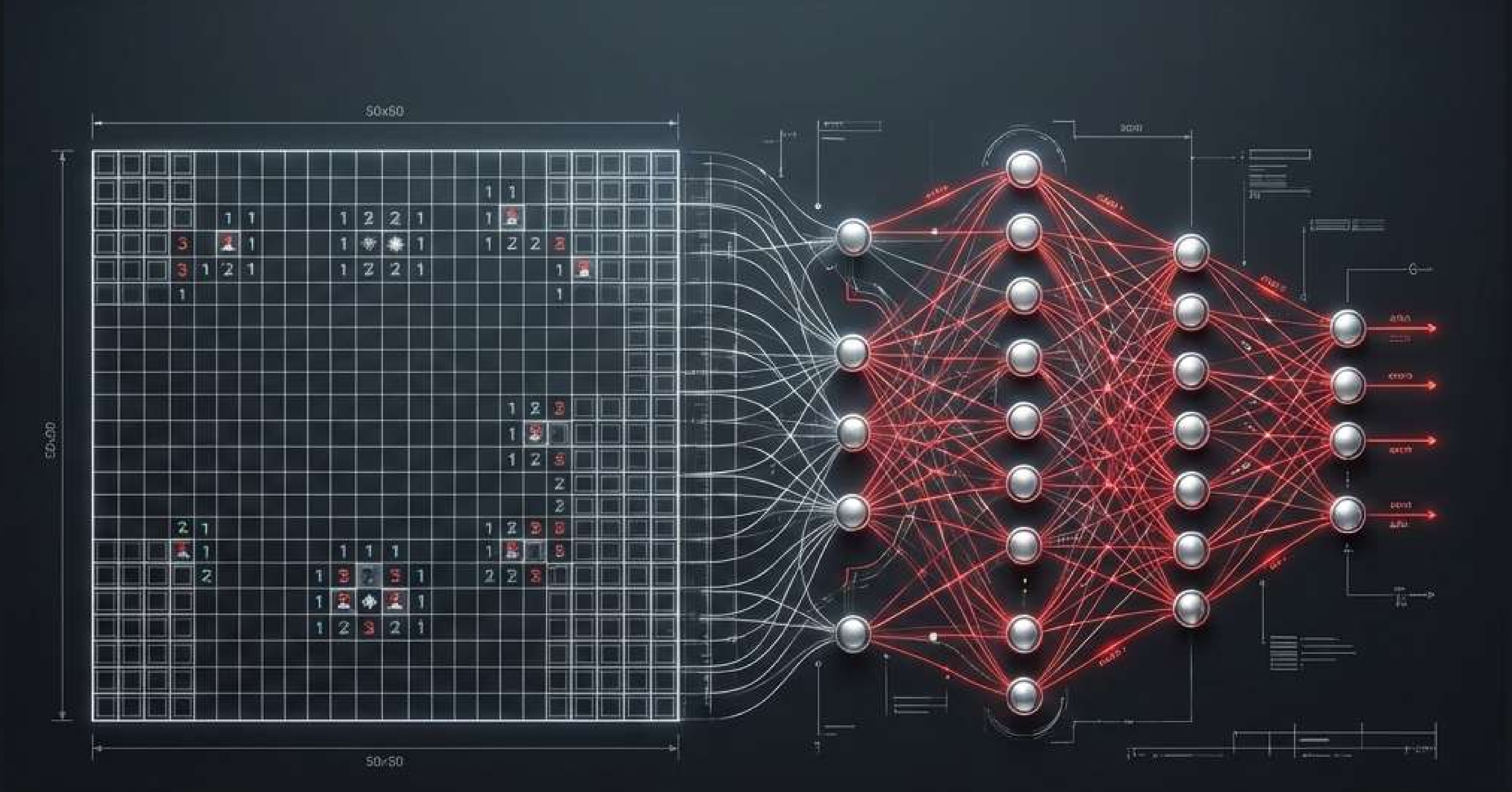


AMD Pervasive AI Developer Contest

Gaming the Models: The Minesweeper LLM Agent

Achieving Expert-Level Logic via SFT & GRPO on AMD MI300x

Team Spambots

Members: Shivam Dwivedi, Janesh Kapoor, Tushar Chandra, Varsha Bhaskar

High-Velocity Engineering on the MI300x

90%

Token Reduction via Compact Board Representation

264+

Games Simulated across 32 Model Configurations

238

High Score (Winning Qwen2.5-14B Configuration)

Insight: SFT taught the rules; GRPO taught the strategy.

