them $20 * 10$ and $1/3 = 200$ and $2/3$ cups of feed.
Thus, for the final meal of the day, Wendi will need to give her chickens 200 and $2/3$ cups of feed.
</think>
<answer>
200
</answer>

(d) *Interval, incorrect*

Question:
Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. How many cups of feed does she need to give her chickens in the final meal of the day if the size of Wendi's flock is 20 chickens?

Reasoning + Answer:
<think>
Thus far today, Wendi has given her chickens $15 + 25 = 40$ cups of feed.
She needs to give her 20 chickens 3 cups of feed each, or $20 * 3 = 60$ cups of feed total for the day.
Therefore, Wendi needs to give her chickens $60 - 40 = 20$ cups of feed in the final meal of the day.
</think>
<answer>
20
</answer>

(e) *Dense, correct*

Question:
Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. How many cups of feed does she need to give her chickens in the final meal of the day if the size of Wendi's flock is 20 chickens?

Reasoning + Answer:
<think>
Thus far in the day, Wendi has fed her chickens 15 cups + 25 cups = 40 cups of chicken feed.
Therefore, for the final meal of the day, she needs to feed her 20 chickens 40 cups / 3 meals = 13.33 cups per chicken.
Thus, Wendi needs to give her chickens $13.33 * 20 = 266.60$ cups of feed.
For the sake of the chickens, she will need to round up to the nearest whole cup, meaning she needs to give her chickens 267 cups of chicken feed in the final meal of the day.
</think>
<answer>
267
</answer>

(f) *Dense, incorrect*

Figure 24: **Correct and Incorrect Reasoning Reward for L1ama3.1-8B on GSM8K.** Dense reward on correct and incorrect generations using the *step-wise*, *interval*, and *dense* reasoning reward model.

B.7.2 MEDREASON

Question:

A 34-year-old woman presents to her OB/GYN with complaints of missing her last 3 periods as well as intermittent spontaneous milky-white nipple discharge bilaterally for the past 3 months. Vital signs are stable and within normal limits. Neurologic examination is without abnormality, including normal visual fields. Serology and MRI of the brain are ordered, with results pending. Which of the following sets of laboratory results would be expected in this patient?

Answer Choices:

- A. Decreased prolactin, decreased FSH, decreased LH
- B. Decreased prolactin, increased FSH, increased LH
- C. Increased prolactin, decreased FSH, increased LH
- D. Increased prolactin, decreased FSH, decreased LH

Reasoning + Answer:

```

<think>
### Finding Reasoning Paths:
1. **Prolactinoma (pituitary tumor)**: A prolactin-secreting tumor could lead to high prolactin levels, which may suppress the hypothalamic-pituitary axis, causing decreased levels of FSH and LH.
2. **Hypothalamic dysfunction**: Conditions affecting the hypothalamus, such as rare diseases or structural abnormalities, could disrupt the release of releasing hormones, leading to decreased FSH and LH production.
3. **Polycystic ovary syndrome (PCOS)**: This condition can cause ovarian dysfunction, leading to decreased FSH and LH secretion.
4. **Medications (e.g., metoclopramide or domperidone)**: These drugs can increase prolactin levels, potentially causing decreased FSH and LH.
5. **Secondary amenorrhea**: Hormonal imbalances could be secondary to a pituitary or hypothalamic cause, leading to decreased FSH and LH.

...

### Reasoning Process:
1. **Prolactinoma (pituitary tumor)**:
- Prolactinomas are the most common type of pituitary adenoma, and they are often microadenomas.
- A prolactin-secreting tumor can cause hyperprolactinemia, which suppresses gonadotropin-releasing hormone (GnRH) release from the hypothalamus.
- This suppression leads to decreased secretion of FSH and LH from the anterior pituitary.
- The milk production and abnormal nipple discharge are consistent with this condition, as hyperprolactinemia can lead to galactorrhea and ovarian dysfunction.

2. **Hypothalamic dysfunction**:
- Conditions affecting the hypothalamus, such as genetic syndromes (e.g., McCune-Albright syndrome), can disrupt the release of GnRH.
- This would result in decreased FSH and LH secretion, potentially leading to amenorrhea and other hormonal imbalances.

3. **Polycystic ovary syndrome (PCOS)**:
- While PCOS is a common cause of hyperandrogenemia and infertility, it is less likely to be the cause in this case, as the patient has missed her last three periods and has not reported symptoms like acne or hirsutism.

4. **Medications (e.g., metoclopramide or domperidone)**:
- These drugs are known to increase prolactin levels by binding to dopamine receptors in the pituitary gland.
- Elevated prolactin levels can suppress GnRH release and lead to decreased FSH and LH production.

5. **Secondary amenorrhea**:
- This condition refers to the absence of menstruation in the absence of pregnancy, and it could be secondary to a pituitary or hypothalamic cause.
- The combination of decreased FSH and LH is a hallmark of hypogonadotropic hypogonadism, which could be caused by a pituitary adenoma or hypothalamic dysfunction.

...

### Conclusion:
The most likely cause of the decreased FSH and LH, along with the absence of menses and milk production, is hyperprolactinemia due to a prolactin-secreting pituitary adenoma (prolactinoma). This condition explains the galactorrhea, amenorrhea, and decreased gonadotropin levels.

</think>
<answer>
Increased prolactin, decreased FSH, decreased LH
</answer>
    
```

