

Zero-Shot Game Abstraction: LLM-Guided Information Abstraction for Imperfect-Information Games

Anonymous ACL submission

Abstract

Information abstraction is essential for scaling game-solving algorithms to large imperfect-information games, yet constructing effective abstractions typically requires domain-specific evaluators such as hand-strength calculators or equity estimators. We propose an *Abstraction Agent*, a zero-shot pipeline that uses a large language model (LLM) to discover continuous strategic features from a natural-language game description, score private states on these features, and cluster them into abstraction buckets—without any game-specific evaluators, training data, or game-tree traversal during abstraction construction. The pipeline operates in four phases: feature discovery via constrained prompting with calibration anchors, batched private-state scoring, correlation-based feature selection, and k -means clustering. On two heads-up no-limit Texas hold’em (HUNL) turn endgames from Libratus subgames, the resulting abstractions reduce lifted-strategy exploitability by 33% relative to the expected hand strength (EHS) baseline across 2 granularity levels. We further show qualitatively that the same pipeline produces strategically coherent abstractions in Pot-Limit Omaha, HUNL preflop/flop, and Riichi Mahjong. From an NLP perspective, our work demonstrates *structured knowledge elicitation*: converting implicit strategic knowledge in LLM parameters into explicit numerical features suitable for downstream algorithmic computation. We will release the full pipeline code upon publication.

1 Introduction

Large language models (LLMs) have shown strong capabilities in reasoning, planning, and strategic thinking across diverse domains. A growing body of work explores LLMs as game-playing agents, yet their performance in complex strategic games remains limited by imprecise probability estimation and inconsistent decision-making. We propose using LLMs not as players, but as *feature engineers*

that identify the strategic dimensions along which game states should be compared.

In imperfect-information games, where players must act under uncertainty about opponents’ private states, *information abstraction* is essential for computational tractability. Abstraction groups strategically similar private states into buckets, enabling solvers like counterfactual regret minimization (CFR) to operate on manageable game sizes (Zinkevich et al., 2007). The quality of this grouping is a major determinant of solution quality: merging states with different strategic roles introduces exploitability.

Constructing effective abstractions has traditionally required domain-specific evaluators. In poker, hands are clustered by equity or expected hand strength (EHS) (Gilpin and Sandholm, 2007), metrics that require hand evaluators, Monte Carlo roll-outs, or expert-designed features (Johanson, 2013). Recent advances such as EVPA (Li and Huang, 2025) and WEVA (Li and Huang, 2026a) reduce the dependency on neural training or handcrafted features, but both still require running CFR on the full game tree, which is a prerequisite that is computationally expensive for large games and unavailable when no solver implementation exists. More broadly, many existing methods rely on at least one of: (i) game-specific evaluators, (ii) extensive training data from game-tree traversal, or (iii) closed-source implementations that make reproduction difficult (Brown and Sandholm, 2018).

This motivates our central question: *can an LLM, given only a natural-language description of the game rules, construct useful information abstractions without game-tree traversal during abstraction construction?*

We provide initial evidence that this is possible by proposing an *Abstraction Agent*, a zero-shot pipeline that leverages LLM reasoning to discover and apply strategic features for information abstraction. By “zero-shot” we mean that the pipeline uses

no game-specific training data, rollout evaluator, or game-tree traversal; it receives only a natural-language rules description and textual renderings of private states. The agent operates in four phases: (i) **Feature Discovery**, the LLM proposes continuous (real-valued in $[0, 1]$) strategic features with calibration anchors from a game description; (ii) **Private-State Scoring**, each holding is scored on these features via batched LLM calls; (iii) **Feature Selection**, redundant features are filtered by correlation analysis; (iv) **Clustering**, the resulting feature vectors are partitioned with k -means. The pipeline avoids domain-specific evaluators and manually designed scoring functions, though it still requires a textual game description and rendering functions for private states.

The key insight is that LLMs trained on broad text corpora encode useful strategic regularities about well-studied games. Concepts like “nut potential,” “blocker value,” and “draw vulnerability” exist in the LLM’s knowledge as natural-language descriptions of strategic dimensions. Our pipeline elicits this knowledge as *structured, continuous features* suitable for downstream computation. We call this process *structured knowledge elicitation*: prompting an LLM to convert implicit domain knowledge into explicit, machine-readable numerical variables that can be consumed by non-neural downstream algorithms.

We evaluate on heads-up no-limit Texas hold’em (HUNL) turn endgames from Libratus subgames (Brown and Sandholm, 2018), a standard benchmark from prior poker-solving work, where the agent’s abstraction achieves 33% lower exploitability than EHS on average across 2 subgames and 2 granularity levels. We further show qualitatively that the same pipeline produces strategically coherent abstractions in Pot-Limit Omaha (16,432 canonical hands), HUNL preflop/flop, and Riichi Mahjong (a tile-based imperfect-information game with substantially different state structure from poker).

Our contributions are:

- **A zero-shot abstraction pipeline.** We propose, to our knowledge, the first LLM-based pipeline that converts a natural-language game description into a multi-dimensional information abstraction without hand evaluators, rollout simulators, or game-tree traversal during abstraction construction.
- **Quantitative gains over standard baselines.**

On two HUNL turn subgames, the agent’s abstraction achieves 33% lower exploitability than EHS across granularity levels $K \in \{20, 50\}$ (where K is the number of buckets), without any domain-specific evaluator.

- **Qualitative cross-game portability.** Qualitative analyses indicate that the same pipeline produces strategically coherent abstractions in PLO4, HUNL preflop/flop, and Riichi Mahjong, discovering game-appropriate features in each domain without modification.

2 Related Work

Information Abstraction in Games. Abstraction is a core technique for scaling imperfect-information game solving. In extensive-form games, information sets group states indistinguishable to a player; abstraction further merges information sets deemed strategically similar (Sandholm, 2015). Early work on poker abstraction used hand rank bucketing (Shi and Littman, 2000; Billings et al., 2003). Modern approaches cluster private hands by expected hand strength (EHS) or equity against a uniform opponent range (Gilpin and Sandholm, 2007; Johanson, 2013; Johanson et al., 2013). Potential-aware abstractions (Gilpin et al., 2007; Ganzfried and Sandholm, 2014) additionally consider how hand values change with future public cards. RL-CFR (Li et al., 2024) uses reinforcement learning to optimize *action* abstraction, complementing information abstraction. EVPA (Li and Huang, 2025) trains neural networks on CFR traversal data to estimate per-hand features for online abstraction. WEVA (Li and Huang, 2026a) avoids neural training but still requires game-tree traversal. Our work removes the requirement for game-specific evaluators and game-tree traversal entirely, using LLM-derived features instead.

CFR and Exploitability. Counterfactual regret minimization (Zinkevich et al., 2007) and its variants (Lanctot et al., 2009; Brown and Sandholm, 2019a; Li and Huang, 2026b) are standard algorithms for approximating Nash equilibria in two-player zero-sum games. Superhuman poker AI systems (Moravčík et al., 2017; Brown and Sandholm, 2018, 2019b) combine CFR with abstraction and real-time subgame solving. We use CFR as the downstream solver and evaluate abstractions by the exploitability of the lifted average strategy (Vaughn et al., 2009; Johanson et al., 2013).

Domain-Knowledge-Free Abstraction. Deep CFR (Brown et al., 2019) replaces tabular regret with neural function approximation trained from self-play traversals, avoiding manual feature design but requiring extensive game-specific sampling. WEVA (Li and Huang, 2026a) eliminates neural training but still requires running CFR iterations on the unabridged game. Our approach uses pretrained LLM knowledge as the source of abstraction signal, requiring neither game-specific training nor game-tree traversal, making it applicable to games where no solver implementation exists.

LLMs for Strategic Reasoning. Recent work has explored LLMs as game-playing agents (FAIR), strategic reasoners (Gandhi et al., 2024), and decision-makers in interactive environments. In poker, LLMs have been used as zero-shot players (Zhuang et al., 2025) with mixed results or reasoning (Anonymous, 2026b). Recent work demonstrates that structured prompts can activate latent poker skills in frontier LLMs (Anonymous, 2026a). While these works use LLMs as *players*, our work uses LLMs as *feature engineers* for offline abstraction construction, a complementary role that feeds into traditional game-solving pipelines.

