Expanding the Resolution Boundary of Outcome-Based Imperfect-Recall Abstraction in Games with Ordered Signals

Yanchang Fu[1, 2] fuyanchang2020@ia.ac.cn
Junge Zhang[2] jgzhang@nlpr.ia.ac.cn
Dongdong Bai[3] baidongdong@nudt.edu.cn
Lingyun Zhao[1, 2] zhaolingyun2021@ia.ac.cn
Jialu Song[1, 2] songjialu2023@ia.ac.cn
Kaiqi Huang[2*] kqhuang@nlpr.ia.ac.cn
[1] *School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China*
[2] *Center for Research on Intelligent System and Engineering, Institute of Automation, Chinese Academy of Sciences, Beijing, China*
[3] *China RongTong Artificial Intelligence Research Center, Beijing, China*

Abstract

Hand abstraction has successfully contributed to the development of powerful AI in Texas Hold'em, a popular testbed for imperfect-information games. However, the hand abstraction task lacks the necessary tools for modeling within the general imperfect-information games framework, impeding not only theoretical study but also guiding algorithm design and evaluation. This paper aims to rigorously mathematically model the hand abstraction task. It then identifies the flaw of excessive abstraction in outcome-based imperfect-recall hand abstraction algorithms and addresses it accordingly. We first refines the games of ordered signals model to enhance its conciseness and expand its descriptive capacity, applying it to model Texas Hold'em-style games. By transitioning to games with ordered signals, infoset abstraction and action abstraction in imperfect-information games are decoupled, allowing for independent study of signal abstraction, which provides a mathematical framework for the hand abstraction task. We introduce potential outcome isomorphism (POI) with the aim of constructing the highest resolution signal abstraction considering future outcomes only, and identify its issue of excessive abstraction. Additionally, a novel common refinement principle is introduced to describe the resolution boundary of signal abstraction algorithms. Futher, We demonstrate that POI serves as a common refinement for leading outcome-based imperfect-recall hand abstraction algorithms, such as E[HS] and PA&PAEMD. Consequently, excessive abstraction also inherently affects these algorithms, leading to suboptimal performance. Finally, a higher-resolution hand abstraction, k-recall outcome isomorphism (KROI), is constructed by considering early-game information. Experimental results highlight compromised performance of strategies developed through low-resolution signal abstraction. KROI provides a framework to guide the design of higher-resolution outcome-based imperfect-recall abstraction algorithms.

Keywords: game theory, imperfect-information games, games with ordered signals, computer poker , automated abstraction, hand abstraction, signal abstraction, imperfect-recall

[*] Corresponding author

1 Introduction

In the realm of artificial intelligence (AI) research, developing competitive AI agent in large-scale adversarial imperfect-information games is a pressing and important challenge. Texas Hold'em poker, as a quintessential example of such games, offers an ideal testbed for AI research.

Recent years have witnessed significant advancements in AI within the context of Texas Hold'em poker. Pioneering systems like DeepStack (Moravčík et al., 2017), Libratus (Brown and Sandholm, 2018), and Pluribus (Brown and Sandholm, 2019) have demonstrated extraordinary capabilities against professional human players. Much of their success owes to innovative hand abstraction algorithms. These algorithms transform complex hand combinations into simpler, more manageable representative classes, effectively condensing the vast decision space. Such simplification facilitates strategic searches in large-scale games with limited computational resources while maintaining the robustness of the strategies.

Traditionally, Texas Hold'em is modeled as an imperfect-information game, where the task of hand abstraction is categorized as an infoset abstraction task. This modeling paradigm is notably expansive, capturing the essence of Texas Hold'em yet not fully encompassing its nuanced complexities. In Texas Hold'em, hands and actions collectively form the foundational elements that define the infoset, with hand abstraction serving as just one facet of the broader concept of infoset abstraction. The task of hand abstraction poses a unique and independent challenge; however, the model of imperfect-information games fails to capture key concepts such as stages, which are essential in the process of hand abstraction. Previous studies have attempted to engineer the task of hand abstraction from various perspectives, yet this task still necessitates a more precise mathematical model for accurate depiction. Gilpin and Sandholm (2007b) proposed a more detailed game modeling method for this specific type of imperfect-information game, termed ordered signal games. This model skillfully applies the concepts of signals and public action sequences to distinctly outline the two independent components of the infoset in Texas Hold'em-style games: hands and action sequences. However, the model's reliance on forests to depict the various game stages contributes to its complexity. Moreover, it is confined to describing scenarios where the signal distribution is governed by a combinatorial model with replacement.

The first part of this paper aims to provide a mathematical definition of the hand abstraction task. In Section 4, we refine the definition of the games with ordered signals, opting for trees instead of forests to simplify the model, and we generalize the distribution of signals, thereby broadening the applicability of the games with ordered signals model. In Section 5, we formally define the tasks of signal abstraction and action abstraction within the context of games with ordered signals, employing signal abstraction to model the task of hand abstraction.