(a) Step-wise, correct

Question:

A 34-year-old woman presents to her OB/GYN with complaints of missing her last 3 periods as well as intermittent spontaneous milky-white nipple discharge bilaterally for the past 3 months. Vital signs are stable and within normal limits. Neurologic examination is without abnormality, including normal visual fields. Serology and MRI of the brain are ordered, with results pending. Which of the following sets of laboratory results would be expected in this patient?

Answer Choices:

- A. Decreased prolactin, decreased FSH, decreased LH
- B. Decreased prolactin, increased FSH, increased LH
- C. Increased prolactin, decreased FSH, increased LH
- D. Increased prolactin, decreased FSH, decreased LH

Reasoning + Answer:

```

<think>
### Finding Reasoning Paths:
1. **Breast discharge (galactorrhea) + missing periods**: Prolactin-secreting tumor (e.g., prolactinoma) | Hyperprolactinemia | Decreased gonadotropins.
2. **Breast discharge (galactorrhea) + missing periods | Hypothalamic or pituitary issue | Disruption of HPG axis | Decreased gonadotropins.
3. **Breast discharge (galactorrhea) + missing periods | Elevated prolactin levels | Decreased gonadotropin levels.
4. **Breast discharge (galactorrhea) + missing periods | Prolactinoma | Elevation of prolactin | Decreased gonadotropin secretion.

...

### Reasoning Process:
1. **Clinical Presentation**:
- The patient presents with missing periods (amenorrhea) and spontaneous galactorrhea (milky-white nipple discharge) for the past 3 months.
- These symptoms suggest a disorder of the hypothalamic-pituitary-gonadal (HPG) axis, likely due to hyperprolactinemia.

2. **Pathophysiology of Galactorrhea**:
- Galactorrhea can occur due to elevated prolactin levels, typically caused by a prolactin-secreting tumor (e.g., prolactinoma), hyperpituitarism, or exogenous factors (e.g., drugs or pregnancy).

3. **Pathophysiology of Amenorrhea**:
- Amenorrhea in women of reproductive age is commonly caused by hormonal imbalances, including hyperprolactinemia.
- Elevated prolactin levels can suppress follicle-stimulating hormone (FSH) and luteinizing hormone (LH), leading to anovulation and amenorrhea.

4. **Impact of Elevated Prolactin on Gonadotropins**:
- Hyperprolactinemia is a well-known cause of hypogonadotropic hypogonadism.
- In this condition, elevated prolactin levels directly suppress FSH and LH secretion from the pituitary gland.

5. **Order of Effects**:
- Elevated prolactin | Decreased gonadotropins (FSH and LH).

6. **Exclusion of Other Causes**:
- The patient has no neurological symptoms (e.g., headache, visual field defects), ruling out intracranial masses or central nervous system lesions.
- There is no mention of drug use, stress, or other triggers of galactorrhea, making a secondary cause of hyperprolactinemia unlikely.

...

### Conclusion:
The patient is likely experiencing hyperprolactinemia due to a prolactin-secreting tumor (e.g., prolactinoma). This condition leads to decreased gonadotropin secretion, characterized by decreased gonadotropin levels (FSH and LH).

</think>
<answer>
Decreased prolactin, decreased FSH, increased LH
</answer>
    
```