LLMs as Feature Engineers. The idea of using LLMs to propose features has emerged in tabular learning and AutoML contexts (Hollmann et al., 2023). Our work applies this paradigm to game-theoretic abstraction, where features must satisfy game-theoretic desiderata (spread, independence, coverage) that differ from standard ML feature quality metrics. The key technical challenge is eliciting *calibrated numerical outputs* from LLMs—we address this through calibration anchors and constrained output formatting.

3 Method

We propose an *Abstraction Agent* that constructs information abstractions for imperfect-information games through a four-phase pipeline (Figure 1). The agent is game-agnostic: all domain knowledge is injected through a declarative GameConfig that provides natural-language descriptions of the game rules and context. No hand-strength calculators, equity estimators, or rollout simulators are used.

3.1 Problem Setting

Consider an imperfect-information game at a fixed public state c (e.g., community cards in poker,

visible discards in Mahjong). Each player can hold one of N private states h_1, \dots, h_N (e.g., hole cards or concealed tiles). An *information abstraction* maps each private state to one of K buckets: $\phi : \{h_1, \dots, h_N\} \rightarrow \{1, \dots, K\}$. A solver (e.g., CFR) computes a strategy over the K buckets; this strategy is *lifted* back to the original game by assigning all states in the same bucket the same mixed action—if two private states share a bucket, the solver cannot distinguish them and must prescribe identical behavior. The quality of ϕ is measured by the *exploitability* of the lifted strategy: how much an optimal adversary can gain by exploiting the abstraction’s inability to differentiate merged states.

3.2 Phase 1: Feature Discovery

Given a GameConfig containing the game rules and strategic context, the agent prompts the LLM to propose a set of continuous strategic features $\{f_1, \dots, f_M\}$ (typically $M \in [3, 8]$). Each feature f_j is defined by:

- A **name** (e.g., NUT_CEILING)
- A **description** explaining what the feature measures
- **Calibration anchors** mapping scores 0.0, 0.5, and 1.0 to concrete game situations

The prompt enforces several desiderata: features must be continuous in $[0, 1]$, must produce sufficient spread ($\text{std} \geq 0.15$) across the hand space, must be conceptually independent, and must collectively cover the major strategic axes of the game. When the game has future public cards (e.g., turn in poker), the prompt additionally requests features capturing future-card sensitivity.

This phase requires a single LLM call and is cached for reuse across different K values on the same game.

3.3 Phase 2: Private-State Scoring

For each private holding h_i , the agent constructs a prompt containing:

1. The public context (board cards, pot size)
2. The discovered feature definitions with their calibration anchors
3. A batch of holdings to score (typically 20–30 per call)

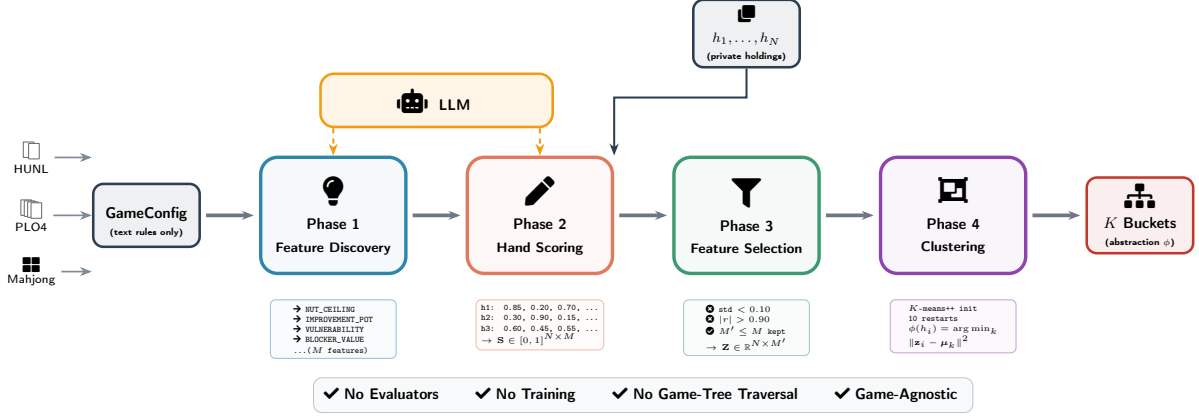


Figure 1: The Abstraction Agent pipeline. Given only a textual GameConfig and a set of private holdings, the agent uses an LLM to discover strategic features (Phase 1), score each holding on those features (Phase 2), filter redundant dimensions (Phase 3), and cluster into K buckets (Phase 4). The pipeline requires no hand evaluators, no training data, and no game-tree traversal. The same code applies to poker, Omaha, and Mahjong without modification.

The LLM returns a structured response assigning each holding a score vector $\mathbf{s}_i = (s_{i,1}, \dots, s_{i,M}) \in [0, 1]^M$. The calibration anchors serve as a shared reference frame, encouraging consistent scoring across batches. Holdings are processed in batches with retry logic and caching to handle API failures gracefully.

After scoring, we obtain a feature matrix $\mathbf{S} \in \mathbb{R}^{N \times M}$.

3.4 Phase 3: Feature Selection

Not all discovered features may be informative or independent in practice. We apply two automated filters:

Variance filter. Features with column standard deviation below a threshold $\tau_\sigma = 0.10$ are dropped as uninformative (the LLM assigned near-constant scores).

Correlation filter. For each pair of surviving features with Pearson correlation $|r| > \tau_r = 0.90$, we drop the feature with lower variance. This removes redundancy while preserving the most informative member of each correlated group.

The surviving features are rescaled by dividing each column by its standard deviation (without mean-centering, since the calibration anchors already provide a shared zero-point), producing the final feature matrix $\mathbf{Z} \in \mathbb{R}^{N \times M'}$ where $M' \leq M$.

3.5 Phase 4: Clustering

We partition the N holdings into K buckets using multi-restart K -means on \mathbf{Z} :

$$\phi(h_i) = \arg \min_k \|\mathbf{z}_i - \mu_k\|^2 \quad (1)$$

Algorithm 1 Abstraction Agent Pipeline

Require: Game config \mathcal{G} , holdings $\{h_1, \dots, h_N\}$, public state c , bucket count K

Ensure: Abstraction $\phi : \{h_i\} \rightarrow \{1, \dots, K\}$

- 1: **Phase 1: Feature Discovery**
- 2: $\{f_1, \dots, f_M\} \leftarrow \text{LLM}(\mathcal{G}.\text{game_description})$
- 3: **Phase 2: Private-State Scoring**
- 4: **for** each batch $B \subset \{h_1, \dots, h_N\}$ **do**
- 5: $\mathbf{s}_i \leftarrow \text{LLM}(c, \{f_j\}, B)$ for $h_i \in B$
- 6: **end for**
- 7: $\mathbf{S} \leftarrow [\mathbf{s}_1; \dots; \mathbf{s}_N] \in \mathbb{R}^{N \times M}$
- 8: **Phase 3: Feature Selection**
- 9: Drop columns with $\text{std} < \tau_\sigma$
- 10: Drop correlated pairs with $|r| > \tau_r$
- 11: $\mathbf{Z} \leftarrow$ standardize surviving columns
- 12: **Phase 4: Clustering**
- 13: $\phi \leftarrow K\text{-means}(\mathbf{Z}, K)$ with multi-restart
- 14: **return** ϕ

where μ_1, \dots, μ_K are cluster centroids. We use K -means++ initialization with 10 random restarts (50 iterations each) and select the partition with minimum within-cluster sum of squares.

3.6 Algorithm Summary

Algorithm 1 summarizes the full pipeline.

3.7 Game-Agnostic Design

The pipeline’s game-agnosticism is achieved through a GameConfig interface specifying the game rules (for feature discovery), a shorter context (for scoring prompts), and two rendering callbacks (hand_to_text, board_to_text). To apply the

agent to a new game, one writes a GameConfig (typically 20–50 lines of natural language) and two rendering functions. No evaluation code, rollout simulators, or domain-specific feature engineering is required.