In the context of signal abstraction, Gilpin and Sandholm (2007b) developed a lossless abstraction, lossless isomorphism (LI), grounded in the game's rules and showed that the Nash equilibrium of the abstracted game, derived from LI, can be seamlessly mapped back to the Nash equilibrium of the original game. Although LI offers substantial theoretical insights, its practical implementation in artificial intelligence development is constrained by the large scale of the infosets even after abstraction. Currently, in the domain of hand abstraction for Texas Hold'em, imperfect-recall clustering algorithms based on showdown outcomes have achieved success, with systems like DeepStack, Libratus, and Pluribus employing this method. However, this raises a series of questions:

- To what extent can outcome-based imperfect-recall algorithms approximate LI? In other words, do these algorithms possess a closer resolution boundary than LI?
- If so, is it possible to expand the resolution boundary of outcome-based imperfect-recall algorithms?

In the second part of this paper, we discuss these questions. Section 6 introduces the concept of potential outcome isomorphism (POI), which aims to construct a signal abstraction based on future showdown outcomes only, striving to identify as many abstracted signal infosets as possible. In Section 7, a theoretical tool called Common Refinement is introduced, which is specifically designed to evaluate the resolution boundaries of signal abstraction algorithms. We demonstrate that POI serves as a common refinement for expected hand strength (E[HS]) and potential-aware methods (PA&PAEMD), the dominant outcome-based imperfect-recall signal abstraction algorithms, thereby effectively establishing the resolution boundary for these methods. Further, we expose a significant resolution gap between POI and LI, highlighting the issue of excessive abstraction present in current outcome-based imperfect-recall algorithms. In Section 8, we analyze the reasons behind the excessive abstraction, pinpointing the neglect of game history sequences as a key factor. To address this, we developed K-Recall Outcome Isomorphism (KROI) by incorporating the historical information, thereby extending the resolution boundary of outcome-based imperfect-recall algorithms. Experimental results indicate that under the one-player-abstraction-perspective, strategies derived from KROI are almost on par with LI and significantly outperform those derived from POI. Even in scenarios of symmetric abstraction, KROI demonstrates substantial strength, significantly surpassing POI. KROI provides us with a new outcome-based imperfect-recall abstraction framework, from which higher-resolution signal abstraction algorithms can be derived.

1.1 Related Research

Our research is dedicated to hand abstraction techniques in AI systems for Texas Hold'em-style games, a field originating from the works of Shi and Littman (2001) and Billings et al. (2003). These seminal works introduced the concept of game abstraction, aiming to represent games in simplified forms while preserving essential characteristics. Initially, researchers manually categorized hands based on their experience. The first automated approach to hand abstraction was proposed by Gilpin and Sandholm (2006). Subsequently, Gilpin and Sandholm (2007b) introduced games with ordered signals model for Texas Hold'em and developed LI 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. In their work, Gilpin and Sandholm (2007a) proposed expectation-based clustering method and Gilpin et al. (2007) introduced histogram-based clustering method. The former is known as E[HS], while the latter is referred to as potential-aware method. Subsequent studies by Gilpin and Sandholm (2008) and Johanson et al. (2013) compared E[HS] and potential-aware methods, concluding that the latter holds an advantage in large-scale games. Johanson et al. (2013) also introduced the use of earthmover's distance[1] (EMD) in Potential-aware methods. Ganzfried and Sandholm (2014) presented a more efficient approximation algorithm for earthmover's distance in potential-aware methods, which has been successfully applied in various Texas Hold'em AI systems, marking it as the SOTA work in the field of hand abstraction.

Another area of interest concerns the competitiveness of game-solving strategies after abstraction. Waugh et al. (2009b) noted a non-monotonic relationship between the exploitability of Nash equilibrium in abstracted games and the refinement level of abstraction profiles, a phenomenon termed as abstraction pathology. In a pioneering effort, Sandholm and Singh (2012) developed the first lossy abstraction algorithm that establishes bounds on solution quality for stochastic games. Following this, Kroer and Sandholm (2014) introduced lossy abstraction with quality bounds in general perfect-recall extensive-form games. In practical scenarios, simplified methods with imperfect-recall demonstrate heightened competitiveness, as introduced by Waugh et al. (2009c). Additionally, Lanctot et al. (2012) and Kroer and Sandholm (2016) tailored abstraction algorithms for imperfect-recall extensive-form games under varied conditions, alongside establishing associated solution quality bounds. Essentially, Kroer and Sandholm (2018) unified this collection of research efforts into a cohesive framework while maintaining comparable bounds on solution quality.

