

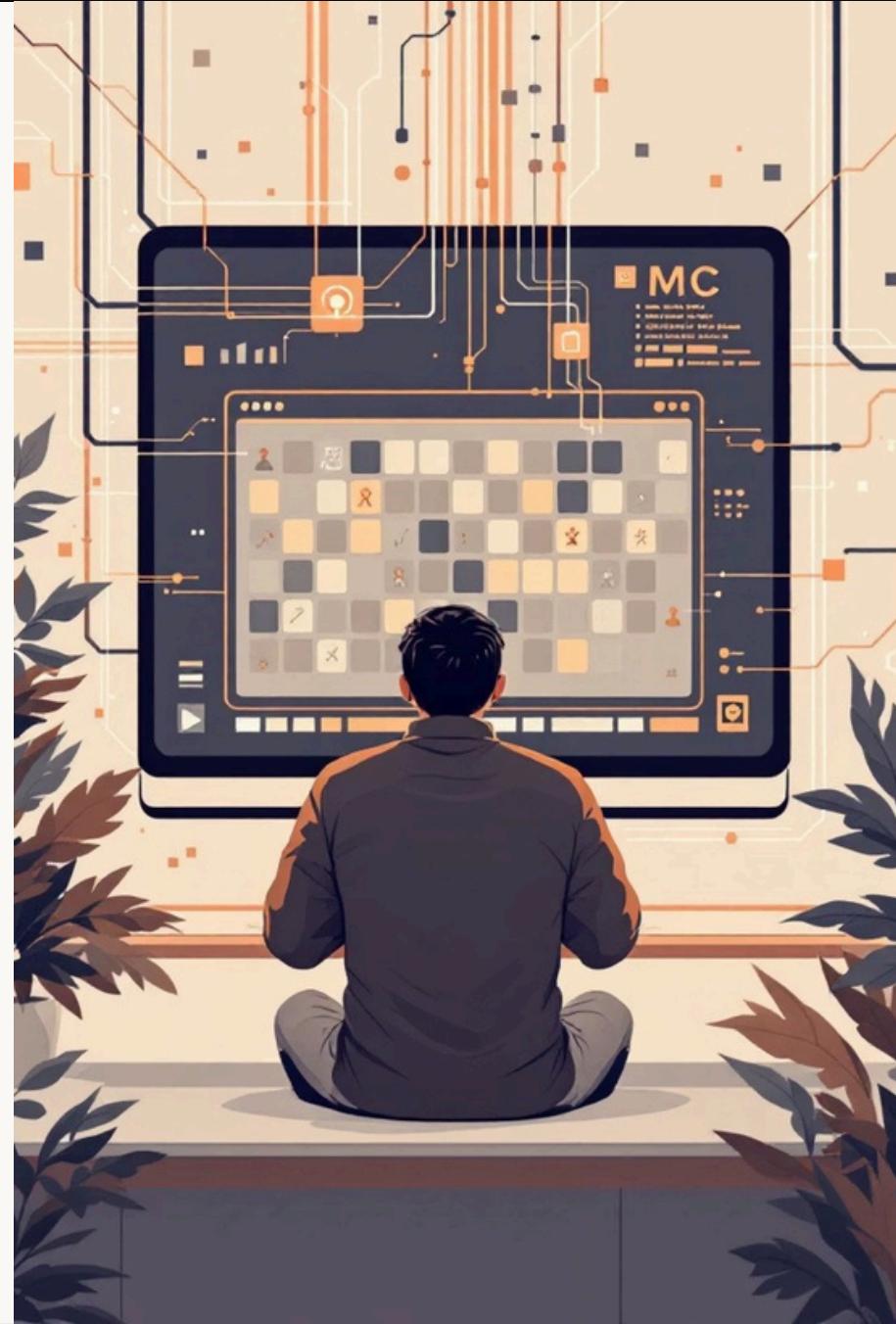
Team Conquer

Team Number 73

- Ritvik Shrivastav
- Prashasth Immanuel
- Kamal Enoch
- Bharat Reddy

GRPO Fine-Tuning of Llama-3.1-8B using Unsloth

Accelerated by AMD MI300X



Problem Statement: Mastering Minesweeper

Our core objective is to engineer an LLM capable of autonomously playing Minesweeper, specifically within the constraints of a Track-2 RL competition.



