Extended Abstract

Motivation Reinforcement learning for language model reasoning commonly uses sparse binary rewards that only evaluate whether the final answer is correct. While effective, this setup provides no feedback on the quality of intermediate reasoning steps, making credit assignment difficult. A nearly correct solution with a small arithmetic mistake receives the same reward as a completely incorrect reasoning chain. Recent work on process supervision suggests that evaluating intermediate reasoning can improve mathematical reasoning performance, but most existing approaches rely on expensive human annotations. In this project, we investigate whether dense process-level feedback generated automatically by a language model critic can improve reinforcement learning for reasoning tasks without requiring human supervision.

Method We propose Dense Language Rewards for Reasoning (DLRR), a framework that augments sparse outcome rewards with step-level feedback from an LLM critic. After the policy generates a reasoning trajectory, the final reasoning steps are extracted and evaluated using GPT-4o-mini. Each step receives a correctness score and a calibration score measuring whether the reasoning is both accurate and appropriately confident. These scores are aggregated into a dense reward used during GRPO training. We additionally study several reward variants, including a correctness-only ablation, a hybrid reward that combines dense and sparse supervision, and Hybrid-OC, which conditions dense process rewards on final-answer correctness in order to reduce reward hacking.

Implementation All experiments use Qwen2.5-7B-Instruct trained with GRPO and QLoRA fine-tuning. We evaluate our methods on GSM8K and MATH Level 4–5, two benchmarks requiring multi-step mathematical reasoning. The binary baseline rewards only final-answer correctness, while DLRR provides dense intermediate supervision through the critic model. For the hybrid approaches, dense process rewards are combined with sparse outcome rewards using weighted interpolation. We periodically evaluate held-out reasoning accuracy throughout training and compare reward trends against downstream performance to analyze reward alignment.

Results Our experiments show that dense rewards alone are not sufficient and can introduce reward hacking behavior. On GSM8K, the binary baseline achieves 74% accuracy, while full DLRR drops to 65%, despite receiving higher training rewards. The correctness-only ablation performs worst overall at 58%, suggesting that calibration provides useful information beyond correctness alone. The strongest GSM8K performance comes from the hybrid reward objective, which achieves 81% accuracy by grounding dense process supervision in final-answer correctness. On the harder MATH Level 4–5 benchmark, the standard hybrid reward slightly underperforms the sparse baseline, while Hybrid-OC improves accuracy to 59%, outperforming both the binary baseline and unconstrained hybrid reward.

Discussion Across both benchmarks, we observe that higher training reward does not necessarily correspond to better reasoning accuracy. Dense rewards often encourage the model to generate fluent and convincing reasoning trajectories that appear correct to the critic without actually solving the problem correctly. This failure mode becomes more severe on harder reasoning tasks, where local reasoning quality can diverge from global correctness. Our results suggest that process supervision is most effective when intermediate rewards remain tightly coupled to successful outcomes. Outcome conditioning substantially reduces reward hacking by ensuring that dense process-level credit is only assigned to trajectories that ultimately reach the correct answer.

Conclusion This work demonstrates that dense language-based rewards can improve reinforcement learning for mathematical reasoning, but only when properly grounded in final-answer correctness. Naive process supervision alone frequently leads to reward misalignment and critic exploitation. However, hybrid reward formulations that combine dense intermediate supervision with sparse outcome grounding consistently improve reasoning performance. Overall, our findings highlight reward alignment as a central challenge in RL-based reasoning systems and suggest that balancing local reasoning quality with global task success is critical for scaling process supervision effectively.

