KrwEmd: Revising the Imperfect Recall Abstraction from Forgetting Everything

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

A recent research has shown that an extreme interpretation of imperfect recall abstraction – completely forgetting all past information – has led to excessive abstraction issues. Currently, there are no hand abstraction algorithms that effectively integrate historical information. This paper aims to develop the first such algorithm. Initially, we introduce the KRWI abstraction for Texas Hold'em-style games, which categorizes hands based on K-recall winrate features that incorporate historical information. Statistical results indicate that, in terms of the number of distinct infosets identified, KRWI significantly outperforms POI, an abstraction that identifies the most abstracted infosets that forget all historical information. Following this, we introduce the KrwEmd algorithm, the first hand abstraction algorithm to effectively use historical information by combining K-recall win rate features and earth mover's distance for hand classification. Experimental studies conducted in the Numeral211 Hold'em environment show that under identical abstracted infoset sizes, KrwEmd not only surpasses POI but also outperforms state-of-the-art hand abstraction algorithms such as Ehs and PaEmd. These findings suggest that incorporating historical information can significantly enhance the performance of hand abstraction algorithms, positioning KrwEmd as a promising approach for advancing strategic computation in large-scale adversarial games.

1 Introduction

Imperfect recall abstraction has proven to be very important for solving large-scale computational games, significantly reducing computational complexity. Recently, AI using imperfect recall abstraction has developed better-than-human strategies for Texas Hold'em testbed—even when using limited computational resources [23, 7, 8].

The task of hand abstraction in Texas Hold'em aims to reduce computational overhead by applying the same strategy to similar hands. In an imperfect recall setting [29, 20], the hand abstraction in the later phase does not strict depend on the results of the hand abstraction in the earlier phase. However, the term **imperfect recall** is often interpreted in an extreme manner in practice. Researchers typically understand it as completely forgetting all past information—in other words, considering only future information—and design abstraction algorithms based on this understanding [16, 17, 19, 15, 14]. There are two major factors that mainly affect the results of abstraction for each phase: the number of clustering centers (i.e. centroids), which can be set man-

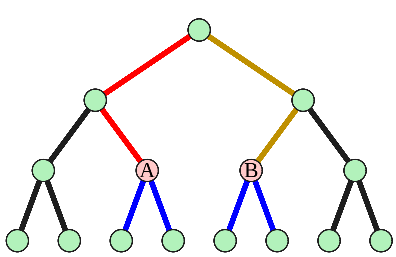


Figure 1: In a 4-phase game hand abstraction task, the current goal is to classify hands A and B.

ually, and the number of distinct features that are used to categorize hands at each phase. Recent research [12] has found that constructing hand features solely based on future information can lead to excessive abstraction. For example, as shown in the Figure 1, two hands: A and B constructed with only future information can have the same hand features. As the game progresses, the rate of feature repetition among different hands gradually increases, while the distribution of distinct hand features assumes a spindle-shaped pattern. Additionally, constructing hand features with historical information in addition to the future may differentiate two hands sharing the same future information and hence makes more features available for clustering as well as enhances the performance of hand abstraction.

However, there still remain two unsolved issues. First, Fu et al. [12] have introduced a K-recall outcome feature, which incorporates historical information. This feature can only identify if elements are identical or not, but it lacks the capability to discern the extent of differences between features. Therefore, it is difficult to adjust the number of clusters appropriately, which makes it challenging to construct an effective hand abstraction algorithm that integrates historical information. Second, due to the inability to modify the number of clusters, Fu et al. [12] only compared the performance between the maximum clusters cases of integration of historical information (KROI) and no integration at all (POI). In this condition, although KROI significantly outperforms POI, the comparison is inconclusive because KROI recognizes more abstracted infosets than POI. Thus, it does not prove that the performance of abstraction algorithms that integrate historical information is necessarily superior under the condition of having the same number of abstracted infosets.

This paper introduces a framework for constructing hand features based on winrates, with the K-recall winrate feature being the most crucial one. Based on this, we developed the K-recall winrate isomorphism (KRWI), an abstraction that integrates historical information. Across the same game phases, KRWI identifies slightly fewer hand features than KROI but significantly more than POI. Importantly, the K-recall winrate feature is capable of discerning the extent of differences between features. Therefore, by combining the earth mover's distance with the K-recall winrate feature, we developed the first hand abstraction algorithm that integrates historical information, named KrwEmd, and designed an efficient computational method. We validated our approach in the Numeral211 game environment, where KrwEmd demonstrated superior performance to POI under the same infosets conditions. Additionally, in clustering settings, KrwEmd also outperformed the Ehs and PaEmd algorithms, with PaEmd being the current state-of-the-art hand abstraction algorithm.

2 Background and Notation

Generally, Texas Hold'em-style poker games are modeled as imperfect information games. However, for the task of hand abstraction, games with ordered signals [18, 12] offer a better theoretical tool. The game with ordered signals is a subclass of imperfect information games in that they further subdivide the nodes (also called histories, states, or trajectories) in imperfect information games into mutually independent signals and public nodes. This allows for each aspect to be studied in isolation. Under this framework, the hand abstraction task in Texas Hold'em-style games is modeled as signal abstraction.

In a game with ordered signals $\tilde{\Gamma} = \left\langle \tilde{\mathcal{N}}, \tilde{H}, \tilde{Z}, \tilde{\rho}, \tilde{A}, \tilde{\chi}, \tilde{\tau}, \gamma, \Theta, \varsigma, O, \omega, \succeq, \tilde{u} \right\rangle$, there is a set of players $\tilde{\mathcal{N}} = \mathcal{N} \cup \{c, pub\}$, which includes not only the main participants $\mathcal{N} = \{1, \dots, N\}$ but also a special nature player c who controls the randomness and an observer player pub who can see everything but doesn't take any actions. The game progresses through a series of public nodes $\tilde{X} = \tilde{H} \cup \tilde{Z}$. Some of these public nodes are terminal public nodes $\tilde{Z}$ where the game ends and outcomes are determined, while the others are non-terminal public nodes $\tilde{H}$. Among the non-terminal public nodes, some are where players make decisions within the action space $\tilde{A}$, and the remaining are chance public nodes where the nature player reveals signals, with the special action $Reveal$ within $\tilde{A}$.

At every non-terminal public node, $\tilde{\rho} : \tilde{H} \mapsto \mathcal{N}c$ (i.e., $\mathcal{N} \cup \{c\}$) specifies which player is responsible for making an action, and $\tilde{\chi} : \tilde{H} \mapsto 2^{\tilde{A}}$ confines the possible actions they can take. When the nature player makes a move, it reveals signals $\theta \in \Theta$ that carry information relevant to the game. These signals are then observed by all players except c, $O(\theta) = (O_1(\theta), \dots, O_N(\theta), O_{pub}(\theta))$, though what they can see might differ.

