Progress Report
210386A

Project Overview

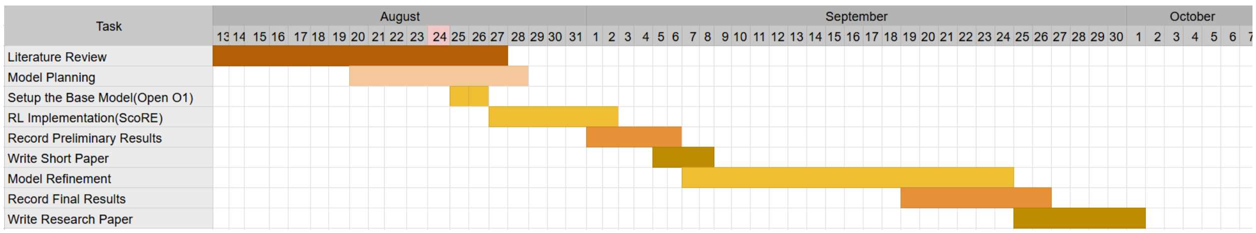
The goal of this project is to improve the reasoning capabilities of the **OpenO1-LLaMA-8B-v0.1** model, one of the two models released under the OpenO1 initiative. This research focuses on leveraging **Reinforcement Learning (RL)** and iterative self-correction methods to achieve incremental but measurable improvements over the baseline model.

The starting point is the **baseline evaluation** of OpenO1-LLaMA-8B-v0.1 across a suite of reasoning and knowledge benchmarks, which demonstrates competitive performance compared to LLaMA3.1-8B-Instruct. Building on this foundation, the project aims to further enhance performance through RL fine-tuning, emphasizing improvements in mathematical reasoning and general problem-solving abilities.

Project Timeline

Phase	Activities
Phase 1: Literature Review	Study existing work on RL for reasoning in LLMs, focusing on RLHF, Expert Iteration, SCoRe, and iterative refinement methods.
Phase 2: Model Planning	Develop a logical methodology based on the literature review to improve base model performance.
Phase 3: Base Model Setup	Set up **OpenO1-LLaMA-8B-v0.1** as the baseline model.
Phase 4: RL Implementation	Implement the SCoRe framework: Stage I – initialize self-correction; Stage II – multi-turn RL with reward shaping to encourage self-correction.
Phase 5: Record Preliminary Results & Short Paper Release	Evaluate the SCoRe model on GSM8K, MATH, MMLU, ARC-C, HellaSwag, and BBH benchmarks. Compare against baseline performance.
Phase 6: Model Refinements	Extend the RL pipeline with iterative self-correction and multi-agent enhancements (MoA and CORY-inspired methods).

Phase 7: Experimental Evaluation	Evaluate improved models on all benchmarks (GSM8K, MATH, MMLU, ARC-C, HellaSwag, BBH) and compare against baseline and SCoRe results.
Phase 8: Research Paper Submission	Document methodology, experiments, results, and analysis in a full research paper.



Methodology Outline

The methodology for improving reasoning performance in **OpenO1-LLaMA-8B-v0.1** is designed around **reinforcement learning (RL), iterative self-correction, and generation-time refinement**. It builds on insights from the literature, starting with **SCoRe-based fine-tuning** and extending to **multi-agent cooperative strategies** and **generation-time correction**.

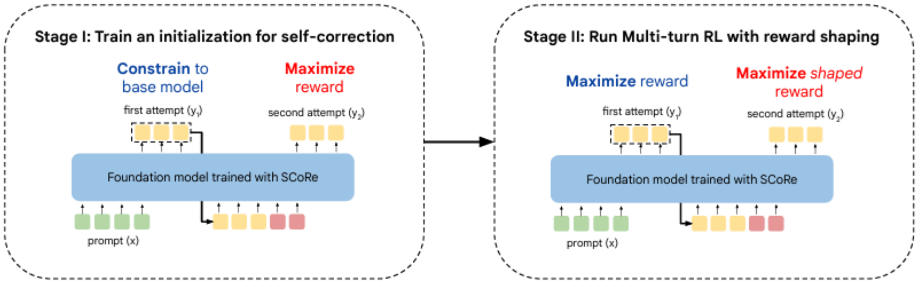
1. Baseline Model Selection

The **OpenO1-LLaMA-8B-v0.1** model is chosen as the baseline due to its strong initial reasoning performance across diverse benchmarks. All subsequent enhancements—including RL fine-tuning and iterative self-correction—are **benchmarked against this baseline** to ensure measurable performance gains.