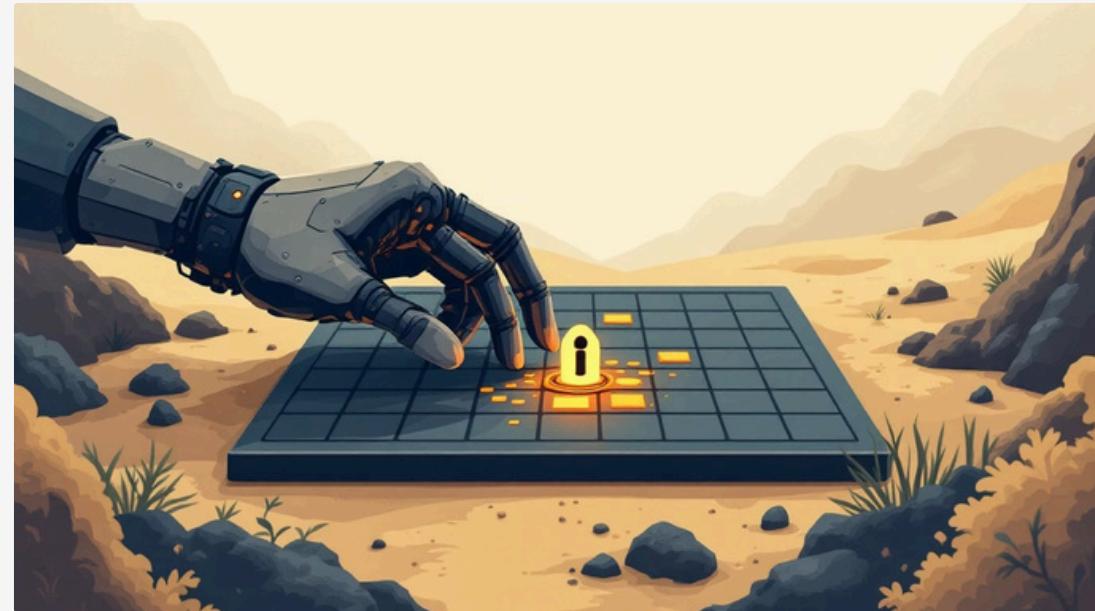
Structured Action Output

The LLM must generate actions in a precise JSON format for system interpretation.



Optimize Full-Game Win Rate

Success is measured by complete board wins, not merely survival or partial clearance.



System Architecture: A Powerful Stack

We leverage a robust, accelerated pipeline to tackle the complex task of RL-driven LLM training for game environments.



Base Model: Llama-3.1-8B

A strong foundation for general language understanding and generation.



Fine-Tuning: GRPO

Utilizing Group Relative Policy Optimization for enhanced stability and exploration.



Acceleration: Unislot

4-bit quantization and LoRA techniques for efficient fine-tuning.