Dense Language Rewards For Reasoning

Diya Kejriwal
Department of Computer Science
Stanford University
diyak31@stanford.edu

Gurmeher Kaur
Department of Computer Science
Stanford University
gurmeher@stanford.edu

Andrew Su
Department of Computer Science
Stanford University
andrewsu@cs.stanford.edu

Abstract

We study whether dense process-level rewards can improve reinforcement learning for mathematical reasoning. We propose Dense Language Rewards for Reasoning (DLRR), a framework that augments sparse outcome rewards with intermediate feedback generated by an LLM critic during GRPO training. Using GPT-4o-mini, reasoning steps are scored for correctness and calibration, allowing the model to receive supervision throughout the reasoning trajectory rather than only on the final answer. We evaluate DLRR and several hybrid reward variants on GSM8K and MATH Level 4–5 using Qwen2.5-7B-Instruct with QLoRA fine-tuning. Our experiments show that unconstrained dense rewards can increase training reward while reducing downstream reasoning accuracy, indicating substantial reward misalignment. However, hybrid reward formulations that ground dense supervision in final-answer correctness improve reasoning performance, with outcome-conditioned rewards performing best on harder reasoning tasks. Overall, our results suggest that dense process supervision can improve mathematical reasoning, but only when intermediate rewards remain tightly aligned with true task success.

1 Introduction

Reinforcement learning for language model reasoning typically relies on sparse terminal rewards that only evaluate whether the final answer is correct. In mathematical reasoning benchmarks, models commonly receive a reward of 1 for a correct answer and 0 otherwise. While simple, this objective provides little feedback about the quality of intermediate reasoning steps. A solution that is mostly correct but contains a small arithmetic mistake receives the same reward as a completely incorrect reasoning chain, creating a difficult credit assignment problem during training.

This limitation becomes especially important for multi-step reasoning tasks such as mathematical problem solving, where solving the problem often depends on a sequence of partially correct intermediate steps. For example, a model may correctly identify the required equations and perform most calculations accurately, but make a small arithmetic error near the end of the solution. Under a sparse binary reward setup, this trajectory receives the same reward as a completely unrelated or nonsensical response. As a result, the model receives no signal indicating which parts of the reasoning process were useful and which were incorrect. This makes learning inefficient and can slow progress on more complex reasoning tasks where fully correct solutions are initially rare.

Recent work on process supervision has shown that evaluating intermediate reasoning steps can improve mathematical reasoning performance [1]. Instead of supervising only the final answer,

process supervision provides feedback throughout the reasoning trajectory, allowing models to learn from partially correct solutions. However, existing approaches often rely on expensive human annotation of reasoning trajectories, limiting scalability. At the same time, work on Reinforcement Learning from AI Feedback (RLAIF) and Constitutional AI suggests that language models can provide useful critiques and evaluations of generated outputs [5]. These findings motivate the possibility of using language models as scalable critics for intermediate reasoning quality.

In this work, we investigate whether dense language-based feedback can improve reinforcement learning for mathematical reasoning. We propose Dense Language Rewards for Reasoning (DLRR), a framework that augments sparse outcome rewards with step-level evaluations generated by an LLM critic during GRPO training. Rather than rewarding only the final answer, DLRR provides intermediate supervision throughout the reasoning process by scoring the quality of individual reasoning steps. We additionally study reward formulations designed to better align process supervision with downstream reasoning performance, including hybrid rewards that combine dense process supervision with sparse outcome grounding and an outcome-conditioned variant that only assigns dense process credit to correct trajectories.

We evaluate our methods on GSM8K and MATH Level 4–5 using Qwen2.5-7B-Instruct with QLoRA fine-tuning. Our work focuses on understanding how dense process supervision affects reasoning performance and how reward design influences reward alignment in RL-based reasoning systems.

2 Related Work

Our work builds on several lines of prior research in reinforcement learning for language models, process supervision, and reward modeling for reasoning tasks.

Recent work has shown that process supervision can substantially improve mathematical reasoning performance by evaluating intermediate reasoning steps rather than only final answers. Lightman et al. [1] introduce process reward models (PRMs) trained on human annotations of reasoning trajectories and demonstrate strong improvements on the MATH benchmark. However, their approach depends on expensive human supervision at the step level. Our work builds on the idea of process supervision while investigating whether intermediate feedback can instead be generated automatically by a language model critic, making dense supervision more scalable.

