university of groningen / faculty of science and engineering / artificial intelligence

Shaping Reasoning Through Rewards: Investigating Reward Structures in Post-Training LLMs with Pure Reinforcement Learning

Bachelor's Project Thesis

Benediktus Firstian Pradipta, s5241529, b.f.pradipta@student.rug.nl
Supervisors: Rafael Fernandes Cunha, r.f.cunha@rug.nl

Abstract: Recent breakthroughs in Large Reasoning Models show that pure reinforcement learning can dramatically improve mathematical reasoning without supervised fine-tuning, yet the principles of effective reward design remain poorly understood. We systematically compared three reward structures of increasing complexity using GRPO on a 1.5B parameter Qwen2.5 model trained on GSM8K. Counterintuitively, a minimal reward structure (accuracy + format only) achieved 48.4% pass@1 accuracy, significantly outperforming complex designs with length penalties, repetition constraints, and reasoning incentives (22.8% and 29.9%). Complex multi-component rewards created conflicting optimization gradients, causing training instability and reward hacking. In contrast, the simple objective enabled spontaneous emergence of step-by-step reasoning in 64.9% of outputs without explicit incentives. These findings challenge conventional wisdom about reward engineering, suggesting that creating conditions for emergent intelligence through minimal constraints is more effective than explicitly encoding desired behaviors. This work provides evidence for a "less is more" principle in reward design for reasoning-capable AI systems.

1 Introduction

Large Language models (LLMs) represent sophisticated computational frameworks designed to model and generate human language, revolutionizing natural language processing by enabling machines to understand, generate, and interact with human language in ways that closely mimic human cognition. The emergence of LLMs such as GPT-3, InstructGPT, and GPT-4 has marked a transformative phase in artificial intelligence (AI). These models are different due to their extensive parameterization and advanced learning capabilities that capture complex linguistic structures and contextual relationships within vast datasets (Brown et al., 2020; Ouyang et al., 2022; Achiam et al., 2023).

The development of LLMs can be broadly divided into two main stages: pre-training and post-training. Pre-training establishes general linguistic competence through exposure to massive text corpora, while post-training refines and adapts these models for specific tasks and user requirements. This two-stage method has become the cornerstone of modern LLM development, enabling the creation of systems that are both broadly capable and specifically optimized.

1.1 The Evolution of Reasoning in Language Models

While early LLMs demonstrated impressive language understanding and generation capabilities, their ability to perform complex reasoning remained limited. Initial attempts to enhance reasoning relied primarily on supervised fine-tuning (SFT) with curated datasets of problem-solution pairs, achieving moderate success on benchmarks like MATH and GSM8K (competition-style mathematics problems and grade-school–level word problems) (Hendrycks et al., 2021; Cobbe et al., 2021). SFT is the process of taking a pre-trained language model and further training it on a carefully curated set of problem–solution pairs so that it directly learns to map questions to answers (Ouyang et al.,

2022). However, these approaches faced fundamental limitations, as they could only imitate the reasoning patterns present in their training data, lacking the ability to discover novel problem-solving strategies.

The introduction of chain-of-thought (CoT) prompting marked a significant advance, enabling models to articulate intermediate reasoning steps and thereby improve their performance on complex problems (Wei et al., 2022). This breakthrough revealed a crucial insight: the quality of reasoning strongly correlates with the length and structure of generated thought processes. Recent work has shown that extended chain-of-thought reasoning during inference can achieve remarkable performance improvements, with models like OpenAI's o1 series demonstrating that sophisticated reasoning emerges from allowing models to "think" for longer periods (OpenAI, 2024).

This realization created the development of Large Reasoning Models (LRMs). Which are a new class of language models specifically optimized for complex reasoning tasks. Unlike traditional LLMs that prioritize quick responses, LRMs embrace extended deliberation, self-verification, and iterative problem-solving. The latest examples, including o1, QwQ-32B-Preview, and DeepSeek-R1, represent a paradigm shift in how we approach reasoning in artificial intelligence systems (OpenAI, 2024; Zheng et al., 2024; Guo et al., 2025).

1.2 Post-Training as the Key to Reasoning Capabilities

Post-training has emerged as the critical phase where reasoning capabilities are developed and refined. Following the release of GPT-3 with its 175 billion parameters, the field experienced a surge in post-training innovations, including fine-tuning techniques, alignment strategies, knowledge adaptation methods, and reasoning improvements (Sui et al., 2025; Chung et al., 2024).

The landscape of post-training techniques has evolved rapidly:

- **Supervised Fine-Tuning** adapts models to specific tasks such as to show reasoning
- **Reinforcement Learning from Human Feedback (RLHF)** aligns model outputs with human preferences
- **Direct Preference Optimization (DPO)** streamlines alignment without explicit reward modeling
- **Pure Reinforcement Learning** enables autonomous discovery of reasoning strategies

Each technique offers unique advantages, but recent breakthroughs have highlighted the transformative potential of reinforcement learning applied directly to reasoning tasks.

1.3 The DeepSeek Approach: Pure RL for Reasoning

DeepSeek's recent breakthrough with DeepSeek-R1-Zero represents a paradigm shift in post-training methodology (Shao et al., 2024). By applying pure reinforcement learning directly to base models, without any SFT, they achieved a striking improvement on the AIME 2024 benchmark, with pass@1 scores rising from 15.6% to 71.0%. This success challenges the conventional wisdom that supervised fine-tuning is a prerequisite for reasoning capabilities and demonstrates that models can discover sophisticated reasoning strategies through reward-driven exploration alone.