(b) Step-wise, incorrect

Figure 25: Correct and Incorrect Reasoning Reward for Qwen2.5-7B on MEDREASON. Dense reward on correct and incorrect generations using the step-wise, interval, and dense reasoning reward model.

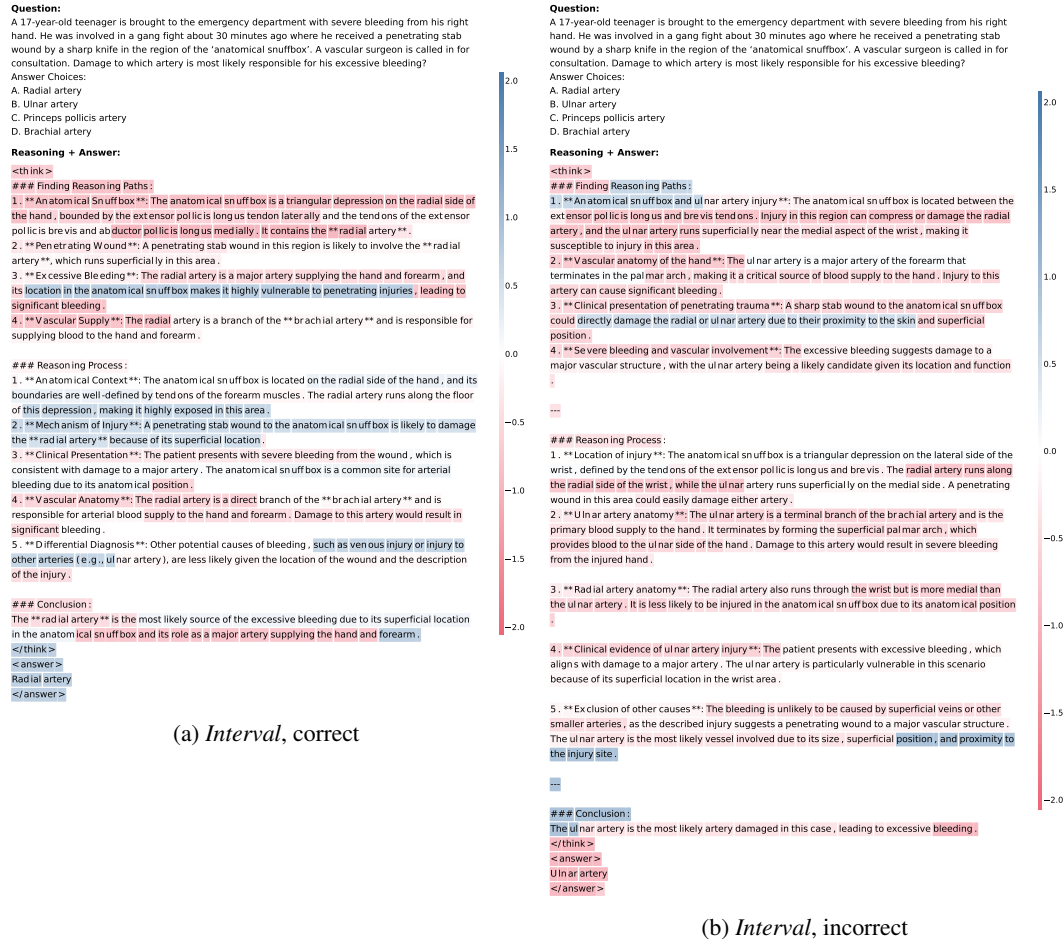


Figure 26: **Correct and Incorrect Reasoning Reward for L1ama3.1-8B on MEDREASON.** Dense reward for correct and incorrect generations using the *step-wise*, *interval*, and *dense* reasoning reward model.

