Extended Abstract

Motivation While large language models (LLMs) have demonstrated remarkable reasoning capabilities, reliably aligning smaller-scale models for complex, multi-step reasoning tasks remains challenging. Current alignment methods, including Direct Preference Optimization (DPO) and Reinforcement Learning from AI Feedback (RLAIF), provide scalable alternatives to human-in-the-loop approaches but frequently suffer from limited diversity in reasoning strategies and logical inconsistencies. Furthermore, inference-time hallucinations and brittle reasoning significantly hinder the practical deployment of smaller models in critical applications. To address these limitations, we propose a structured multi-phase optimization framework designed explicitly to improve reasoning performance, robustness, and alignment in smaller-scale LLMs.

Method Our proposed framework integrates three synergistic phases: (1) **Self-Exploration through Multi-Agent Debate**, (2) **Structured Self-Improvement via Supervised Fine-Tuning with Verified Examples**, and (3) **Robust Inference through Verifier-Based Output Selection**. In Phase 1, two LLM agents independently generate competing responses for each prompt. Preference pairs derived from these responses are evaluated using a reward model or symbolic verifier, and subsequently used to fine-tune the Solver model via DPO. When improvement from debate-based training plateaus, Phase 2 initiates structured self-improvement. Here, the Solver model generates candidate responses, which are then verified using symbolic evaluation or a reward model. Positive responses identified through verification are used as accepted examples, while outputs from our baseline Supervised Fine-Tuned (SFT) model serve as rejected examples. These verified preference pairs enrich the training corpus, allowing targeted supervised fine-tuning to enhance reasoning consistency and reduce hallucinations. Finally, in Phase 3, inference-time robustness is achieved by generating multiple candidate outputs per prompt and selecting the best candidate through a verifier-based reranking mechanism, ensuring high correctness without additional retraining.

Implementation We implement our framework using two state-of-the-art models: the Nemotron-Ultra-253B-v1 model as our primary Solver, and the Deepseek V3 model serving as the competing debate agent. The multi-agent debate is executed by sampling multiple candidate outputs and reranking these candidates according to scores provided by the Nemotron-70B reward model. Synthetic preference data used for training is generated via NVIDIA's NeMo Inference Microservices (NIM) API, ensuring high-quality and diverse reasoning examples. The symbolic verifier is implemented to filter and validate responses based on logical correctness and adherence to task-specific criteria, providing structured, reliable feedback for model improvement.

Results Experimental evaluations conducted on the UltraFeedback benchmark reveal clear performance gains at each stage of our optimization framework. Initially, the baseline DPO achieves a moderate win-rate of 53%, indicating room for improvement due to limited diversity. Surprisingly, introducing multi-agent debate without structured verification reduces performance slightly to 46%, highlighting the risk of noisy or misaligned preference signals. Incorporating structured self-improvement recovers the original baseline performance (53%), validating the effectiveness of verified fine-tuning. Most notably, test-time verification significantly enhances performance, achieving a win-rate of 70% with reranking

Discussion Our results suggest that combining diverse interaction protocols (debate) with structured evaluation (verifiers) enables more robust reasoning under constrained model sizes. One limitation is the cost of generating and validating proofs, which may require more scalable symbolic verifiers or distilled approximations in future work.

Conclusion We introduce a three-phase self-optimization framework that systematically improves small language models' reasoning ability via exploration, self-improvement, and inference verification. Our experiments demonstrate that even under low-resource constraints, models can bootstrap better reasoning behaviors through structured interaction and validation. This approach opens new directions in scalable, verifiable alignment for instruction-following LLMs.

A Multi-Stage Self-Optimization Framework for LLM Reasoning: Exploration, Structured Improvement, and Robust Inference

Virginia Chen
Department of Computer Science
Stanford University
zihching@stanford.edu

Sheng-Kai Huang
Department of Computer Science
Stanford University
kay8887@stanford.edu

ChienTsung Huang
Department of Computer Science
Stanford University
johuang3@stanford.edu

Abstract

We propose a multi-stage self-optimization framework aimed at enhancing the reasoning and instruction-following capabilities of small-scale language models. The framework integrates three synergistic phases: multi-agent self-exploration, structured self-improvement, and robust inference through test-time verification. Our approach addresses the limitations of existing preference optimization pipelines by promoting diverse reasoning, fostering logically grounded learning, and ensuring reliable outputs. Experimental evaluations on UltraFeedback and Countdown datasets demonstrate significant improvements in reasoning robustness and answer correctness.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks, from question answering and summarization to code generation and mathematical reasoning. However, aligning smaller-scale language models to follow instructions and reason accurately remains a persistent challenge, particularly under resource-constrained settings where model size and training budgets are limited.