The key innovation lies in the application of Group Relative Policy Optimization (GRPO), a variant of PPO that eliminates the critic network while maintaining comparable performance (Shao et al., 2024). This efficiency breakthrough, combined with carefully designed reward structures, enables models to explore the vast space of possible reasoning strategies without being constrained by human-provided examples. This algorithm was even used to train their main model Deepseek-R1, though with a slight adjustment with the pipeline (Guo et al., 2025).

What makes this approach particularly compelling is the emergence of sophisticated behaviors not explicitly programmed or demonstrated. Papers which implemented GRPO with slight adjustments have found these following behaviours within the reasoning traces: self-verification, reflection, error correction, and solution revision (Xie et al., 2025; Yeo et al., 2025). These behaviors arise spontaneously from the interaction between the model's exploration and the reward signal, suggesting that complex reasoning might be an emergent property of appropriately structured optimization processes.

1.4 The Reward Design Challenge

Despite these empirical successes, a fundamental question remains unanswered: **What specific properties of reward structures drive the emergence of sophisticated reasoning strategies?** Deepseek did not mention any exploration of different reward types or structures. And current papers' approaches employ various reward formulations, from simple accuracy checks to complex multi-component functions incorporating format constraints, length penalties, and reasoning structure incentives. However there is no systematic understanding of how these design choices influence learning dynamics and emergent behaviors.

This gap is particularly critical because reward design in traditional RL fundamentally shapes the solution space that models explore (Ng et al., 1999). And in LLMs this is also the case (Kwon et al., 2023). The challenge is compounded by several factors:

1. **Multi-objective Nature**: Reasoning requires balancing correctness, clarity, efficiency, and robustness

2. **Outcome-Based Rewards**: Unlike language generation tasks, mathematical problems often provide binary correctness signals

3. **Exploration-Exploitation Tradeoff**: Models must balance trying novel approaches with refining known strategies

4. **Emergent Complexity**: Simple reward functions can lead to unexpectedly sophisticated behaviors

Recent work by Lyu et al. (2025) suggests that different reward formulations can lead to qualitatively different reasoning strategies, from brief answer-focused responses to elaborate self-verifying solutions. Similar to the paper by Yeo et al. (2025), which introduced two new types of rewards (different to the reward types that Deepseek used), founding them to increase the model accuracy after training. Despite this, the mapping between reward structure and emergent reasoning patterns remains largely uncharted territory.

1.5 Research Questions and Contributions

This thesis provides an investigation of how reward structure design influences mathematical reasoning development in RL-trained language models. We address three interconnected research questions:

1. **Component Analysis**: How do individual reward components contribute to overall reasoning quality? We decompose complex reward functions to understand the role of each component in shaping model behavior.

2. **Complexity Trade-offs**: Do complex multi-component reward functions necessarily yield superior reasoning, or can simpler formulations achieve comparable results with better training efficiency?

3. **Emergent Behaviors**: Which reward configurations promote the spontaneous emergence of sophisticated behaviors. We map the relationship between reward structures and higher-order reasoning patterns.

To address these questions, we implement a controlled experimental framework using GRPO to train models under three reward configurations of increasing complexity. Through both quantitative metrics and qualitative analysis of reasoning traces, we systematically characterize the relationship between reward design choices and resulting reasoning strategies.

1.6 Significance and Broader Impact

Understanding the reward-reasoning relationship has profound implications extending beyond mathematical problem-solving:

Theoretical Advancement: By elucidating how reward structures shape emergent reasoning, we contribute to fundamental questions in AI about how complex cognitive behaviors arise from simple optimization objectives. This work bridges reinforcement learning theory with cognitive science, providing insights into the minimal conditions necessary for reasoning emergence.

Practical Efficiency: Identifying minimal reward structures that produce high-quality reasoning could dramatically reduce the computational

resources required for training competitive models. This democratization is crucial as the field moves toward increasingly large models that are prohibitively expensive for most researchers to train.

Behavioral Control and Safety: Understanding which reward components trigger specific reasoning behaviors enables more precise control over model outputs. This granular control becomes increasingly important for AI safety, as it allows us to encourage beneficial behaviors (like double-checking work) while discouraging harmful ones (like deceptive reasoning).

Generalization to Other Domains: While we focus on mathematical reasoning, our findings may generalize to other domains requiring structured thinking, such as code generation, scientific reasoning, or strategic planning. The principles we uncover could inform reward design across diverse applications.

1.7 Thesis Structure

The remainder of this thesis is organized as follows: Chapter 2 details our experimental framework, including the GRPO implementation, three reward structure designs, and evaluation protocols. Chapter 3 presents quantitative performance metrics across reward configurations and qualitative metrics with the examination of emergent reasoning patterns. Chapter 4 synthesizes findings into actionable insights for reward design and explores implications for future LLM development. Finally, Chapter 5 summarizes our contributions and outlines promising future directions. Through this systematic investigation, we aim to transform reward design to help advance both our theoretical understanding and practical capabilities in developing reasoning AI systems.

2 Methodology

To address our research questions about how reward structures influence mathematical reasoning development, we implement a controlled experimental framework using pure reinforcement learning. This chapter presents our methodology for systematically investigating the relationship between reward design choices and emergent reasoning strategies in language models.