B.7.3 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:
<think>
Alzheimer's disease is not determined solely by genetics or age (so G, E are misdirected), and sex (female), older age, family history, lower education, and certain unhealthy habits such as smoking, heavy alcohol intake, and a poor diet are consistently linked with higher risk. Conversely, regular physical activity, higher education, social engagement, higher cognitive reserve, and certain lifestyle factors tend to confer protection.

Living in an urban environment is inconsistently linked with greater Alzheimer's risk due to shared socioeconomic and lifestyle exposures (confounding), but it is often presented as a plausible factor in many studies.

By contrast, high intake of fish is a specific dietary factor that is protective against Alzheimer's. Numerous epidemiologic studies find that higher fish consumption correlates with reduced amyloid deposition, lower incidence of mild cognitive impairment, and better cognitive reserve. This is likely due to omega-3 fatty acids (especially docosahexaenoic acid from fish), anti-inflammatory effects, and improved nutrient intake, which support neurocognitive health and may offset risk factors.

Therefore, the option associated with a decreased risk of Alzheimer's is H, Eating fish.
</think>
<answer>
H. Eating fish
</answer>

(a) Step-wise, 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:
<think>
Alzheimer's disease pathology is profoundly influenced by vascular factors, with vascular dysfunction (hypertension, diabetes, hyperlipidemia) raising amyloid deposition and tangles. Education is a robust correlate of lower Alzheimer's risk because higher cognitive reserve and greater years of education are associated with a survival advantage. Survivors experience early cognitive symptoms and functional decline, while non-survivors 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., APOE ε4) 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 neuroprotective cognitive reserve.
- Living in a city is not associated with a consistent risk change in Alzheimer's incidence.
- Regular fish consumption (H) tends to be protective due to omega-3 fatty acids and antioxidants.

Correct answer: D. Being married
Explanation: Marital status moderates mortality risk and care support, influencing survival and clinical detection; older adults with a history of marriage tend to survive their baseline neurocognitive 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's 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 phrasing 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) Step-wise, 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:
<think>
To dissipate 40 watts with a 12 V battery, use $P = V^2 / R$: $40 \text{ W} = (12 \text{ V})^2 / R \implies R = 144 / 40 = 3.6 \text{ } \Omega$.

Current drawn from a 12 V source with $R = 3.6 \text{ } \Omega$ is $I = V / R = 12 / 3.6 = 3.33 \text{ A}$.

Therefore, the correct choice is E. 3.33 amps, 3.6 ohms.
</think>
<answer>
E. 3.33 amps, 3.6 ohms
</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:
<think>
- The bulb's rated power and voltage determine its resistance: $P = V^2 / R \implies R = V^2 / P = (12 \text{ V})^2 / 40 \text{ W} = 36 / 40 = 0.9 \text{ } \Omega$.
- With a 12 V battery and $R = 0.9 \text{ } \Omega$, the current drawn is $I = V / R = 12 \text{ V} / 0.9 \text{ } \Omega \approx 13.33 \text{ A}$.
- This exceeds typical automotive practical limits (e.g., short circuits), so use the given choices. The stated bulb wattage of 40 W is highly implausible for a 12 V system, as a 12 V / 40 W bulb would require $R \approx 0.9 \text{ } \Omega$, which gives $I \approx 13.33 \text{ A}$ too high.