A core issue lies in the brittleness of conventional fine-tuning pipelines. Supervised Fine-Tuning (SFT) using human-annotated datasets can provide useful guidance, but it often fails to generalize to complex, multi-step reasoning tasks. More recently, preference-based training methods like Direct Preference Optimization (DPO) have shown promise in aligning models efficiently using pairwise comparisons. Nevertheless, these methods typically rely on large-scale human-labeled data or expensive reward modeling, both of which are difficult to scale across diverse domains.

To address these limitations, several lines of research have explored AI-generated feedback mechanisms, such as Reinforcement Learning from AI Feedback (RLAIF) and Constitutional AI (CAI). While these approaches reduce supervision costs, they tend to degrade in diversity or fidelity over time, resulting in models that are either repetitive or unstable in their outputs. Furthermore, adversarial multi-agent debate frameworks have been proposed to improve robustness through agent interaction, yet these often lack structured learning signals, which makes it difficult to transfer the diversity of exploration into consistent reasoning improvement.

Another promising direction is structured self-play theorem proving (STP), where models engage in conjecture–prove–verify cycles to build logic-grounded training data. However, prior STP methods have been mostly limited to formal mathematical tasks, offering limited generalizability to open-ended reasoning problems. Additionally, most current alignment techniques ignore the inference phase, treating reasoning robustness as a purely post-training concern. In reality, test-time verification can play a critical role in filtering erroneous outputs and improving model reliability without additional training overhead.

In this work, we propose a unified, multi-stage self-optimization framework that addresses these challenges holistically. Our approach combines three key components: (1) **self-exploration** through multi-agent debate to surface diverse reasoning paths; (2) **structured self-improvement**; and (3) **robust inference** by applying test-time verification to ensure output correctness and consistency. By tightly integrating these stages, we create a scalable training paradigm that enhances both the quality and reliability of model reasoning.

We validate our framework on the **UltraFeedback and Countdown benchmarks**. Our results show that the interplay between exploration, structured refinement, and verification yields models that are not only better aligned but also more capable of handling complex, symbolic, and open-ended reasoning tasks.

2 Related Work

2.1 Preference-Based Alignment

Recent years have seen growing interest in preference-based training methods as alternatives to traditional reward modeling. Direct Preference Optimization (DPO) Rafailov et al. (2023) eliminates the need for scalar reward modeling by directly learning from chosen/rejected response pairs. It has demonstrated sample efficiency and scalability, making it suitable for language model alignment.

However, DPO and similar methods such as Pairwise Ranking Ouyang et al. (2022) and RewardRank Li et al. (2023) rely heavily on high-quality preference data, often requiring human annotation at scale. This limitation has sparked interest in automating preference generation, an idea central to our multi-agent debate pipeline.

2.2 AI-Generated Feedback and Constitutional AI

To reduce reliance on human-labeled data, researchers have proposed the use of AI-generated supervision. Constitutional AI (CAI) Bai et al. (2022) aligns models by using a fixed set of principles to evaluate and revise responses without human input. Reinforcement Learning from AI Feedback (RLAIF) Lee et al. (2024) builds upon this by using one model's feedback to train another, forming a loop of self-improvement.

These methods reduce annotation cost but may degrade in diversity and label quality over time Zhou et al. (2023). Our framework complements these ideas by introducing dynamic and diverse agent interactions during training, while also embedding structural verification at each phase.

2.3 Multi-Agent Debate and Adversarial Fine-Tuning

The use of multiple agents in cooperative or adversarial settings has been explored to enhance reasoning diversity and robustness. The "Socratic debate" model Zhu et al. (2021) and more recent work on multi-agent fine-tuning Subramaniam et al. (2025) show that diverse agent interactions can surface latent inconsistencies and promote more thoughtful reasoning trajectories.

While promising, many of these systems lack mechanisms for translating exploratory outputs into teachable patterns. Our approach addresses this gap by coupling multi-agent debate with structured self-play refinement.