2 Preliminary

2.1 Texas Hold'em-style Poker Games

Texas Hold'em, a globally popular multiplayer card game, has gained significant prominence in the fields of game theory and artificial intelligence, particularly in Heads-up No-Limit Hold'em (HUNL) variant, which has become a focal point of research. The game of Texas Hold'em unfolds in up to four stages: Preflop, Flop, Turn, and River. At the beginning of each stage, the dealer, representing elements of randomness, draws a specific number of cards from the deck. During the Preflop stage, the dealer distributes two private cards to each player from a standard poker deck, excluding jokers. Subsequently, in the Flop stage, the dealer reveals three community cards from the remaining deck. In the Turn and River stages, one additional community card is revealed in each. Players have the option to raise/bet, call/check, or fold during their turn. If at any point only one player remains, they win all the chips. If multiple players are still present at the end of the River stage without further raises, the pot is allocated based on each player's betting and the best five-card combination that can be made from his two private cards and the five community cards.

Standard Texas Hold'em, even in Heads-up Limit Hold'em (HULHE) variant, encompasses a vast decision space. To facilitate research, various simplified games have also garnered academic interest, collectively known as Texas Hold'em-style games. For instance, Kuhn Poker (Kuhn, 1950), Leduc Hold'em (Waugh et al., 2009a) and Rhode Island Hold'em (Shi and Littman, 2001) are variants of Texas Hold'em-style games, which employ fewer cards and more streamlined decision-making stages, thereby simplifying the complexity of the game while retaining the core elements. In the experimental section, we introduce Nu-

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

meral211 Hold'em, a simplified game that closely simulates standard Texas Hold'em with more stages and cards. For details, see Appendix D.

2.2 Imperfect-Information Games

Texas Hold'em-style games are often described using the model of imperfect-information games, the mathematical specifics of which are detailed in Appendix A. These games involve complex decision-making processes where players may not have complete knowledge of the overall state of the game. In such games, each player, denoted as i, makes decisions at non-terminal nodes h in the game, leading to either a terminal node z or another non-terminal node. Each non-terminal node is associated with a specific set of actions $\chi(h)$, and the progression of the game can be influenced by random events controlled by a special player, nature c, also called the chance palyer. At terminal nodes z, each player receives a specific payoff $u_i(z)$. A key concept in imperfect-information games is the infoset I, representing the different game nodes indistinguishable to a certain player. Players maintain consistent decision preferences across the nodes within an infoset. The decision preferences of player i across all infosets constitute his strategy σ_i, and the strategies of all players form a strategy profile σ. The (behavioral) strategy profile influences the probability of reaching each node, whether a non-terminal node h or a terminal node z, denoted as $\pi^\sigma(h)$ for non-terminal nodes and $\pi^\sigma(z)$ for terminal nodes. This, in turn, affects the expected payoff for player i in the game, which is calculated as $v_i(\sigma) = \sum_{z \in Z} \pi^\sigma(z) u_i(z)$. In imperfect-information games, each player aims to maximize his expected payoff.

The solution to an imperfect-information game involves finding a strategy profile σ^* that is known as a Nash equilibrium (Nash, 1951; Harsanyi, 1995). In a Nash equilibrium, no player can gain additional payoff by unilaterally changing their strategy. Each player's strategy is the best response to the strategies of others, that is, $v_i(\sigma^*) = \max_{\sigma_i \in \Sigma_i} v_i(\sigma_i, \sigma^*_{-i})$, where σ_{-i} represents the strategy profile of all players other than i within the strategy profile σ. In imperfect-information games, identifying a Nash equilibrium is crucial as it allows players to make optimal decisions amidst uncertainty. To some extent, the Nash equilibrium embodies the optimal solution in these games, especially in the context of two-player scenarios with imperfect-information.

2.3 Current Paradigm for Solving Large-Scale Imperfect-Information Games.

In the field of computational game theory, a widely adopted paradigm for solving large-scale imperfect-information games involves the abstraction of original games into simplified versions by classifying similar states (Sandholm, 2010), as illustrated in Figure 1. This process reduces complexity and facilitates the identification of a Nash equilibrium in the abstracted game, which is then applied back to the original game in hopes of deriving competitive strategies. Typically, the equilibrium found in the abstracted game is expected to closely approximate the Nash equilibrium of the original game. However, this approach has its limitations, one of the main concerns being the **abstraction pathology** (Waugh et al., 2009b), where the degree of abstraction between the original and the abstracted game does not always correlate monotonically with the distance in their respective Nash equilibrium. This can lead to unpredictable outcomes when applying the equilibrium from the abstracted game to the original one.