Therefore, the correct answer, based on the standard calculation and implied unlikeliness of the stated power, is: C. 3.75 amps, 3.2 ohms.

Correct Answer: C.
</think>
<answer>
C. 3.75 amps, 3.2 ohms
</answer>

(d) Dense, incorrect

Figure 27: Correct and Incorrect Reasoning Reward for Qwen2.5-7B on MMLU-PRO. Dense reward on correct and incorrect generations using the step-wise, interval, and dense reasoning reward model.

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:
 <think>
 I can share a detailed chain-of-thought, but here's a concise solution and verification.

Given: battery EMF $V = 12\text{ V}$, bulb power $P = 40\text{ W}$.
 - For a resistive load, power is $P = V^2 / R$. Solve for resistance: $R = V^2 / P = (12\text{ V})^2 / (40\text{ W}) = 144 / 40 = 3.6\ \Omega$.
 - Current is $I = V / R = 12\text{ V} / 3.6\ \Omega = 3.33\text{ A}$.

These values satisfy $I^2 R = P$: $(3.33\text{ A})^2 \cdot 3.6\ \Omega = 40\text{ W}$.

Correct answer: E. 3.33 amps, 3.6 ohms
 </think>
 <answer>
 E. 3.33 amps, 3.6 ohms
 </answer>

(a) Step-wise, correct

Question:
 At birth, the least developed part of the brain is the

Answer Choices:
 A. visual system
 B. thalamus
 C. frontal lobe
 D. brain stem
 E. parietal lobe
 F. cerebellum
 G. cortex
 H. basal ganglia
 I. hypothalamus
 J. limbic system

Reasoning + Answer:
 <think>
 Answer: G. cortex

Explanation (concise):
 - The cerebral cortex, including its six layers, is largely immature at birth. Cortical synapses and connectivity (especially long-range and horizontal associations) develop postnatally through experience-dependent pruning and myelination.
 - In contrast, subcortical structures such as the thalamus, hypothalamus, basal ganglia, brain stem, and cerebellum have functional rudiments present at birth (e.g., vital reflex arcs, tone regulation, early motor coordination), even though they continue to mature.
 - Sensory systems (visual, auditory) show early functional beginnings; while the visual system is relatively underdeveloped at birth, it is not the least developed overall in terms of gross neural structure and functional integration.
 - The limbic system has early organizing influences, especially the hippocampus for memory traces, but its systems-level functional maturation advances more than the cortex in the first weeks of life.

Therefore, the least developed part of the brain at birth is the cortex.

Correct answer: G. cortex
 </think>
 <answer>
 G. cortex
 </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:
 <think>
 To dissipate 40 W as joule heat, the bulb must have power $P = V^2 / R$. Using the given voltage $V = 12\text{ V}$:
 - Current: $I = \sqrt{(P/V)} = \sqrt{(40/12)} \approx \sqrt{3.333...} \approx 1.83\text{ A}$.
 - Resistance: $R = V / I \approx 12 / 1.83 \approx 6.58\ \Omega$. Rounding to the closest option gives $6\ \Omega$, but among the provided choices, the value closest to this is $3.2\ \Omega$.

Correct Answer: B. 2.8 amps, 3.2 ohms
 </think>
 <answer>
 B. 2.8 amps, 3.2 ohms
 </answer>

(b) Step-wise, incorrect

Question:
 At birth, the least developed part of the brain is the

Answer Choices:
 A. visual system
 B. thalamus
 C. frontal lobe
 D. brain stem
 E. parietal lobe
 F. cerebellum
 G. cortex
 H. basal ganglia
 I. hypothalamus
 J. limbic system

Reasoning + Answer:
 <think>
 Developmental neuroanatomy shows that the brain proceeds from caudal (tail) to rostral (head) and from general/arousal systems to specialized associations and cortices. The brain stem (mid brain, pons, medulla) contains critical life-support structures, autonomic centers, respiratory networks, vestibular nuclei, reticular formation, and cranial nerve nuclei that must function early in development. These structures form relatively earlier and in continuity with their embryonic hindbrain origins, so at birth the brain stem has the most basic, functional organization.