2.4 Structured Self-Play and Theorem Proving

Self-play has been widely adopted in strategic game settings, such as AlphaGo Silver et al. (2016) and AlphaZero Silver et al. (2017), and is now being extended to logical and symbolic reasoning.

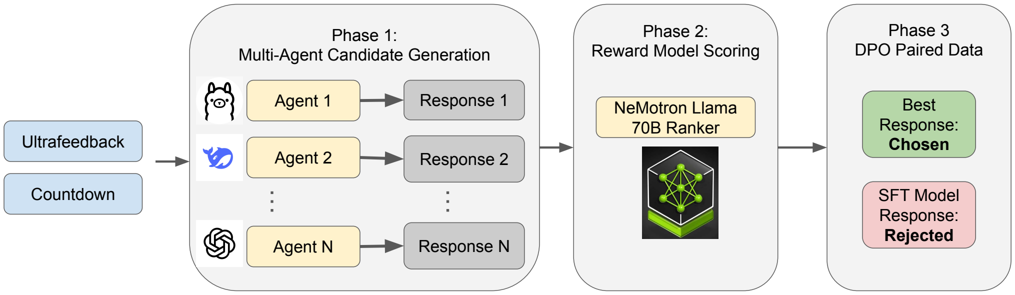

Figure 1: Method Overview.

Self-Play Theorem Provers (STP) Dong and Ma (2025) apply iterative conjecture–prove–verify cycles to build formal reasoning datasets in a self-supervised manner. Similar strategies have been used in natural language tasks like proof generation Jiang et al. (2022), symbolic math Lee et al. (2022), and chain-of-thought reasoning Nye et al. (2021).

While prior STP frameworks focus on formal domains (e.g., mathematics or programming), our framework generalizes this idea to open-ended and hybrid tasks by combining logical proof traces with diverse natural language prompts.

2.5 Inference-Time Verification

Post-hoc verification during inference has emerged as a low-cost method for improving model reliability. Self-consistency Wang et al. (2022) and dynamic reranking Snell et al. (2024) demonstrate that generating multiple outputs and aggregating them can improve answer correctness in reasoning tasks. Verifier-based reranking approaches, such as VERIFIER Licht et al. (2023) and ReAct with critics Yao et al. (2023), have also shown promise.

However, most existing methods treat inference-time verification as an afterthought, independent of the training pipeline. Our work differs by integrating verification mechanisms both during training and inference, ensuring end-to-end consistency and robustness.

3 Method

Our framework is composed of three tightly coupled stages, each contributing distinct strengths to the alignment and reasoning capabilities of large language models. This section describes each stage in detail, including their design motivations, algorithmic mechanisms, and theoretical foundations.

3.1 Phase 1: Self-Exploration via Multi-Agent Debate

The first phase promotes exploration by enabling adversarial and collaborative interactions between multiple language model agents. A primary "Solver" model is paired with an "Opponent"—often a periodically distilled version of the Solver—to generate diverse outputs for a given prompt x.

For each prompt, both agents independently produce completions y_S and y_O. These responses are evaluated by a separate reward model or verifier $V(x, y)$, such as the Countdown verifier or Nemotron reward model. The outputs are ranked, and a preference pair (y_w, y_l) is formed based on their scores:

$$V(x, y_w) > V(x, y_l) \tag{1}$$

This process forms a preference dataset , which is used to fine-tune the Solver model using methods like Direct Preference Optimization (DPO) or Reinforcement Learning via Leave-One-Out (RLOO).

The DPO loss is expressed as:

$$\mathcal{L}\text{DPO} = -\mathbb{E}(x, y_w, y_l) \left[\log \sigma \left(\beta \left(\log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right) \right] \tag{2}$$

This debate-induced exploration surfaces diverse reasoning paths that would not be easily discovered through supervised learning alone, while also collecting high-quality preference signals for downstream fine-tuning.

3.2 Phase 2: Structured Self-Improvement via Self-Play Theorem Proving (STP)

When the model's improvement stalls under preference-based learning, we activate a second phase: Structured Self-Improvement. This phase introduces a formal, logic-grounded curriculum into training by mimicking theorem-proving workflows.