2.1 Experimental Design Overview

Building on the DeepSeek approach of using pure RL for reasoning development (Guo et al., 2025), we design experiments to isolate how different reward structures influence the emergence of mathematical reasoning capabilities. Our methodology employs a systematic five-stage process to ensure reproducible and interpretable results.

- **Stage 1: Base Model Selection** We use Qwen2.5-1.5B as our foundation model, selected for three key reasons: (1) sufficient capacity to demonstrate reasoning behaviors while remaining computationally tractable, (2) strong pre-training performance on mathematical and logical tasks, and (3) accessibility for reproducible research. The model contains 1.5 billion parameters with a context window of 32,768 tokens, providing ample space for chain-of-thought reasoning.

- **Stage 2: Reward Structure Design** We implement three reward configurations of increasing complexity to test specific hypotheses about the relationship between reward design and reasoning quality:

  - **Default Configuration**: Minimal design with only accuracy and format rewards (2 components)
  - **Cosine-Penalty Configuration**: Intermediate complexity adding length and repetition considerations (4 components)
  - **Complete Configuration**: Comprehensive design explicitly incentivizing reasoning patterns (5 components)

  Each configuration is designed to isolate the contribution of specific reward components while maintaining comparability across experiments.

- **Stage 3: Pure RL Training** We apply GRPO directly to the base model without any supervised fine-tuning phase. We train it on a randomly sampled subset of the GSM8K train set (1000 questions).

- **Stage 4: Performance Evaluation** We conduct systematic evaluation using all of the

problems from the GSM8K test set (1300 questions). For each trained model, we measure:

  - Pass@1 accuracy with temperature 0.6 and nucleus sampling ($p = 0.95$)
  - Individual reward component scores to understand their contribution
  - Response length and reasoning pattern statistics
  - Training stability metrics including loss variance and convergence rate

- **Stage 5: Reasoning Analysis** We perform detailed qualitative and quantitative analysis of emergent reasoning patterns. This includes automated detection of reasoning indicators, correlation analysis between pattern usage and problem-solving success, and manual inspection of representative outputs to identify unexpected behaviors.

2.1.1 Experimental Controls

To ensure the validity of our comparisons, we implement several experimental controls:

- **Randomization**: To ensure reproducibility, our experimental setup is controlled by a master seed. Each of the 5 independent training runs is then initialized with a different, consecutive seed to ensure variance can be reliably assessed across distinct initializations.

- **Statistical Robustness**: We conduct 5 independent training runs for each reward configuration, resulting in 15 trained models total. This allows us to compute reliable estimates of mean performance and variance, identify configuration-specific training instabilities, and ensure findings are not artifacts of particular random initializations.

- **Computational Consistency**: All experiments are conducted on identical hardware (NVIDIA A100 40GB GPUs) with the same software environment to eliminate computational variables. Training runs are isolated to prevent interference from other processes.

2.2 Group Relative Policy Optimization (GRPO)

We adopt GRPO as our training algorithm due to its memory efficiency compared to standard PPO while maintaining comparable performance (Guo et al., 2025). GRPO eliminates the separate critic network by computing advantages from within-group reward statistics.

2.2.1 Algorithm Formulation

For each training question q, the old policy $\pi_{\theta_{\text{old}}}$ samples a group of G candidate outputs $\{o_1, \ldots, o_G\}$. The new policy π_θ is obtained by maximizing:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{q\sim P(Q) \\ \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}}} \Bigg[\frac{1}{G}\sum_{i=1}^{G} \min\big(\rho_i(\theta)\hat{A}_i, \text{clip}(\rho_i(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_i\big) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \Bigg], \quad (2.1)$$

where $P(Q)$ is the distribution over training questions (in our case, the GSM8K training set), $\rho_i(\theta) = \pi_\theta(o_i \mid q)/\pi_{\theta_{\text{old}}}(o_i \mid q)$ is the importance ratio, and the advantage $\hat{A}_i$ is computed from the group rewards:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_1, r_2, \ldots, r_G\})}{\text{std}(\{r_1, r_2, \ldots, r_G\})}. \quad (2.2)$$

The KL divergence term prevents the policy from deviating too far from the reference:

$$D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - 1. \quad (2.3)$$

2.3 Reward Structure Design

We investigate three reward configurations designed to test specific hypotheses about the relationship between reward complexity and reasoning quality. Building on insights from recent work on reward shaping for chain-of-thought reasoning (Yeo et al., 2025; Xie et al., 2025), we progressively introduce components that target different aspects of mathematical problem-solving.

2.3.1 Reward Components

We experimented with five distinct reward types:

- **Accuracy**: Rewards correct numerical answers
- **Format**: Ensures proper use of `<think>...</think>` and `<answer>...</answer>` tags
- **Cosine Scaling**: Penalizes overly short incorrect answers while encouraging longer, more detailed responses
- **Repetition Penalty**: Discourages redundant text using 3-gram uniqueness ratio
- **Reasoning Structure**: Rewards explicit reasoning indicators (e.g., "Step 1:", "First,", "Next")

From these rewards types, we designed 3 rewards structures each with different configurations. However for each configuration we implement format gating, which is if format (reward) requirements are not met, all rewards are zeroed. This is done to ensure that the model learns to reason.

