A Reinforcement Learning Pipeline for Financial Reasoning: Comparing Deep RL vs. Heuristic Bandits

Jinyoon Kim[1] **Scarlett Yu**[1] **Donggen Li**[1]
[1]Department of Computer Science, University of Virginia
{xna8aw, bce9ka, eaz7wk}@virginia.edu

Abstract

This project conducts a comprehensive ablation study on applying Reinforcement Learning (RL) to the Financial Question Answering (FinQA) task. To isolate the effects of model complexity and learning paradigms, we separate our research into two parallel tracks. **Track 1** (Deep RL) investigates the efficiency and stability of PPO and GRPO on Large Language Models (Llama-3.2-3B) for discriminative ranking tasks. **Track 2** (Heuristic Bandits) explores exploration-exploitation trade-offs using non-contextual multi-armed bandits. Additionally, we present an **Extension Study** applying RLOO and DPO to weaker models (TinyLlama-1.1B) to test the hypothesis that RL benefits are inversely proportional to base model capability. Our findings indicate that for capable small models (3B), Supervised Fine-Tuning (SFT) is more efficient and stable than RL, whereas RL methods provide significant gains for weaker learners (1B).

1 Introduction

Financial Question Answering (FinQA) requires complex multi-step reasoning, retrieving evidence from unstructured text, and performing numerical calculations. While Large Language Models (LLMs) have shown promise in this domain, ensuring their reliability and efficiency remains a challenge.

This project aims to answer three key research questions:

1. **Efficiency vs. Stability:** How do different RL algorithms (PPO vs. GRPO) compare in terms of computational efficiency and training stability?
2. **The "RL Gap":** Does Reinforcement Learning provide meaningful gains over strong Supervised Fine-Tuning baselines for discriminative reasoning tasks?
3. **Model Capability:** Does the impact of RL training vary significantly between "strong" small models (3B) and "weak" small models (1B)?

To answer these, we developed a unified training framework that supports diverse algorithms (SFT, PPO, GRPO, RLOO, DPO) and evaluated them on the FinQA dataset.

2 Related Work

Financial Reasoning & Datasets. The domain of financial NLP has evolved from simple sentiment analysis to complex reasoning tasks that require hybrid processing of text and structured tables. The **FinQA** dataset [1] serves as a primary benchmark for this task, requiring models to retrieve evidence and perform arithmetic calculations. While large proprietary models perform well on such tasks, smaller open-source models often struggle with the strict syntactic requirements of generating executable reasoning programs.

Reinforcement Learning for Reasoning. Reinforcement Learning from Human Feedback (RLHF) has become the standard for aligning LLMs. **PPO** [4] is the dominant algorithm but imposes significant memory overhead due to the need for a separate value network and a frozen reference model. To address this, recent works have proposed more efficient alternatives. **DPO** [**?**] optimizes preferences directly without a reward model, while **GRPO** (Group Relative Policy Optimization), introduced in DeepSeekMath [5], eliminates the critic by using group-based outcome averaging as a baseline. Similarly, **RLOO** (REINFORCE Leave-One-Out) [**?**] reduces variance in REINFORCE-style estimators, making it particularly suitable for reasoning tasks with sparse rewards.

Discriminative Reranking vs. Generation. A key challenge in applying RL to reasoning is the difficulty of exploration in a vast generative action space. **Cobbe et al.** [2] demonstrated that training a "verifier" (or ranker) to judge the correctness of candidate solutions is often more sample-efficient than training a generator to produce the correct solution from scratch. Our Track 1 methodology aligns with this paradigm: by reformulating the reasoning task as a discriminative ranking problem among K candidates, we bypass the instruction-following difficulties inherent in small generative models.

Efficient Fine-Tuning. To enable these experiments on consumer-grade hardware, we rely on Low-Rank Adaptation (**LoRA**) [3], which freezes the pre-trained model weights and injects trainable rank decomposition matrices, significantly reducing the memory footprint of RL training.

3 Methodology

We propose a multi-track approach to isolate the effects of contextual understanding (Deep RL) versus simple statistical learning (Bandits).

3.1 Task Definition and Data

All experiments utilize the **FinQA** dataset. The raw data consists of financial reports containing text segments and structured tables. We preprocess these into `(Context, Question, Answer)` triples.