The progression from one public node to another is clearly defined $\tilde{\tau} : \tilde{H} \times \tilde{A} \mapsto \tilde{X}$, ensuring that the game's structure is sequential and predictable. Similarly, the signals are revealed according to a probability distribution $\varsigma : \Theta \mapsto \Delta(\Theta)$, which specifies the likelihood of the next signal given the current one. We use $\tilde{h} \sqsubseteq \tilde{h}'$ to indicate that $\tilde{h}$ is a predecessor of $\tilde{h}'$, and $\theta \sqsubseteq \theta'$ to indicate that θ is a predecessor of θ'. Each phase of the game is the number of times nature player has revealed signals, denoted by $\gamma : \tilde{X} \mapsto \mathbb{N}^+$. $\mathfrak{r} = \{\gamma(\tilde{x}) \mid \tilde{x} \in \tilde{X}\}$ represents the phases that a game with ordered signals may go through. Since the root is a chance public node, we have $\min \mathfrak{r} = 1$.

At the end of the game, players receive their payoffs based on the signals and the terminal public node, represented by $\tilde{u} = (\tilde{u}_1, \dots, \tilde{u}_N)$, where $\tilde{u}_i : \Theta \times \tilde{Z} \mapsto \mathbb{R}$. Additionally, each player's survival status is determined at these terminal public nodes, denoted by $\omega = (\omega_1, \dots, \omega_N)$, where $\omega_i : \tilde{Z} \mapsto \{true, false\}$. The signals possess a partial order within their subset, terminal signals $\tilde{\Theta}$, indicated by $\succeq: \tilde{\Theta} \times \mathcal{N} \times \mathcal{N} \mapsto \{true, false\}$. It is required that for any terminal signal $\theta \in \tilde{\Theta}$ and terminal public nodes $\tilde{z} \in \{\tilde{z}' \in \tilde{Z} \mid \omega_i(\tilde{z}') = \omega_j(\tilde{z}') = true\}$, if $\succeq (\theta, i, j) = true$, then $\tilde{u}_i(\theta, \tilde{z}) \geq \tilde{u}_j(\theta, \tilde{z})$.

Players make decisions based on their observations of signals and the current non-terminal public node. A player may have the same observation for different signals, forming a signal infoset for signals they cannot distinguish. For a player $i \in \mathcal{N}$, the signal infoset for a signal θ is denoted as $\vartheta_i(\theta) = \{\theta' \in \Theta \mid O_i(\theta) = O_i(\theta') \wedge O_{pub}(\theta) = O_{pub}(\theta')\}$. Specifically, for the nature player, $\vartheta_c(\theta) = \{\theta' \in \Theta \mid O_{pub}(\theta') = O_{pub}(\theta)\}$. We abuse the notation $\vartheta \in \mathit{\Theta}_i$ to represent a signal infoset, where for any player $i \in \mathcal{N}$, $\mathit{\Theta}_i$ is a partition of Θ, representing the collection of player i's signal infosets. $\mathit{\Theta}_i^{(1)}, \dots, \mathit{\Theta}_i^{(|\mathfrak{r}|)}$ are the collections of player i's signal infosets for each phase, and they form a partition of $\mathit{\Theta}_i$. In games with ordered signals, the signals describe all private information. The signal infoset, combined with public nodes, can be transformed into the infoset of an imperfect information game. Fu et al. [12] detailed this transformation process.

The game with ordered signals model allows us to study the issue of signal abstraction independently. For this purpose, we introduce a signal (infoset) abstraction profile, $\alpha = (\alpha_1, ., \alpha_N)$, where for each player $i \in \mathcal{N}$, α_i is a partition of Θ called the signal (infoset) abstraction. Any $\hat{\vartheta} \in \alpha_i$ then is said to be an abstracted signal infoset for player i, and it can be further divided into several signal infosets within $\mathit{\Theta}_i$. These finer signal infosets collectively form a partition of $\hat{\vartheta}$. In general, two signal abstractions cannot be directly compared in terms of performance, but in a few specific cases there does exist a special relationship between them, which is called refinement. Consider two abstractions α_i and β_i. If $\forall \hat{\vartheta} \in \beta_i$, there exists one or more abstracted signal infosets in α_i such that the union of these forms a partition of $\hat{\vartheta}$, then we said that α_i refines β_i, symbolically $\alpha_i \sqsupseteq \beta_i$. The signal abstracted game $\tilde{\Gamma}^\alpha$ was derived by substituting $\mathit{\Theta}_i$ with α_i across all $\tilde{x} \in \tilde{X}$.

Perfect/imperfect recall originally describes a property of imperfect information games, indicating that players do not need to remember all the information they have observed throughout the game. Since games with ordered signals are a subset of imperfect information games, we derived the concept of signal perfect/imperfect recall from them. A player i in a game $\tilde{\Gamma}$ is said to have signal perfect recall if, for any $\theta_1', \theta_2' \in \vartheta'$, any predecessor θ_1 of θ_1' has a corresponding predecessor θ_2 of θ_2' such that $\theta_2 \in \vartheta(\theta_1)$. If all players have signal perfect recall, the game $\tilde{\Gamma}$ is said to have signal perfect recall. For a game $\tilde{\Gamma}$ with signal perfect recall, if α_i is the signal abstraction of player $i \in \mathcal{N}$, let $(\alpha_i, \mathit{\Theta}_{-i})$ denote the signal abstraction profile where player i adopts the signal abstraction α_i while other players do not do abstraction. If $\tilde{\Gamma}^{(\alpha_i, \mathit{\Theta}_{-i})}$ retains signal perfect recall, then α_i is considered a signal abstraction with perfect recall; otherwise, it is an signal abstraction with imperfect recall.

In games with ordered signals, the strategy π_i for player i maps from a non-terminal public node and a signal infoset to a probability distribution over actions, with the strategy profile denoted as $\pi = (\pi_1, \dots, \pi_N)$. When all players adopt the strategy profile π, the expected sum of future rewards, also known as expected value, for player i at public node $\tilde{x}$ and signal θ is denoted as $v_i^\pi(\theta, \tilde{x})$, and the expected value for the entire game is denoted as $v_i(\pi)$. A Nash equilibrium is a strategy profile where no player can obtain a higher expected value by changing their strategy. Formally, π^* is a Nash equilibrium if for every player i, $v_i(\pi^*) = \max_{\pi_i} v_i(\pi_i, \pi_{-i}^*)$, where π_{-i} denotes the strategies of all players except i. In two-player zero-sum scenarios, the exploitability of π is denoted as $\epsilon(\pi) = \frac{\max_{\pi_1'} v_i(\pi_1', \pi_2) + \max_{\pi_2'} v_i(\pi_1, \pi_2')}{2}$.