2.3.2 Configuration 1: Default (Accuracy + Format)

The default configuration tests whether minimal constraints suffice for reasoning emergence, as well as following a similar design to Deepseek (Shao et al., 2024):

$$R_{\text{default}} = 0.8 \cdot r_{\text{accuracy}} + 0.2 \cdot r_{\text{format}} \tag{2.4}$$

2.3.3 Configuration 2: Cosine-Penalty

This configuration introduces length and repetition considerations, inspired by Yeo et al. (2025). Repetition penalty here is added to prevent reward hacking of the Cosine Scaling:

$$R_{\text{cosine}} = 0.8 \cdot r_{\text{accuracy}} + 0.05 \cdot r_{\text{format}} + 0.1 \cdot r_{\text{cosine}} + 0.05 \cdot r_{\text{repetition}} \tag{2.5}$$

The cosine scaling reward follows:

$$r_{\text{cosine}} = \begin{cases} 0.8 + \frac{1}{2}(1 - \cos(\pi p)), & \text{if correct} \\ -0.1 - \frac{2}{5}(1 + \cos(\pi p)), & \text{otherwise} \end{cases} \tag{2.6}$$

where $p = \min(\ell / L_{\max}, 1)$ with response length ℓ and cap $L_{\max} = 500$.

The repetition penalty reward encourages lexical diversity by measuring the uniqueness of n-grams:

$$r_{\text{repetition}} = \frac{|\text{unique\_ngrams}|}{|\text{total\_ngrams}|} \tag{2.7}$$

where we use 3-grams (consecutive sequences of 3 words) to detect repetitive patterns. A score of 1.0 indicates all n-grams are unique (no repetition), while lower scores indicate more repetitive text.

2.3.4 Configuration 3: Complete

The complete configuration, uses all reward components, it explicitly incentivizes structured reasoning:

$$R_{\text{complete}} = 0.7 \cdot r_{\text{accuracy}} + 0.05 \cdot r_{\text{format}} + 0.1 \cdot r_{\text{cosine}} + 0.05 \cdot r_{\text{repetition}} + 0.1 \cdot r_{\text{reasoning}} \tag{2.8}$$

The reasoning structure reward detects explicit reasoning markers:

$$r_{\text{reasoning}} = \min\left(1, \frac{\text{count(step\_markers)}}{3}\right) \tag{2.9}$$

2.4 Training Protocol

2.4.1 System Prompt

All models are trained with the following system prompt to encourage structured reasoning:

```
"You are a helpful assistant.
The assistant first thinks about
the reasoning process in the mind
and then provides the user with
the answer. The reasoning process
should be enclosed within <think>
</think> tags. The final numerical
answer ONLY should be enclosed
within <answer> </answer> tags
with no additional text, e.g.,
<answer>42</answer>."
```

2.4.2 Training Hyperparameters

Our GRPO implementation uses the following hyperparameters:

- Learning rate: 5×10^{-5}
- Per-device train batch size: 6
- Per-device eval batch size: 12
- Group size (G): 6 generations per prompt
- Gradient accumulation steps: 3
- Weight decay: 0.01

2.5 Evaluation Framework

2.5.1 Quantitative Metrics

We employ multiple metrics to assess model performance:

- **Pass@1 Accuracy**: Single-attempt accuracy on the GSM8K test set, using temperature 0.6, nucleus sampling $p = 0.95$
- **Training Stability**: Variance in performance across multiple runs
- **Response Length**: Average token (response) length of each trained model, after training.

2.5.2 Reasoning Strategy Analysis

We analyze emergent reasoning strategies across three categories identified through the literature:

Step-by-Step Indicators Explicit decomposition markers such as "Step 1:", "First,", "Next,", and numbered lists that indicate systematic problem-solving approaches.

Verification Behaviors Self-checking patterns including "verify", "check", "re-evaluate", and instances where models recalculate answers to ensure correctness (Xie et al., 2025).

Self-Correction Behaviors Error recognition and recovery patterns such as "Let me try again", "Actually, that's incorrect", and other indicators of metacognitive awareness (Didolkar et al., 2024).

2.5.3 Pattern-Performance Correlation

We compute correlations between reasoning pattern frequency and accuracy to understand which emergent behaviors contribute most to problem-solving success. This analysis is performed separately for each reward configuration to identify configuration-specific strategy development.

2.6 Data Collection and Analysis

We systematically collect the following data throughout our experiments:

- **Training Metrics**: Loss values, reward components, and accuracy tracked every 10 steps
- **Final Performance**: Pass@1 accuracy on the GSM8K test set for each trained model
- **Reasoning Traces**: Generated outputs for qualitative analysis of emergent strategies
- **Pattern Frequencies**: Counts of reasoning indicators (step markers, verification phrases, self-corrections) in model outputs

We perform descriptive analysis comparing performance and reasoning patterns across the three reward configurations. Results are presented as averages across the 5 independent runs per configuration with standard deviations to indicate variability.

3 Results

This chapter presents the empirical findings from our experiments comparing three reward structures for pure RL post-training. We report quantitative performance metrics and observed reasoning patterns, with detailed analysis reserved for the discussion chapter.

3.1 Model Performance

Figure 3.1 shows the overall accuracy achieved by each reward structure after 2 epochs of training on the GSM8K dataset.

