Events vs. Score-based Learning for Splendor

Optimizing Value Networks and MCTS Integration (AlphaZero-style extension)

Team: Yehao Yan, Qianyun Shen, Xinyan Guo

1) Motivation

Splendor is a turn-based strategy game characterized by resource management and engine building. It has a clear two-phase rhythm: early turns build an engine (tokens + discounts), later turns convert that engine into prestige. Compared with perfect-information games, Splendor also includes partial observability (hidden reserved cards) and stochasticity (deck/card replenishment).

Key challenges:

- **Sparse reward**: prestige points often stay at 0 for the first 10-20 turns, so score-only learning receives delayed feedback.
- **Partial observability**: opponents' hidden reserved cards and unknown future market information complicate state evaluation.
- **Stochasticity**: random card replenishment after purchases (deck order hidden) increases uncertainty.
- **High branching factor**: many legal actions each turn (take tokens, buy, reserve, etc.), hurting pure search.

Why this matters: score-based agents struggle to distinguish a "good 0-point state" (engine-rich) from a "bad 0-point state", and vanilla Monte Carlo Tree Search (MCTS) can become short-sighted under stochastic branching.

2) Core idea & hypothesis

We compare **score-based** reinforcement learning with **event-based** reinforcement learning. Instead of rewarding only point gains, we define a small set of meaningful in-game events (e.g., buying key cards, efficient reservation, engine milestones) to densify feedback.

Hypothesis: event-based signals yield faster learning and higher win rate, but may bias toward short-term gains; planning with MCTS guided by a value/policy network can restore long-horizon reasoning.

3) Methods

A. Baseline (score-based RL)

- Input: state vector (tokens, discounts, visible cards, nobles, etc.).
- Target: value network trained from score-based returns.

B. Proposed (events-based RL)

- Input: state vector + **event vector** (engine-building + tactical progress).
- Target: event-value / shaped return; compare convergence and final strength.

C. Planning extension (MCTS + value networks)

- Value network evaluates leaf states in MCTS (Statistical Forward Planning).
- Optional AlphaZero-style upgrade: policy + value guiding search.

4) Evaluation plan

Win-rate benchmark

- Target: events-based agent achieves **> 60%** win rate vs score-based baseline.
- 500 matches: 250 first player + 250 second player to reduce first-move bias.

Ablations

- Score-only: state vector.
- Events: state + event vector (same network capacity).
- With vs without MCTS integration.

5) Compute budget & feasibility

- Hardware: RTX 4090 GPU + 16-core / 32-thread CPU.
- Feasible overnight training; supports multiple ablations.

6) Deliverables

- Reproducible code: environment/simulator + training + evaluation scripts.
- Models: score-based, events-based, and MCTS-integrated variants.

References

- Mnih et al. (2015). Human-level control through deep reinforcement learning. *Nature*.
- Bravi & Lucas (2020). Rinascimento: using event-value functions for playing Splendor.