2. Reinforcement Learning Fine-Tuning (SCoRe)

The first stage of improvement leverages the **SCoRe framework** (Kumar et al., 2024), a multi-turn RL approach designed to train models to **iteratively refine their outputs**.Key insights from SCoRe include:

- **Two-Stage Approach:**



1. **Stage I – Initialization for Self-Correction:**
   - The model is trained to **decouple first- and second-attempt distributions**.
   - The first attempt is constrained to mimic the base model (via KL-divergence), while the second attempt is optimized for high-reward outputs.
   - This ensures the model **does not collapse to trivial solutions** and is prepared for multi-turn self-correction.
2. **Stage II – Multi-Turn RL with Reward Shaping:**
   - Both attempts are jointly optimized.
   - Reward shaping incentivizes **progress in self-correction**, assigning positive bonuses for correcting errors in the second attempt and penalties for regressing from correct to incorrect responses.
   - This encourages learning a **nuanced self-correction strategy** rather than simply optimizing first-attempt responses.

This two-stage SCoRe fine-tuning ensures that the model **learns to self-correct iteratively** and generalizes across new reasoning tasks.

3. Iterative Self-Correction

Building on SCoRe fine-tuning, **iterative self-correction** further improves reasoning by enabling the model to **reflect and refine its answers** through multiple iterations. Two approaches are explored:

1. **Mixture-of-Agents (MoA) Inspired Iteration**

    - **Mechanism:** Deploys multiple instances of the fine-tuned model as agents in a layered architecture.
    - **Process:**
        - First-layer agents generate diverse candidate outputs.
        - Higher-layer agents aggregate and refine outputs using candidates from previous layers.
        - Iteration continues for several cycles to promote **self-correction using prior outputs**.
    - **Benefit:** Enhances reasoning consistency, quality, and collaborative self-reflection.
2. **CORY-Inspired Multi-Agent Cooperative RL**
    - **Mechanism:** Duplicates the model into a **pioneer** and an **observer** agent, promoting cooperative learning.
    - **Process:**
        - Pioneer generates a response.
        - Observer generates a response conditioned on both the input and the pioneer's output.
        - Agents periodically exchange roles, and policies are updated using cooperative RL rewards.
    - **Benefit:** Encourages exploration of diverse strategies, mitigates distribution collapse, and improves policy robustness.

Together, these iterative and cooperative frameworks allow the model to **self-correct earlier errors**, integrate diverse perspectives, and refine outputs more effectively than single-pass fine-tuning.

4. Generation-Time Correction

Given the impracticality of retraining extremely large models for every improvement, **generation-time correction** is applied to enhance output quality without modifying model weights. Two strategies are considered:

- **Generate-then-Rank:** Sample multiple candidate outputs and select the best based on a **critic model or external feedback**. Examples include **DIVERSE, LEVER, and CodeT**.

- **Feedback-Guided Decoding:** Provide **step-level feedback during generation**, enabling intermediate error correction. Critic models can leverage **human feedback, trained verifiers, external metrics, external knowledge, or self-evaluation**, as implemented in **Tree-of-Thought, GRACE, and RAP**.

By combining generation-time correction with iterative self-correction, the model gains the ability to **refine reasoning dynamically during inference**, complementing the improvements achieved through SCoRe-based fine-tuning and multi-agent RL.

5. Evaluation and Selection

- All methods are **benchmarked against the original OpenO1-LLaMA-8B-v0.1 baseline**.

- Iterative self-correction variants (MoA-inspired and CORY-inspired) are compared to identify the **best-performing strategy**.

- Performance metrics include **accuracy, reasoning consistency, error recovery**, and **robustness across multi-turn tasks**.