3 Related Work

Our research focuses on hand abstraction techniques in AI systems for Texas Hold'em-style games (i.e. the signal abstraction in games with ordered signals), building on the initial works of Shi and Littman [25] and Billings et al. [4]. These seminal works introduced the concept of game abstraction, which aims to simplify games while preserving essential characteristics. The researchers started by manually forming hand buckets as a result of their expertise with game-playing strategy. The first automated hand abstraction was that of Gilpin and Sandholm [16]. Later, a model of games with ordered signals was given for Texas Hold'em by Gilpin and Sandholm [18]; lossless isomorphism (LI) was developed with signal rotation. Despite the elegance of LI, its low compression rates hinder its application in large-scale games, whereas lossy abstraction shows potential for such application. An expectation-based clustering method was proposed by Gilpin and Sandholm [17] in their work, and a histogram-based clustering method was introduced by Gilpin et al. [19]. The former is known as Ehs, while the latter is referred to as the potential-aware method. Subsequent studies by Gilpin and Sandholm [15] and Johanson et al. [20] compared Ehs and potential-aware methods, concluding that the latter holds an advantage in large-scale games. Johanson et al. [20] also introduced the use of earth mover's distance[1] (EMD) in potential-aware methods. Ganzfried and Sandholm [14] introduced a more efficient approximation algorithm for earth mover's distance in potential-aware methods (PaEmd). Brown et al. [9] further applied PaEmd to distributed environments for solving large-scale imperfect-information games. This paradigm has found success in Texas Hold'em AI systems and is considered state-of-the-art in hand abstraction. Very recently, Fu et al. [12] proposed several novel tools, such as abstraction resolution and common refinement. They introduced two signal abstraction: one is the potential outcome isomorphism (POI), which identifies the maximum number of abstracted signal infosets considering future information only; The other is the K-recall outcome isomorphism (KROI), which identifies the maximum number of abstracted signal infosets considering historical information. They emphasized that current imperfect recall signal abstraction algorithms, which consider only future information, are prone to excessive abstraction. However, they did not provide practical signal abstraction algorithms.

Other abstraction techniques for decision-making problems include action abstraction [13, 6, 21] and general imperfect recall abstraction [10, 11] in extensive-form games, as well as state abstraction and action abstraction in reinforcement learning [1, 2].

4 Winrate Isomorphism

The first contribution of this paper is an isomorphism framework of winrate-based features, including the potential winrate isomorphism (PWI) and the k-recall winrate Isomorphism (KRWI). Compared with outcome-based features, winrate-based features offer a streamlined approach, focusing exclusively on the distribution of loss, draw, and win outcomes of signals emanating from a signal infoset (and its predecessors) as it evolves towards the terminal signals. Winrate-based features are numerical vectors of consistent length. In this section, an identical Winrate-based feature uniquely determines an abstracted signal infoset. It is worth noting that the similarity of Winrate-based features reflects the similarity among signal infosets, allowing for clustering based on these features (see Section 5).

Both PWI and KRWI share the similar isomorphism construction process for player i in phase r, as illustrated in algorithm 1. The difference lies only in the construction operator for the winrate-based features, FEATURE, used in lines 5 and 12. The isomorphism construction process starts by iterating through all signal infosets of $\Theta_i^{(r)}$ and collecting their features. Next, these features are deduplicated and stored in lexicographical order within set $\mathcal{C}_i^{(r)}$, which is implemented as a vector data structure. Within $\mathcal{C}_i^{(r)}$, the index of a feature serves as an identifier for an abstracted signal infoset. Then, by utilizing a hash table $\mathcal{CI}_i^{(r)}$, we can identify an abstracted signal infoset's identifier based on its feature. In the final step, we traverse $\Theta_i^{(r)}$ again, associating the identifier of a signal infoset with the identifier of its corresponding abstracted signal infoset, and this relationship is recorded in $\mathcal{D}_i^{(r)}$, an isomorphism map. The function $Index_i(r, \cdot)$ is a domain-specific mapping that assigns a unique identifier to each signal infoset at phase r, within the numeric range of 0 to $|\Theta_i^{(r)}| - 1$. In

[1] `https://en.wikipedia.org/wiki/Earth_mover%27s_distance`

Algorithm 1 Isomorphism Constructor

Require:

$r = 1, \ldots, R$. Phases.

$\Theta_i^{(r)}$. Signal infoset space for player i.

$Index_i(r, \cdot) : \Theta_i^{(r)} \mapsto \mathbb{N}$. Signal infoset index function for player i.

procedure ISOMORPHISMCONSTRUCTOR(r, $\Theta_i^{(r)}$, FEATURE($\cdot$))
  Initialize $\mathcal{C}_i^{(r)}$ vector as empty.
  Initialize $\mathcal{D}_i^{(r)}$ array arbitrarily with length $|\Theta_i^{(r)}|$.
  for $\vartheta \in \Theta_i^{(r)}$ **do**
    $feature \leftarrow$ FEATURE(ϑ).
    Append $feature$ to $\mathcal{C}_i^{(r)}$.
  end for
  Eliminate duplicates from $\mathcal{C}_i^{(r)}$.
  Sort the elements of $\mathcal{C}_i^{(r)}$ in lexicographical order.
  Construct hash table $\mathcal{CI}_i^{(r)}$ from $\mathcal{C}_i^{(r)}$. Store the index $lexid$ and value $feature$ of $\mathcal{C}_i^{(r)}$ in $\mathcal{CI}_i^{(r)}$ as key-value pairs $(feature, lexid)$.
  for $\vartheta \in \Theta_i^{(r)}$ **do**
    $feature \leftarrow$ FEATURE(ϑ), $idx \leftarrow Index_i(r, \vartheta)$.
    Update $\mathcal{D}_i^{(r)}[idx]$ with $\mathcal{CI}_i^{(r)}[feature]$.
  end for
  return $(\mathcal{C}_i^{(r)}, \mathcal{D}_i^{(r)})$.
end procedure

Texas Hold'em-style games, one optional approach for implementing this function is through lossless isomorphism [18, 27].

4.1 Potential Winrate Isomorphism

Potential winrate isomorphism (PWI) is a signal abstraction that classify signal infosets based on its potential winrate features. These features focus on the distribution of a player's winrate over terminal signals after passing through a given signal infoset, without considering the history of how the player reached the signal infoset. Specifically, for player i in phase r, the potential winrate feature associated with $\vartheta \in \Theta_i^{(r)}$ is defined as

$$pf_i^{(r)}(\vartheta) = (pf_i^{(r),0}(\vartheta), pf_i^{(r),1}(\vartheta), \ldots, pf_i^{(r),N}(\vartheta)), \tag{1}$$

where

- $pf_i^{(r),0}(\vartheta)$ denotes the probability that player i ranks lower than least one other player in the terminal signals, after passing through ϑ.
- $pf_i^{(r),l}(\vartheta)$, for $l > 0$, denotes the probability that player i ranks no lower than any other player and ranks higher than exactly $l - 1$ other players in the terminal signals, after passing through ϑ.

In the terminal phase, the winrate feature is calculated by directly statisticing the game outcomes for players in the given signal infoset. Moreover, in the non-terminal phases, we use a recursive approach to simplify the computation of the winrate feature, thereby avoiding the need to enumerate every signal infoset down to the terminal phase. The recursive formula is

$$pf_i^{(r),l}(\vartheta) = \sum_{\substack{\vartheta^{(r+1)} \in \Theta_i^{(r+1)} \\ \vartheta \sqsubseteq \vartheta^{(r+1)}}} pf_i^{(r+1),l}(\vartheta^{(r+1)}) Pr\{\vartheta^{(r+1)} | \vartheta\} \tag{2}$$

	Preflop	Flop		Turn			River			
Recall	0	0	1	0	1	2	0	1	2	3
KRWI	169	1028325	1123442	1850624	34845952	37659309	20687	33117469	529890863	577366243
KROI	100	1137132	1241210	2337912	38938975	42040233	20687	39792212	586622784	638585633
W/O (%)	100.0	90.43	90.51	79.16	89.49	89.58	100.0	83.23	90.33	90.41

Table 1: The number of abstracted signal infosets identified by KRWI, and KROI in each phase and k of HUNL&HUNLE, with W/O indicating the ratio identified by PWI and POI.