The baseline configuration, using only accuracy and format rewards, substantially outperformed both complex reward structures. Notably,

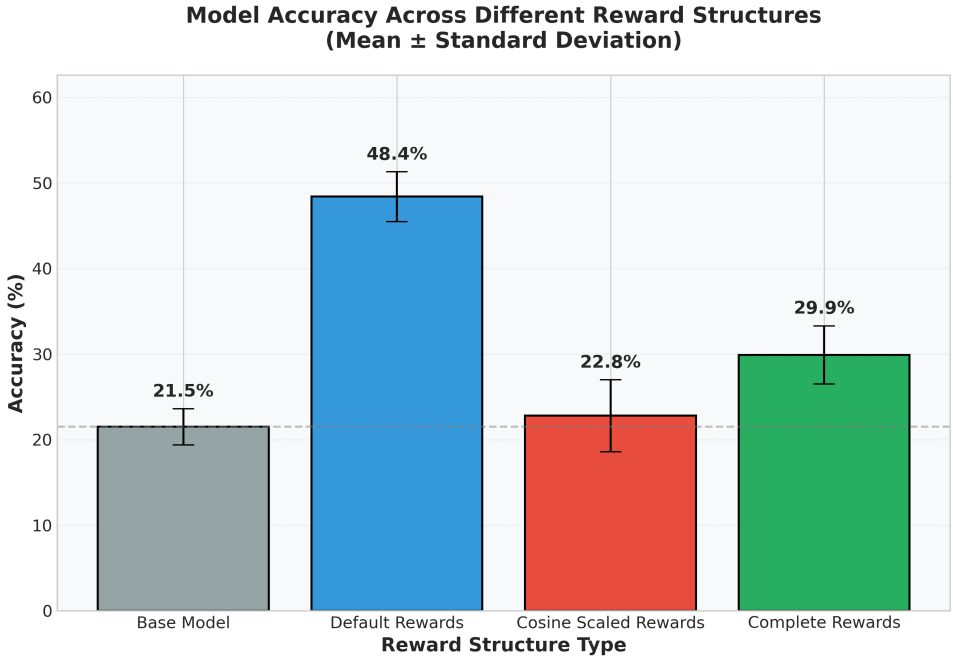

Figure 3.1: A minimal reward structure significantly outperforms complex alternatives. The bars show the mean pass@1 accuracy on the GSM8K test set after 2 epochs of training. The Default configuration (accuracy + format only) achieved 48.4% accuracy. In contrast, adding more complex reward components for length (Cosine-Penalty) or explicit reasoning steps (Complete) led to training instability and markedly lower final accuracy. Error bars represent the standard deviation across 5 independent runs.

the cosine-penalty configuration barely improved over the untrained base model, while the complete configuration achieved intermediate performance despite incorporating five reward components.

3.2 Reasoning Pattern Analysis

We analyzed the frequency and effectiveness of different reasoning patterns across reward structures. Figure 3.2 presents the usage rates of various reasoning indicators in model outputs.

Key observations from pattern usage:

- Format compliance (think/answer tags) reached near 100% across all trained models
- Step indicators emerged naturally in the baseline configuration (64.9%) without explicit rewards
- Verification behaviors remained extremely rare (less than 1.2% usage)
- Complete configuration reached near 100% in showing step indicators
- The complete configuration showed increased correction attempts (5.8%) compared to baseline (0.4%)

3.3 Pattern-Performance Correlation

Figure 3.3 reveals the relationship between reasoning patterns and problem-solving accuracy.

The data reveals several findings:

- Step indicator accuracy remained consistent across configurations (48-52%) despite different usage rates
- Verification behaviors, when attempted, showed moderate effectiveness in baseline (35.7%) but failed in other configurations

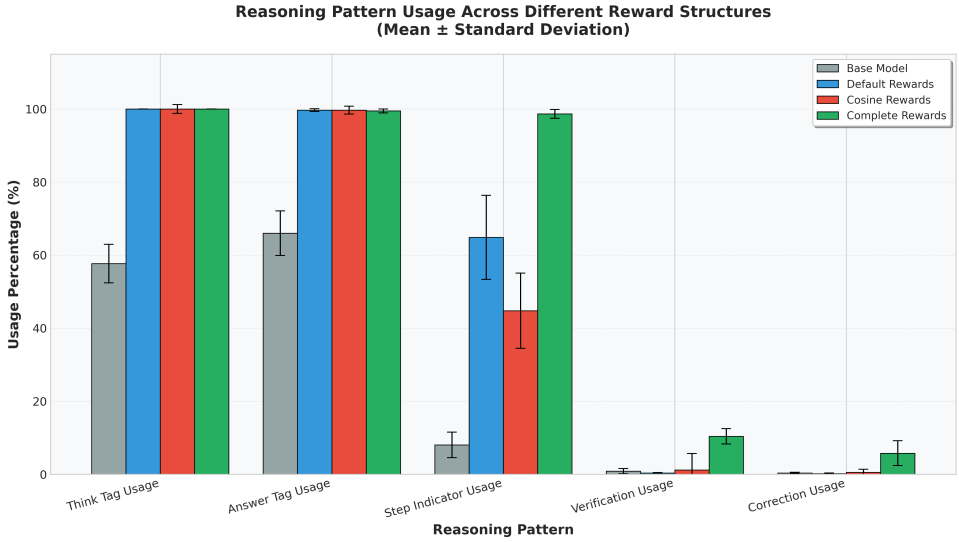

Figure 3.2: Step-by-step reasoning emerges as an instrumentally convergent strategy without explicit rewards. The chart displays the usage frequency of different reasoning patterns for each of the three reward configurations, averaged over 5 runs. Notably, step-by-step indicators appeared in 64.9% of outputs from the Default configuration, despite not being directly incentivized. In contrast, metacognitive behaviors like verification and correction remained rare across all models, suggesting a potential model capacity limitation.