- Generation-time correction is evaluated to measure **incremental improvements during inference**.

- Target benchmarks include:

    - **GSM8K** – mathematical reasoning with grade-school level problems.

    - **MATH** – advanced mathematical problem solving.

    - **MMLU** – multi-domain knowledge and understanding.

    - **Hellaswag** – commonsense reasoning.

    - **ARC-C** – challenging scientific reasoning questions.

    - **BBH (Big-Bench Hard)** – complex reasoning and general AI capabilities.

Literature Review

Large Language Models (LLMs) have demonstrated remarkable performance across natural language understanding and generation tasks. However, enabling LLMs to **self-correct and refine outputs** remains a significant research challenge. Recent studies have explored approaches spanning **prompting, reinforcement learning (RL), generation-time correction, and multi-agent frameworks**. This review synthesizes key findings relevant to improving reasoning performance in LLMs.

4.1 Prompting for Intrinsic Self-Correction

Naïve prompting for self-correction often **degrades performance** (Huang et al., 2023; Qu et al., 2024; Tyen et al., 2024; Zheng et al., 2024), contradicting earlier claims that simple prompt-based corrections could succeed (Kim et al., 2023; Madaan et al., 2023; Shinn et al., 2023). Failures typically stem from mismatched assumptions, such as the availability of ground-truth answers or reliance on weak initial prompts. For example, in **code self-repair**, even strong LLMs fail to correct errors when only partial feedback is available (Olausson et al., 2023). These studies highlight the **limitations of relying solely on prompting** for intrinsic self-correction.

4.2 RL for LLM Fine-Tuning

Reinforcement Learning from Human Feedback (RLHF) has been widely applied to fine-tune LLMs, aligning outputs with human preferences (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022). RLHF typically trains a reward model to score outputs and applies **policy optimization**, most commonly via Proximal Policy Optimization (PPO). Variants such as **ReST** (Gulcehre et al., 2023), **Reward-Ranked Fine-tuning** (Dong et al., 2023), and **AlpacaFarm** (Dubois et al., 2023) show that **high-reward response fine-tuning** with standard cross-entropy loss can achieve comparable performance.

Beyond prompting, fine-tuning using **revision demonstrations**—either from human annotators (Saunders et al., 2022) or stronger models (Qu et al., 2024; Ye et al., 2023)—has proven effective. Approaches like **SCoRe** (Kumar et al., 2024) leverage multi-turn RL with **reward shaping**, training models to iteratively refine outputs while preventing behavior collapse. Other frameworks, such as **GLoRE** (Havrilla et al., 2024b) and **Self-Correction models** (Welleck et al., 2023; Akyürek et al., 2023; Paul et al., 2023), train separate correction models but introduce additional deployment complexity. Overall, **RL-based fine-tuning for intrinsic self-correction remains a cornerstone strategy**, especially when reward functions can guide model-generated outputs effectively.

4.3 Generation-Time Correction

Given the **impracticality of retraining extremely large LLMs**, generation-time correction has emerged as a practical alternative for improving outputs without modifying model weights. Two main strategies have been explored:

- **Generate-then-Rank:** Multiple candidate outputs are sampled, and the best one is selected using a **critic model** or external feedback. Examples include **DIVERSE** (Li et al., 2023), **LEVER** (Ni et al., 2023), and **CodeT** (Chen et al., 2023).

- **Feedback-Guided Decoding:** Step-level feedback is provided during generation, allowing models to correct **intermediate reasoning steps**. Examples include **Tree-of-Thought** (Yao et al., 2023), **GRACE** (Khalifa et al., 2023), and **RAP** (Hao et al., 2023). Critic models can leverage **human feedback, trained verifiers, external metrics, external knowledge, or self-evaluation** to guide generation efficiently.

Generation-time correction complements training-time fine-tuning by enabling **real-time refinement of outputs**, especially for large or closed-source models where retraining is infeasible.