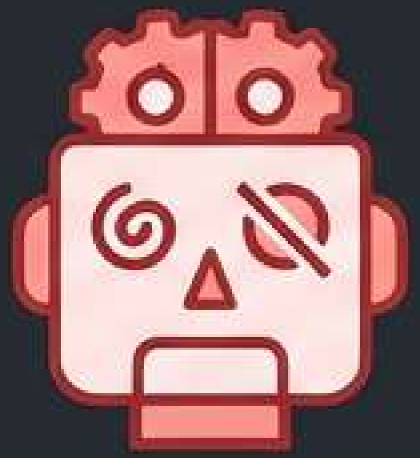
Tech Stack: Unsloth (2x speedup) | TRL Library | AMD Instinct MI300x (256GB HBM3)

Engineering Editorial

The Challenge: Why LLMs Fail at Minesweeper

The Input

The Failure



// 01. Spatial Reasoning: Converting 1D text to 2D adjacency is non-trivial.

// 02. State Tracking: Must distinguish 'Hidden', 'Flagged', and 'Numbered' states.

// 03. Token Economy: Standard formats exhaust context immediately.

Baseline Performance: GPT-4 Win Rate ~0% (NAACL 2024)

Innovation I: Cracking the Context Window

Standard JSON (50x50 Board) = ~8,000 Tokens

```
{"row": 0, "col": 0, "val": "unknown"},
{"row": 0, "col": 1, "val": "1"},
{"row": 0, "col": 2, "val": "unknown"},
{"row": 1, "col": 0, "val": "F"},
{"row": 1, "col": 1, "val": "2"},
{"row": 1, "col": 2, "val": "unknown"},
{"row": 0, "col": 0, "val": "F"},
{"row": 0, "col": 1, "val": "2"},
{"row": 0, "col": 0, "val": "F"},
{"row": 0, "col": 1, "val": "2"},
{"row": 0, "col": 2, "val": "unknown"},
{"row": 0, "col": 2, "val": "unknown"},
{"row": 1, "col": 0, "val": "F"},
{"row": 1, "col": 1, "val": "2"},
{"row": 1, "col": 2, "val": "2"},
{"row": 1, "col": 3, "val": "unknown"},
{"row": 1, "col": 0, "val": "F"},
```

Compact Format (50x50 Board) = ~800 Tokens

```
|...1.F....|2..F1...|
|...F3.2...|1..1F...|
|...1.F....|2..F1...|
|...1.F....|2..F1...|
|...1.F....|2..F1...|
|...F3.2...|1..1F...|
|...1.F....|2..F1...|
|...F3.2...|1..1F...|
|...F3.2...|1..1F...|
|...1.F....|2..F1...|
|...1.F....|2..F1...|
|...F3.2...|1..1F...|
|...F3.2...|1..1F...|
|...1.F....|2..F1...|
|...1.F....|2..F1...|
|...1.F....|2..1F...|
```

Impact: Enabled full 50x50 board training within MI300x context limits.

Engineering Editorial

Innovation II: The Expert Data Engine

Raw Board State → **Constraint Solver** (Python) → **Logical Labels** → **Training Data**

Constraint Solver (Python): Propagates constraints & subset analysis.

Logical Labels: Distinguishes "Lucky Guess" (+10) vs "Logic" (+15).

Training Data: 100% Correct SFT Samples.