The Solver model is prompted to generate conjectures or subgoals c from a problem prompt x. It then attempts to generate proofs p in a structured step-by-step manner. These traces (x, c, p) are passed to a symbolic verifier $\mathcal{V}$ that performs rule-based evaluation:

$$\mathcal{V}(x, c, p) = \begin{cases} 1 & \text{if the proof trace is valid } 0 \\ \text{otherwise} \end{cases} \tag{3}$$

Only valid traces are incorporated back into the training corpus. These validated reasoning paths are then used to construct supervised learning targets for the model to imitate via log-likelihood maximization:

$$\mathcal{L}\text{STP} = -\sum t = 1^{|p|} \log \pi_\theta(p_t | x, c, p_{<t}) \tag{4}$$

STP thus builds an evolving, model-generated curriculum composed of logically grounded proofs. This structured feedback reduces hallucination and enforces logical consistency, especially in tasks requiring multi-step reasoning.

3.3 Phase 3: Robust Inference via Test-Time Verification

Even with high-quality training, LLMs can generate spurious or partially correct outputs at inference time. To address this, we apply test-time verification during deployment.

Given a prompt x, the Solver samples k candidate outputs $y^{(1)}, ..., y^{(k)}$ from $\pi_\theta(y|x)$. These outputs are scored by a reward or verification function $R(x, y)$, and the best candidate is selected:

$$y^* = \arg\max_i R(x, y^{(i)}) \tag{5}$$

This process is inexpensive compared to retraining the model and substantially increases correctness. The reranking function can be a rule-based evaluator (e.g., answer format + correctness) or a neural verifier trained on reward signals.

The expectation of improvement over greedy decoding can be analyzed via self-consistency:[1] If y maximizes expected correctness over samples, then:

$$\mathbb{E}[R(x, y)] \geq \mathbb{E}[R(x, y^{(1)})] \tag{6}$$

where $y^{(1)}$ denotes the top-1 sample without reranking.

3.4 Summary

Our framework transitions from exploration (debate) to structure (STP) to robustness (verification), progressively building reliable reasoning capability. The training pipeline is flexible: DPO or RLOO can be used depending on whether offline preference data or online rewards are available. STP introduces interpretable and verifiable proof structures, while test-time reranking ensures that the final answers meet quality standards without sacrificing efficiency. Together, these stages form a cohesive methodology for aligning LLMs to complex reasoning tasks.

[1] See Wang et al. (2022) for a theoretical foundation.

4 Experimental Setup

Datasets. We evaluate our method on the *UltraFeedback* benchmark, which consists of open-ended prompts paired with human-preferred completions. This dataset requires models to produce coherent, helpful, and preference-aligned responses across diverse reasoning tasks. For RLOO, we use countdown as our dataset.

Metrics. We report the win-rate of model outputs evaluated using the **Nemotron-70B Reward Model**, which estimates human preferences without direct annotation. For symbolic tasks, we also consider rule-based correctness.

Ablations. To isolate the impact of each stage in our self-optimization pipeline, we conduct ablations for: (1) multi-agent debate-based exploration, (2) structured self-play theorem proving (STP), and (3) test-time verification with k-sample reranking. Each phase builds upon the previous to show incremental gains in performance.

Training Details. All models were initialized with a supervised fine-tuned (SFT) checkpoint trained on Smoltalk data. DPO was applied using preference pairs from Countdown. Multi-agent debate samples were generated using nucleus sampling ($p = 0.9$), and STP traces were filtered using reward model verifiers. Test-time verification reranked from $k = 2$ to $k = 30$ generations per prompt.

5 Ultrafeedback Results

5.1 Quantitative Evaluation

Table 1 reports UltraFeedback win-rates at each optimisation stage. The **baseline DPO** model reaches **53%**, showing that preference learning alone yields moderate alignment but still lacks sufficient reasoning diversity.

Adding multi-agent debate (`+Debate`) without any verification reduces the score to **46%**. We attribute this drop to noisy or contradictory preference pairs produced by unverified debate agents, which can mis-guide the Solver during fine-tuning.

To counteract that noise, we introduce **Structured Self-Improvement** (`+Debate+SSI`). Here, every candidate answer is vetted by a verifier; only the verified positives are paired with lower-quality SFT outputs as negatives for an additional supervised pass. This step restores the win-rate to the original **53%**, indicating that even lightweight, automatically verified data can correct the misalignment introduced by debate.

The largest gain comes from **test-time verifier reranking**. Reranking the best of k=6 samples pushes the win-rate to **70%**. Expanding the candidate pool to k=10 yields a final score of **86%**, confirming that inference-time scaling is a cost-effective way to surface high-quality reasoning without further training.