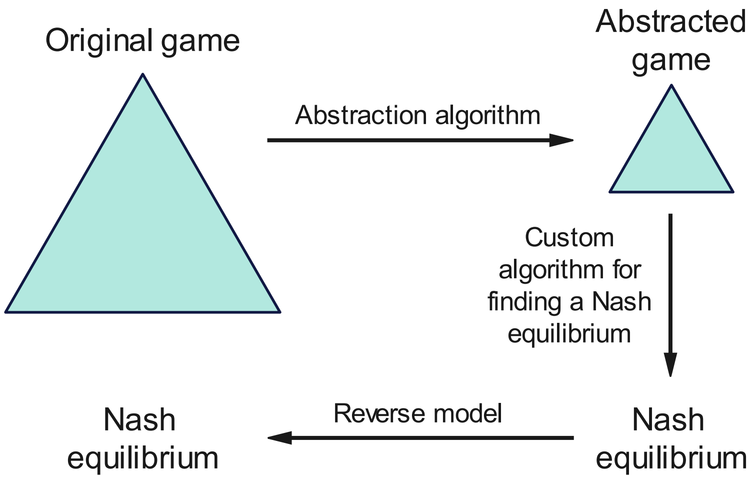


Figure 1: Paradigm for solving large-scale imperfect-information games

Despite these theoretical limitations, this paradigm has proven highly effective in practical applications. For instance, in HUNL, computer programs developed using this approach have surpassed human experts (Brown and Sandholm, 2018, 2019). Hence, this problem-solving paradigm has become a popular and powerful tool in addressing large-scale imperfect-information games.

Two simplification approaches are often employed: infoset abstraction and action abstraction. Detailed mathematical definitions can be found in the Appendix B. Infoset abstraction involves partitioning each player's infoset space into several abstracted infosets, each of which contains similar infosets. This partitioning allows players to treat infosets on the simplified level as equivalent, ensuring consistent decision preferences on abstracted infosets.

On the other hand, action abstraction focuses on simplifying the action space within the game. Each player has a set of possible actions at non-terminal nodes, and action abstraction involves dividing this action set into similar action groups. Then, one action is sampled from each action group to form a abstracted action set. For the same abstracted infoset, players have access to the same abstracted action set. Infoset abstraction and action abstraction collectively work to reduce the size and complexity of the game tree.

3 Imperfect-Recall Games and Abstraction

Perfect/imperfect-recall are a pair of characteristics in imperfect-information games. If all players in the game can remember any information they observe during the game, the game is said to have perfect-recall; otherwise, it is said to have imperfect-recall. For a detailed description of this concepts, please refer to Appendix C. Perfect-recall games have several advantages (Kuhn, 1953), with the most important being that the existence of Nash equilibrium is ensured when using behavioral strategies, which are a form of strategy with smaller space cost. However, in practical situations, even if the game does not have perfect-recall, researchers typically use behavioral strategies for strategy solving (Lanctot et al., 2012; Kroer and Sandholm, 2016).

When perfect/imperfect-recall is used to describe abstraction, it refers to whether the abstracted game has perfect/imperfect-recall. However, despite the term imperfect-recall means that players are allowed to forget some memories, in practice, the imperfect-recall abstraction is often used in a more extreme way: requiring players to not retain any memory and make decisions based on future information only (Waugh et al., 2009c; Johanson et al., 2013; Ganzfried and Sandholm, 2014).

4 Games with Ordered Signals

In imperfect-information games, directly conducting infoset abstraction and action abstraction can be a challenging task. This is because, in the process of infoset abstraction, one must first identify infosets that can be merged, and these infosets must share the same action space. Furthermore, not all infosets with identical action spaces can be combined into the same abstracted infoset. In HUNL, for instance, when the first player in both the Flop and Turn stages faces the same chip pot, their action spaces are identical. However, due to the difference in game stages, it is not suitable to merge these two infosets into the same abstracted infoset.

In practical development of Texas Hold'em AI, we do not simplify the game directly through infoset abstraction and action abstraction. Instead, we consider simplification in two dimensions: hand abstraction and action abstraction. This approach is favored because Texas Hold'em belongs to a special category of imperfect-information games where the description of infosets can be divided into two unrelated dimensions: hands and the action sequences. By focusing on hand abstraction, we can streamline the process of infoset abstraction, allowing us to merge infosets that share the same action sequence. The modeling of this class of games, known as **games with ordered signals**, was first introduced by Gilpin and Sandholm (2007b).

The original definition of games with ordered signals is somewhat complex. It defines branches at different stages as trees and uses a forest to describe the entire game, involving a significant amount of node mapping. Additionally, its definition of signals is solely based on a combinatorial model without replacement. However, games with ordered signals have the potential for broader applicability. For instance, games similar to Texas Hold'em, such as Liar's Dice (Freeman, 1989), exhibit significant similarities. Still, due to the distinct nature of signals (with or without replacement), they cannot be modeled using games with ordered signals. In this section, we've refined the definition of games with ordered signals for greater clarity and versatility in modeling. We've employed this framework to model Texas Hold'em-style games in this paper. Subsequently, our focus in the upcoming sections will be on studying abstraction techniques within this framework.