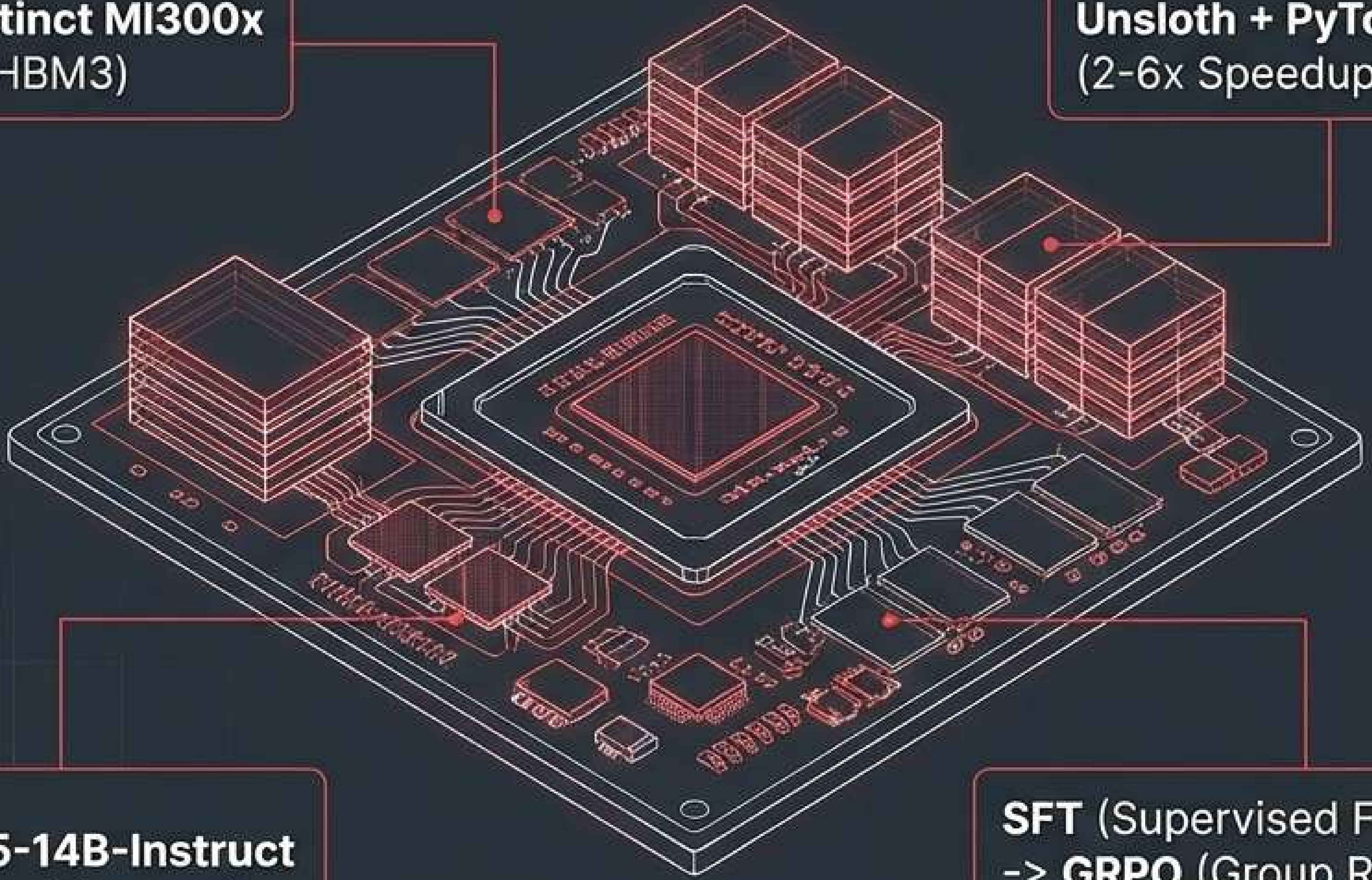
Solver Stats	
Solver Win Rate (50x50):	56%
Moves per Game:	2,200+
Logic Verification:	100%

Architecture & Hardware Strategy

Engineering Editorial

AMD Instinct MI300x
(256GB HBM3)

Unsloth + PyTorch + TRL
(2-6x Speedup)



Qwen2.5-14B-Instruct
(Dense Architecture)

SFT (Supervised Fine-Tuning) -> **GRPO** (Group Relative Policy Optimization)

Why Qwen?

Dense 14B parameters offered superior reasoning depth compared to MoE architectures (GPT-OSS-20B) for this task.

Engineering Editorial

Training Phase 1: Supervised Fine-Tuning (SFT)