	Preflop	Flop	Turn	River
LI	169	1286792	55190538	2428287420
PWI	169	1028325	1850624	20687
POI	169	1137132	2337912	20687
W/O (%)	100.0	90.43	79.16	100.0

Figure 2: The number of abstracted signal infosets identified by LI, PWI, and POI in each phase of HUNL&HUNLE, with W/O indicating the ratio identified by PWI and POI.

The PWI algorithm is derived from the POI algorithm [12], and the details of the PWI algorithm are elaborated in Appendix A.1. Both algorithms use the potential winrate feature to distinguish between different abstracted signal infosets in the terminal phase. However, unlike POI, PWI also uses the potential winrate feature in non-terminal phases to identify different abstracted signal infoset classes, while POI relies on the potential outcome feature (which captures the distribution of the abstracted signal infoset class for future signal infoset). In non-terminal phases, the potential winrate feature is a simplified version of the potential outcome feature. Unsurprisingly, PWI also results in excessive abstraction similar to POI. As shown in Table 2, in heads-up limit hold'em (HULHE) and heads-up no-limit hold'em (HUNL), the number of abstracted signal infosets identifiable by lossless isomorphism increases with each phase, indicating that the game becomes increasingly complex. However, the number of abstracted signal infosets identifiable by PWI and POI first increases and then decreases, showing a spindle-shaped pattern. And we observed that when only future information is considered, winrate-based features may lead to greater information loss compared to outcome-based features. For instance, in the River phase, the number of abstracted signal infosets identified by PWI is only 79.16% of that identified by POI.

4.2 K-Recall Winrate Isomorphism

As Fu et al. [12] mentioned, supplementing historical information can enhance the ability of signal abstraction to identify abstracted signal infosets. Inspired by KROI's construction approach, we developed the k-recall winrate isomorphism (KRWI). The key difference is that instead of using k-recall outcome features to distinguish between different signal infosets, KRWI utilizes k-recall winrate features.

In a game with signal perfect recall, all signals within the signal infoset ϑ have their predecessors at phase r', which belong to the identical signal infoset ϑ'. For player i at phase r, the signal infoset $\vartheta \in \Theta_i^{(r)}$ has a k-recall winrate feature ($k < r$) represented as a numerical array with a dimension of $(k+1)(N+1)$:

$$rf_i^{(r,k)}(\vartheta) = (pf_i^{(r)}(\vartheta); pf_i^{(r-1)}(\vartheta); \dots; pf_i^{(r-k)}(\vartheta)) \tag{3}$$

When r' is less than r, $pf_i^{(r')}(\vartheta)$ denotes the potential winrate feature for the predecessor signal infoset ϑ' of ϑ at phase r'. Since we have stored all the potential winrate features of $\vartheta \in \Theta_i^{(r)}$ through $\mathcal{PC}_i^{(r)}, \mathcal{PD}_i^{(r)}$ and assigned them unique identifiers in Algorithm A1. To save storage space and facilitate retrieval, what we actually store is

$$rfi_i^{(r,k)}(\vartheta) = (\mathcal{PD}_i^{(r)}[\vartheta], \mathcal{PD}_i^{(r-1)}[\vartheta], \dots, \mathcal{PD}_i^{(r-k)}[\vartheta])) \tag{4}$$

$\mathcal{PD}_i^{(r')}[\vartheta]$ is the identifier for the potential winrate feature of the predecessor ϑ' of ϑ in the r' phase, $r' \leq r$. For algorithm details, please refer to Appendix A.2.

Just as the potential winrate feature is a simplified version of the potential outcome feature, the k-recall winrate feature is a simplified version of the k-recall outcome feature. Table 1 shows the number of signal infosets that KRWI and KROI can identify and their ratio in HUNL&HULHE. We were pleasantly surprised to find that while the ratio of PWI to POI resolution can drop below 80%, when k is set to its maximum value, i.e. $r-1$, the ratio of KRWI to KROI resolution can reach nearly 90% at a minimum, with most of the information preserved. Also, we can easily observe that the number of abstracted signal infosets identified by KRWI is much higher than that identified by POI.

5 K-Recall Winrate Abstraction with Earth Mover's Distance

Fu et al. [12] introduced potential and k-recall outcome features, referred to as outcome-based features, to distinguish different abstracted signal infosets. In the previous section, we developed potential and k-recall winrate features, termed winrate-based features, for the same purpose. In these two methods, Each unique feature corresponds to a single abstracted signal infoset. Intuitively, we can infer that feature similarity might reflect the similarity among abstracted signal infosets, enabling further abstraction and compression for application in large-scale games. However, assessing similarity with outcome-based features is challenging because the identification code indicates only the category, without reflecting the degree of similarity. In contrast, winrate-based features represent winrates, which are inherently comparable, allowing for an easy definition of distances between them.

For the signal information sets ϑ, ϑ' of player i at phase r, we can define the distance of their k-recall winrate feature as

$$d(rf_i^{(r,k)}(\vartheta), rf_i^{(r,k)}(\vartheta')) = \sum_{j=0}^{k} w_j \cdot \mathrm{Emd}(pf_i^{(r-j)}(\vartheta), pf_i^{(r-j)}(\vartheta')) \tag{5}$$

Among Equation (5), Emd is the operator used to calculate the earth mover's distance (EMD) [24]. The EMD calculates the distance between two histograms using optimal transport theory. Since it requires solving linear programming equations, the computational complexity of the EMD is sensitive to the dimensionality of the histograms, and approximate algorithms are usually used for larger-scale problems. However, the dimensionality of winrate-based features is small, with a dimension of 3 in a two-player scenario, so we attempt to use a fast algorithm for accurately computing the EMD [5]. $w_0, \ldots, w_k$ are hyperparameters used to control the importance of EMD at each phase $r, \ldots, r-k$. We use the KMeans++ algorithm [3], combined with the distance of their k-recall winrate feature, to cluster the abstracted signal infosets of KRWI. We named this algorithm KrwEmd.

Although calculating EMD on small-dimensional histograms is already very fast, clustering actual Texas Hold'em still faces a significant computation. For example, for the River phase of HUNL&HULHE, the clustering input size of the KRWI abstracted signal infoset is approximately 5.8×10^8. When we set the number of centroids to 20000, a single Kmeans++ iteration takes about 19000 core hours on a computer with a 2.40GHz clock frequency, which is a significant time cost. Therefore, we need to find ways to reduce this time cost. We have developed an accelerated algorithm, please refer to Appendix A.3 for details.

6 Experimental Setup

We conducted experiments on the Numeral211 Hold'em [12] testbed. Numeral211 is a two-player three-phase Taxes Hold'em-style game with more complex hand systems than the Leduc Hold'em [26] and Rhode Island Hold'em [25] test environments, making it suitable for studying hand abstraction issues. Detailed rules are included in Appendix B. Table 3 shows the number of abstracted signal infosets recognized by KRWI and KROI, along with lossless isomorphism, in Numeral211 Hold'em.