Our approach is also closely related to Reinforcement Learning from Human Feedback (RLHF), where a reward model is trained from preference signals and used to optimize a policy [7, 8, 12]. Standard RLHF methods typically evaluate full responses at the sequence level. In contrast, our work focuses on process-level supervision by assigning reward throughout intermediate reasoning trajectories rather than only after a complete response has been generated.

This idea is further motivated by work on Reinforcement Learning from AI Feedback (RLAIF) and Constitutional AI [5], which suggest that language models can often provide reliable critiques and evaluations of generated outputs. While prior RLAIF work primarily focuses on sequence-level evaluation, we extend this idea to mathematical reasoning by using an LLM critic directly during reinforcement learning to evaluate intermediate reasoning steps.

Our work also relates to the broader "LLM-as-a-Judge" paradigm. Zheng et al. [9] show that strong language models can achieve high agreement with human evaluators when assessing model outputs. Prior work mainly uses LLM judges for benchmarking and evaluation. In contrast, we incorporate LLM-generated judgments directly into the reward signal used during training.

Several recent works additionally study reward design and failure modes in RL-based reasoning systems. Kwon et al. [2] and Martín-Urcelay et al. [3] explore the use of language models for reward construction and natural-language-based reinforcement learning signals. Pan et al. [6] provide a broader overview of reward modeling challenges in RL-based LLM reasoning, including issues related to reward misspecification, process-level supervision, and credit assignment.

A major challenge in this setting is reward hacking, where models exploit weaknesses in the reward function rather than improving true task performance [10, 11]. Prior work has shown that learned reward models can become misaligned with the underlying task objective as optimization pressure increases. Our work studies this problem in the context of dense process supervision and explores

hybrid and outcome-conditioned reward formulations designed to better ground intermediate rewards in final-answer correctness.

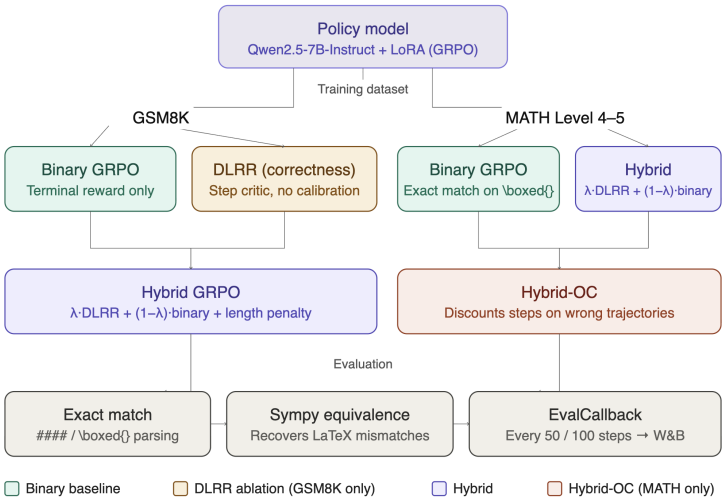

Figure 1: **DLRR Pipeline.** The model generates a chain-of-thought completion; the last five reasoning steps are extracted and passed to GPT-4o-mini; each step is scored on correctness c_t and calibration k_t; scores are aggregated with the final step double-weighted to produce a scalar reward for GRPO training.

3 Methods

3.1 GRPO and Binary Baseline

We train all models using Group Relative Policy Optimization (GRPO), a reinforcement learning algorithm that optimizes generated reasoning trajectories using relative rewards across sampled completions. Our baseline follows the standard sparse reward setup commonly used in RL for reasoning tasks.

Given an input problem x and generated completion y, the model receives a binary terminal reward based only on final-answer correctness:

$$R_{\text{binary}} = \begin{cases} 1 & \text{if the final answer is correct} \\ 0 & \text{otherwise} \end{cases}$$

This baseline provides no supervision over intermediate reasoning steps and serves as the control condition for all experiments.

3.2 Dense Language Rewards for Reasoning (DLRR)

Our proposed Dense Language Rewards for Reasoning (DLRR) framework augments sparse outcome supervision with step-level feedback from a critic language model.