3.8 Prompt Design Principles

Three design choices ensure reliable outputs: (1) *Calibration anchors* map 0.0/0.5/1.0 to concrete game situations, grounding the LLM in consistent reference points across batches; (2) *Constraint specification* enforces SPREAD, INDEPENDENCE, and COMPLETENESS as explicit requirements for feature discovery; (3) *Batch formatting* uses numbered lists with fixed format (“0: FEAT1=0.xx FEAT2=0.yy”), enabling reliable parsing. The calibration anchors serve a dual role: they normalize scores across batches (analogous to few-shot examples) and provide interpretable semantics for each feature’s scale.

4 Experimental Setup

4.1 Domains

HUNL Endgames. We evaluate on two turn endgames derived from Libratus-style subgames (Brown and Sandholm, 2018). Each subgame specifies a public board, pot size, and private hands reaching the decision point. Turn boards are the most strategically complex street: hand rankings can change when the final card is revealed, creating multi-dimensional strategic value (current strength, draw potential, vulnerability) that a single scalar metric cannot capture. Private hands are filtered by reach probability to retain strategically relevant holdings, evaluated at $K \in \{20, 50\}$ buckets (4 conditions total).

PLO4 Preflop (Qualitative). We apply the agent to Pot-Limit Omaha with 4 private cards at the pre-flop stage. The full hand space is $\binom{52}{4} = 270,725$ raw combinations, reduced to 16,432 canonical representatives after suit isomorphism, clustered with $K = 30$. No exploitability is computed; we inspect whether discovered features capture recognized PLO strategic concepts.

HUNL Preflop and Flop (Qualitative). We apply the agent to HUNL preflop (169 canonical hands, $K = 10$) and flop (1,176 valid holdings, $K = 20$) to test cross-street generalization without pipeline modification.

Riichi Mahjong (Qualitative). To demonstrate applicability beyond card games, we apply the agent to Japanese Riichi Mahjong (MahjongRepository, 2024), a tile-based game with 136 tiles, chance nodes, and fundamentally different mechanics from poker. We fix a mid-game scenario (turn 8, East round, 14 visible discards) and sample 200 random 13-tile hands ($K = 10$). This tests whether the pipeline discovers meaningful features for a game with chance nodes and no shared structure with poker.

4.2 Baselines

Expected Hand Strength (EHS). Each hand is assigned its expected equity against a uniform opponent range over all possible river cards, clustered by this scalar using K -means. Requires a hand evaluator and Monte Carlo rollout.

Random. Hands assigned to buckets uniformly at random (lower bound).

LLM-Embed (zero-shot). The LLM generates text descriptions of each hand, descriptions are embedded with a sentence encoder, and embeddings are clustered with K -means. Tests whether raw embeddings capture useful structure without explicit feature discovery.

Potential-Aware (PA). Hands clustered in 2D space of (EHS, $\sqrt{\text{Var}[\text{HS}]}$), where variance is computed across all river cards (Ganzfried and Sandholm, 2014). Requires a hand evaluator plus full river enumeration.

4.3 Evaluation Metric

For HUNL endgames, we report **exploitability** of the lifted strategy as a fraction of the pot. The lifted strategy is obtained by running CFR for 2,000 iterations on the abstract game and mapping the resulting strategy back to the original game. Lower exploitability indicates a better abstraction. We also report the **ratio** of agent exploitability to EHS exploitability; ratio < 1 means the agent outperforms EHS.

4.4 Implementation Details

We use GPT-5.5 via the OpenAI-compatible API (temperature 0.0, deterministic decoding). Feature discovery uses a single LLM call per game; private-state scoring uses batches of 20–30 holdings with 3-attempt retry logic. We define *parse rate* as the percentage of LLM scoring responses that match the

Table 1: HUNL turn endgame exploitability (fraction of pot). Lower is better. **Ratio** = Agent / EHS; values below 1.0 indicate the agent outperforms EHS. All runs achieve 100% parse rate.

Board	K	Agent	EHS	Ratio
7♠ 9♥ 9♣ T♣	20	0.0228	0.0335	0.68
7♠ 9♥ 9♣ T♣	50	0.0118	0.0192	0.62
T♠ 6♥ A♥ 7♣	20	0.0862	0.1088	0.79
T♠ 6♥ A♥ 7♣	50	0.0356	0.0592	0.60
Average ($K=20$)		—	—	0.74
Average ($K=50$)		—	—	0.61
Overall Average		—	—	0.67

required structured format without manual correction. The feature matrix is filtered (min std = 0.10, max correlation = 0.90) and rescaled before K -means++ clustering (10 restarts, 50 iterations each). CFR uses vanilla alternating updates for 2,000 iterations; exploitability is computed exactly by best-response traversal over the unabstrated game tree. All clustering uses a fixed random seed; the full pipeline completes in under 60 seconds per subgame excluding CFR time. Prompts are identical across all games except the GameConfig content; full prompts are provided in Appendix C. We cache the first feature-discovery output per game; clustering stability across 10 random seeds yields mean pairwise ARI of 0.77.

5 Results

5.1 HUNL Endgames

Table 1 presents the main quantitative results on turn endgames, where future cards create multi-dimensional strategic value that EHS cannot capture. The Abstraction Agent outperforms the EHS baseline in all 4 experimental conditions, with an average exploitability ratio of 0.67 (33% improvement). The improvement is consistent across both $K = 20$ (average ratio 0.74) and $K = 50$ (average ratio 0.61), with larger gains at finer granularity where the additional feature dimensions provide more discriminative power.

Comparison with Other Methods. Table 2 compares the Abstraction Agent against additional baselines at $K = 20$. The agent substantially outperforms LLM-Embed (58% lower exploitability), demonstrating that feature discovery extracts more useful structure than raw text embeddings. It also outperforms EHS (32% and 21% improvement on SG1/SG2) despite EHS requiring a domain-specific

Table 2: Comparison of abstraction methods at $K = 20$ (exploitability as fraction of pot). The Abstraction Agent outperforms LLM-Embed without requiring any domain-specific evaluator, and also outperforms EHS which requires a hand evaluator.

Method	SG1	SG2
Random	0.2235	0.6513
LLM-Embed (zero-shot)	0.0544	0.2063
Abstraction Agent (ours)	0.0228	0.0862
<i>Domain-specific methods (require hand evaluator):</i>		
EHS	0.0335	0.1088
PA (EHS + Variance)	0.0149	0.0564

hand evaluator. The PA method with access to hand evaluators and full river enumeration achieves lower exploitability, as expected; the agent bridges a substantial portion of the gap using only LLM-derived features.

Feature Discovery. Across both turn subgames, the LLM consistently discovers 7 features. After correlation-based filtering ($|\rho| > 0.9$ threshold), 6 features are typically retained. Five features appear in both subgames’ final selection: CURRENT_MADE_STRENGTH, IMPROVEMENT_POTENTIAL, VULNERABILITY_TO_OUTDRAWS, NUT_CEILING, and BLOCKER_VALUE. Two features, RIVER_EQUITY and RIVER_STABILITY, are sometimes filtered due to high correlation with other features.

Feature–EHS Correlation. Feature–EHS correlations (Appendix G) confirm that while CURRENT_MADE_STRENGTH correlates highly with EHS ($r = 0.87$ – 0.93), features like IMPROVEMENT_POTENTIAL ($r = -0.30$ to 0.27) capture orthogonal strategic dimensions.

5.2 PLO4 Preflop Clustering

The agent discovers 7 features for PLO4 preflop, of which 5 are selected after filtering: RANK_AND_PAIR_STRENGTH, SUITEDNESS_AND_FLUSH_POTENTIAL, STRAIGHT_CONNECTIVITY, ALL_CARD_COORDINATION, and DOMINATION_RISK_RESISTANCE. These align closely with recognized PLO strategic concepts (Billings et al., 2003), despite the agent having no poker-specific code. We cluster the 16,432 canonical PLO4 hands with $K = 30$, achieving 100% parse rate. Representative clusters are shown in Appendix H.