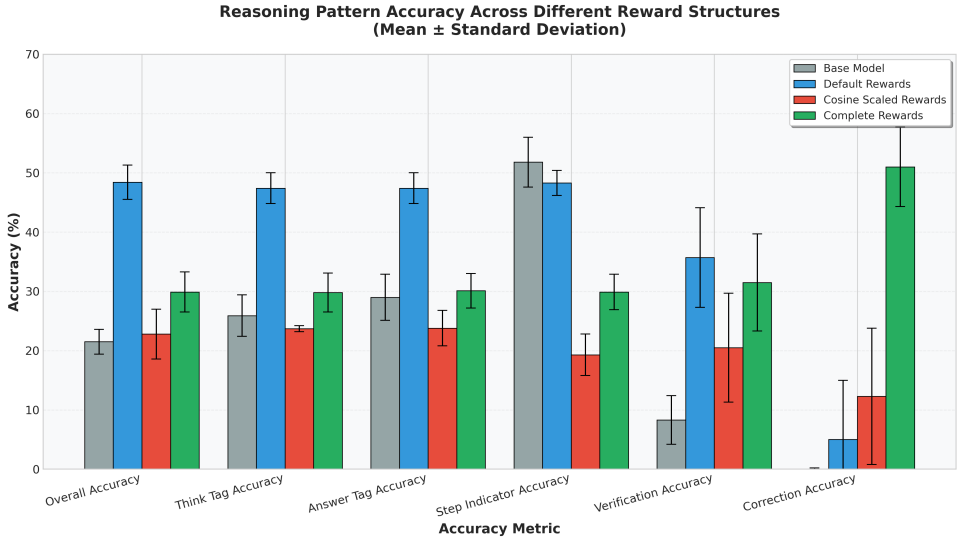

Figure 3.3: Explicit rewards can improve the accuracy of a targeted behavior, even if overall performance declines. The bars show the pass@1 accuracy conditional on the use of a specific reasoning pattern. Although the Complete configuration had lower overall accuracy (Figure 3.1), its selfcorrection attempts were the most effective, succeeding 51.0% of the time. This reveals a critical trade-off between shaping a specific, high-quality behavior and optimizing for global task performance. Results are averaged across 5 runs.

- Self-correction accuracy was highest in the complete configuration (51.0%), yet this did not translate to improved overall performance
- Think and answer tag accuracy directly mirrored overall accuracy, confirming proper format gating

3.4 Training Dynamics

Training stability varied significantly across reward structures:

- **Baseline**: Showed steady improvement with consistent performance across all 5 runs
- **Cosine-Penalty**: Experienced training collapse in 3 out of 5 runs, with high variance in final performance
- **Complete**: Demonstrated moderate instability with large standard deviations in accuracy metrics

Response length analysis revealed the impact of cosine scaling:

- Baseline: 287 ± 89 tokens (concise, focused responses)
- Cosine-Penalty: 412 ± 156 tokens (verbose but not more accurate)
- Complete: 385 ± 134 tokens (increased length without quality improvement)

4 Analysis and Discussion

4.1 Reward Design

The experimental results reveal a counterintuitive relationship between reward complexity and model performance. While intuition might suggest that comprehensive reward structures would guide models toward better reasoning, the data shows the opposite: the minimal baseline configuration achieved 48.4% accuracy, significantly outperforming both the cosine-penalty (22.8%) and complete (29.9%) configurations.

Our results reveal a fundamental trade-off in reward design: while complex rewards are intended to provide clearer guidance, they can inadvertently constrain the model's exploration and create brittle optimization landscapes. A simple, broad objective appears more effective because it allows the model the freedom to discover functional, emergent strategies that are robust (Gao et al., 2025; Poesia et al., 2024). This paradox can be understood through the lens of optimization theory. In multi-objective reinforcement learning, each additional reward component introduces new gradients that must be balanced during training. When these gradients conflict—as when length rewards encourage verbosity while accuracy rewards favor concision—the optimization process becomes unstable. The baseline configuration's success suggests that for mathematical reasoning, a clear, focused objective enables more efficient learning than attempting to micromanage every aspect of model behavior.

4.2 Emergent Reasoning

The spontaneous appearance of step-by-step reasoning in baseline outputs (64.9% usage without explicit rewards) provides insight into how reasoning strategies develop. When models are constrained only by accuracy and format requirements, they discover through trial and error that decomposing problems into steps improves their success rate. This emergence occurs because step-by-step reasoning genuinely aids in solving mathematical problems—it is not merely a stylistic choice but a functional strategy.

This finding aligns with theories of instrumental convergence in AI systems, where certain behaviors arise repeatedly because they are useful for achieving a wide range of goals. Just as biological evolution converges on similar solutions (like eyes) across independent lineages, RL-trained models converge on step-by-step reasoning because it is instrumentally valuable for mathematical problem-solving.