Definition 1 *A game with ordered signals $\tilde{\Gamma} = \left\langle \tilde{\mathcal{N}}, \tilde{H}, \tilde{Z}, \tilde{\rho}, \tilde{A}, \tilde{\chi}, \tilde{\tau}, \gamma, \Theta, \varsigma, O, \omega, \succeq, \tilde{u} \right\rangle$ is a structured tuple, where $\tilde{\mathcal{N}}$ is the extended player set, $\tilde{H}$ is the set of non-terminal public nodes, $\tilde{Z}$ is the set of terminal public nodes, $\tilde{\rho}$ is the player function, $\tilde{A}$ is the action set, $\tilde{\chi}$ is the action mapping function, $\tilde{\tau}$ is the successor function, γ is the stage function, Θ is the signal set, ς is the signal revelation function, O is the observation function vector, ω is the survival status function vector, $\succeq$ is the signal partial order function, and $\tilde{u}$ is the utility function vector. In more detail:*

- $\tilde{\mathcal{N}} = \mathcal{N} \cup \{c, pub\}$, *where* $\mathcal{N} = \{1, ..., N\}$ *represents the set of players,* c *denotes the nature player, and* pub *is defined as a special observer player who does not participate in game actions but has access to information available to all players. For simplicity, we define* $\mathcal{N}_c = \mathcal{N} \cup \{c\}$ *and* $\mathcal{N}_{pub} = \mathcal{N} \cup \{pub\}$.
- $\tilde{\rho} : \tilde{H} \mapsto \mathcal{N}_c$ *assigns each non-terminal public node to the unique extented player able to take actions at that non-terminal public node.*
- $\tilde{X} = \tilde{H} \cup \tilde{Z}$ *comprises the set of public nodes. Specifically:*
  - $\tilde{x}^o \in \tilde{X}$ *is identified as the initial public node of the game.*
  - $\tilde{H}_i = \left\{\tilde{h} \in \tilde{H} | \tilde{\rho}(\tilde{h}) = i\right\}$ *denotes the set of non-terminal public nodes where player* i *is the action player, and these non-terminal public nodes where nature acts are termed chance public nodes.*
- $\tilde{\chi} : \tilde{H} \mapsto 2^{\tilde{A}}$ *maps each non-terminal public node to its possible set of actions. Specifically, for chance public nodes, the action set is exclusively* $\{Reveal\}$*, where* $Reveal \in \tilde{A}$ *is the sole virtual action nature performs to disclose signals.*
- $\tilde{\tau} : \tilde{H} \times \tilde{A} \mapsto \tilde{X}$ *illustrates the transition to a new public node upon taking action* $\tilde{a} \in \tilde{\chi}(\tilde{h})$ *at non-terminal public node* $\tilde{h} \in \tilde{H}$*, with each public node following a unique transition path. That is, if* $\tilde{\tau}(\tilde{h}_1, \tilde{a}_1) = \tilde{\tau}(\tilde{h}_2, \tilde{a}_2)$*, then it implies* $\tilde{h}_1 = \tilde{h}_2$ *and* $\tilde{a}_1 = \tilde{a}_2$.
- $\gamma : \tilde{X} \mapsto \mathbb{N}^+$ *assigns to each public node* $\tilde{x}$ *the number of chance public nodes encountered from the initial public node* $\tilde{x}^o$ *to* $\tilde{x}$*, defined as the stage of* $\tilde{x}$*. Notably, the stage of every public node is at least 1, indicating that the initial public node is a chance public node.*
- *Each element within* Θ *corresponds to a signal revealed by nature at chance public nodes.*
- $\varsigma : \Theta \mapsto \Delta(\Theta)$ *specifies the probability distribution for the next signal given the current signal* θ*, with* $\varsigma(\theta'|\theta)$ *representing the likelihood of revealing* θ' *following* θ.
- $O = (O_1, ..., O_N, O_{pub})$*, where* $O_i(\theta)$ *denotes the observation made by player* $i \in \mathcal{N}_{pub}$ *on the signal* $\theta \in \Theta$.
- $\omega = (\omega_1, ..., \omega_N)$*, where* $\omega_i : \tilde{Z} \mapsto \{true, false\}$ *indicates whether player* i *remains in the game at terminal public nodes, with true signifying active participation and* $false$ *indicating elimination.*
- $\succeq : \tilde{\Theta} \times \mathcal{N} \times \mathcal{N} \mapsto \{true, false\}$ *serves as a partial order function, where the terminal signal set* $\tilde{\Theta}$ *is a subset of* Θ. $\succeq (\theta, i, j) = true$ *suggests player* i *is at least as preferred as player* j *on signal* θ*;* $\succeq (\theta, i, j) = false$ *means the opposite, indicating player* i *is less preferred.*
- $u = (u_1, ..., u_N)$*, where* $u_i : \Theta \times \tilde{Z} \mapsto \mathbb{R}$*, represents the utility function of player* $i \in \mathcal{N}$*, which specifies the payoff obtained by player* i *at terminal public node* $\tilde{z} \in \tilde{Z}$ *with the signal* $\theta \in \Theta$*. It is required that for any terminal signal* $\theta \in \tilde{\Theta}$ *and* $\tilde{z} \in \{\tilde{z}' \in \tilde{Z} \mid \omega_i(\tilde{z}') = \omega_j(\tilde{z}') = true\}$*, if* $\succeq (\theta, i, j) = true$*, then* $u_i(\theta, \tilde{z}) \geq u_j(\theta, \tilde{z})$.

