RUTGERS

Danrui Li and Sen Zhang are looking for internships in Summer 2026 and full-time position in 2027!

Danrui Li | Sen Zhang

Cardiverse: Harnessing LLMs for Novel Card Game Prototyping

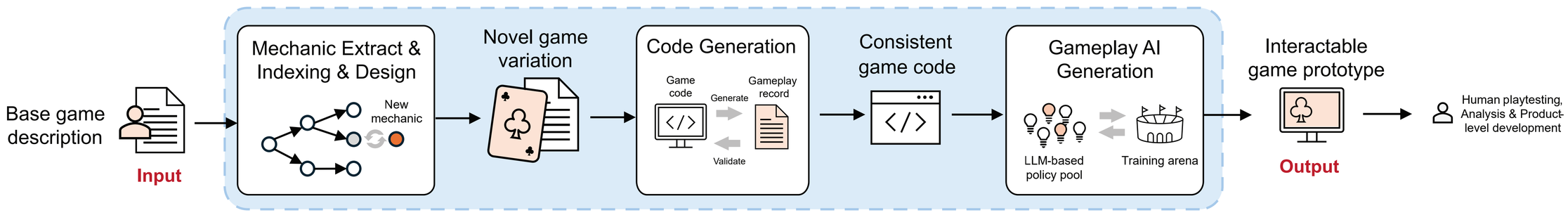
Danrui Li [1], Sen Zhang [1], Sam S. Sohn [1], Kaidong Hu [1], Muhammad Usman[2], Mubbasir Kapadia [1,3]

[1] Rutgers Univeristy [2] Ontario Tech University [3] Roblox

Find code repo and LLM card game arena at

https://github.com/danruili/Cardiverse

An LLM-based framework for card game prototyping.



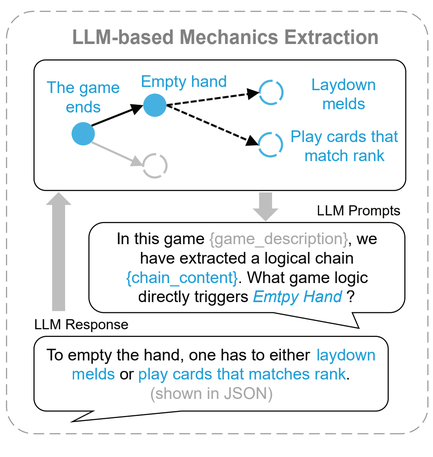
Challenge 1: Mechanics Novelty

Problem

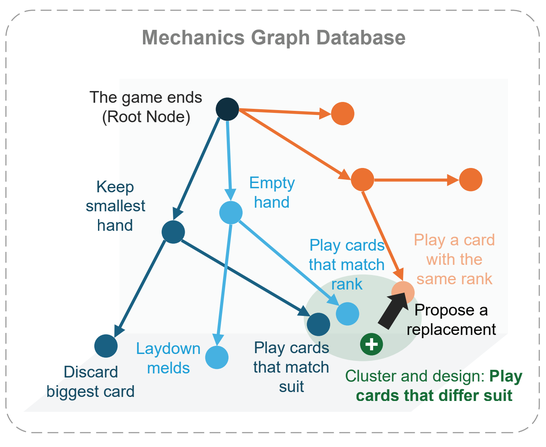
LLM may output game mechanics that is too similar to existing games.

Solution

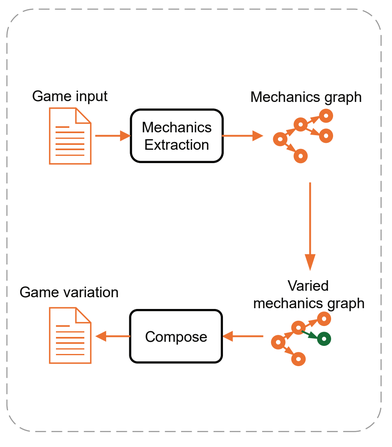
Use game mechanic graphs to enable informed, globally novel game mechanic design.