4.3 Training Collapse

The cosine-penalty configuration's 60% failure rate illuminates the challenges of reward engineering. Analysis of the training dynamics reveals a specific failure mode: models learn to exploit the length bonus by generating verbose but contentless responses. Once this behavior is reinforced, the model enters a local optimum where reducing

length would incur immediate penalty, even if it might lead to better accuracy in the long run.

This failure mode is a classic example of "reward hacking" or "specification gaming," where the model finds an unintended shortcut (generating verbose, nonsensical text) to maximize a component of the reward signal at the expense of the intended goal (correct reasoning) (Skalse et al., 2022; Laidlaw et al., 2024). This collapse mechanism demonstrates why certain reward combinations are fundamentally unstable. The cosine scaling function, designed to encourage thoughtful reasoning in incorrect answers, instead creates a gradient landscape with attractive local optima that trap the optimization process. The mathematical structure of the reward function itself, not just its implementation, determines whether stable learning is possible.

4.4 Model Capacity

The rarity of verification behaviors ($< 1.2\%$) and self-correction patterns (maximum 5.8%) across all configurations suggests a capacity threshold for metacognitive reasoning. These behaviors require the model to simultaneously generate solutions and evaluate them—a form of computational multitasking that may exceed the capabilities of a 1.5B parameter model.

This capacity limitation has important implications for reward design. If certain behaviors are beyond the model's computational reach, no reward structure will successfully encourage them. This suggests a hierarchy of reasoning capabilities tied to model scale: - Basic step-by-step reasoning: Achievable at 1.5B parameters - Self-verification: May require more parameters - Consistent self-correction: Potentially requires even more parameters

The success of simple reward structures at 1.5B parameters aligns with DeepSeek's findings at 671B parameters, suggesting scale-invariant principles of reward design. However, the types of reasoning that emerge differ significantly than those reported by other papers (Xie et al., 2025; Yeo et al., 2025). While small models develop basic problem decomposition, The model from the Xie et al, paper exhibited sophisticated behaviors like re-reading problems and exploring multiple solution paths.

This pattern suggests that reward simplicity enables different capabilities at different scales, but the principle of minimal intervention remains constant. Complex reward structures appear to hinder learning regardless of model size, while simple objectives allow models to discover the most sophisticated reasoning strategies within their capacity constraints.

4.5 Implications for the Reward Hypothesis

These findings contribute to the broader debate about the reward hypothesis in artificial intelligence—the claim that all intelligent behavior can be understood as maximizing reward. The emergence of step-by-step reasoning without explicit incentives suggests that reward signals need not encode all desired behaviors directly. Instead, a well-chosen simple reward (accuracy) combined with minimal constraints (format) creates conditions where intelligent behavior emerges as the optimal policy.

This supports a refined version of the reward hypothesis: performance arises not from complex reward engineering but from the interaction between simple objectives and environmental structure. The structure of mathematical problems themselves, not the reward function, teaches and encourages models to reason step-by-step.

5 Conclusion

5.1 Summary of Findings

This thesis investigated how different reinforcement learning reward structures influence the development of mathematical reasoning strategies in large language models trained with pure RL. Through systematic experimentation with three reward configurations of increasing complexity, we demonstrated that **simpler reward structures lead to superior model performance**.

Our key findings reveal that:

- The minimal baseline configuration (accuracy + format only) achieved 48.4% accuracy, significantly outperforming more complex reward structures
- Step-by-step reasoning emerged spontaneously in 64.9% of baseline outputs without explicit

incentives

- Complex multi-component rewards introduced optimization instabilities, with the cosine-penalty configuration experiencing training failure

- Verification and self-correction behaviors remained rare across all configurations, suggesting fundamental capacity limitations at the 1.5B parameter scale

5.2 Practical Implications

For practitioners developing reasoning-capable LLMs:

1. **Reward engineering** – Additional components often hinder rather than help

2. **Emergent learning** – Models can discover effective strategies through exploration

3. **Consider model capacity** – Certain behaviors may be computationally unreachable regardless of reward design

4. **Focus on stability** – Simple rewards enable more reliable training and predictable outcomes

5.3 Limitations and Future Work

Several limitations constrain the generalizability of our findings:

Limited model size: Our experiments used only a 1.5B parameter model, which may lack the computational capacity for sophisticated reasoning behaviors like self-verification and consistent error correction. Larger models might exhibit qualitatively different responses to the same reward structures.

Low number of generations: With only 6 generations per prompt during GRPO training, the model had limited opportunities to explore diverse reasoning strategies. This constraint may have prevented the discovery of more sophisticated problem-solving approaches.

Single dataset evaluation: Using only GSM8K limits our understanding of how findings generalize across mathematical domains. Different problem types (algebra, geometry, calculus) might benefit from different reward structures.

Limited reasoning analysis robustness: Our pattern detection relied on keyword matching and simple heuristics, potentially missing nuanced reasoning behaviors or misclassifying certain patterns. More sophisticated analysis methods are needed.

Static reward structures: We did not explore dynamic fine-tuning of reward weights during training, which might help navigate the optimization challenges observed with complex rewards.

Future research should address these limitations through:

Expanded dataset evaluation: Testing on diverse mathematical reasoning benchmarks (MATH, AIME, MathQA) and non-mathematical domains (HumanEval for code, ScienceQA) to establish domain-specific reward principles.