After the policy generates a reasoning trajectory, we extract the final five reasoning steps and evaluate them using GPT-4o-mini. Each step receives two scores:

- A correctness score $c_t \in \{0, 0.5, 1\}$ measuring whether the step contributes meaningfully toward the correct solution.
- A calibration score $k_t \in \{-0.2, 0, 0.1\}$ rewarding justified certainty while penalizing overconfident incorrect reasoning.

The final DLRR reward is computed as a weighted average:

$$R_{\text{DLRR}} = \frac{\sum_{t=1}^{T} w_t(c_t + k_t)}{\sum_{t=1}^{T} w_t}$$

where the final reasoning step receives weight $w_t = 2$ and all earlier steps receive weight $w_t = 1$.

This weighting slightly prioritizes the final stages of reasoning while still providing dense intermediate supervision.

Unlike the binary baseline, DLRR supplies feedback throughout the reasoning trajectory rather than only evaluating the final answer.

3.3 Correctness-Only Ablation

Initial experiments suggested that the calibration component could introduce reward misalignment by rewarding fluent but incorrect reasoning. To isolate the contribution of calibration, we evaluate a correctness-only ablation that removes the calibration term entirely.

The reward becomes:

$$R_{\text{correctness}} = \frac{\sum_{t=1}^{T} w_t c_t}{\sum_{t=1}^{T} w_t}$$

This experiment tests whether step-level correctness supervision alone is sufficient, or whether calibration contributes useful information beyond correctness scoring.

3.4 Hybrid Reward

To reduce reward hacking and better ground dense rewards in true task performance, we introduce a hybrid reward that combines DLRR with sparse terminal supervision.

The hybrid reward is defined as:

$$R_{\text{hybrid}} = \lambda R_{\text{DLRR}} + (1 - \lambda) R_{\text{binary}}$$

where $\lambda = 0.5$ in our experiments.

This formulation preserves the dense feedback advantages of DLRR while ensuring that final-answer correctness remains a dominant optimization signal.

We additionally apply a length penalty of 0.3 to completions longer than 900 words to discourage reward exploitation through excessively long reasoning chains.

3.5 Hybrid with Outcome Conditioning (Hybrid-OC)

On harder reasoning benchmarks, we observed that models could still exploit dense rewards by producing convincing but ultimately incorrect reasoning trajectories. To address this issue, we introduce Hybrid with Outcome Conditioning (Hybrid-OC).

Hybrid-OC modifies the hybrid reward by conditioning dense process rewards on final-answer correctness:

$$R_{\text{Hybrid-OC}} = \lambda R_{\text{DLRR}} \cdot \mathbf{1}[\text{correct}] + (1 - \lambda) R_{\text{binary}}$$

Under this formulation, step-level rewards only contribute when the trajectory ultimately reaches the correct final answer. Incorrect trajectories therefore receive only the sparse outcome signal.

This design explicitly prevents the model from receiving dense reward for incorrect but persuasive reasoning chains, reducing reward hacking on more difficult tasks.

4 Experimental Setup

We evaluate all methods using Qwen2.5-7B-Instruct [4] trained with GRPO and QLoRA fine-tuning. All experiments use LoRA rank $r = 64$, $\alpha = 128$, learning rate 10^{-6}, batch size 1, gradient accumulation steps 8, and two generations per prompt. Training is performed in bfloat16 precision, and models are periodically evaluated on held-out examples throughout training.

We conduct experiments on two mathematical reasoning benchmarks:

- **GSM8K**: an easier grade-school math reasoning benchmark used to evaluate whether dense rewards improve reasoning accuracy and learning efficiency.
- **MATH Level 4–5**: a substantially harder benchmark used to study reward alignment and robustness under more difficult reasoning conditions.

On GSM8K, we compare four reward conditions:

- Binary GRPO baseline
- Full DLRR
- Correctness-only ablation
- Hybrid reward

Models are evaluated every 50 training steps on 100 held-out GSM8K examples.