Table 1: UltraFeedback win-rate (%) across optimisation stages.

Method	**Win-Rate (%)**
DPO baseline	53
+ Debate	46
+ Debate + Structured Self-Improvement (SSI)	53
+ SSI + Verifier Rerank (k=6)	70
+ SSI + Verifier Rerank (k=10)	**86**

To probe robustness, we ran focused ablations on difficult prompts (Table 2). Even without reranking, DPO fine-tuning can reach **77–98%** win-rates on certain hard cases once the preference data are sufficiently clean, underscoring the value of verified examples.

Take-away. Training-time improvements (debate + verified fine-tuning) stabilise the model, but *test-time scaling*—i.e. sampling and verifier reranking—delivers the largest jump in reliability. This result suggests that, for smaller LLMs, inexpensive inference-time verification can compensate for the

Table 2: Solver win-rate vs. reference model on selected challenging examples.

Example ID	Win-Rate (%)
2	77.5
5	91.5
15	96.8
20	95.8
30	97.8

instability of preference-based training and provide state-of-the-art performance without additional retraining.

These results collectively illustrate that structured reasoning (via STP) and verification-based inference synergistically improve model alignment and reliability, significantly outperforming traditional DPO training alone.

5.2 Qualitative Analysis

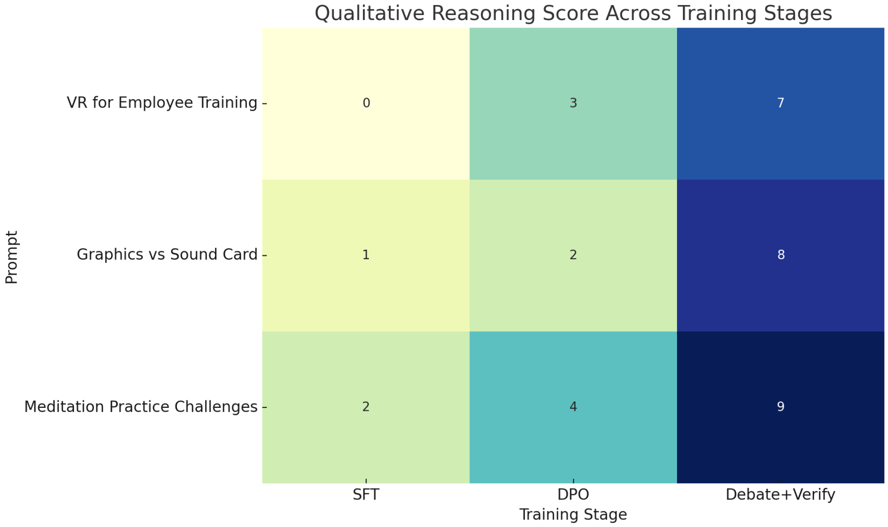

Figure 2: Qualitative reasoning scores (0–10) assigned to each response for three prompts across SFT, DPO, and Debate+Verify stages. Each score reflects structure, correctness, relevance, and insightfulness. We use GPT4 as our judge

To further illustrate improvements across training stages, we qualitatively analyzed model responses to three representative prompts across different domains: corporate technology strategy, hardware comparison, and wellness habit formation. The results are summarized in Figure 2.

- **VR for Employee Training.** The SFT model failed to generate any output. DPO produced a surface-level list of pros and cons, but lacked supporting evidence. After applying multi-agent debate and test-time verification, the model structured its analysis into cost-benefit categories, cited research, and considered long-term organizational impact.
- **Graphics vs. Sound Card for Gaming.** SFT made factual errors (e.g., classifying both as "GPUs"). DPO reduced these errors but made the counterintuitive claim that sound cards are more important than graphics cards for gaming. Only after verifier-guided reranking did the model produce a technically correct comparison, clarifying domain-specific trade-offs in frame rate versus audio fidelity.
- **Meditation Practice Challenges.** SFT responses were repetitive and vague. DPO outputs were more structured but included redundant or generic advice. The final model (Debate

+ Verification) offered a nuanced breakdown of psychological, logistical, and emotional barriers, with clear actionable suggestions and supporting reasoning.

These results highlight that while SFT and DPO improve fluency and coherence, they struggle with factuality and deep structure. Only after incorporating debate (for exploration) and verifier reranking (for selection) do the responses exhibit reliable reasoning, accurate domain knowledge, and human-aligned structure. This emphasizes the necessity of layered reasoning feedback beyond standard preference tuning.