5.3 Riichi Mahjong Mid-Game Clustering

To demonstrate game-agnosticism beyond poker, we apply the agent to Japanese Riichi Mahjong, a tile-based game with fundamentally different mechanics. In a mid-game scenario (turn 8, East round, 14 visible discards, ~ 50 tiles remaining in the wall), we sample 200 random 13-tile hands and cluster with $K = 10$. The agent discovers 6 features: `CURRENT_SPEED` (shanten + effective tiles), `ACCEPTANCE_BREADTH` (number of improving tile types), `HAND_VALUE_POTENTIAL` (yaku/han scoring upside), `WAIT_QUALITY` (tenpai waiting pattern quality), `DEFENSIVE_RESILIENCE` (safety for defensive play), and `NEXT_DRAW_SWING_POTENTIAL` (how much one random draw changes strategic value). The last feature directly captures the chance-node dynamics of tile draws from the wall. After correlation filtering, 5 features are retained (100% parse rate). Clusters separate hands by strategic role: Cluster 9 (high speed, 1–2 shanten, wide acceptance) vs. Cluster 5 (low speed, high defense, 3–4 shanten). Representative clusters are shown in Appendix K. The pipeline requires no Mahjong-specific code, only a text description of the rules.

5.4 HUNL Flop Clustering

We apply the agent to a HUNL flop board ($A\spadesuit K\heartsuit Q\diamondsuit$, 1,176 valid holdings, $K = 20$). The agent discovers 7 features, of which 5 are selected after filtering. Parse rate is 97.9%. The resulting clusters separate sets, top two pair, nut straight draws, and backdoor flush draws into distinct groups, suggesting recognition of draw potential as distinct from made hand strength. Representative clusters are shown in Appendix I.

5.5 HUNL Preflop Clustering

We further apply the agent to HUNL preflop (169 canonical hands, $K = 10$). The agent discovers 7 features, all retained after filtering. Parse rate is 100%. The resulting clusters separate premium pairs (TT+, AKo) from small pairs (22–66), suited connectors from offsuit junk, and group suited aces with suited connectors based on shared flush potential. Full cluster details are in Appendix F.

5.6 Cross-Game Summary

Table 3 summarizes the agent’s performance across all evaluated games, distinguishing quantitative from qualitative evidence.

Table 3: Cross-game summary. The same pipeline is applied to four game settings. Quant. = exploitability; Qual. = cluster inspection.

Game	N	K	Eval	Key Result
HUNL Turn	$\sim 1K$	20, 50	Quant.	33% \downarrow vs EHS
PLO4 Preflop	16,432	30	Qual.	PLO features
HUNL Flop	1,176	20	Qual.	Draw/made sep.
HUNL Preflop	169	10	Qual.	Premium groups
Mahjong	200	10	Qual.	Chance features

We emphasize that cross-game results outside HUNL turn are qualitative: they assess strategic coherence of the discovered features and clusters rather than downstream exploitability. The “unchanged pipeline” refers to the algorithm, prompts, selection, and clustering procedure; only the `GameConfig` content and text rendering callbacks differ across games.

5.7 Parse Rate and Robustness

Across all experiments (HUNL turn/flop/preflop, PLO4, and Riichi Mahjong), the overall parse rate exceeds 99% (100% on turn, preflop, PLO4, and Mahjong; 97.9% on flop). Each feature produces 33–61 unique values per subgame with standard deviations of 0.11–0.25, and score distributions adapt to board texture (e.g., `NUT_CEILING` averages 0.78 on flush-draw-heavy boards versus 0.48 on paired boards).

6 Analysis

6.1 Why Multi-Feature Abstraction Outperforms EHS

EHS compresses all strategic information into a single scalar, losing information about vulnerability, nut potential, and blocker effects. The feature–EHS correlation analysis (Appendix G) confirms this: while `CURRENT_MADE_STRENGTH` correlates highly with EHS ($r = 0.87$ – 0.93), features like `IMPROVEMENT_POTENTIAL` ($r = -0.30$ to 0.27) capture orthogonal dimensions. By clustering in multi-dimensional space, the agent avoids merging hands with similar EHS but different strategic roles.

6.2 Granularity Effects

Increasing K from 20 to 50 consistently improves the agent’s ratio ($0.74 \rightarrow 0.61$). Finer abstractions allow the multi-dimensional feature space to be partitioned more precisely, while one-dimensional EHS gains less from additional buckets once major

strength tiers are separated. Clustering is stable across random seeds: mean pairwise ARI is 0.77 overall (10 seeds per condition), with $K = 50$ more stable (0.77–0.88) than $K = 20$ (0.63–0.80).

6.3 Feature Importance via Leave-One-Out

We perform leave-one-out ablation: for each feature, we remove it and re-cluster, measuring ARI between full and reduced clusterings. Results (Appendix J) show that VULNERABILITY_TO_OUTDRAWS and BLOCKER_VALUE contribute the most unique clustering signal (lowest ARI when removed), while RIVER_EQUITY has the least impact, validating the automated selection.

6.4 Comparison with Potential-Aware Methods

Potential-aware abstractions (Ganzfried and Sandholm, 2014) extend EHS by incorporating future-card distributions; EVPA (Li and Huang, 2025) learns neural features online during CFR. Our agent’s IMPROVEMENT_POTENTIAL and VULNERABILITY_TO_OUTDRAWS capture analogous information through natural-language reasoning. The PA baseline (Table 2) achieves lower exploitability, as expected for a method with exact equity calculators. Our method’s value lies in achieving competitive performance *without* game-specific evaluators or game-tree traversal.

6.5 Scope of Zero-Shot Abstraction

The PA method outperforms our agent when hand evaluators exist. The zero-shot setting is the realistic use case for (1) games without existing solver implementations, (2) rapid prototyping, and (3) games where equity computation is intractable. Our Mahjong experiment exemplifies case (1): no open-source Mahjong CFR solver exists, yet the pipeline produces meaningful strategic clusters in under 60 seconds.

6.6 Role of LLM Prior Knowledge

The pipeline’s effectiveness depends on the LLM having internalized relevant strategic concepts from its training corpus. For well-documented games like poker, this prior knowledge is rich: concepts such as “nut potential” and “blocker value” appear extensively in strategy literature. For less-documented games, the LLM may propose less informative features. Our Mahjong experiment provides a partial test: while Mahjong strategy literature exists, it is far less extensive than poker’s,

yet the agent still discovers meaningful features (shanten-based speed, defensive resilience). The method’s applicability to truly novel or undocumented games remains an open question.

7 Conclusion

We presented an Abstraction Agent that uses LLMs to discover and score strategic features for information abstraction from natural-language game descriptions, without game-specific evaluators or game-tree traversal. On two HUNL turn endgames, the agent achieves 33% lower exploitability than EHS across 2 granularity levels. The same pipeline produces qualitatively coherent abstractions in PLO4, HUNL preflop/flop, and Riichi Mahjong, discovering game-appropriate features in each domain without modification. These results suggest that LLMs can serve not only as decision-makers but as structured interfaces for converting natural-language domain knowledge into formal computational representations. We will release the full pipeline code upon publication. Future work includes scaling to full HUNL games, evaluating with multiple LLM backbones, and iterative refinement where the agent adjusts features based on solver feedback.

Limitations

Our work has several limitations. First, the quantitative evaluation is limited to two HUNL turn subgames; broader validation across more game instances and game types remains future work. Second, the agent’s advantage is strongest on streets with future cards; on the river, where EHS equals exact equity, multi-dimensional features provide limited additional benefit. Third, the method relies on GPT-5.5’s strategic knowledge and may vary with different LLM backbones; we do not yet evaluate across multiple models or prompt variants. Fourth, cross-game results (PLO4, Mahjong) are qualitative and assess strategic coherence rather than downstream exploitability. Fifth, while the pipeline avoids game-specific evaluators, it still requires a textual game description and rendering functions for private states, which presupposes some understanding of the game’s structure.

Ethics Statement

This work proposes a method for constructing information abstractions in imperfect-information games. The primary application domain is poker,

which involves gambling. Although our experiments are research-oriented and evaluated on standard academic benchmarks (Libratus subgames), improved poker abstractions could in principle be used in automated gambling systems. We release code for reproducibility in controlled research settings but do not provide deployment tools for real-money play, and we encourage use only in academic contexts. The method itself is game-agnostic and has applications beyond gambling, including security games, negotiation, and multi-agent coordination.