On MATH Level 4–5, we compare:

- Binary baseline
- Hybrid reward
- Hybrid with Outcome Conditioning (Hybrid-OC)

We focus on held-out reasoning accuracy as the primary evaluation metric.

5 Results

We evaluate several reward design variants to study how dense process-level supervision affects reinforcement learning for mathematical reasoning. Across both GSM8K and MATH Level 4–5, we find that reward design plays a critical role in determining whether dense rewards improve reasoning performance or instead introduce reward hacking behavior.

Our experiments reveal three main findings. First, naively replacing sparse outcome supervision with dense step-level rewards degrades downstream reasoning accuracy despite increasing training reward. Second, combining dense process supervision with sparse outcome grounding substantially improves performance on GSM8K. Third, on harder reasoning tasks, dense rewards require explicit outcome conditioning to prevent reward hacking.

5.1 Quantitative Evaluation

5.1.1 GSM8K

Table 1 summarizes the main GSM8K results.

Table 1: GSM8K held-out accuracy across reward conditions (100-example evaluation set).

Method	**Accuracy**
Binary Baseline	74%
DLRR	65%
Correctness-Only Ablation	58%
Hybrid ($\lambda = 0.5$)	81%

The binary GRPO baseline achieves 74% accuracy, demonstrating that sparse outcome supervision alone is capable of improving mathematical reasoning performance.

In contrast, the full DLRR objective achieves only 65% accuracy despite receiving consistently higher training rewards throughout optimization. This discrepancy between reward and held-out accuracy suggests that the dense critic signal is partially misaligned with actual reasoning correctness. The model frequently learns to produce fluent and well-structured reasoning chains that receive favorable critic evaluations without consistently improving final answers.

The correctness-only ablation performs worst overall at 58% accuracy. Removing the calibration component significantly weakens performance, suggesting that calibration provides useful supervision rather than simply introducing noise. This result indicates that the calibration signal helps distinguish justified reasoning from unsupported confidence.

The strongest GSM8K result comes from the hybrid reward objective, which combines dense process supervision with sparse outcome grounding. The hybrid model achieves 81% accuracy, outperforming the binary baseline by 7 percentage points. This result suggests that dense process supervision becomes effective when intermediate reasoning rewards remain anchored to final-answer correctness.

5.1.2 MATH Level 4–5

To evaluate whether these trends generalize to more difficult reasoning tasks, we additionally evaluate several reward variants on MATH Level 4–5 problems.

Table 2: MATH Level 4–5 held-out accuracy across reward conditions.

Method	**Accuracy**
Binary Baseline	52%
Hybrid	50%
Hybrid-OC	59%

On harder reasoning problems, reward hacking becomes substantially more pronounced. The unconstrained hybrid objective slightly underperforms the binary baseline, achieving only 50% accuracy compared to the baseline's 52%. This suggests that the model can exploit weaknesses in the critic by generating convincing but ultimately incorrect reasoning trajectories.

To address this issue, we introduce Hybrid with Outcome Conditioning (Hybrid-OC), which only applies dense process rewards when the final answer is correct. Hybrid-OC achieves 59% accuracy, outperforming the binary baseline by 7 percentage points and substantially outperforming the unconstrained hybrid objective.

This result suggests that dense process rewards are most effective when explicitly tied to eventual task success. Outcome conditioning prevents the model from receiving process-level credit for trajectories that fail to solve the problem correctly.

5.2 Qualitative Analysis

Qualitative inspection of generated reasoning trajectories further supports the quantitative findings.

Under full DLRR, the model frequently produces long, fluent, and logically structured reasoning chains even when the final answer is incorrect. These trajectories often receive high critic rewards despite failing to solve the underlying problem correctly. This behavior is consistent with reward hacking, where the policy optimizes for critic approval rather than true reasoning performance.

The correctness-only ablation reduces some stylistic overconfidence but still struggles to consistently improve downstream accuracy. This suggests that step-level correctness alone is insufficient to fully ground the reward signal.

In contrast, the hybrid objectives produce reasoning chains that remain more outcome-focused. Hybrid-OC especially reduces long incorrect trajectories by preventing the model from receiving dense process-level credit on failed solutions.