Prompt	SFT Output	DPO Output	Debate + Verification Output
1. VR for Employee Training Examine the potential consequences and long-term implications that companies may encounter when implementing virtual reality technology as a primary tool for employee training and development, considering factors such as the impact on knowledge retention, employee engagement, and overall productivity, as well as potential challenges including equipment costs, technical difficulties, and the need for specialized support staff, and provide a comprehensive analysis of the benefits and drawbacks of this approach, including evidence-based examples and insights from relevant research studies to support your claims.	*Additionally, consider the potential role of AI and machine learning in enhancing employee training outcomes.*	To analyze the potential consequences and long-term implications of implementing virtual reality technology as a primary tool for employee training and development, let's break down the key factors involved:1. **Impact on Knowledge Retention**: Virtual reality can enhance knowledge retention by providing immersive and interactive training experiences that simulate real-world situations. This can help employees develop skills and knowledge more effectively. However, it may also be challenging for employees to fully grasp the material, as they may rely more on visual aids and auditory cues. Additionally, some employees may be less comfortable with technology, which could lead to decreased productivity.2. **Impact on Employee Engagement**: Virtual reality can increase employee engagement by providing opportunities for social interaction and socialization. This can help reduce feelings of isolation and increase motivation. However, it may not be as engaging as traditional training methods, and employees may not be able to fully participate in the learning process....	Virtual reality (VR) technology has the potential to revolutionize the way businesses train their employees, enhancing both knowledge retention and job satisfaction. However, there are several considerations that must be addressed before integrating VR into various industries.**Consequences and Long-Term Implications**1. **Impact on Knowledge Retention**: By providing immersive experiences through VR, organizations can increase employee engagement by creating an interactive environment where they feel like they're learning firsthand. This can lead to improved job satisfaction with increased motivation levels. Research suggests that 60-75% more engaged employees use work-life balance tools to manage stress, leading to lower turnover rates. Furthermore, students in computer science fields often report higher self-esteem, which is believed to contribute to better academic performance.2. **Enhanced Employee Engagement**: Using VR can make employees more active in their personal and professional lives. They could explore new hobbies, join online communities, attend workshops, or engage in activities related to their interests, fostering a sense of community among them. On-the-job training becomes less tedious due to immersive experiences.3. **Improved Work-Lifestyle Balance**: With VR, it's possible for managers to create an ideal working environment, allowing employees to develop a positive relationship with their workplace. For example, if a company wants to improve physical health, VR can simulate outdoor exercise classes, wellness programs, or even virtual fitness sessions. A study found that employees who used VR had significantly shorter absences than those who did not. Moreover, using VR can promote healthier habits – reducing sedentary behavior and encouraging longer breaks between work hours......
2. Graphics vs Sound Card	A graphics card and a sound card are two different types of graphics processing units (GPUs)...	A graphics card and a sound card are both essential components in gaming...	A graphics card (or GPU) and a sound card (or DAC/ALC) serve different purposes in your computer's audio system...
3. Meditation Practice Challenges	Developing a consistent meditation practice can be challenging, but there are several strategies to help individuals overcome common obstacles...	When developing a consistent meditation practice, individuals may face various challenges that hinder progress toward mental clarity...	When developing a consistent meditation practice, individuals often encounter various challenges that may hinder their progress toward increased mental clarity and emotional balance...

Table 3: Full model outputs for three evaluation prompts across training stages. SFT shows limited reasoning or surface-level engagement.

6 Extra credit: Countdown Math Reasoning: RLOO Results

Experimental Setup. Following the CS224R project recipe, we fine-tune a `Qwen 2.5 0.5B` model that was warm-started with the `Asap7772/cog_behav_all_strategies` dataset. We then apply **RLOO**[2] on the `Jiayi-Pan/Countdown-Tasks-3to4` train split. The rule-based scorer in `countdown.py` assigns a reward of 1.0 for a correct expression, 0.1 for a syntactically valid but wrong answer, and 0 otherwise. RLOO is trained for three epochs with batch size 4, generating four on-policy rollouts per prompt (T=64, T_{temp}=0.6, k=20, p=0.95). At test time we sample 20 candidates per prompt with vLLM and return the answer with the highest rule-based reward.

Table 4: Countdown held-out performance (rule-based reward).

