Reasoning Model for No-Limit Hold’em Poker

Project Overview

- Develop a **reasoning model** optimized for poker.
- Goal: **Maximize Expected Value (EV)** through **strategic and adaptive play**.
- Applications in **game theory, economic modeling, and negotiation**.

1. Success Metrics

✅ Expected Value per Hand (EV)
✅ Win Rate vs. Opponents
✅ Unpredictability & Robustness
✅ Computational Efficiency

Constraints

- **Computational Limits** – Real-time decision-making is required.
- **Data Availability** – Must gather diverse hand histories.

2. Risks and Challenges

⚠️ **Data Quality** – Hands may end without revealing all cards.

⚠️ **Combinatorial Explosion** – Over **56 billion** hand possibilities per player.

Prior Work

- **PokerBench** – LLMs trained to play professional poker.
- **PokerGPT** – Lightweight solver leveraging LLMs.
- **Pluribus** – Near-GTO multiplayer strategies.

3. Technical Approach

- 🔹 **Fine-tuning LLM** (DeepSeek R1) for action decisions.
- 🔹 **Self-Play with PyPokerEngine** for iterative training.

🎯 **Goal:** Train model to process poker hand states & choose optimal actions.

4. Mathematical Formulation

Maximize Expected Value (EV)

$$\max \mathbb{E}[\text{Winnings per action}]$$

Minimize Regret

$$\min \sum_{t=1}^{T} (\text{Best Possible Outcome} - \text{Chosen Action Outcome})$$

✅ Constraints: **Bankroll management, time limits, opponent inference.**

5. Algorithm Choice & Justification

🤔 **Why a Reasoning Model?**

✔ More **efficient** and **adaptive** than GTO solvers.

✔ Avoids excessive **computation per hand**.

✔ Can generalize across diverse **poker hands & scenarios**.

6. Implementation Strategy

Using PyTorch

1️⃣ **Fine-Tuning Model** – Train with pot odds, hand strength, & opponent tendencies.
2️⃣ **PyPokerEngine Integration** – Simulate hands and update parameters via reward signals.

7. Validation Methods

🔎 Comparisons to GTO Strategies

🃏 Testing Against Bots & Humans

📊 Measuring Win Rate & Decision Accuracy

8. Resource Requirements

💻 **GPU/Cloud Compute** for model training.

📂 **Historical Hand Histories** for training data.

⏳ **Time & Budget** – Need efficient real-time inference.

9. Initial Results

❌ **GPT-2: 0% Accuracy** – Could not output strategic actions.

✔️ **Larger Models (GPT-4, Reasoning Models)** performed significantly better.

Performance Metrics

- ✅ **Win Rates vs. Baselines**
- ✅ **Decision Accuracy (vs. GTO strategies)**
- ✅ **Consistency Across Game Scenarios**

10. Current Limitations

🚧 **Insufficient Compute** – More powerful models needed.

🚧 **Unconstrained LLM Output** – Must restrict responses to concise actions.

11. Next Steps

🎯 **Restrict LLM Output** – Limit to "check", "fold", "raise 20", etc.

🎯 **Explore More Powerful Models** – DeepSeek-R1, GPT-4, or domain-specific models.

🎯 **Enhance Data Pipeline** – Annotate and systematically train with additional plays.

🎯 **Optimize Efficiency** – Reduce computational load in training & inference.

🎯 **Investigate Modern RL Techniques** – Explore **PPO (Proximal Policy Optimization)** & **GRPO (General Reinforcement Policy Optimization)** for training.

🎯 **Conduct Literature Review** – Study **reasoning models** like **TinyZero** and **DeepSeek** for potential adaptation.

🎯 **Run & Fine-Tune TinyZero Locally** – Load **TinyZero** and apply **RL fine-tuning** using our existing dataset.

12. Open Questions

- ❓ **Beyond Fine-Tuning** – How can RL or hierarchical reasoning improve performance?
- ❓ **Constrained Output** – Best method to limit model responses?
- ❓ **Validating Partial Information** – How to infer hidden opponent cards?

13. Alternative Approaches

- 💡 **Standalone Reasoning Models** (non-LLM)
- 💡 **Pure Reinforcement Learning**
- 💡 **Hybrid Approach** – Mix of GTO solvers & RL

14. Key Learnings

✅ **Small LLMs are unreliable** for structured decision-making.
✅ **GTO models are too computationally expensive** for real-time play.
✅ **Reasoning-based models show promise** for strategic adaptability.

Final Thoughts

🚀 By leveraging reasoning models, self-play, and structured training, we aim to create a powerful No-Limit Hold’em AI.

🎵 Project Playlist: Spotify Link