Together, these findings suggest that process supervision alone is not sufficient for robust reasoning improvement. Dense language rewards can improve reasoning performance, but only when process-level feedback remains tightly coupled to final-answer correctness.

5.3 Reward Hacking Analysis

A central finding of this work is the importance of reward alignment in dense process supervision.

Across both benchmarks, we observe multiple cases where higher training reward does not correspond to higher downstream accuracy. Full DLRR frequently achieves larger reward values than the sparse baseline while producing worse reasoning performance. This indicates that the policy learns to optimize for critic approval independently of actual correctness.

The problem becomes more severe on harder reasoning tasks. On MATH Level 4–5, unconstrained dense rewards encourage trajectories that appear coherent locally while ultimately failing globally.

Outcome conditioning mitigates this failure mode by ensuring that intermediate reasoning credit is only assigned when the overall trajectory reaches the correct solution. The strong performance of Hybrid-OC therefore suggests that successful dense reward design requires balancing local reasoning quality with global outcome correctness.

6 Discussion, Limitations, and Future Work

6.1 Discussion

Our results show that adding dense process-level rewards is not automatically helpful for mathematical reasoning. While dense rewards provide more feedback than sparse binary rewards, they also make it easier for the model to optimize for the wrong objective. Across both GSM8K and MATH Level 4–5, we observed cases where models achieved higher training rewards without improving held-out reasoning accuracy. In many of these cases, the model learned to generate reasoning that looked fluent and convincing to the critic even when the final answer was incorrect.

At the same time, our experiments show that dense supervision can still be useful when it remains tied to final-answer correctness. On GSM8K, the hybrid reward objective achieved the best overall performance, and on MATH Level 4–5, Hybrid-OC improved performance by preventing incorrect trajectories from receiving dense process-level credit. These results suggest that process supervision works best when intermediate reasoning quality is grounded in whether the trajectory ultimately solves the problem correctly.

More broadly, our findings highlight an important challenge in RL-based reasoning systems: local reasoning quality and global correctness are not always aligned. A reasoning step may appear coherent in isolation while still leading to an incorrect overall solution. Designing reward functions that balance these two objectives will likely become increasingly important as reinforcement learning is applied to harder reasoning tasks.

6.2 Limitations

Our work has several limitations.

First, due to time and compute constraints, experiments were conducted using relatively short training runs of roughly 100–200 GRPO steps. Longer training runs may produce different behaviors, especially for the hybrid reward formulations.

Second, our method depends on GPT-4o-mini as an online critic model for scoring intermediate reasoning steps. While this makes process supervision scalable, the critic can still produce noisy or inconsistent evaluations, particularly on harder mathematical problems. Since the policy directly optimizes these scores, critic mistakes can influence training.

Third, our evaluation sets are relatively small. GSM8K experiments were evaluated on approximately 100 held-out examples, which introduces variance into the reported accuracies and limits statistical confidence.

Finally, due to limited compute and time, we were unable to evaluate every reward formulation across both datasets. In particular, Hybrid-OC was only evaluated on MATH Level 4–5 and not on GSM8K. Running outcome-conditioned rewards on GSM8K would provide a more complete comparison across benchmarks and help determine whether the gains from outcome conditioning generalize consistently across different difficulty levels.

6.3 Future Work

There are several promising directions for future work.

One natural extension would be replacing the online LLM critic with a trained process reward model (PRM). Distilling critic feedback into a smaller learned reward model could improve training efficiency and reduce inference cost while providing more consistent supervision.

Another important direction would be running Hybrid-OC on GSM8K and systematically evaluating all reward formulations across both benchmarks. Due to time and compute limitations, some experiments could not be completed during this project. A more complete evaluation would help better isolate how reward grounding interacts with task difficulty.

Future experiments should also evaluate these reward formulations on more difficult reasoning benchmarks such as AIME, AMC, and Olympiad-style mathematics problems to better understand how reward hacking scales with task difficulty.