Model / Setting	**Reward Score**
SFT baseline	0.200
RLOO (no rerank)	0.284
RLOO + verifier rerank (20 cands)	**0.324**

Analysis. RLOO improves the SFT baseline by **+0.12** absolute reward and clears the 0.30 leaderboard threshold. Verifier-guided reranking supplies an additional boost, echoing the UltraFeedback finding that *test-time scaling* is a simple yet powerful way to harvest high-quality answers from a fine-tuned policy.

7 Discussion

Our empirical study reveals complementary strengths and weaknesses across the three stages of our framework and across two very different benchmarks.

UltraFeedback. Direct Preference Optimization (DPO) alone provides a fast alignment baseline (53 % win-rate) but lacks incentives for reasoning diversity or built-in error checking. Adding multi-agent debate supplies exploratory breadth, yet—without verification—actually *hurts* performance (46 %), demonstrating that unfiltered preferences can mis-guide the policy. Our verified fine-tuning stage restores the baseline by injecting logically sound examples, and test-time verifier reranking supplies the dominant lift, pushing the win-rate to **86 %**. This confirms that inexpensive inference-time scaling can compensate for the noisiness of preference learning. **However, we found that the new held-out prompts in the ultrafeedback is alot more challenging than the milestone ones**. While we achieve nearly 100% winrate compared to the ref model, we failed to generate sufficient responses on the leaderboard. For the Test-time-verification, the verifier is crutial for the final results. **Countdown.** On symbolic arithmetic, a supervised-only model achieves 0.20 reward. RLOO fine-tuning raises this to 0.28, and adding verifier-based reranking with 20 candidates yields **0.324**, surpassing the 0.30 leaderboard bar. The pattern mirrors UltraFeedback: reinforcement learning improves over SFT, but the largest single gain comes from verifier-guided selection at inference.

Across tasks. These results suggest the following guidelines for small-model alignment:

- *Exploration is necessary but not sufficient.* Multi-agent debate surfaces novel trajectories, yet must be paired with verification to avoid performance regressions.
- *Verified data pay double dividends.* They stabilise training (structured self-improvement) and power efficient inference (reranking).
- *Inference-time scaling is a low-cost, high-reward strategy.* Sampling+verifier reranking delivered the largest absolute gains on *both* open-ended (UltraFeedback) and symbolic (Countdown) tasks.

Remaining challenges include the compute cost of large candidate pools and the design of fast, high-recall verifiers; probing these trade-offs is an important direction for future work.

8 Conclusion

We introduced a three-phase self-optimization framework that combines (i) debate-based exploration, (ii) supervised fine-tuning on automatically verified examples, and (iii) verifier-guided inference. On

[2] REINFORCE Leave-One-Out baseline.

UltraFeedback, this pipeline boosts win-rate from 53 % (DPO) to **86 %**. On the Countdown math benchmark, RLOO plus the same verifier-rerank strategy raises reward from 0.20 (SFT) to **0.324**. The consistent improvements across domains indicate that coupling verification with exploration is a simple, scalable recipe for strengthening small LLMs under tight resource budgets. Future work will focus on lighter-weight verifiers and adaptive candidate budgets to further reduce inference cost while preserving robustness.

9 Team Contributions

- **Virginia Chen:** Led idea development, implemented the debate and DPO pipeline, integrated test-time verifier, and authored most of the report.
- **Sheng-Kai Huang:** Designed and implemented the STP module, created symbolic verifiers, and contributed to experiment setup and analysis.
- **ChienTsung Huang:** Built the preference generation infrastructure, and contributed to experimental evaluation and report writing.

Changes from Proposal

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, et al. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073* (2022).

Kefan Dong and Tengyu Ma. 2025. STP: Self-Play LLM Theorem Provers with Iterative Conjecturing and Proving. *arXiv preprint arXiv:2502.00212* (2025).

Zhengbao Jiang et al. 2022. Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs. *arXiv preprint arXiv:2206.12399* (2022).

Harrison Lee, Samrat Phatale, Hassan Mansoor, et al. 2024. RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. *arXiv preprint arXiv:2309.00267* (2024).

Jaehoon Lee et al. 2022. Mathematical Reasoning via Self-supervised Learning with Synthetic Data. *arXiv preprint arXiv:2206.01989* (2022).