	Preflop	Flop		Turn		
LI	100	2260		62020		
Recall	0	0	1	0	1	2
KRWI	100	2234	2248	3957	51000	51070
KROI	100	2250	2260	3957	51176	51228
W/O (%)	100.0	99.29	99.47	100.0	99.67	99.69

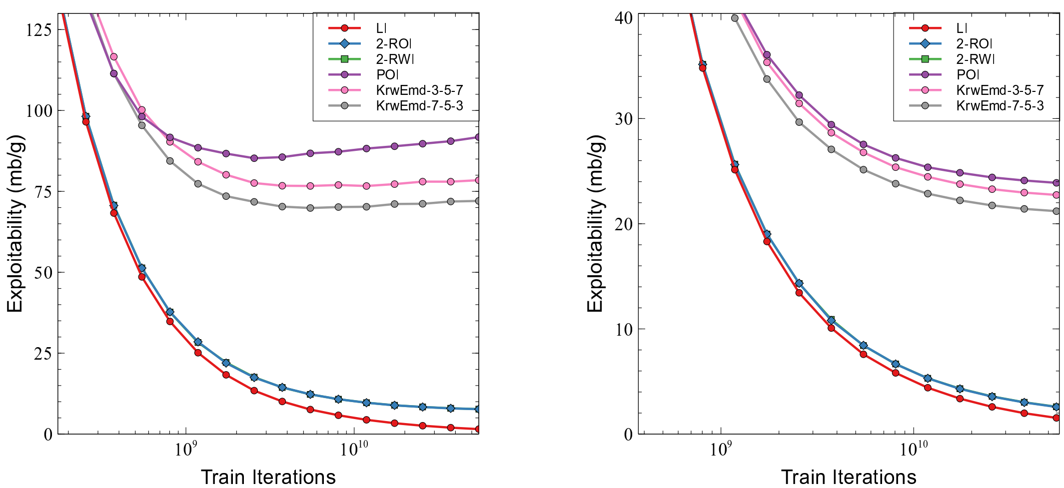
Figure 3: The number of abstracted signal infosets identified by LI, PWI, and POI in each phase of HUNL&HUNLE, with W/O indicating the ratio identified by PWI and POI.

Let $\alpha = (\alpha_1, \alpha_2)$ be the signal abstraction we would like to assess. We will test the strength of the signal abstraction by measuring exploitability of the approximate equilibrium derived using the

CSMCCFR algorithm [30, 22] in different abstracted signal infoset scales. We gauge the performance over exploitability. For doing that, we consider both symmetric and asymmetric abstraction scenarios.

In this symmetric abstraction setting, we measure the exploitability of approximate equilibrium that is yielded when both the players in the game employ signal abstraction in the original game. However, it may lead to the abstraction pathology [28]. To avoid such problems, we illustrate the theoretical performance of the signal abstraction under evaluation through asymmetric abstraction. The approximate equilibrium in the signal abstracted games $\tilde{\Gamma}^{(\alpha_1,\Theta_2)}$ and $\tilde{\Gamma}^{(\Theta_1,\alpha_2)}$ is obtained to obtain $\pi^{*,1}$ and $\pi^{*,2}$, respectively. Finally, we concat the two strategies to get $\pi' = (\pi_1^{*,1}, \pi_2^{*,2})$ and check the exploitability of π'.

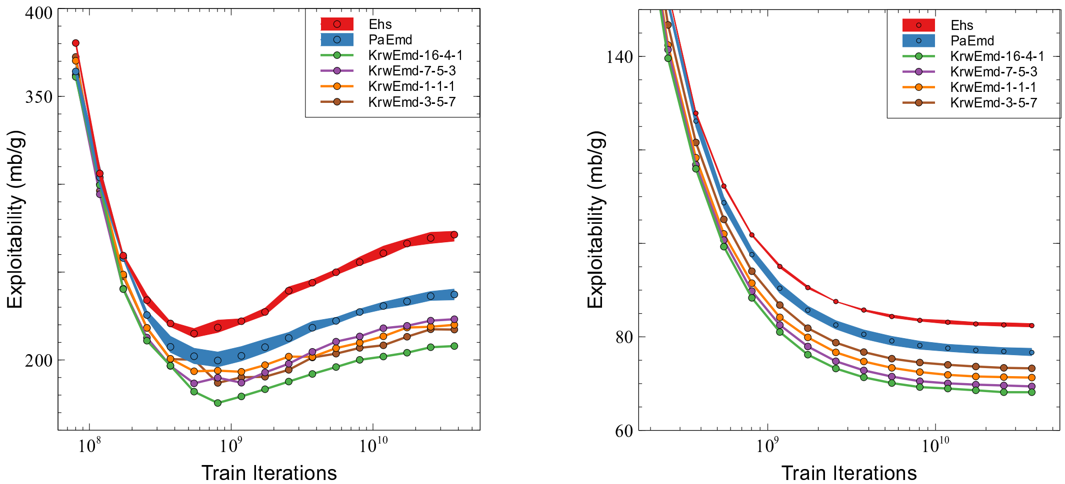
7 Experiment



(a) (b)

Figure 4: Full abstraction setting experiment, trained for 5.5×10^{10} iterations.

Firstly, we provide an evaluation of the performance of KRWI (2-RWI) compared with KROI (2-ROI) and POI (0-ROI) approaches and lossless isomorphism. We keep the most abstracted signal infosets identified under the full abstraction setting. Note that POI is the common refinement of existing signal abstraction algorithms that only consider future information. And, since previous works cannot control the number of abstracted infoset, they cannot justify their performance in that considering historical information in signal abstraction was better than that in signal abstraction with the same number of abstracted infoset. To investigate this issue, we included KrwEmd and set the clustering scale to be consistent with POI. Note here, that 2-RWI and 2-ROI share the same capability of infoset recognition in Preflop and Flop, while POI is only a little bit worse than 2-RWI and 2-ROI in Flop. Thus, we can directly allow clustering of KrwEmd abstraction use the abstracted signal infosets identified by POI in Preflop and Flop, and only perform clustering in River. Here, we design four sets of hyper-parameters: (w_0, w_1, w_2), i.e., exponentially decreasing: $(16, 4, 1)$, linearly decreasing: $(7, 5, 3)$, constant: $(1, 1, 1)$, and increasing: $(3, 5, 7)$ in the importance of historical information. We only show the result of best- and worst-performing parameters (to make the figure neat). The full figures appear in the Appendix C. Figure 4a shows the result of symmetric abstraction, while Figure 4b shows the result of asymmetric abstraction. We observed that both symmetric and asymmetric abstractions maintained consistent abstraction performance without abstraction pathologies. As expected, overfitting was observed in the symmetric abstraction scenario while in the asymmetric scenario overfitting was significant only for POI. The performance difference between 2-RWI and 2-ROI is small, which means that under the full abstraction setting, using simple winrate-based features instead of complex outcome-based features can achieve nearly the same performance. Even with the worst parameter configuration (increasing importance), KrwEmd with the same number of abstracted signal inforsets as POI still outperforms POI.