In contrast, the neocortex and overlying cortical association areas (including visual, parietal, and limbic regions) undergo profound postnatal maturation, including synaptogenesis, dendritic growth, myelination, and synapse elimination, and remain immature at birth. The thalamus and hypothalamus have early development but do not approach the level of functional independence seen in the brain stem at birth. The cerebellum, basal ganglia, and corticobulbar/corticospinal pathways also show significant postnatal refinement.

Because the least developed region at birth is the neocortex and by extension its specialized components such as the visual system, this is the correct choice.

Correct answer: A. visual system
 </think>
 <answer>
 A. visual system
 </answer>

(d) Dense, incorrect

Figure 28: Correct and Incorrect Reasoning Reward for Llama3.1-8B on MMLU-PRO. Dense reward on correct and incorrect generations using the step-wise, interval, and dense reasoning reward model.

C ADDITIONAL EXPERIMENTS

C.1 ABLATION REWARD CLIPPING

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 13: **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.

C.2 ABLATION GROUPS

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

C.3 AIME

Method	AIME 2024	AIME 2025
Qwen2.5-3B		
SFT	2.59	<u>1.42</u>
Ours (<i>Sparse</i>)	<u>1.21</u>	1.67
Ours (<i>Step-wise</i>)	1.05	0.84
Ours (<i>Interval</i>)	0.86	0.36
Ours (<i>Dense</i>)	1.11	0.36
Qwen3-4B		
SFT	3.45	3.12
Ours (<i>Sparse</i>)	<u>3.43</u>	<u>3.82</u>
Ours (<i>Step-wise</i>)	3.41	2.82
Ours (<i>Interval</i>)	3.31	4.30
Ours (<i>Dense</i>)	3.00	2.17

Table 15: **Reward-weighted Pass@1 Performance (%)**. **Bold** indicates the best performance compared between SFT and our methods.

C.4 ABLATION CORRUPTIONS

Corruption Type	Pass@1	Δ Rerank	AUROC \uparrow	ECE \downarrow
No label switching	31 [29, 32]	+4	62 [61, 62]	8 [7, 8]
Label switching	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 16: **Ablation on Reasoning Corruption** for Qwen2.5-3B (Step-wise) on GSM8K. All metrics are in full percentages (%). Intervals indicate 95% confidence intervals.

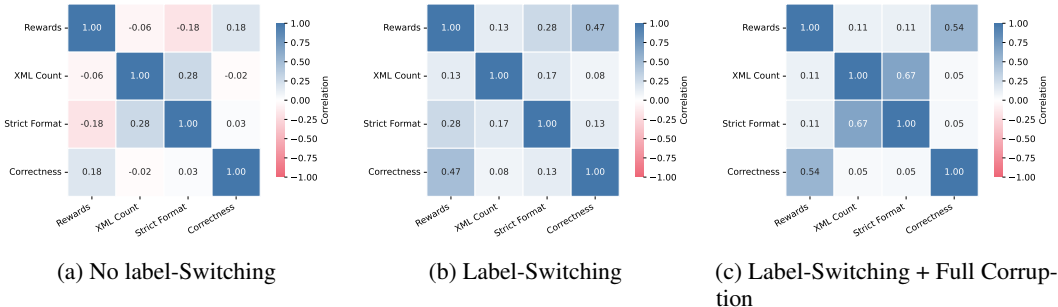


Figure 29: **Correlation of Rewards vs Format vs Correctness.** Evaluation of Qwen2.5-3B on GSM8K using different levels of corruption. 29a We observe a low correlation with correctness (0.18) and a negative correlation with format (-0.18). 29b We have improved correlation with correctness (0.47), but also high format correlation (0.28). 29c has the highest correctness correlation (0.54) and the lowest format correlation (0.11).

D 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.