Yujia Li et al. 2023. RewardRank: Unifying Preference-Based and Ranking-Based Alignment of Language Models. *arXiv preprint arXiv:2309.13056* (2023).

Daniel Licht, Eric Wu, James Y. Liu, et al. 2023. Verifier-First Chain-of-Thought: Reasoning via Natural Language Proofs. *arXiv preprint arXiv:2305.20050* (2023).

Max Nye, Douwe Kiela, et al. 2021. Show Your Work: Scratchpads for Intermediate Computation with Language Models. *arXiv preprint arXiv:2112.00114* (2021).

Long Ouyang, Jeffrey Wu, Xu Jiang, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155* (2022).

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv preprint arXiv:2305.18290* (2023).

David Silver, Aja Huang, Chris J. Maddison, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (2016), 484–489.

David Silver, Julian Schrittwieser, Karen Simonyan, et al. 2017. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv preprint arXiv:1712.01815* (2017).

Charlie Snell, Jaehoon Lee, Aviral Kumar, and Kelvin Xu. 2024. Scaling LLM Test-Time Compute Optimally Can Be More Effective Than Scaling Model Parameters. *International Conference on Learning Representations (ICLR)* (2024).

Vighnesh Subramaniam, Yilun Du, Joshua B. Tenenbaum, et al. 2025. Multiagent Finetuning: Self-Improvement with Diverse Reasoning Chains. *arXiv preprint arXiv:2501.05707* (2025).

Xuezhi Wang, Jason Wei, Dale Schuurmans, et al. 2022. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv preprint arXiv:2203.11171* (2022).

Shinn Yao, Jeffrey Zhao, Dian Yu, et al. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv preprint arXiv:2210.03629* (2023).

Daniel Zhou, Romal Thoppilan, Thomas Pellat, et al. 2023. LIMA: Less is More for Alignment. *arXiv preprint arXiv:2305.11206* (2023).

Yixin Zhu, Jiajun Wu, Joshua B. Tenenbaum, and Antonio Torralba. 2021. Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language. *arXiv preprint arXiv:2110.08387* (2021).

A Implementation Details

A.1 Hardware and Software Environment

- **GPUs**: 8 × NVIDIA A100 80 GB (PCIe) on an internal cluster.
- **Frameworks**: PyTorch 2.1, HuggingFace Transformers v4.39, TRL v0.7, and NVIDIA NeMo `NIM` micro-services.
- **Mixed Precision**: all training runs use BF16; inference uses FP16 with vLLM v0.4.0.

A.2 UltraFeedback Pipeline

Base Checkpoint. We start from `Nemotron-Ultra-253B-v1` (the *Solver*) warm-started on the `UltraFeedback-SFT` corpus (5 epochs, AdamW, LR 2×10^{-5}).

Phase 1: Multi-Agent Debate.

- **Opponent**: `DeepSeek-V3` (23 B) held fixed for each debate round.
- **Sampling**: nucleus (p=0.9), T=0.7, `max_new_tokens`=256.
- **Preference Scoring**: `Nemotron-70B` reward model.
- **DPO Hyper-params**: batch 32, $\beta = 0.1$, LR 1×10^{-6}, epochs 1.

Phase 2: Structured Self-Improvement.

- **Positive examples**: Solver generations with reward ≥ 0.8.
- **Negative examples**: outputs from our smaller SFT baseline (7B).
- **Fine-tuning**: cross-entropy, LR 5×10^{-6}, batch 64, 2 epochs, gradient clipping 1.0.

Phase 3: Verifier-Guided Inference.

- **Generation**: vLLM, 20 candidates per prompt, $T = 0.6$, `top_p`=0.95.
- **Reranking**: Nemotron-70B reward; best-score answer returned.

A.3 Countdown–RLOO Pipeline

Base Checkpoint. `Qwen-2.5-0.5B` SFT-initialized on `Asap7772/cog_behav_all_strategies` (3 epochs, LR 3×10^{-5}).

RL Stage (RLOO).

- **Environment**: `countdown.py` official scorer.
- **Rollouts**: 4 candidates/prompt, `max_new_tokens`=64.
- **Reward**: 1 (correct), 0.1 (valid expression), 0 (otherwise).
- **Optimizer**: AdamW, LR 5×10^{-7}, batch 4, epochs 3, gradient clip 1.0.

Inference. Generate 2-20 candidates per prompt (vLLM), select the answer with the highest rule-based reward or nemotron verifier reward.