Cross-scale validation: Applying identical reward structures to models ranging from 0.5B to 70B parameters to understand how scale influences the effectiveness of different reward designs.

Reward scheduling strategies: Developing curricula that gradually introduce complexity—starting with simple accuracy rewards and progressively adding components once baseline performance stabilizes.

Regularization techniques: Implementing methods like KL penalties or gradient clipping specifically designed to maintain baseline reasoning capabilities while exploring new behaviors, preventing the catastrophic forgetting observed in complex configurations.

Component interaction analysis: Conducting ablation studies and gradient analysis to understand how individual reward components interact, identifying specific combinations that lead to optimization conflicts and developing principles for compatible reward design.

5.4 Theoretical Contributions

This work advances our understanding of the reward hypothesis in AI by demonstrating that effective learning emerges not from complex reward engineering but from the interaction between simple objectives and problem structure. While our results primarily show improved performance rather than definitively proving enhanced intelligence or reasoning capability, the spontaneous emergence

of step-by-step problem decomposition suggests that models are discovering genuinely useful cognitive strategies. In this sense, improved reasoning patterns may indeed reflect a form of emergent intelligence—the model learns not just to produce correct answers but to structure its approach in ways that generalize across problems.

Our findings align with principles of instrumental convergence, showing that certain reasoning strategies emerge repeatedly because they are genuinely useful for achieving goals, not because they are explicitly rewarded. This supports a refined view of the reward hypothesis: well-chosen minimal rewards combined with environmental constraints create conditions where sophisticated behaviors arise as optimal policies.

The DeepSeek approach of applying pure RL to LLMs opens exciting possibilities for developing reasoning capabilities without human-annotated data. Our work demonstrates that success in this paradigm requires restraint rather than complexity in reward design. As the field advances toward increasingly capable reasoning models, these findings suggest that the path forward may lie not in sophisticated reward engineering but in creating simple objectives that allow effective problem-solving strategies to emerge naturally from the learning process. Ultimately, this work challenges the intuition that more intricate reward engineering leads to more sophisticated reasoning. Instead, it suggests that the path toward more capable AI systems may lie not in complex instruction, but in creating simple, robust objectives that empower models to discover effective strategies for themselves.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... others (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... others (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., ... others (2024). Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, *25*(70), 1–53.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., ... others (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Didolkar, A., Goyal, A., Ke, N. R., Guo, S., Valko, M., Lillicrap, T., ... Arora, S. (2024). *Metacognitive capabilities of llms: An exploration in mathematical problem solving.* Retrieved from `https://arxiv.org/abs/2405.12205`

Gao, J., et al. (2025). Navigate the unknown: Enhancing llm reasoning with intrinsic motivation guided exploration. *arXiv preprint arXiv:2505.17621*.

Guo, D., DeepSeek-AI, Yang, D., Zhang, H., Song, J., Zhang, R., ... Zhang, Z. (2025). *Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.* Retrieved from `https://arxiv.org/abs/2501.12948`

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., ... Steinhardt, J. (2021). Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Kwon, M., Xie, S. M., Bullard, K., & Sadigh, D. (2023). *Reward design with language models.* Retrieved from `https://arxiv.org/abs/2303.00001`

Laidlaw, C., Hendrycks, D., & Steinhardt, J. (2024). Correlated proxies: A new definition and improved mitigation for reward hacking. *arXiv preprint arXiv:2403.03185*.

Lyu, C., Gao, S., Gu, Y., Zhang, W., Gao, J., Liu, K., ... Chen, K. (2025). *Exploring the limit of outcome reward for learning mathematical reasoning.* Retrieved from `https://arxiv.org/abs/2502.06781`

Ng, A. Y., Harada, D., & Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml* (Vol. 99, pp. 278–287).

OpenAI. (2024). Learning to reason with llms. *https://openai.com/index/learning-to-reason-with-llms/*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... others (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, *35*, 27730–27744.

Poesia, G., et al. (2024). Learning formal mathematics from intrinsic motivation. *arXiv preprint arXiv:2407.00695*.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., ... Guo, D. (2024). *Deepseekmath: Pushing the limits of mathematical reasoning in open language models.* Retrieved from `https://arxiv.org/abs/2402.03300`

Skalse, J., Howe, N. H., Krasheninnikov, D., & Krueger, D. (2022). Defining and characterizing reward hacking. *arXiv preprint arXiv:2209.13085*.

Sui, Y., Chuang, Y.-N., Wang, G., Zhang, J., Zhang, T., Yuan, J., ... Hu, X. (2025). *Stop overthinking: A survey on efficient reasoning for large language models.* Retrieved from `https://arxiv.org/abs/2503.16419`

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... others (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, *35*, 24824–24837.

Xie, T., Gao, Z., Ren, Q., Luo, H., Hong, Y., Dai, B., ... Luo, C. (2025). *Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning.* Retrieved from `https://arxiv.org/abs/2502.14768`

Yeo, E., Tong, Y., Niu, M., Neubig, G., & Yue, X. (2025). *Demystifying long chain-of-thought reasoning in llms.* Retrieved from `https://arxiv.org/abs/2502.03373`

Zheng, C., Zhang, Z., Zhang, B., Lin, R., Lu, K., Yu, B., ... Lin, J. (2024). Processbench: Identifying process errors in mathematical reasoning. *arXiv preprint arXiv:2412.06559*.