References

- Anonymous. 2026a. PokerSkill: Expert-level poker play from pure language models. *Under review*.
- Anonymous. 2026b. Solver-guided reasoning for mixed-equilibrium strategies. *Under review*.
- Darse Billings, Neil Burch, Aaron Davidson, Robert Holte, Jonathan Schaeffer, Terence Schauenberg, and Duane Szafron. 2003. Approximating game-theoretic optimal strategies for full-scale poker. In *International Joint Conference on Artificial Intelligence*, pages 661–668.
- Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. 2019. Deep counterfactual regret minimization. In *International Conference on Machine Learning*, pages 793–802.
- Noam Brown and Tuomas Sandholm. 2018. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424.
- Noam Brown and Tuomas Sandholm. 2019a. Solving imperfect-information games via discounted regret minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1829–1836.
- Noam Brown and Tuomas Sandholm. 2019b. Superhuman ai for multiplayer poker. *Science*, 365(6456):885–890.
- Meta Fundamental AI Research Diplomacy Team (FAIR)[†], Anton Bakhtin, Noam Brown, Emily Dian, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, and 1 others. 2022. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074.
- Kanishk Gandhi, Dorsa Sadigh, and Noah D Goodman. 2024. Strategic reasoning with language models. *arXiv preprint arXiv:2305.19165*.
- Sam Ganzfried and Tuomas Sandholm. 2014. Potential-aware imperfect-recall abstraction with earth mover’s distance in imperfect-information games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.
- Andrew Gilpin and Tuomas Sandholm. 2007. Better automated abstraction techniques for imperfect information games, with application to texas hold’em poker. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8.
- Andrew Gilpin, Tuomas Sandholm, and Troels Bjerre Sørensen. 2007. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of texas hold’em poker. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 50.
- Noah Hollmann, Samuel Müller, and Frank Hutter. 2023. Large language models for automated data science: Introducing caafe for context-aware automated feature engineering. *Advances in Neural Information Processing Systems*, 36:44753–44775.
- Michael Johanson. 2013. Measuring the size of large no-limit poker games. *arXiv preprint arXiv:1302.7008*.
- Michael Johanson, Neil Burch, Richard Valenzano, and Michael Bowling. 2013. Evaluating state-space abstractions in extensive-form games. In *International Conference on Autonomous Agents and Multi-Agent Systems*, pages 271–278.
- Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. 2009. Monte carlo sampling for regret minimization in extensive games. *Advances in neural information processing systems*, 22.
- Boning Li, Zhixuan Fang, and Longbo Huang. 2024. RL-cfr: improving action abstraction for imperfect information extensive-form games with reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, pages 27752–27770.
- Boning Li and Longbo Huang. 2025. Efficient online pruning and abstraction for imperfect information extensive-form games. In *The Thirteenth International Conference on Learning Representations*.
- Boning Li and Longbo Huang. 2026a. Effective, efficient, and general information abstraction for imperfect-information extensive-form games. *arXiv preprint arXiv:2605.10900*.
- Boning Li and Longbo Huang. 2026b. Real-time parallel counterfactual regret minimization. *arXiv preprint arXiv:2605.19928*.
- MahjongRepository. 2024. mahjong: Implementation of japanese riichi mahjong hand calculation. <https://github.com/MahjongRepository/mahjong>. MIT License, version 1.4.0.

Matěj Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. 2017. DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513.

Tuomas Sandholm. 2015. Abstraction for solving large incomplete-information games. In *AAAI Conference on Artificial Intelligence*.

Jiefu Shi and Michael L Littman. 2000. Abstraction methods for game theoretic poker. In *Computers and Games*, pages 333–345.

Kevin Waugh, Martin Zinkevich, Michael Johanson, Morgan Kan, David Schnizlein, and Michael H Bowling. 2009. A practical use of imperfect recall. In *Symposium on Abstraction, Reformulation and Approximation*.

Richard Zhuang, Akshat Gupta, Richard Yang, Aniket Rahane, Zhengyu Li, and Gopala Anumanchipalli. 2025. PokerBench: Training large language models to become professional poker players. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26175–26182.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. 2007. Regret minimization in games with incomplete information. *Advances in Neural Information Processing Systems*, 20.

A Game Descriptions

This appendix provides the complete game descriptions used as input to the Abstraction Agent. These descriptions constitute the *only* domain knowledge provided to the system; no game-specific evaluation code, hand evaluators, or equity calculators are used.

A.1 Heads-Up No-Limit Texas Hold’em (HUNL)

Cards and Deal. A standard 52-card deck with 13 ranks ($2 < 3 < \dots < K < A$) and 4 suits ($\spadesuit, \heartsuit, \diamondsuit, \clubsuit$). Each player receives 2 private (hole) cards. Five community cards are dealt face-up in stages: flop (3), turn (1), river (1).

Hand Formation. At showdown, each player forms the best 5-card combination from their 2 hole cards and 5 community cards (best 5 of 7).

Hand Rankings (strongest to weakest).

1. **Straight Flush:** five consecutive ranks, all same suit
2. **Four of a Kind:** four cards of one rank

3. **Full House:** three of one rank + two of another

4. **Flush:** five cards of one suit, not consecutive

5. **Straight:** five consecutive ranks, mixed suits

6. **Three of a Kind:** three of one rank

7. **Two Pair:** two different pairs

8. **One Pair:** one pair

9. **High Card:** none of the above

Betting Structure. No-limit: players may bet any amount up to their remaining stack at any time. Betting rounds occur after each community card stage.

Turn Endgame Setting. In our turn experiments, 4 community cards are visible and one more will be revealed. The key strategic consideration is that hand rankings *can change* with the final card; draws may complete, and currently strong hands may be overtaken.

River Endgame Setting. All 5 community cards are visible. Hand rankings are fully determined. Strategic considerations focus on current showdown strength, nut proximity, and blocker effects.

A.2 Pot-Limit Omaha (PLO4)

Cards and Deal. Standard 52-card deck. Each player receives **4 private cards** (compared to 2 in HUNL). Five community cards are dealt in stages: flop (3), turn (1), river (1).

Critical Rule. At showdown, each player **must use exactly 2** of their 4 private cards combined with **exactly 3** of the 5 community cards. This is the fundamental difference from HUNL; players cannot use 1, 3, or 4 private cards.

Hand Rankings. Same 9-category hierarchy as HUNL (straight flush through high card).

Betting Structure. Pot-limit: the maximum bet is the current pot size.

Preflop Setting. In our PLO4 experiments, no community cards have been dealt. Hand value is entirely about *post-flop potential*. The hand space is $\binom{52}{4} = 270,725$ raw combinations, reduced to 16,432 canonical representatives after suit isomorphism.

A.3 Riichi Mahjong (Japanese Competitive Mahjong)

Tiles. 136 tiles total: 34 unique types \times 4 copies. Number tiles: three suits of 1–9 (Man/Characters, Pin/Circles, Sou/Bamboo). Honor tiles: 4 winds (East, South, West, North) and 3 dragons (White, Green, Red).

Winning Condition. A complete hand (14 tiles) consists of 4 sets + 1 pair, where each set is either a sequence (three consecutive tiles of the same suit) or a triplet (three identical tiles). Special patterns: Seven Pairs, Thirteen Orphans.

Shanten and Tenpai. Shanten number = minimum tile exchanges to reach tenpai (ready to win). Tenpai (shanten = 0) means one more tile completes the hand. Lower shanten = closer to winning.

Chance Nodes. Each turn, a player draws one tile randomly from the remaining wall (~ 70 tiles initially, ~ 40 – 50 mid-game). This draw can: complete a winning hand, reduce shanten, enable higher-value scoring patterns (yaku), or provide no improvement. The draw is the primary source of randomness after the initial deal.

Scoring (Han/Fu System). Hand value depends on yaku (scoring patterns): Riichi, Tanyao (all simples), Pinfu, Honitsu (half flush), Chinitsu (full flush), etc. More han = exponentially higher payment. Mangan (5+ han) = 8000 points.