4.4 Multi-Agent and Iterative Self-Reflection

Recent work explores **multi-agent and iterative frameworks** to improve self-correction and reasoning:

- **Mixture-of-Agents (MoA)** (Wang et al., 2024) employs multiple LLM instances in a **layered architecture**. Early-layer agents generate diverse candidate outputs, while higher-layer agents aggregate and refine responses, allowing **iterative self-correction** using prior outputs and multiple perspectives.

- **Agent-R** (Yuan et al., 2025) introduces **iterative self-training with Monte Carlo Tree Search (MCTS)** to construct correction trajectories. By identifying error steps dynamically within a trajectory, the model can **perform timely revisions**, improving recovery from long-horizon errors.

- **CORY** (Ma et al., 2024) applies **sequential cooperative multi-agent RL**, duplicating the model into a **pioneer and an observer** that periodically exchange roles. This approach promotes cooperative learning, improves **policy robustness**, mitigates distribution collapse, and often outperforms PPO in real-world reasoning tasks.

These frameworks inspire methodologies like **SCoRe-based fine-tuning**, followed by **iterative self-correction** and **multi-agent cooperative RL**, forming the backbone of strategies for improving reasoning and self-correction in LLMs.

References

[1] Open-Source-O1, "Open-O1 Deployment," GitHub, 2025. [Online]. Available: https://github.com/Open-Source-O1/Open-O1/blob/main/Deployment/app.py

[2] A. Havrilla, Y. Du, S. C. Raparthy, C. Nalmpantis, J. Dwivedi-Yu, M. Zhuravinskyi, E. Hambro, S. Sukhbaatar, and R. Raileanu, "Teaching Large Language Models to Reason with Reinforcement Learning," *arXiv preprint arXiv:2408.13296v1*, 2024. [Online]. Available: https://arxiv.org/html/2408.13296v1#bib.bib72

[3] R. Ma, P. Wang, C. Liu, X. Liu, J. Chen, B. Zhang, X. Zhou, N. Du, and J. Li, "S2R: Teaching LLMs to Self-verify and Self-correct via Reinforcement Learning," 2025.

[4] A. Kumar, V. Zhuang, R. Agarwal, Y. Su, J. D. Co-Reyes, A. Singh, K. Baumli, S. Iqbal, C. Bishop, R. Roelofs, L. M. Zhang, K. McKinney, D. Shrivastava, C. Paduraru, G. Tucker, D. Precup, F. Behbahani, and A. Faust, "Training Language Models to Self-Correct via Reinforcement Learning," 2024.

[5] X. Chen, M. Lin, N. Schärli, and D. Zhou, "Teaching Large Language Models to Self Debug," 2023.

[6] L. Pan, M. Saxon, W. Xu, D. Nathani, X. Wang, and W. Y. Wang, "Automatically Correcting Large Language Models: Surveying the landscape of diverse self-correction strategies," 2023.

[7] J. Wang, J. Wang, B. Athiwaratkun, C. Zhang, and J. Zou, "Mixture-of-Agents Enhances Large Language Model Capabilities," 2024.

[8] H. Ma, T. Hu, Z. Pu, B. Liu, X. Ai, Y. Liang, and M. Chen, "Coevolving with the Other You: Fine-Tuning LLM with Sequential Cooperative Multi-Agent Reinforcement Learning," in *Advances in Neural Information Processing Systems 37 (NeurIPS 2024) Main Conference Track*, 2024.

[9] A. Havrilla, S. Raparthy, C. Nalmpantis, J. Dwivedi-Yu, M. Zhuravinskyi, E. Hambro, and R. Raileanu, "GLoRe: When, Where, and How to Improve LLM Reasoning via Global and Local Refinements," *arXiv*, Feb. 2024, revised Jun. 2024. [Online]. Available: https://arxiv.org/abs/2402.XXXX.

[10] C. Gulcehre, T. Le Paine, S. Srinivasan, K. Konyushkova, L. Weerts, A. Sharma, A. Siddhant, A. Ahern, M. Wang, C. Gu, W. Macherey, A. Doucet, O. Firat, and N. de Freitas, "Reinforced Self-Training (ReST) for Language Modeling," *arXiv*, Aug. 2023, revised Aug. 2023. [Online]. Available: https://arxiv.org/abs/2308.XXXX