Hardware: AMD MI300X GPUs

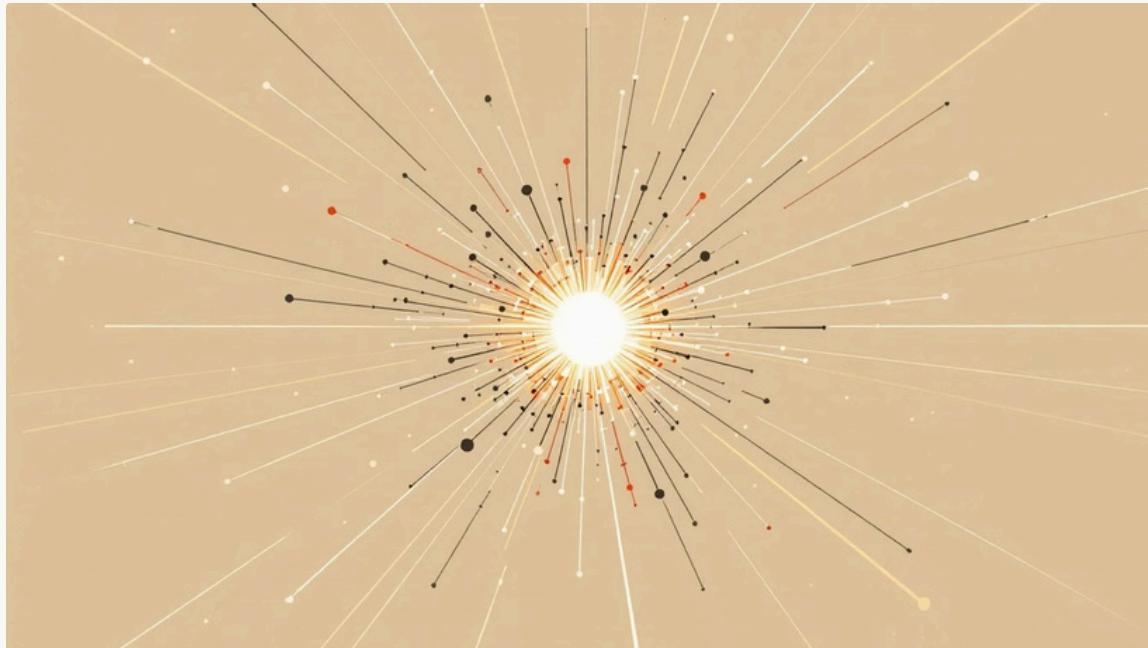
High-performance computing power for rapid experimentation and training.



Custom Environment

Tailored Minesweeper simulator for targeted RL training.

Why GRPO? Enhancing Stability and Exploration



- Multiple Completions

Generates several potential actions per board state, providing a richer learning signal.

- Relative Reward Comparison

Compares rewards within a generated group, improving policy gradient stability.

- Encourages Exploration

Reward variance within groups naturally fosters a more exploratory policy.

Group Relative Policy Optimization (GRPO) offers significant advantages over standard PPO methods, particularly in environments requiring nuanced decision-making.

Reward Shaping Strategy: Guiding the Agent

Carefully crafted reward signals are crucial for efficient learning in sparse-reward environments like Minesweeper.

Valid JSON Output

Direct reward for adhering to the required output format.

Legal Move Reward

Positive reinforcement for making valid game moves.

Safe Reveal Bonus

Reward for uncovering safe cells, progressing towards the goal.

Penalties

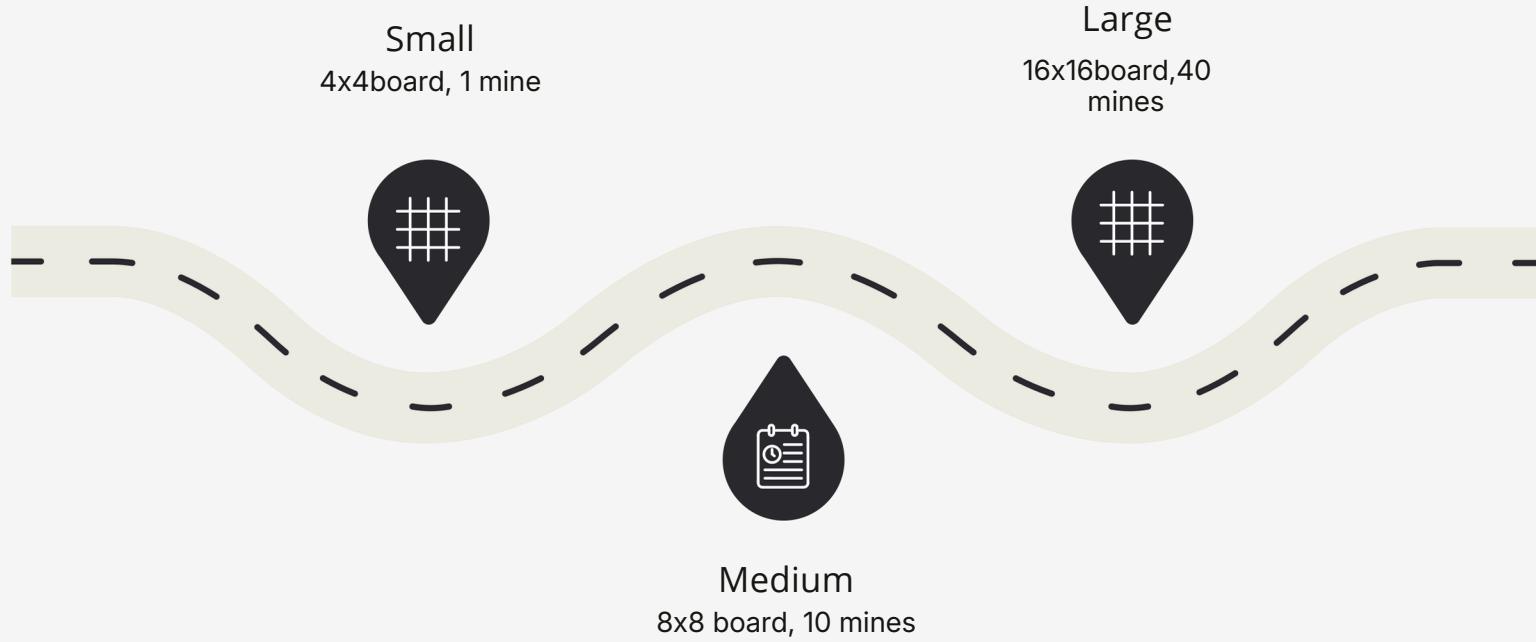
Negative rewards for bombs, invalid actions, or redundant moves.