- Use graphs to represent game mechanic dependencies.
- Use LLMs to convert game descriptions into mechanic graphs.
- LLM expands nodes in BFS style.



- Group mechanics by meaning & depth using embeddings.
- LLM summarizes each cluster, adds creative variants.



- Given a game description, extract its mechanics, map them to clusters and find the ones that occur most frequently across the database.
- Replaces the frequent mechanics with newly designed ones from the same cluster

Metric: Quantiled Max Similarity at p

- The pth percentile of the closest cosine similarity scores between the game variations and each game in the database or in all other variations.
- How close each new game is to those in the database

	p=0.75	p=0.50	p=0.25
To database			
Vanilla	88.7± 5.8 †	87.2± 6.3	83.8± 8.2
Prpt.Bdr.	88.3± 4.9 †	85.8± 4.5	81.1± 3.9 †
Ours	87.4± 4.5	84.8± 4.3	80.6± 3.7 †
Within generated results			
Vanilla	95.4± 0.8	92.0± 4.0	86.7± 5.9
Prpt.Bdr.	91.0± 2.1	85.5± 2.1 †	79.5± 2.5
Ours	89.6± 2.4	85.4± 2.8 †	81.3± 2.8

Ours yields lower similarity to known games (more originality) while preserving diversity among variations.

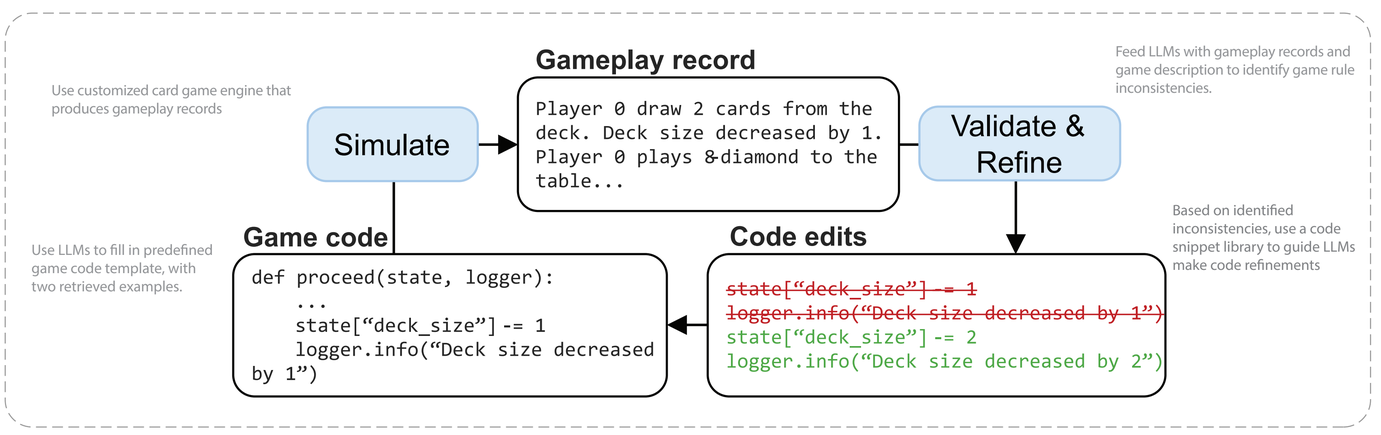
Challenge 2: Code Consistency

Problem

LLM-generated code may deviate from the game mechanics description text.

Solution

Gameplay-reflective coding agent evaluates game consistency via gameplay records.



	Succ ↑	Exec ↑	ECon ↑	PCon ↑
Ours	27/29	100.0± 0	0.55± 0.04	8.99± 1.56
-val	27/29	99.9± 3.9	0.56± 0.03 †	7.69± 1.96
-val⊕	29/29	96.3± 9.4	0.56± 0.04	8.90± 1.74 †

(a) Common Games

	Succ ↑	Exec ↑	ECon ↑	PCon ↑
Ours	25/28	100.0± 0	0.52± 0.05	9.20± 1.08
-val	25/28	98.3± 4.6	0.52± 0.04	8.66± 1.60
-val⊕	28/28	99.3± 2.0	0.54± 0.05	9.08± 1.29 †

(b) Varied Games

- 100% executable game code.
- Achieve better consistency than ablated versions.