Strategic Dimensions. Hand progression speed (shanten + acceptance width), hand value potential (yaku routes), tile efficiency (acceptance count), defensive safety (safe discard availability), and waiting pattern quality (remaining copies of winning tiles). We use the mahjong Python library ([MahjongRepository, 2024](#)) for shanten calculation and hand evaluation.

B PLO4 Preflop Hand Taxonomy

PLO4 preflop hands are characterized along multiple strategic dimensions. Table 4 and Table 5 provide a comprehensive taxonomy.

B.1 Suit Patterns

B.2 Structural Types

B.3 Key PLO4 Concepts

Nut Potential. In PLO, the “nuts” (best possible hand) is critical because multi-way pots are common. Hands that can make the *nut* flush (holding

Table 4: PLO4 suit pattern classification. Suit structure determines flush potential.

Pattern	Notation	Description
Double-suited	ds (2+2)	Two suited pairs; two flush draws. Strongest.
Single-suited	ss (3+1)	Three of one suit; one flush draw.
Suited pair	sp (2+1+1)	One suited pair; weaker flush potential.
Rainbow	r (1+1+1+1)	All four suits; no flush potential.

Table 5: PLO4 structural hand types with examples and strategic properties.

Type	Example	Strategic Properties
Premium pairs + suited	AA-KK ds	Top pair + flush draws; strongest
Top rundowns	AKQJ, KQJT	Max straight connectivity
Suited connectors	8h7h6s5s	Flush + straight; strong playability
High pairs + dangles	AA-xx rainbow	High pair, limited coordination
Mid rundowns	T987, 9876	Good connectivity, lower nut potential
Suited aces	As-xxx	Nut flush draw potential
Trips	AAA-x	Reduced pair potential; weak in PLO
Low connected	5432, 6543	Straight potential but rarely nut
Disconnected low	7h3d2c9s	No coordination; weakest

the Ace of a suit) or nut straight (top of a straight) are significantly more valuable than hands that can only make non-nut versions.

Card Coordination. All 4 cards should “work together,” contributing to the same straights or flushes. A hand like $A\spadesuit K\spadesuit Q\heartsuit J\heartsuit$ (double-suited, connected) has maximum coordination, while $A\spadesuit 7\heartsuit 3\clubsuit 2\heartsuit$ has a “dangler” (disconnected card) that reduces effective hand strength.

Domination Risk. Hands where flush or straight draws can only make non-nut hands are “dominated”; they may hit their draw but still lose to a higher version. Small pairs without backup draws are particularly vulnerable.

The Must-Use-Two Rule. The requirement to use exactly 2 private cards fundamentally changes hand evaluation compared to HUNL. For example, holding four cards of one suit does *not* help make a flush (you can only use 2), and holding trips reduces pair potential (only 1 card of that rank remains in the deck).

C Prompt Templates

This section provides the exact prompt templates used by the Abstraction Agent. All prompts are game-agnostic; domain knowledge enters only through the GameConfig fields.

C.1 Feature Discovery Prompt

SYSTEM:

```

936 You are a game-theory analyst specializing in
937 imperfect-information games. Your task is to
938 propose continuous strategic features for
939 grouping private holdings.
940
941 USER:
942 ## Game Description
943 {config.game_description}
944
945 ## Task
946 Propose {M} continuous features in [0, 1] for
947 evaluating private holdings in this game.
948
949 ## Requirements
950 1. SPREAD: Each feature must produce sufficient
951 variance (std >= 0.15) across the hand space.
952 2. INDEPENDENCE: Features should capture different
953 strategic dimensions (pairwise |r| < 0.7).
954 3. COMPLETENESS: Together, features should cover
955 the major axes of strategic variation.
956 4. CALIBRATION: Provide anchors mapping 0.0, 0.5,
957 and 1.0 to concrete game situations.
958 {5. FUTURE AWARENESS (if has_future_cards):
959 Features should capture both current state AND
960 potential for change when new information is
961 revealed. Note: {future_card_note}}
962
963 ## Anti-Patterns (DO NOT propose)
964 - Binary features (e.g., "has a pair: yes/no")
965 - Features that are subsets of another
966 - Features requiring external computation
967
968 ## Output Format (JSON)
969 [{"name": "FEATURE_NAME",
970 "description": "what it measures",
971 "anchors": {"0.0": "...", "0.5": "...",
972 "1.0": "..."}]}

```

973 C.2 Hand Scoring Prompt

```

974 SYSTEM:
975 {config.game_context}
976
977 You will score private holdings on the following
978 features. Use the calibration anchors as reference
979 points for consistent scoring.
980
981 Features:
982 {for each feature f:
983   - {f.name}: {f.description}
984     0.0 = {f.anchors["0.0"]}
985     0.5 = {f.anchors["0.5"]}
986     1.0 = {f.anchors["1.0"]}
987 }
988
989 USER:
990 {board_to_text(board)}
991
992 Score each holding below. Output format:
993 INDEX: FEAT1=0.xx FEAT2=0.yy ...
994
995 {for i, hand in enumerate(batch):
996   {i}: {hand_to_text(hand, board)}}

```

996 C.3 Example Scoring Output

```

997 For a HUNL turn board 7♠9♥9♣T♣:
998 0: CURRENT_MADE_STRENGTH=0.85
999   IMPROVEMENT_POTENTIAL=0.30
1000   VULNERABILITY=0.45 NUT_CEILING=0.90
1001   BLOCKER_VALUE=0.60 RIVER_STABILITY=0.70

```

```

1: CURRENT_MADE_STRENGTH=0.20
   IMPROVEMENT_POTENTIAL=0.80
   VULNERABILITY=0.15 NUT_CEILING=0.40
   BLOCKER_VALUE=0.25 RIVER_STABILITY=0.20
...
1002
1003
1004
1005
1006

```

1007 D Experimental Details

1008 D.1 Libratus Subgame Specifications

1009 Table 6 provides details of the two Libratus-derived
1010 turn subgames used in our quantitative HUNL eval-
1011 uation.

Table 6: Libratus subgame specifications. Both are turn boards (4 community cards, 1 more to come). Hands are filtered by reach probability.

ID	Board	Street	Pot
1	7♠9♥9♣T♣	Turn	500
2	T♠6♥A♥7♣	Turn	4780

1012 D.2 Board Texture Analysis

1013 **Subgame 1** (7♠9♥9♣T♣): Paired board with
1014 club flush draw and open-ended straight draws. The
1015 pair creates full house potential; the connected tex-
1016 ture enables many draws.

1017 **Subgame 2** (T♠6♥A♥7♣): Ace-high board
1018 with heart flush draw. The disconnected low cards
1019 create a wide range of possible hand strengths.

1020 D.3 CFR Solver Configuration

- 1021 • Algorithm: Vanilla CFR with alternating up-
1022 dates
- 1023 • Iterations: 2,000
- 1024 • Exploitability: Exact best-response computa-
1025 tion over the unabstracted game tree
- 1026 • Lifted exploitability: Strategy from abstract
1027 game mapped back to original game, then
1028 best-response computed

1029 D.4 LLM API Configuration

- 1030 • Model: GPT-5.5 (primary), GPT-5.4 and GPT-
1031 5.4-mini (comparison)
- 1032 • Temperature: 0.0 (deterministic)
- 1033 • Batch size: 20–30 hands per API call
- 1034 • Retry logic: 3 attempts with exponential back-
1035 off
- 1036 • Caching: All LLM responses cached by
1037 (model, prompt hash) for reproducibility

E Discovered Features by Game

E.1 HUNL Turn Features

The following features are consistently discovered across turn subgames:

1. **CURRENT_MADE_STRENGTH** (retained in both subgames)
Measures the current 5-card hand ranking relative to all possible holdings.
Anchors: 0.0 = no pair/weak high card; 0.5 = top pair/overpair; 1.0 = set or better.
2. **IMPROVEMENT_POTENTIAL** (retained in both subgames)
Probability of improving to a significantly stronger hand with the river card.
Anchors: 0.0 = no draws; 0.5 = gutshot or weak draw; 1.0 = open-ended straight + flush draw.
3. **VULNERABILITY_TO_OUTDRAWS** (retained in both subgames)
Risk that currently strong hands will be overtaken by opponent draws.
Anchors: 0.0 = invulnerable (nuts); 0.5 = moderate risk; 1.0 = highly vulnerable to multiple draws.
4. **NUT_CEILING** (retained in both subgames)
How close the hand is to the best possible hand on this board.
Anchors: 0.0 = far from nuts; 0.5 = second/third nuts; 1.0 = current nuts.
5. **BLOCKER_VALUE** (retained in both subgames)
Whether the hand blocks opponent's strong holdings or draws.
Anchors: 0.0 = no blocking; 0.5 = blocks one draw; 1.0 = blocks multiple strong hands.
6. **RIVER_STABILITY** (retained in 1/2 subgames)
How stable the hand's relative strength is across possible river cards.
Anchors: 0.0 = highly volatile; 0.5 = moderate stability; 1.0 = rank unchanged by any river.
7. **RIVER_EQUITY** (sometimes filtered)
Expected equity after the river card is revealed.
Often filtered due to high correlation with CURRENT_MADE_STRENGTH.