Training Curriculum

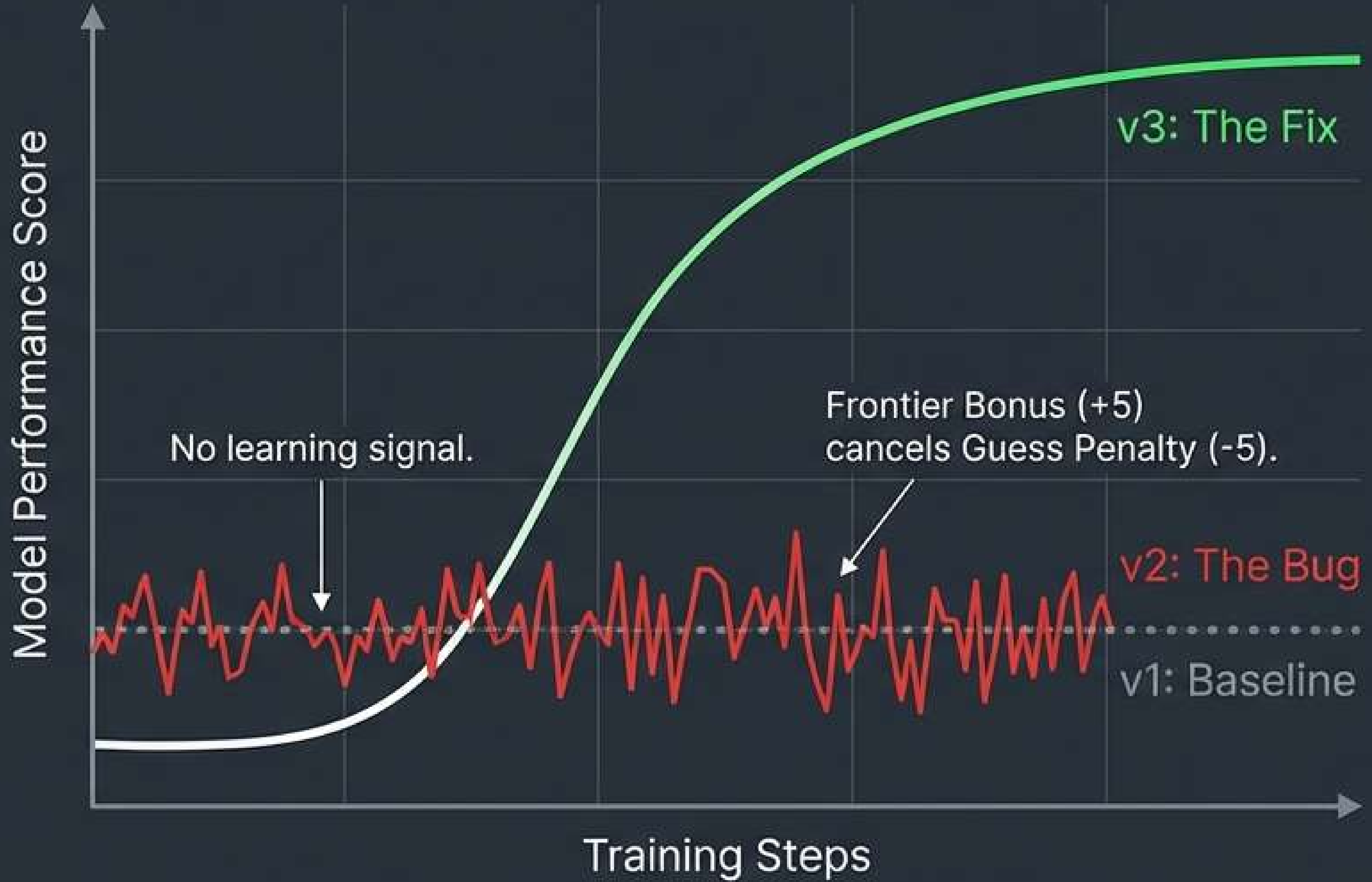
20%	50%	30%
Early Game (Open Areas)	Mid Game (Critical Reasoning)	Late Game (Endgame Logic)

- **Goal:** Teach JSON syntax and valid move rules.
- **Dataset:** 15,000 samples (Square, Rectangular, Tall variants).
- **Hyperparameters:** LoRA Rank 64, Alpha 128.

Result

Model plays legally but lacks winning strategy. It mimics, but does not yet think.

Training Phase 2: The Reward Engineering Loop



The Fix Details

- Random Guess: -15 (Harsh Penalty)
- Logical Deduction: +30 (Massive Reward)
- Mine Hit: -100 (Immediate Fail)

Asymmetric penalties forced the model to value logic over luck.

Prompt Engineering: 6 Strategies to Enforce Logic

V1: Simple Instruction

Umple the Instruction, simple Instruction.

V2: Constraint Logic Focus

Constraint logic stretain Logic format targets.

V3: Aggressive 'DO NOT' Warnings

Aggressive syndras or "DO NOT" warnings.

Winner: Explicitly listing valid targets removed coordinate hallucinations.

V4: Step-by-Step Verification

Diarmentane Step-by-step coordinate levels.

V5: Annotated Board

Enumerates valid targets.

V6: CoT Self-Correction

Selfrvcition targets remove coordinate hallucinations.

Evaluation Methodology

	Strategy V1	Strategy V2	Strategy V3	Strategy V4	Strategy V5	Strategy V6
Base Qwen	✓	✓	✓	✓	☑	✓
GPT-OSS-20b	✓	✓	✓	✓	☑	✓
Qwen Phase 1 (SFT)	☑	☑	☑	☑	☑	☑
Qwen Phase 2 (GRPO)	☑	☑	☑	☑	☑	☑