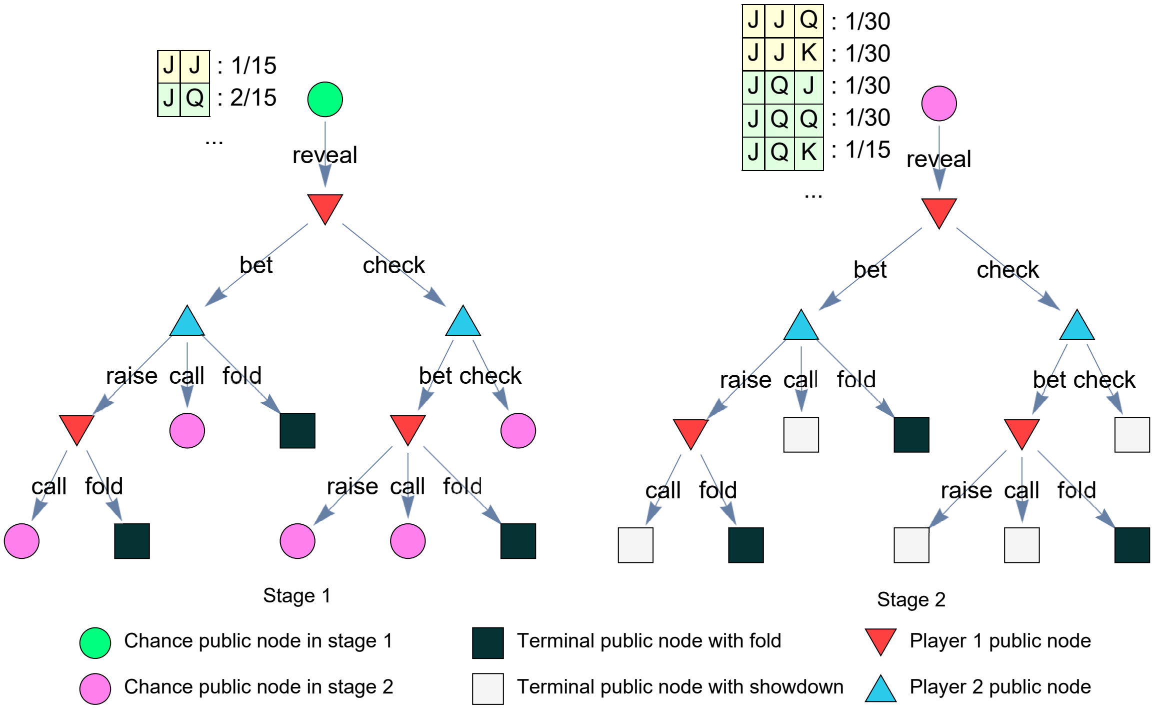


Figure 2: Game tree with ordered signal for Leduc Hold'em

Figure 2 illustrates the game tree with ordered signal for Leduc Hold'em. According to rules of Leduc Hold'em rules, see Appendix D.1, all subtrees rooted at chance public nodes in the second stage are identical. For the sake of simplicity in visualization, we represent the Leduc game tree as a forest of two trees. However, it is worth noting that the game with ordered signals model defined in Definition 1 is a complete tree. We can concatenate a subtree from the first stage with five subtrees from the second stage to form the complete Leduc game tree. It is important to distinguish between two types of terminal public nodes. When $\tilde{z}$ is a terminal public node with fold, it implies $\sum_{i=1}^{N} \omega_i(\tilde{z}) = 1$, indicating that only one player remains in the game. On the other hand, when $\tilde{z}$ is a terminal public node with showdown, it means that more than one player is still participating in the game, and their payoffs need to be determined through a showdown. In Leduc Hold'em, Rhode Island Hold'em, HULHE, HUNL, and the Numeral211 Hold'em introduced in the experimental section, all terminal public nodes with fold are located in the final stage. Meanwhile, the signal set is solely dependent on the stage, implying that the terminal signal set $\tilde{\Theta}$ in these games corresponds exclusively to the signal set of the final stage. Unlike the definition in Gilpin and Sandholm (2007b), signals in Leduc Hold'em, as defined in Definition 1, refer to the complete set of cards currently dealt. For example, JQ in the first stage (the first player dealt a J, the second player dealt a Q) and JQK in the second stage (the first player dealt a J, the second player dealt a Q, and the community card is K) are signals in their respective stages. In the case of JQK in the second stage, the cards dealt to each player and the individual community card are observations of the signal from their respective perspectives.

That is, if $\theta =$ JQK, then

$$\begin{cases} \vartheta_1(\theta) = \text{J*K} \\ \vartheta_2(\theta) = \text{Q*K} \\ \vartheta_{pub}(\theta) = \text{**K} \end{cases} .$$

We can define other games with ordered signals where signal unveiling follows various probability distributions, not just the model of sampling without replacement, as we are familiar with.