(a) (b)

Figure 5: Performance comparison of KrwEmd versus other imperfect recall signal abstraction algorithms considering only future information, trained for 3.7×10^{10} iterations.

Next, we compared the performance of KrwEmd with the currently applied signal abstraction algorithms Ehs and PaEmd. It should be noted that POI is the common refinement both for Ehs and PaEmd, meaning that the maximum number of abstracted signal infosets they can recognize will not exceed that of POI. Thus, we set a compression rate that is 10 times lower than that of POI, while not performing abstraction for Preflop. The final number of abstracted infosets is set to $(100, 225, 396)$. To exclude the influence of random events on performance, we generated 3 sets of abstractions for Ehs and PaEmd each. KrwEmd used hyperparameters $(w_{3,0}, w_{3,1}, w_{3,2}; w_{2,0}, w_{2,1})$ in Flop and River, which are exponentially decreasing $(16, 4, 1; 4, 1)$, linearly decreasing $(7, 5, 3; 5, 3)$, constant $(1, 1, 1; 1, 1)$, and increasing $(3, 5, 7; 5, 7)$ in the importance of historical information. Additionally, since PaEmd uses approximate EMD calculations, its approximate distance is asymmetric, making it difficult for the algorithm to converge. We truncated after 1000 iterations on a single core, with an average cost of 1427.7s, while Ehs and KrwEmd both achieved convergent clustering results, requiring an average of 12.3 and 96.7 iterations, with average time costs of 11.2s and 341.4s, respectively.

Figure 5a shows the results of symmetric abstraction experiments, while Figure 5b shows the results of asymmetric abstraction experiments. We observed that both symmetric and asymmetric abstractions maintained consistent abstraction performance, similar to the full abstraction scenario, without significant abstraction pathologies. The experimental results show that KrwEmd's performance is far superior to that of Ehs and PaEmd under all parameter settings. Our experiments also confirmed that, despite PaEmd's convergence issues, it is indeed a more effective abstraction algorithm than Ehs. Additionally, we further validated that the importance of historical information decreases progressively from bottom to top, although this time the best-performing parameter was exponentially decreasing rather than linearly decreasing as in the previous experiment.

These two experiments validate that considering historical information is indeed more effective than considering future information only in signal abstraction even in imperfect recall setting.

8 Conclusion

This research introduces the first imperfect recall signal abstraction algorithm that considers historical information. This algorithm has the ability to adjust the scale of the abstracted signal infosets. Based on this, we fully verified that the imperfect recall signal abstraction and abstraction algorithms considering historical information is superior to that only considering future information. Therefore, the KrwEmd algorithm has replaced the PaEmd algorithm and become the SOTA in this field. Based on the KrwEmd algorithm, we are expected to build a stronger Texas Hold'em AI.

References

[1] David Abel. A theory of state abstraction for reinforcement learning. In *AAAI conference on artificial intelligence*, volume 33, pages 9876–9877, 2019.

[2] David Abel, Nate Umbanhowar, Khimya Khetarpal, Dilip Arumugam, Doina Precup, and Michael Littman. Value preserving state-action abstractions. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1639–1650, 2020.

[3] David Arthur and Sergei Vassilvitskii. k-means++ the advantages of careful seeding. In *ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 1027–1035, 2007.

[4] D Billings, N Burch, A Davidson, R Holte, J Schaeffer, T Schauenberg, and D Szafron. Approximating game-theoretic optimal strategies for full-scale poker. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 3, pages 661–668, 2003.

[5] Nicolas Bonneel, Michiel van de Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement Interpolation Using Lagrangian Mass Transport. *ACM Transactions on Graphics (SIGGRAPH ASIA 2011)*, 30(6), 2011.

[6] Noam Brown and Tuomas Sandholm. Regret transfer and parameter optimization. In *AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[7] Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

[8] Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365 (6456):885–890, 2019.

[9] Noam Brown, Sam Ganzfried, and Tuomas Sandholm. Hierarchical abstraction, distributed equilibrium computation, and post-processing, with application to a champion no-limit texas hold'em agent. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 7–15, 2015.

[10] Jiří Čermák, Branislav Bošansky, and Viliam Lisy. An algorithm for constructing and solving imperfect recall abstractions of large extensive-form games. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 936–942, 2017.

[11] Jiří Čermák, Viliam Lisỳ, and Branislav Bošanskỳ. Automated construction of bounded-loss imperfect-recall abstractions in extensive-form games. *Artificial Intelligence*, 282:103248, 2020.

[12] Yanchang Fu, Junge Zhang, Dongdong Bai, Lingyun Zhao, Jialu Song, and Kaiqi Huang. Expanding the resolution boundary of outcome-based imperfect-recall abstraction in games with ordered signals. *arXiv preprint arXiv:2403.11486*, 2024.

[13] Sam Ganzfried and Tuomas Sandholm. Action translation in extensive-form games with large action spaces: axioms, paradoxes, and the pseudo-harmonic mapping. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 120–128, 2013.

[14] Sam Ganzfried and Tuomas Sandholm. Potential-aware imperfect-recall abstraction with earth mover's distance in imperfect-information games. In *AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[15] Andrew Gilpin and Thomas Sandholm. Expectation-based versus potential-aware automated abstraction in imperfect information games: An experimental comparison using poker. In *National Conference on Artificial Intelligence (NCAI)*, volume 3, pages 1454–1457, 2008.

[16] Andrew Gilpin and Tuomas Sandholm. A competitive texas hold'em poker player via automated abstraction and real-time equilibrium computation. In *National Conference on Artificial Intelligence (NCAI)*, volume 21, page 1007. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[17] Andrew Gilpin and Tuomas Sandholm. Better automated abstraction techniques for imperfect information games, with application to texas hold'em poker. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1–8, 2007.

[18] Andrew Gilpin and Tuomas Sandholm. Lossless abstraction of imperfect information games. *Journal of the ACM (JACM)*, 54(5):25–es, 2007.

[19] Andrew Gilpin, Tuomas Sandholm, and Troels Bjerre Sørensen. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of texas hold'em poker. In *National Conference on Artificial Intelligence (NCAI)*, volume 22, page 50. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

[20] Michael Johanson, Neil Burch, Richard Valenzano, and Michael Bowling. Evaluating state-space abstractions in extensive-form games. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 271–278, 2013.

[21] Christian Kroer and Tuomas Sandholm. Discretization of continuous action spaces in extensive-form games. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 47–56, 2015.

[22] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte carlo sampling for regret minimization in extensive games. *International Conference on Neural Information Processing Systems (NeurIPS)*, 22, 2009.

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

[24] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40:99–121, 2000.

[25] Jiefu Shi and Michael L Littman. Abstraction methods for game theoretic poker. In *Computers and Games: Second International Conference, CG 2000 Hamamatsu, Japan, October 26–28, 2000 Revised Papers 2*, pages 333–345. Springer, 2001.

[26] Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: opponent modelling in poker. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 550–558, 2005.

[27] Kevin Waugh. A fast and optimal hand isomorphism algorithm. In *AAAI Workshop on Computer Poker and Incomplete Information*, 2013.