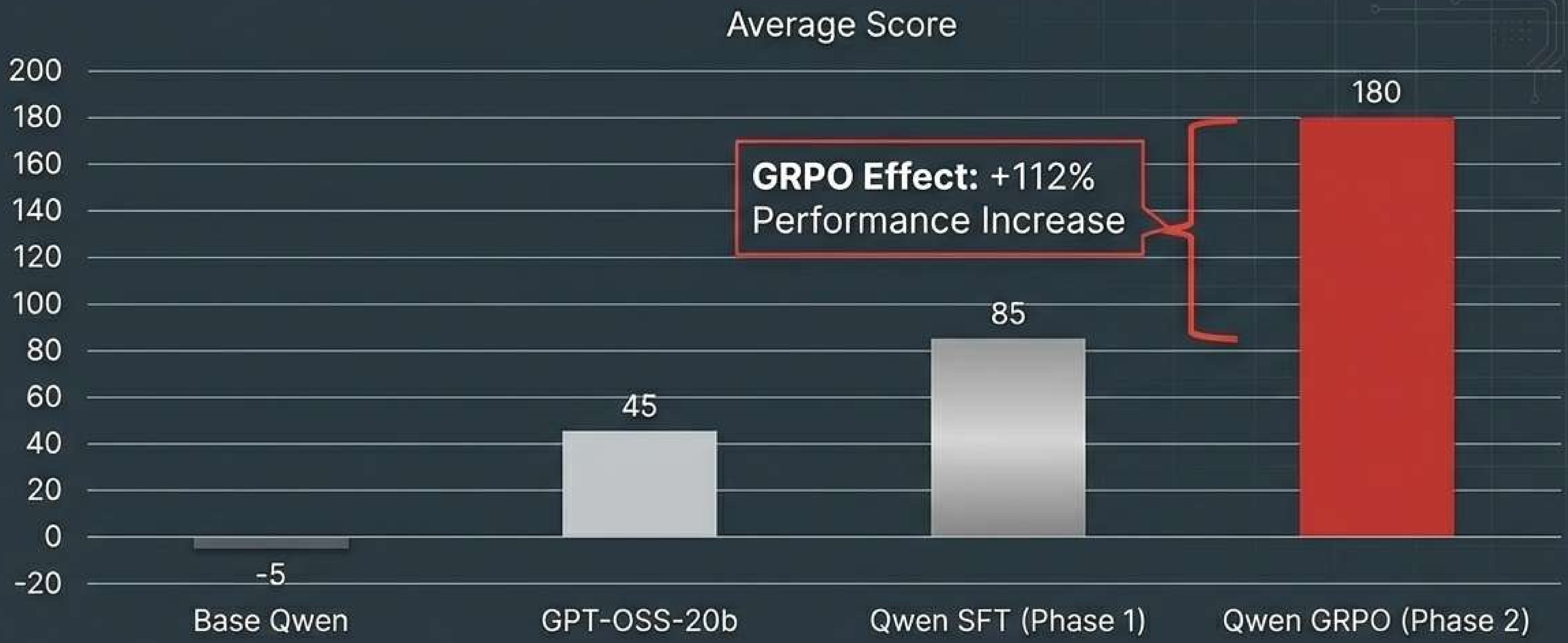
32
Configurations

264+
Full Games Played

Metric:
Cumulative Score

Rigorous testing to isolate the impact of architecture vs. prompting.

Results: The Impact of SFT & GRPO



The model needs to play to learn, not just read.

Winning Configuration: Qwen GRPO + Prompt V5 Qwen GRPO

By removing ambiguity in coordinate mapping, Strategy 5 allowed the model to focus 100% of compute on logic.

Inside the Winning Output

```json
{
  "think": "Cell (4,5) is a '2'. It touches 1 flag and
  1 unknown. 2-1=1. The unknown must be a mine.",
  "action": "flag",
  "row": 4,
  "col": 6
}
```

Constraint Reasoning acts as a logic buffer.

Commitment happens only after reasoning.

Mechanism inspired by "Think Inside the JSON" (arXiv:2502.14905).

Engineering Lessons & Post-Mortem

✅ What Worked

- **Compact Representation** is non-negotiable for 50x50 boards.
- **SFT + GRPO**: SFT stabilizes syntax, GRPO optimizes strategy.
- **Asymmetric Reward Penalties** (v3).

❌ What Failed

- **MoE Architectures**: GPT-OSS-20b struggled with deep reasoning compared to dense Qwen.
- **Frontier Bonuses**: Cancelled out random guess penalties in RL.
- **Standard JSON**: Immediate context overflow.

The Future of Reasoning Agents

Team Spambots demonstrated that LLMs can solve complex spatial reasoning tasks through efficient representation and rigorous reward engineering.

“We didn’t just teach a model to play Minesweeper; we taught it to think.”

Powered by

AMD Instinct MI300x

Team Spambots