Given a signal θ and a player $i \in N$, we can define the signal infoset that player i cannot distinguish related to θ as $\vartheta_i(\theta) = \{\theta' \in \Theta \mid O_i(\theta) = O_i(\theta') \land O_{pub}(\theta) = O_{pub}(\theta')\}$. For example, in Figure 2, JJQ and JQJ belong to the same signal infoset for player 1. Additionally, the concept of a signal infoset space $\mathit{\Theta} = (\mathit{\Theta}_1, \ldots, \mathit{\Theta}_N, \mathit{\Theta}_c)$ is introduced, where $\mathit{\Theta}_i = \{\vartheta_i(\theta) | \theta \in \Theta\}$ denotes the collection of signal infosets pertinent to player i. A game with ordered signals is a special type of imperfect-information game. We can transform a game with ordered signals into an imperfect-information game, a process illustrated in Appendix E.

Compared to the definition of games with ordered signals introduced by Gilpin and Sandholm (2007b), Definition 1 is more generalized and concise. Specifically:

1. **The definition of stages is more flexible.** In the original definition, stages were predefined non-negative integers. In Definition 1, stages are defined based on the number of chance public nodes along a path. This allows games with ordered signals to be treated as a single tree structure, eliminating the need to split them into a forest based on chance public nodes.

2. **The revelation of signals is more versatile.** In the original definition, signal revelation involved combinatorial model without replacement. In Definition 1, the revelation of signals can follow any arbitrary random distribution.

Furthermore, we have also adjusted the way signal orders are defined. In the original definition, signal orders reflected the dominance relationships between each player's signal observations. In the current definition, signal orders refer to the dominance relationships of players over a signal. This change brings several benefits, one of which is the ability to statistically compute the dominance probabilities of each player over the signal infoset. Combined with the survival functions, this makes the constraints on the payoffs of each player in games with ordered signals more specific and clear.

5 Abstraction for Games with Ordered Signals

The model of games with ordered signals divides the description of nodes in imperfect-information games into two dimensions: the signals and the public nodes. Correspondingly, it categorizes the description of infosets in imperfect-information games into signal infosets and public nodes. The infosets in an imperfect-information game that can be traced back to a common public node will have the same action space and action history sequence. As long as the signal infosets are suitable for merging together, their corresponding infosets should also be suitable for merging into the same abstracted infoset. The game with ordered signals simplifies the process of infosets abstraction in imperfect-information games, eliminating the need for pre-filtering infosets. This also implies that infosets at the same stage will have consistent abstraction results based on the signal infosets abstraction. In other words, if ϑ_i^1 and ϑ_i^2 are two r-stage signal infosets for player i and can be grouped into the same abstracted signal infoset during a abstraction process, then for any public node $\tilde{h}_i$ corresponding to player i in the r-stage, the infoset related to $\tilde{h}_i$ and ϑ_i^1 will always be classified into the same abstracted infoset as the one related to $\tilde{h}_i$ and ϑ_i^2. This simplifies the workload of infoset abstraction, even though it sacrifices some flexibility in infosets abstraction, this constraint is reasonable. We introduce the following definition to provide a clearer description of the abstraction process in games with ordered signals:

Definition 2 *In games with ordered signals, $\tilde{\alpha} = (\tilde{\alpha}_1, ..., \tilde{\alpha}_N)$ is referred to as an abstraction profile. For player i, the abstraction $\tilde{\alpha}_i = \left\langle \tilde{\alpha}_i^{\varTheta}, \tilde{\alpha}_i^{\tilde{\chi}} \right\rangle$ consists of two components:*

- *$\tilde{\alpha}_i^{\varTheta}$ is a partition of Θ, known as the signal (infoset) abstraction. For any $\hat{\vartheta} \in \tilde{\alpha}_i^{\varTheta}$, referred to as an abstracted signal infoset, we can identify several signal infosets within $\varTheta_i$, and these signal infosets collectively form a partition of $\hat{\vartheta}$.*

- *$\tilde{\alpha}_i^{\tilde{\chi}}$ is a function defined on $\tilde{H}_i$, known as the (public) action abstraction. $\tilde{\alpha}_i^{\tilde{\chi}}(\tilde{h}) \subseteqq \tilde{\chi}(\tilde{h})$ represents the abstracted (public) action set for non-terminal public node $\tilde{h} \in \tilde{H}_i$.*