E.2 PLO4 Preflop Features

1. **RANK_AND_PAIR_STRENGTH** (retained)
High-card value and pair potential of the 4-card holding.
Anchors: 0.0 = low disconnected; 0.5 = mid pair; 1.0 = AA/KK coordinated.
2. **SUITEDNESS_FLUSH_POTENTIAL** (retained)
Suit structure and nut flush draw potential.
Anchors: 0.0 = rainbow; 0.5 = single suited; 1.0 = double-suited.
3. **STRAIGHT_CONNECTIVITY** (retained)
How many different straights the hand can make.
Anchors: 0.0 = no straight potential; 0.5 = one-gap; 1.0 = 4-card rundown.
4. **ALL_CARD_COORDINATION** (retained)
Degree to which all 4 cards work together.
Anchors: 0.0 = 1+ danglers; 0.5 = 3 coordinated; 1.0 = all 4 coordinated.
5. **DOMINATION_RISK_RESISTANCE** (retained)
Resistance to being dominated by higher versions of the same draw.
Anchors: 0.0 = non-nut; 0.5 = nut in one dimension; 1.0 = multi-nut.

F HUNL Preflop Cluster Details

Table 7 shows the complete clustering of 169 canonical HUNL preflop hands into $K = 10$ clusters. The agent discovers 7 features, all retained after filtering: HIGH_CARD_POWER, PAIR_AND_SET_VALUE, SUITED_FLUSH_POTENTIAL, STRAIGHT_CONNECTIVITY, DOMINATION_RESILIENCE, EQUITY_REALIZATION_STABILITY, and RUNOUT_VOLATILITY. Parse rate is 100%.

The clustering demonstrates strategic coherence: Cluster 8 isolates the strongest starting hands (premium pairs and AKo), Cluster 5 groups premium suited broadways with flush and straight potential, and Cluster 0 groups all small-to-medium pairs together (sharing similar set-mining value). The agent correctly separates suited from offsuit hands (Clusters 1/4/6 vs 7/9/2), reflecting the importance of flush potential in preflop hand evaluation.

Table 7: Complete HUNL preflop clusters ($K = 10$, 169 canonical hands, sorted by ID). The agent separates premium pairs, suited connectors, offsuit broadways, and junk hands without any poker-specific code.

ID	Representative Hands	Size	Characteristics
0	22, 33, 44, 55, 66, 77, 88, 99	8	Small/medium pairs (set-mining value)
1	A2s–A5s, 54s–76s	19	Suited aces + suited connectors
2	32o–T2o	19	Offsuit junk (weakest hands)
3	QTo, KTo, ATo, QJo, KJo, AJo	12	Offsuit broadways
4	K2s–K4s, Q2s–Q4s, T4s–J4s	25	Suited high + low kicker
5	ATs, QJs, KJs, AJs, KQs	10	Premium suited broadways
6	32s–92s, T2s	23	Suited junk (low cards)
7	A3o–A5o, 54o–76o	18	Offsuit aces + connectors
8	TT, JJ, QQ, KK, AKo, AA	6	Premium pairs + AKo
9	J2o–K3o, A2o	29	Offsuit high + low kicker

G Feature–EHS Correlation

Table 8: Pearson correlation between LLM features and EHS across turn subgames. Low-correlation features capture strategic dimensions beyond hand strength.

Feature	SG1	SG2
Current_Made_Str.	0.87	0.93
Improvement_Pot.	0.27	−0.30
Vuln._To_Outdraws	0.68	0.31
Nut_Ceiling	0.68	0.14
River_Stability	−0.36	0.52
Blocker_Value	0.65	0.46

H PLO4 Preflop Cluster Examples

Cluster 27 contains exclusively rainbow hands with high pair strength (AAKK or AA+K) but zero flush potential, while Cluster 10 groups low suited connectors with maximum straight connectivity, demonstrating that the LLM-discovered features capture the multi-dimensional nature of PLO hand evaluation. Cluster 4 (top rundowns like AKQJ) and Cluster 21 (trips/quads) are cleanly separated despite both having high rank strength, because the connectivity and coordination features distinguish them.

I HUNL Flop Cluster Examples

J Feature Ablation Results

K Riichi Mahjong Cluster Details

Table 13 shows the features discovered by the agent for Riichi Mahjong mid-game clustering. Table 14 shows representative clusters.

Table 9: Complete PLO4 preflop clusters ($K = 30$, 16,432 canonical hands, sorted by ID). The agent separates hands by rank strength, suitedness, connectivity, and coordination without poker-specific code.

ID	Size	Representative Hands	Characteristics
0	226	$A\spadesuit Q\heartsuit 9\clubsuit 2\spadesuit, A\spadesuit Q\heartsuit 7\heartsuit 2\heartsuit$	Medium cards, single-suited
1	60	$A\spadesuit A\heartsuit K\spadesuit Q\heartsuit, A\spadesuit A\heartsuit K\spadesuit Q\heartsuit$	Premium AA+broadways, suited, connected
2	184	$A\spadesuit K\spadesuit 6\spadesuit 5\spadesuit, A\spadesuit K\spadesuit 6\heartsuit 5\heartsuit$	AK double-suited + low connectors
3	141	$A\spadesuit A\heartsuit 9\clubsuit 8\spadesuit, A\spadesuit A\heartsuit 9\clubsuit 8\clubsuit$	AA + connected side cards
4	74	$A\spadesuit K\spadesuit Q\heartsuit J\spadesuit, A\spadesuit K\spadesuit Q\heartsuit J\heartsuit$	Top rundowns, max connectivity
5	192	$A\spadesuit K\spadesuit J\heartsuit 5\spadesuit, A\spadesuit K\spadesuit J\heartsuit 5\clubsuit$	AK suited + gapped dangler
6	326	$A\spadesuit K\spadesuit 6\spadesuit 4\heartsuit, A\spadesuit K\spadesuit 6\heartsuit 4\spadesuit$	AK suited + low disconnected
7	85	$A\spadesuit K\heartsuit Q\spadesuit 7\heartsuit, A\spadesuit K\heartsuit Q\spadesuit 6\heartsuit$	AKQ rainbow, low dangler
8	34	$A\spadesuit A\heartsuit K\spadesuit Q\heartsuit, A\spadesuit A\heartsuit K\spadesuit J\heartsuit$	AA + Broadway rundown, rainbow
9	168	$A\spadesuit K\heartsuit 9\clubsuit 3\heartsuit, A\spadesuit K\heartsuit 9\clubsuit 2\heartsuit$	AK rainbow, gapped dangler
10	87	$A\spadesuit 8\spadesuit 7\spadesuit 6\spadesuit, A\spadesuit 8\spadesuit 7\spadesuit 6\heartsuit$	Low suited connectors, max rundown
11	258	$A\spadesuit A\heartsuit A\spadesuit 2\spadesuit, A\spadesuit K\spadesuit 9\spadesuit 3\heartsuit$	Mixed: trips/AK suited, gapped
12	89	$A\spadesuit A\heartsuit K\spadesuit K\heartsuit, A\spadesuit A\heartsuit K\spadesuit K\clubsuit$	Double-paired (AAKK), double-suited
13	165	$A\spadesuit A\heartsuit K\spadesuit 9\spadesuit, A\spadesuit A\heartsuit K\spadesuit 9\clubsuit$	AA + K suited, medium kicker
14	138	$A\spadesuit K\spadesuit J\heartsuit 6\heartsuit, A\spadesuit K\spadesuit J\heartsuit 5\heartsuit$	AK double-suited + Broadway
15	238	$A\spadesuit K\spadesuit 9\spadesuit 3\spadesuit, A\spadesuit K\spadesuit 9\spadesuit 2\spadesuit$	AK mono-suited, medium cards

Table 10: PLO4 preflop clusters (continued).