Evaluation Metrics: We employ a unified evaluation suite measuring:

- **Exact Match (EM):** Whether the prediction exactly matches the gold answer.
- **Numeric Accuracy:** Whether the predicted value is within a tolerance (5–10%) of the ground truth.
- **Logical Validity:** Whether the reasoning program is syntactically correct.

3.2 Track 1 Approach: Discriminative Reranking

Authored by Jinyoon Kim

Our initial approach attempted Generative RL (training models to write JSON programs), which resulted in a 12% parse rate failure. To overcome this, we pivoted to a **Discriminative (Ranking)** formulation. Instead of modeling $P(w_t|w_{<t}, c)$, we model the probability of selecting the best candidate answer y from a set $Y = \{y_1, ..., y_K\}$:

$$P(y_k|x, Y) = \text{softmax}(f_\theta(x, y_k)) \tag{1}$$

This shift reduces the cognitive burden on the model from *construction* to *verification*.

Action Space ($K = 8$). For every question, we generate a discrete pool of 8 candidates:

- **Gold (1x):** The ground truth answer.
- **Similar (Nx):** Plausible distractors (e.g., values perturbed by $\pm 10\%$).
- **Corrupted (Mx):** Obvious errors (e.g., wrong operations).

Dense Reward Function. We designed a composite scalar reward signal:

$$R_{total} = 1.0 \cdot \mathbb{I}_{exact} + 0.9 \cdot \mathbb{I}_{numeric} + 0.5 \cdot \mathbb{I}_{logic} + 0.3 \cdot \mathbb{I}_{format} \tag{2}$$

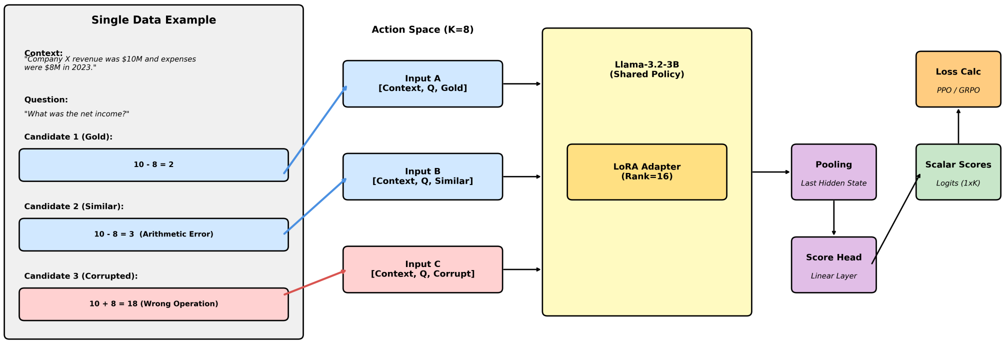
Single Data Example
Context:
"Company X revenue was $10M and expenses were $8M in 2023."
Question:
"What was the net income?"
Candidate 1 (Gold):
10 - 8 = 2
Candidate 2 (Similar):
10 - 8 = 3 (Arithmetic Error)
Candidate 3 (Corrupted):
10 + 8 = 18 (Wrong Operation)
Action Space (K=8)
Input A [Context, Q, Gold]
Input B [Context, Q, Similar]
Input C [Context, Q, Corrupt]
Llama-3.2-3B (Shared Policy)
LoRA Adapter (Rank=16)
Pooling
Last Hidden State
Score Head
Linear Layer
Scalar Scores
Logits (1xK)
Loss Calc
PPO / GRPO

Figure 1: **Track 1 System Architecture.** Left: A concrete example of the action space, showing Gold, Similar, and Corrupted candidates. Right: The data flow where the Llama-3 model acts as a shared policy to output scalar scores for each candidate, which are optimized via PPO or GRPO.

3.3 Track 2 Approach: Heuristic Bandits

Authored by Scarlett Yu

In this track, we study a lightweight approach using non-contextual multi-armed bandits. We have a finite set of arms $\mathcal{A} = \{0, 1, 2\}$ where each arm corresponds to a fixed heuristic:

- **Arm 0 (Token Overlap):** Selects the cell with the largest token-set intersection.
- **Arm 1 (Max Value):** Selects the largest numeric value in the table.
- **Arm 2 (Random):** Selects a uniformly random cell.