[28] Kevin Waugh, David Schnizlein, Michael Bowling, and Duane Szafron. Abstraction pathologies in extensive games. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 2, pages 781–788, 2009.

[29] Kevin Waugh, Martin Zinkevich, Michael Johanson, Morgan Kan, David Schnizlein, and Michael Bowling. A practical use of imperfect recall. In *Symposium on Abstraction, Reformulation and Approximation (SARA)*, 01 2009.

[30] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pages 1729–1736, 2007.

Algorithm A1 Potential Winrate Isomorphism

Require:

$r = 1, \ldots, R$. Phases.

$\Theta_i = \bigcup_{r=1}^{R} \Theta_i^{(r)}$. Signal infoset space for player i.

$Index_i(r, \cdot) : \Theta_i^{(r)} \mapsto \mathbb{N}$. Signal infoset index function for player i.

procedure PotentialWinrateIsomorphism(Θ_i)
  for $r = R$ to 1 **do**
    if $r == R$ **then**
      FeatureFunc $\leftarrow$ PotentialWinrateFeatureLastPhase($\cdot$).
    else
      FeatureFunc $\leftarrow$ PotentialWinrateFeature($\cdot$, r, $\mathcal{PC}_i^{(r+1)}$, $\mathcal{PD}_i^{(r+1)}$).
    end if
    $(\mathcal{PC}_i^{(r)}, \mathcal{PD}_i^{(r)}) \leftarrow$ IsomorphismConstructor(r, $\Theta_i^{(r)}$, FeatureFunc).
  end for
  return $(\mathcal{PC}_i^{(1)}, \mathcal{PD}_i^{(1)}), \ldots, (\mathcal{PC}_i^{(R)}, \mathcal{PD}_i^{(R)})$.
end procedure
procedure PotentialWinratesFeatureLastPhase(ϑ)
  return $pf_i^{(R)}(\vartheta)$ ▷ compute according Equation (1)
end procedure
procedure PotentialWinrateFeature(ϑ, r, $\mathcal{PC}_i^{(r+1)}$, $\mathcal{PD}_i^{(r+1)}$)
  $feature_\vartheta \leftarrow$ zero array with length $N + 1$
  for $\vartheta' \in \Theta_i^{(r+1)}$, such that $\exists \theta' \in \vartheta', \exists \theta \in \vartheta$: $\varsigma(\theta'|\theta) > 0$ **do**
    $idx \leftarrow Index_i(r + 1, \vartheta')$, $abs \leftarrow \mathcal{PD}_i^{(r+1)}[idx]$, $feature_{\vartheta'} \leftarrow \mathcal{PC}_i^{(r+1)}[abs]$.
    for $j = 0$ to N **do**
      $feature_\vartheta[j] \leftarrow feature_\vartheta[j] + feature_{\vartheta'}[j] Pr\{\vartheta'|\vartheta\}$
    end for
  end for
end procedure

A Algorithm Details

A.1 Potential Winrate Isomorphism

Algorithm A1 describes the computation process for potential winrate isomorphism. This algorithm operates in reverse, starting from the game's final phase R.

A.2 K-Recall Winrate Isomorphism

Algorithm A2 constructs the k-recall winrate isomorphism using the k-recall winrate feature. This process requires the prior construction of the potential winrate isomorphism map $\mathcal{PD}_i^{(r)}$ using Algorithm A1.

A.3 Accelerating Distance Computing for K-Recall Winrate Features

According to Equation (5), we note that the distance calculation between a k-recall winrate isomorphism class and a centroid's k-recall winrate feature can be decomposed into k+1 pairs of potential winrate feature EMD calculations. The potential winrate feature of the hand remains unchanged, while only the potential winrate feature of the centroid changes. Decomposing the calculation into the EMDs of potential winrate features involves significantly fewer computations than directly calculating the EMD of two k-recall winrate features. Specifically, for the River phase of HUNL&HULHE, we have the compression ratio as $\frac{169+1028325+1850624+20687}{529890863} = \frac{2899805}{529890863} = 0.0054725$.

Algorithm A3 describes how we accelerate the batch EMD computation between a centroid and all KRWI classes' k-recall winrate features. It should be noted that the K-recall winrate feature involved in the calculation of the centroid in the algorithm is in the form of Equation (3), while the K-recall winrate feature in $\mathcal{RC}^{(r,k)}$ is in the form of Equation (4). This method reduced the computational

Algorithm A2 K-Recall Winrate Isomorphism

Require:

$r = 1, \dots, R$. Phases.

$\Theta_i^{(r)}$. Signal infoset space for player i.

$Index_i(r, \cdot) : \Theta_i^{(r)} \mapsto \mathbb{N}$. Signal infoset index function for player i.

$\mathcal{PD}_i^{(r)} : \mathbb{N} \mapsto \mathbb{N}$. Potential winrate isomporphism map.

```
procedure KRecallWinrateIsomorphism(Θ_i, k)
    for r = 1 to R do
        k' ← Min(r − 1, k).
        FeatureFunc ← KRecallWinrateFeature(·, r, k').
        (RC_i^(r,k'), RD_i^(r,k')) ← IsomorphismConstructor(r, Θ_i^(r), FeatureFunc).
    end for
    return (RC_i^(1,0), RD_i^(1,0)), ..., (RC_i^(k+1,k), RD_i^(k+1,k)), ..., (RC_i^(R,k), RD_i^(R,k)).
end procedure
procedure KRecallWinratesFeature(ϑ, r, k)
    initial a empty vector feature.
    for s = r to r − k do
        ϑ' ← the predecessor signal infoset of ϑ in the s phase for player i.
        idx ← Index_i(s, ϑ'), abs ← PD_i^(s)[idx].
        Append feature with abs.
    end for
    return feature
end procedure
```

cost of EMD from 19000 core hours to approximately 104 core hours, which is significantly lower than the time cost of summarizing the distance for each KRWI class, which is about 524 core hours and is an unavoidable $O(1)$ cost.

The distance batch calculation for each centroid can be processed independently and distributed across tens of multi-core computer (e.g. 96-core computers), with each computer responsible for calculating the features of some centroids in one iteration, which are then aggregated. Using this technique, we can reduce an iteration to a few hours, which is acceptable for Texas Hold'em AI training.

B Numerall211 Hold'em Rules

Numeral211 Hold'em is played according to the following rule:

1. **Ante:** Each player antes 5 chip into the pot at the start of the hand.
2. **Hole Card:** Both players are dealt one private card face down, known as the hole card.
3. **Deck:** The deck consists of a standard poker deck, excluding the Jokers, Kings, Queens, and Jacks, resulting in a total of 40 cards. There are four suits: spades (♠), hearts (♡), clubs (♣), and diamonds (♢), each containing ten cards numbered 2 through 9, and including the ten (T) and ace (A).
4. **First Betting Phase:** Following the distribution of hole cards, a phase of betting occurs. Players can choose to check or bet, with the bet size set at 10 chips.
5. **Flop:** After the initial betting phase, a single community card, termed the flop, is revealed from the deck.
6. **Second Betting Phase:** Another phase of betting takes place after the flop, with the bet size increasing to 20 chips.
7. **Turn:** After the Second betting phase, another community card, termed the turn, is revealed from the deck.
8. **Third Betting Phase:** Another phase of betting takes place after the turn, with the bet size still set at 20 chips.