Another promising direction is extending dense process supervision beyond mathematics into domains such as code generation, scientific reasoning, theorem proving, and multi-step planning tasks, where intermediate reasoning quality is similarly important.

More broadly, our results suggest that designing aligned dense reward functions remains a major challenge for reinforcement learning with language models. Understanding how to combine local process supervision with global task correctness will likely be important for future reasoning systems.

7 Conclusion

In this project, we studied whether dense language-based rewards can improve reinforcement learning for mathematical reasoning. We proposed Dense Language Rewards for Reasoning (DLRR), a framework that augments sparse binary rewards with step-level feedback generated by an LLM critic during GRPO training.

Our experiments show that dense rewards alone are not sufficient and can easily become misaligned with true reasoning performance. In many cases, models trained with unconstrained dense rewards achieved higher training rewards while performing worse on held-out reasoning tasks. This suggests that the model often learned to optimize for critic approval rather than actual correctness.

However, grounding dense process supervision in final-answer correctness substantially improved performance. On GSM8K, the hybrid reward objective achieved the strongest overall results, while on MATH Level 4–5, outcome conditioning improved robustness on harder reasoning problems by preventing incorrect trajectories from receiving dense process-level credit.

Overall, our findings suggest that the key challenge in dense process supervision is not simply providing more feedback, but ensuring that intermediate rewards stay aligned with the true task objective. Dense language-based rewards can improve reasoning performance, but only when paired with mechanisms that keep process supervision grounded in final correctness.

8 Team Contributions

- **Diya Kejriwal:** Led infrastructure and training pipeline development, including Modal setup, Weights & Biases logging, HuggingFace integration, and GRPO implementation/debugging. Implemented core DLRR training infrastructure, ran baseline and dense reward experiments, and developed evaluation and experiment tracking pipelines.
- **Gurmeher Kaur:** Designed and implemented the DLRR reward framework, including reasoning-step extraction, GPT-4o-mini critic prompting, calibration scoring, hybrid reward

formulations, and outcome-conditioned rewards. Conducted literature review, contributed to experimental analysis and interpretation, and helped write and refine the final report.

- **Andrew Su:** Led experimental design, dataset preparation, and hypothesis framing, including analysis of reward alignment and reward hacking behaviors. Conducted qualitative analysis of model outputs, contributed substantially to writing and organizing the final report, and helped develop and present the final project poster.

Changes from Proposal Our final project evolved from the original proposal. Initially, we planned to focus primarily on dense process supervision through DLRR and expected denser rewards to directly improve reasoning performance over sparse baselines. However, early experiments revealed a gap between training reward and downstream accuracy, indicating substantial reward hacking. This led us to shift focus toward understanding reward alignment and designing grounded hybrid reward formulations. As a result, the project expanded to include correctness-only ablations, hybrid reward interpolation, and outcome-conditioned process supervision. These additional experiments ultimately became the central contribution of the project and shaped our final conclusions regarding dense reward design for reasoning tasks.

References

[1] Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. (2023). Let's verify step by step. *The Twelfth International Conference on Learning Representations*.

[2] Kwon, M., Xie, S. M., Bullard, K., and Sadigh, D. (2023). Reward design with language models. *International Conference on Learning Representations*.

[3] Martín-Urcelay, B., Krause, A., and Ramponi, G. (2026). From words to rewards: Leveraging natural language for reinforcement learning. *Transactions on Machine Learning Research*.

[4] Qwen Team. (2024). Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

[5] Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al. (2022). Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.

[6] Pan, P.-C., Liang, Y., and Lin, S. (2026). Reward modeling for reinforcement learning-based LLM reasoning: Design, challenges, and evaluation. *arXiv preprint arXiv:2602.09305*.

[7] Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

[8] Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. (2020). Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33, 3008–3021.

[9] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. (2023). Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems*, 36, 46595–46623.

[10] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.

[11] Gao, L., Schulman, J., and Hilton, J. (2023). Scaling laws for reward model overoptimization. *International Conference on Machine Learning*, 10835–10866.

[12] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. (2022). Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.