We use an ϵ-greedy policy: with probability ϵ_t choose an arm uniformly at random, and with probability $1 - \epsilon_t$ choose the greedy arm $\arg\max_{a \in \mathcal{A}} Q_t(a)$. The exploration rate decays over time over the training horizon T:

$$\epsilon_t \;=\; \max\!\left(0,\; \epsilon_0\left(1 - \frac{t}{T}\right)\right), \qquad \epsilon_0 \approx 0.5.$$

We maintain an action-value estimate $Q_t(a)$ for each arm. After selecting a_t and observing reward r_t (binary relevance), we update only the chosen arm via:

$$Q_{t+1}(a_t) \;\leftarrow\; Q_t(a_t) \;+\; \alpha_t\big[r_t - Q_t(a_t)\big]$$

3.4 Reinforcement Learning Algorithms

Proximal Policy Optimization (PPO). PPO [4] is the standard on-policy RL algorithm used for aligning LLMs. It optimizes a policy π_θ to maximize expected reward while constraining the update to stay close to the pre-trained behavior policy π_{ref}. The objective function is given by:

$$L^{PPO}(\theta) = \mathbb{E}_{(x,y)\sim D_{\pi_{\theta_{old}}}} \left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right] - \beta \text{KL}(\pi_\theta || \pi_{ref}) \qquad (3)$$

where $r_t(\theta) = \frac{\pi_\theta(y|x)}{\pi_{\theta_{old}}(y|x)}$ is the probability ratio, A_t is the generalized advantage estimate, and ϵ is a clipping hyperparameter (typically 0.2).

Computational Cost: PPO requires four models in memory: the active Policy, the frozen Reference model, the Value function (Critic), and usually a separate Reward model. This high VRAM requirement is a primary bottleneck for training 3B+ parameter models on consumer hardware.

Group Relative Policy Optimization (GRPO). GRPO [5] serves as a more efficient alternative to PPO for reasoning tasks. It eliminates the need for a separate value function (Critic) model. Instead, for each question q, it samples a group of G outputs and estimates the advantage A_i directly from group statistics:

$$A_i = \frac{r_i - \text{mean}(\{r_1, ..., r_G\})}{\text{std}(\{r_1, ..., r_G\}) + \epsilon} \tag{4}$$

Direct Preference Optimization (DPO). *(To be completed by Teammate)* DPO [**?**] optimizes preferences directly without a reward model using the implicit reward formulation.

REINFORCE Leave-One-Out (RLOO). *(To be completed by Teammate)* RLOO [**?**] uses a variance-reduced estimator where the baseline is the average reward of the other $K-1$ samples.

4 Experimental Setup

Track 1 (Deep RL). We trained a **Llama-3.2-3B** model using **LoRA (Rank=16)**. We conducted two experimental runs:

- **Run 001 (Standard):** 10 epochs, frequent evaluation.
- **Run 002 (Extended):** 20 epochs, to test convergence stability.

The baseline model (SFT) was trained using Cross-Entropy loss to maximize the likelihood of the Gold candidate.

Extension (Weak Learners). We utilized **TinyLlama-1.1B** fine-tuned with LoRA on CPU to compare RLOO and DPO against SFT.

Track 2 (Bandits). We trained an ε-greedy bandit for $10,000$ episodes, with ε decaying linearly from 0.5 to 0.

5 Results and Analysis

5.1 Track 1 Results: The SFT Ceiling (3B Model)

Analysis by Jinyoon Kim

We compared SFT, PPO, and GRPO on the Llama-3.2-3B model.

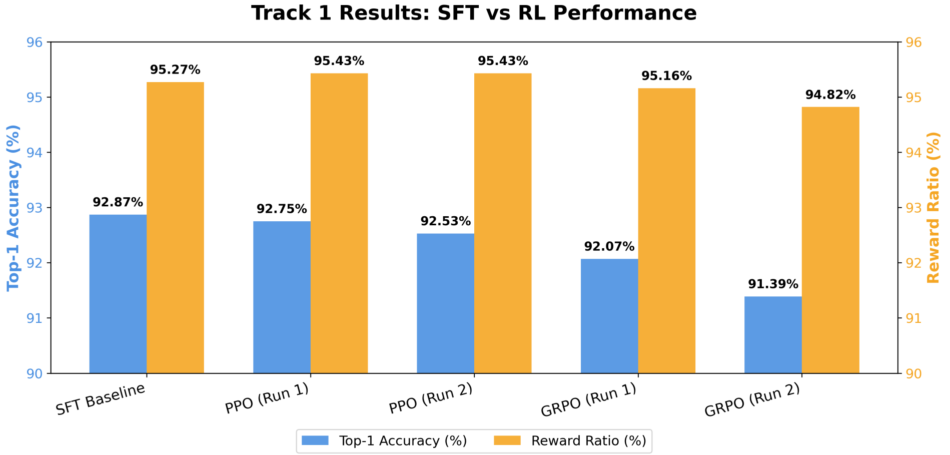

Figure 2: **Performance Comparison (Accuracy vs. Reward Ratio).** The SFT baseline (left) achieved the highest accuracy. While PPO maintained stability (blue bars), GRPO degraded in extended training runs (rightmost bar).