Terminal Win Bonus

Significant positive reward upon successfully clearing the entire board.

Curriculum Learning: From Simple to Complex

A progressive learning strategy ensures the LLM builds foundational skills before tackling more intricate scenarios.



- Start with small boards (e.g., 4x4 with 1 mine) to establish basic policy.
- Gradually increase board size and mine density as performance improves.
- Ensures early positive learning signals, preventing policy collapse.
- Reduces the chances of the agent resorting to random guessing on complex boards.

Evaluation Methodology: Measuring Success

Rigorous evaluation ensures accurate assessment of the trained agent's performance and generalization capabilities.

1

Deterministic Inference

Utilizing greedy decoding to ensure consistent and reproducible action selection.

2

Full-Episode Gameplay

Evaluating the agent across entire games, from start to finish, for comprehensive results.

3

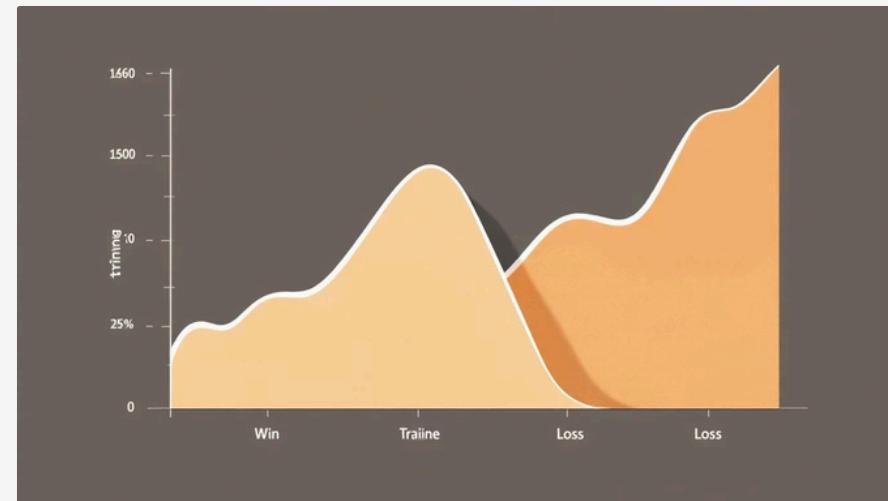
Key Metrics

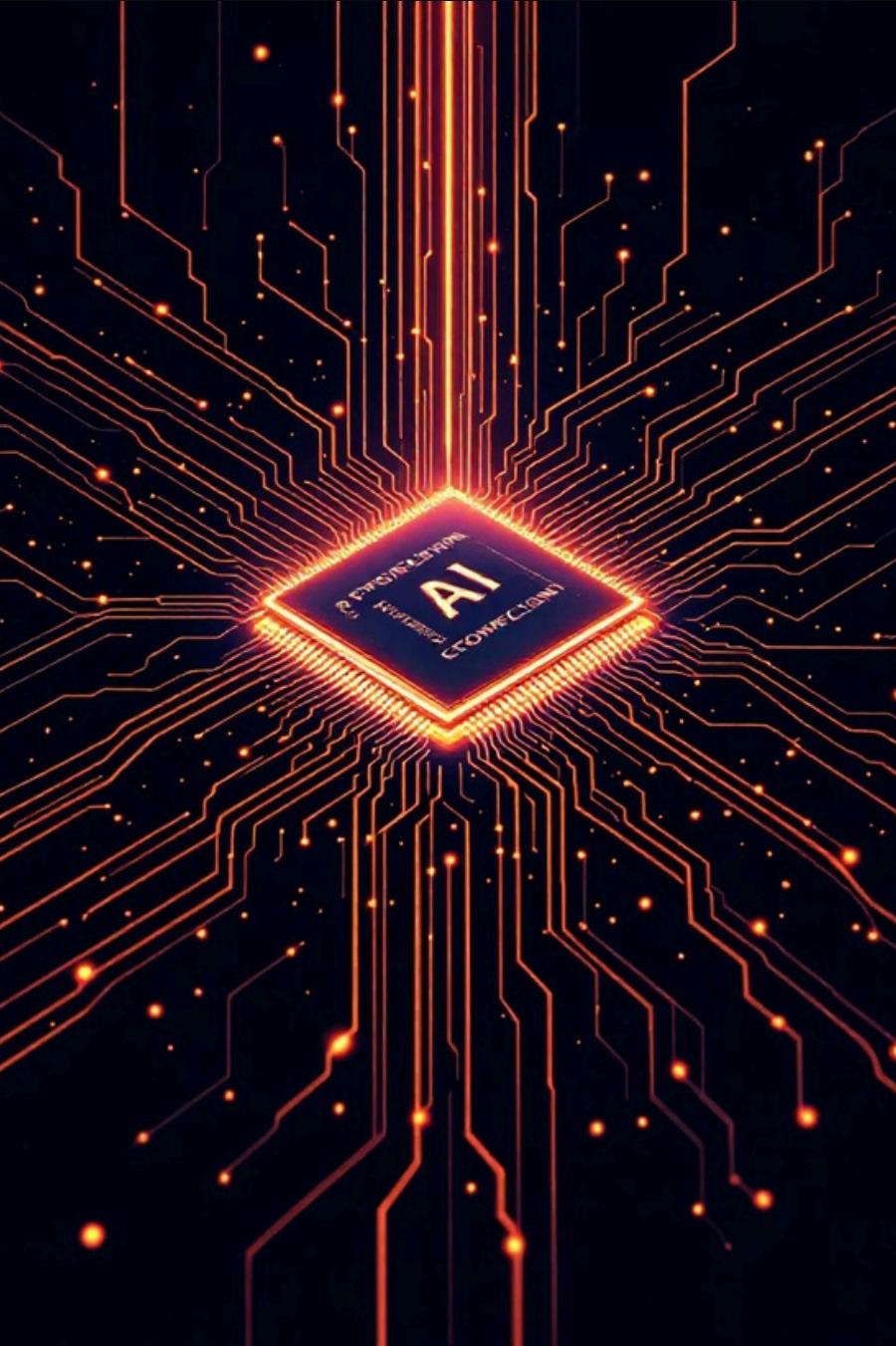
Tracking Win Percentage, Loss Percentage, and Invalid Action Percentage.

4

Periodic Evaluation

Regular checks during training to monitor progress and identify potential issues.





Key Learnings & Insights

Our experiments yielded valuable insights into the effective training of RL-driven LLM agents for game environments.

Reward Shaping is Paramount

Effective reward design significantly outweighs marginal gains from increased model size.

Exploration is Critical

Mechanisms promoting exploration are essential for discovering optimal policies.

Curriculum Accelerates Convergence

Structured curriculum learning speeds up the training process.

Unsloth + MI300X Power

This combination enables rapid iteration and experimentation, crucial for RL development.