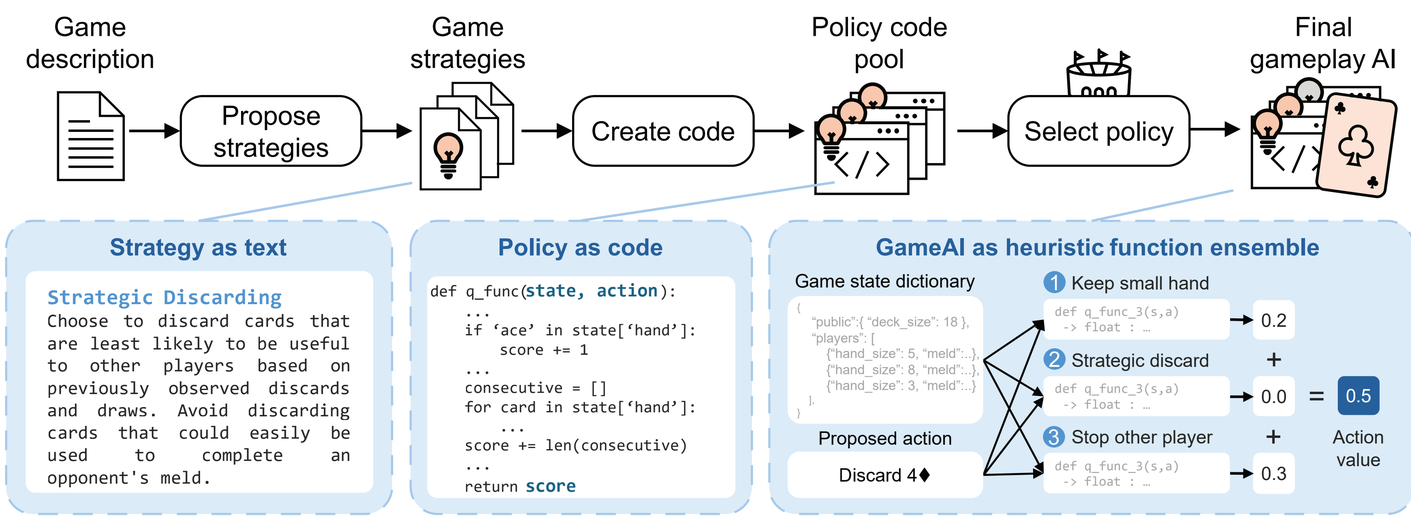
Challenge 3: GameAI Scalability

Problem

LLM-based agents are expensive to run and traditional RL requires long training time + strong hardware

Solution

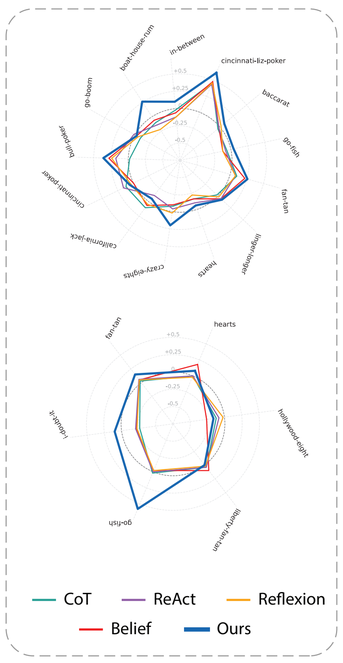
A LLM-based pipeline to create an ensemble of heuristic action-value functions, which doesn't require LLM calls in deployment.



Given the text description of card game, use LLMs to propose a set of game strategies.

Use LLMs to convert game strategies into Python function code, returning a score given the current game state and proposed action.

Iteratively optimize the selection of functions, evaluating each heuristic's win rate, adding the most effective ones step by step.



Our method beat CoT, ReAct, Reflexion, and BeliefAgent in the win rate advantage to random players.