ID	Size	Representative Hands	Characteristics
16	133	$A\spadesuit J\heartsuit 3\heartsuit 2\heartsuit, A\spadesuit J\heartsuit 3\heartsuit 2\clubsuit$	Low cards + A-high flush draw
17	43	$A\spadesuit J\heartsuit 9\clubsuit 8\heartsuit, A\spadesuit J\heartsuit 9\clubsuit 7\heartsuit$	Medium connected, rainbow
18	222	$A\spadesuit K\heartsuit 9\clubsuit 2\spadesuit, A\spadesuit K\heartsuit 8\heartsuit 7\heartsuit$	AK + medium suited side cards
19	128	$A\spadesuit K\spadesuit Q\heartsuit 7\spadesuit, A\spadesuit K\spadesuit Q\heartsuit 8\heartsuit$	AKQ double-suited, connected
20	274	$A\spadesuit K\heartsuit 8\heartsuit 4\heartsuit, A\spadesuit K\heartsuit 8\heartsuit 4\clubsuit$	AK + medium suited, dangler
21	69	$A\spadesuit A\heartsuit A\spadesuit A\heartsuit, A\spadesuit A\heartsuit A\spadesuit K\heartsuit$	Trips/quads (AAA+), rainbow
22	190	$A\spadesuit A\heartsuit 6\spadesuit 5\spadesuit, A\spadesuit A\heartsuit 6\spadesuit 5\clubsuit$	AA + low suited connectors
23	167	$A\spadesuit J\heartsuit 3\heartsuit 2\spadesuit, A\spadesuit T\spadesuit 9\spadesuit 5\spadesuit$	Medium suited, connected
24	108	$A\spadesuit K\spadesuit Q\heartsuit Q\heartsuit, A\spadesuit K\spadesuit Q\heartsuit 9\spadesuit$	AKQ suited, max coordination
25	166	$A\spadesuit K\spadesuit T\heartsuit 3\spadesuit, A\spadesuit K\spadesuit 8\heartsuit 6\spadesuit$	AK suited + offsuit gapped
26	231	$A\spadesuit J\spadesuit 9\spadesuit 3\spadesuit, A\spadesuit J\spadesuit 9\spadesuit 3\heartsuit$	A-high suited, medium cards
27	32	$A\spadesuit A\heartsuit K\spadesuit K\heartsuit, A\spadesuit A\heartsuit K\spadesuit 9\heartsuit$	AA + K rainbow, no flush
28	106	$A\spadesuit K\heartsuit T\spadesuit 7\spadesuit, A\spadesuit K\heartsuit T\spadesuit 7\clubsuit$	AK suited + connected middle
29	106	$A\spadesuit A\heartsuit A\spadesuit K\spadesuit, A\spadesuit A\heartsuit A\spadesuit Q\spadesuit$	Trips (AAA) + suited kicker

Table 11: Complete HUNL flop clusters (board: $A\spadesuit K\heartsuit Q\heartsuit$, $K = 20$, 1,176 hands, sorted by ID). The agent separates hands by made strength, draw potential, and blocker value.

ID	Representative Hands	Size	Characteristics
0	Ad 9s, Ad 9d, Ad 8c	80	Top pair A, strong kicker
1	Js Ts, Js Td, Jd Tc	16	Nut straight (Broadway)
2	9s 4d, 9s 3c, 9s 2h	162	Air (9x low kicker, no draw)
3	Ah 5h, Ah 4h, Ks 9s	138	Medium made + backdoor draws
4	9d 9c, 8s 8h, 6s 6d	25	Underpairs (below board)
5	Ad Ac, Ks Kd, Qs Qc	8	Sets (AA/KK/QQ)
6	8c 7s, 8c 6d, 8c 5c	86	Junk (8x and below, no interaction)
7	Td 5d, Td 4s, Td 3c	29	T-kicker, weak draws
8	Ks 9d, Ks 8c, Ks 7d	153	Second pair K, weak kicker
9	Ks 4d, Ks 3s, Ks 2d	33	Second pair K, very weak kicker
10	Ad Ks, Ad Qh, Ac Kd	21	Top two pair (AK, AQ)
11	Ac 9s, Ac 9d, Ac 8c	35	Top pair A + backdoor nut flush
12	9s 9d, 9s 9c, 8s 8d	23	Medium pocket pairs (multi-suit)
13	8d 6s, 8d 5c, 8d 4s	39	Air + faint backdoor diamond
14	Jc 9s, Jc 8d, Jc 7c	43	J-kicker, straight potential
15	Ks Js, Ks Jd, Ks Ts	35	Second pair K + strong kicker J/T
16	9s 8s, 9s 7d, 9s 6c	161	Largest junk group (below T)
17	Ts 3c, Ts 2d, Td 9s	23	T-kicker mixed, bottom straight draw
18	Ad Js, Ad Td, Ad Tc	27	Top pair A + strong kicker J/T (diamond)
19	Jd 7d, Jd 6s, Jd 5c	39	J-kicker + weak backdoor

Table 12: Feature importance via leave-one-out ablation. Lower ARI = larger impact when removed. Averaged across all subgame/K conditions where the feature is selected.

Feature Removed	Mean ARI
Vulnerability_To_Outdraws	0.7558
Blocker_Value	0.7607
Current_Made_Strength	0.7638
Nut_Ceiling	0.7716
Improvement_Potential	0.7866
River_Stability	0.7959
River_Equity	0.8122

Table 13: Features discovered for Riichi Mahjong mid-game (turn 8, East round). 6 discovered, 5 selected after correlation filtering.

Feature	Description	Selected
CURRENT_SPEED	Shanten number combined with effective tile count	✓
ACCEPTANCE_BREADTH	Number of different tile types that improve the hand	✓
HAND_VALUE_POTENTIAL	Expected scoring upside (yaku routes, dora, han count)	✓
WAIT_QUALITY	Quality of the likely winning wait (remaining copies, readability)	✓
DEFENSIVE_RESILIENCE	Ability to avoid dealing into opponent's hand	✓
NEXT_DRAW_SWING	How much one random draw changes strategic value	×

Table 14: Complete Riichi Mahjong clusters ($K = 10$, 200 hands, 100% parse rate). Sorted by cluster ID. Clusters separate hands by strategic role: offensive speed vs. defensive resilience vs. value potential.

ID	Size	Representative Hands	Dominant Trait
0	27	11578m 156p 13347s; 1167m 33p 12567s GR	Wide acceptance, scattered (3–4 shanten)
1	20	112467899m 4p 16s W; 14567m 569p 6688s W	High speed + high value connected
2	22	11149m 5579p 668s WH; 11369m 369p 17s SSR	Very slow, high defense (4 shanten)
3	16	1129m 39p 4777s WGR; 1147m 37p 278s EESR	High value, slow (honor pairs, yakuhai)
4	23	1136m 2578p 1455s S; 1233558m 49p 156s E	Average hands (moderate all features)
5	33	1119m 1p 157889s WR; 1244m 1389p 34s SWG	Highest defense, slow (many honors)
6	14	11378m 5688p 2s GRR; 145668m 138p 688s E	Highest value potential (pairs/triplets)
7	23	1239m 1245p 2667s W; 127m 13346678p SR	High acceptance, straight-type
8	15	12467m 489p 12339s; 1278m 388p 3479s RR	Balanced moderate (all features ~ 0.5)
9	7	12456m 356p 56s WWR; 2367m 2346p 12455s	Fastest (1–2 shanten, wide waits)