Algorithm A3 Distance Batch

Require:

$r = 1, \ldots, R$. Phases.
$\mathcal{RC}_i^{(r,k)} : \mathbb{N} \mapsto \mathbb{N}^{k+1}$. K-recall winrate feature set.
$\mathcal{PC}_i^{(r)} : \mathbb{N} \mapsto [0,1]^{N+1}$. Potential winrate feature set.
$\mathcal{PD}_i^{(r)} : \mathbb{N} \mapsto \mathbb{N}$. Potential winrate isomporphism map.
$rc = (pc^{(r)}, \ldots, pc^{(r-k)})$. K-recall winrate feature of the input centroid.

Ensure:

Distances of all k-recall winrate feature with centroid.

procedure DISTANCEBATCH($w_0, \ldots, w_k$, rc, r, k)
  Initial phase s empty earth mover's distance vector $EmdDis^{(s)}$ for $s = r, \ldots, r-k$.
  Initial empty output distance vector Dis.
  for t = 0 to k **do**
    for pf in $\mathcal{PC}_i^{(s)}$ **do**
      Append $EmdDis^{(r-t)}$ with Emd$(pf, rc[t])$
    end for
  end for
  for rfi in $\mathcal{RC}_i^{(r,k)}$ **do**
    $dis \leftarrow 0$.
    for t = 0 to k **do**
      $dis \leftarrow dis + w_t * EmdDis^{(r-t)}[\mathcal{PD}_i^{(r-t)}[rfi[t]]]$.
    end for
    Append Dis with dis.
  end for
  return Dis.
end procedure

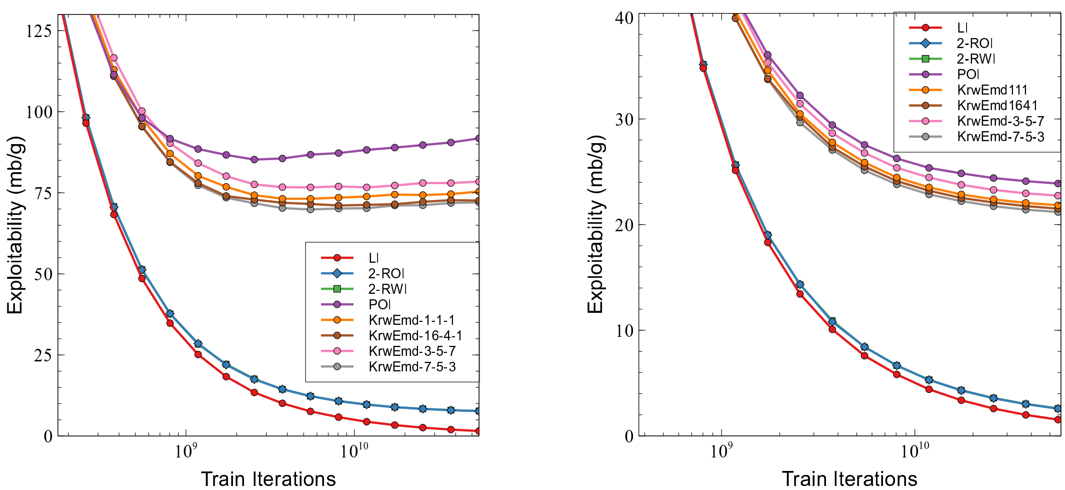
9. **Showdown:** If neither player folds, a showdown occurs. Players reveal their cards, aiming to form the best possible hand. The player with the highest-ranked hand wins the pot. In the case of a tie, the pot is split evenly. The Table 2 show the hand ranks of Numeral211 Hold'em.
10. **Betting Options:** Throughout the game, players have options to fold, call, or raise. In each betting phase, the total sum of bets and raises is limited to a maximum of 4, with fixed bet sizes of 10 chips in the first phase and 20 chips in the last two betting phases.

C Supplementary Data for Experiment 1

Figure 6 show all of the result in experiment 1.

Rank	Hand	Prob.	Description	Example
1	Straight flush	0.00321	3 of cards with consecutive rank and same suit. Ties are broken by highest card.	T♠9♠8♠2♣
2	Three of a kind	0.01587	3 of cards with the same rank. Ties are broken by the card's rank.	T♠T♡T♣2♣
3	Straight	0.04347	3 of cards with consecutive rank. Ties are broken by the highest card rank.	T♠9♡8♣2♢
4	Flush	0.15799	3 of cards with the same suit. Ties are broken by the highest card rank, then second highest card rank, then third highest card rank.	T♠8♠6♠2♣
5	Pair	0.34065	2 of cards with the same rank. Ties are broken by the rank of the pair, then by the rank of the third card.	T♠T♡8♣2♢
6	High card	0.43881	None of the above. Ties are broken by comparing the highest ranked card, then the second highest ranked card, and then the third highest ranked card	T♠8♡6♣2♢

Table 2: Hand ranks of Numeral211 Hold'em



(a) (b)

Figure 6: All data within experiment 1

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have clearly defined our scope and contributions in both the abstract and introduction sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.

- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [NA]

   Justification: This paper introduces a novel hand abstraction algorithm that has been experimentally validated to outperform the previous state-of-the-art (SOTA) algorithm, PaEmd, and no significant flaws have been identified thus far.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper introduces a novel algorithm and validates its effectiveness through experiments, without involving theory or proofs.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.

- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided detailed information on the testbed, evaluation metrics, and experimental scenarios in Section 6. Additionally, specific experimental parameters and equipment are given in Section 7. Therefore, we have included sufficient details in the paper to reproduce the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The experiments in this paper are time-consuming, and we utilized a large number of machines to simultaneously conduct various parts of the experiments. Currently, we do not have a ready-to-use script for one-click deployment of the experiments (the time required to run on a single computer is unacceptable). In the future, we will open-source this work and provide the code to reproduce these experiments.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: As stated in the reproducibility statement, we have provided detailed information on the testbed, evaluation metrics, and experimental scenarios in Section 6. Additionally, specific experimental parameters and equipment are given in Section 7.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the long experimental time and limited sample size, error bars cannot be provided. In the first experiment (Figure 4), the baseline settings adopt fixed abstraction settings and have a large performance gap, so the performance of strategies solved by CSMCCFR is stable and not easily affected by random factors. In the second experiment (Figure 5), random factors may indeed affect individual experimental data. Therefore, we sampled the control group multiple times and drew the performance range, which is far lower than the performance of our algorithm under the worst parameters, which also proves the effectiveness of the algorithm.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss the computational resources and time cost of the experiments in Sections 5, 7, and Appendix A.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We carefully review the NeurIPS Code of Ethics and ensure that the research aligns with it in all aspects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the impact of this work in Section 8. As noted, this work represents the SOTA in hand abstraction algorithms and could be used to create more powerful Texas Hold'em AI.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work focuses solely on introducing a more efficient algorithm for hand abstraction and does not involve the release of data or models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: This paper provides comprehensive citations for all comparative methods involved, and all comparison experiments were re-implemented without using existing tools or code.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release any new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.