Table 1: Track 1 Results: SFT Baseline vs. RL Agents

Model Configuration	Top-1 Accuracy	Reward Ratio	∆ from SFT
SFT Baseline	**92.87%**	**95.27%**	–
PPO (Run 001)	92.75%	95.43%	-0.12%
GRPO (Run 001)	92.07%	95.16%	-0.80%
GRPO (Run 002)	91.39%	94.82%	-1.48%

Contrary to our initial hypothesis, **RL training failed to outperform the SFT baseline**. Our analysis identifies three key reasons:

1. **The "SFT Ceiling":** The ranking task, while semantically complex, proved structurally simple for the SFT model, which achieved ∼93% accuracy immediately. This left very little exploration room for RL agents to discover novel strategies.
2. **Stability vs. Efficiency Trade-off:** PPO remained highly stable (within 0.3% of baseline) due to the KL-penalty anchoring it to the reference model. GRPO, lacking this anchor, degraded significantly in the extended run (-1.48%), suggesting it began overfitting to the reward proxy or "gaming" the noise in the candidate pool.

5.2 Extension Results: The RL Gap (1B Model)

Analysis by Donggen Li

To test if RL benefits weaker models, we repeated the experiment with TinyLlama-1.1B.

Table 2: TinyLlama-1.1B: Baseline vs. RL Methods

Method	Accuracy	Avg Reward	Reward Gap
Baseline (SFT)	0.105	0.601	0.699
DPO	0.360	1.028	0.272
RLOO	**0.460**	**1.057**	**0.243**

Analysis. Unlike the 3B model, the 1B model benefited massively from RL. RLOO improved accuracy by **+35.5 percentage points** over SFT. This confirms that RL is a powerful amplifier for weak learners, even if it has diminishing returns on stronger models.

5.3 Track 2 Results: Bandit Learning

Analysis by Scarlett Yu

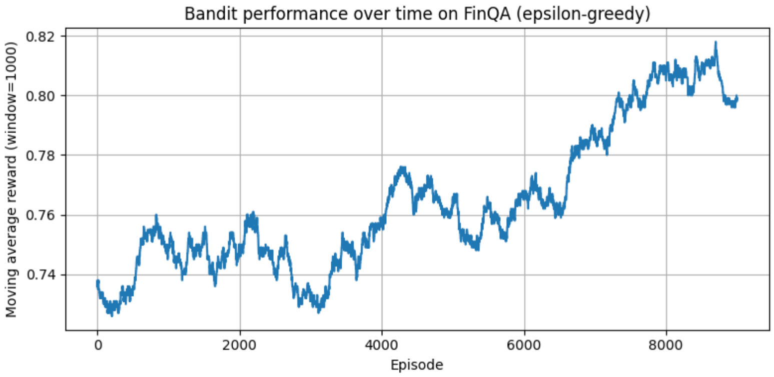

Figure 3: Moving-average reward during non-contextual bandit training.

Early in training, the bandit explores all arms randomly. As ε decays, the agent learns to ignore the Random arm (Arm 2) and converges on the Token Overlap and Max Value heuristics. This demonstrates that simple ε-greedy bandits can reliably identify optimal heuristics from noisy signals without contextual features.

6 Conclusion

This project demonstrates that the effectiveness of Reinforcement Learning in financial reasoning is highly context-dependent.

- For **capable models (3B) on discriminative tasks**, RL adds complexity without gains; SFT is superior.
- For **weaker models (1B)**, RL algorithms (RLOO/DPO) provide massive boosts (+35%), validating RL for alignment.
- For **heuristic selection**, simple bandits successfully identify strategies without deep learning costs.

References

[1] Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. Finqa: A dataset of numerical reasoning over financial data, 2022.

[2] 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, 2021.

[3] 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. In *International Conference on Learning Representations*, 2022.

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[5] 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, 2024.