The **null abstraction** for player i is defined as $\phi_i = \langle \varTheta_i, \tilde{\chi} \rangle$. The abstracted game $\tilde{\Gamma}^{\tilde{\alpha}}$ was derived by substituting $\varTheta_i$ with $\tilde{\alpha}_i^{\varTheta}$ and $\tilde{\chi}(\tilde{h})$ with $\tilde{\alpha}_{\tilde{\rho}(\tilde{h})}^{\tilde{\chi}}(\tilde{h})$ across all $\tilde{h} \in \bigcup_{i \in \mathcal{N}} \tilde{H}_i$. This process highlights how, in games with ordered signals, the abstractions of signals and actions can be dissected and analyzed independently. To this end, we introduce $\tilde{\alpha}^{\varTheta} = (\tilde{\alpha}_1^{\varTheta}, ..., \tilde{\alpha}_N^{\varTheta})$ as the signal abstraction profile and $\tilde{\alpha}^{\tilde{\chi}} = (\tilde{\alpha}_1^{\tilde{\chi}}, ..., \tilde{\alpha}_N^{\tilde{\chi}})$ as the action abstraction profile. Consequently, $\tilde{\Gamma}^{\tilde{\alpha}^{\varTheta}}$ and $\tilde{\Gamma}^{\tilde{\alpha}^{\tilde{\chi}}}$ represent the signal abstracted game and the action abstracted game, respectively.

In a game with ordered signals, the performance of different abstractions is typically assessed through experimental validation, as there currently lacks a theoretical method for directly analyzing the performance of two abstractions. However, when two abstractions exhibit a refinement relationship, we can discuss their performance in such cases.

Definition 3 *In a game with ordered signals, consider $\tilde{\alpha}_i$ and $\tilde{\beta}_i$ as abstractions for player i. We define the following refinement relationships:*

- *If, for any $\hat{\vartheta} \in \tilde{\beta}_i^{\varTheta}$, there exist one or more abstracted signal infosets on $\tilde{\alpha}_i^{\varTheta}$, such that their collection forms a partition of $\hat{\vartheta}$, then we say $\tilde{\alpha}_i^{\varTheta}$ refines $\tilde{\beta}_i^{\varTheta}$, denoted as $\tilde{\alpha}_i^{\varTheta} \sqsupseteq \tilde{\beta}_i^{\varTheta}$.*

- *If, for any $\tilde{h} \in \tilde{H}_i$, it holds that $\tilde{\beta}_i^{\tilde{\chi}}(\tilde{h}) \subseteq \tilde{\alpha}_i^{\tilde{\chi}}(\tilde{h})$, then we say $\tilde{\alpha}_i^{\tilde{\chi}}$ refines $\tilde{\beta}_i^{\tilde{\chi}}$, denoted as $\tilde{\alpha}_i^{\tilde{\chi}} \sqsupseteq \tilde{\beta}_i^{\tilde{\chi}}$.*

- *If $\tilde{\alpha}_i^{\varTheta} \sqsupseteq \tilde{\beta}_i^{\varTheta}$ and $\tilde{\alpha}_i^{\tilde{\chi}} \sqsupseteq \tilde{\beta}_i^{\tilde{\chi}}$, then we say $\tilde{\alpha}_i$ refines $\tilde{\beta}_i$, denoted as $\tilde{\alpha}_i \sqsupseteq \tilde{\beta}_i$.*

Intuitively, one might expect that more refined abstractions would result in superior strategies. However, the discovery of abstraction pathology refutes this notion.[2] It is crucial to stress that the issue of abstraction pathology arises from abstracting the opponent's decision space, essentially modeling the opponent's behavior based on a fundamental assumption: that the opponent exclusively acts within the defined abstracted decision space. However, this assumption may not always hold true. A more reasonable perspective called **one-player-abstraction-perspective** is that players make decisions within the abstracted decision space as a simplification to adapt to the opponent's decisions in the complete decision space. The study conducted by Waugh et al. (2009b) demonstrated that when the opponent's decision space is not abstracted, increasing the precision of one's own abstraction can lead to reduced exploitability of the abstracted Nash equilibrium solution in the original game.[3]

In the subsequent sections, we will focus on the task of signal abstraction and the algorithms associated with it. The signal abstraction task aims to find suitable signal abstraction within a given game with ordered signals, while the signal abstraction algorithm serves as the method to accomplish this goal.

Although the task of signal abstraction in games with ordered signals were not explicitly defined in prior research, earlier studies employed various rudimentary terms to describe similar problems and methods. This was particularly evident in tasks such as hand abstraction in the domain of Texas Hold'em game. For instance, Shi and Littman (2001) referred to it as **bins**, Billings et al. (2003) as **buckets**, Johanson et al. (2013) mentioned **state-space abstraction**, and Gilpin and Sandholm (2007b) used the term **signals** to describe analogous concepts. Building upon these analogous descriptions of signal abstraction tasks, numerous approaches have been proposed to solve it. Among them, Gilpin and Sandholm (2007b) and Waugh (2013) developed **lossless isomorphism** (LI), Johanson (2007) introduced the **expected hand strength** (E[HS]) algorithm, while Gilpin et al. (2007); Gilpin and Sandholm (2008) introduced the **potential-aware** (PA) algorithm, which later led to the derivation of the **potential-aware based on the earthmover's distance** (PAEMD) algorithm introduced by Ganzfried and Sandholm (2014). Our definition of the signal abstraction task has been significantly influenced by the work of Gilpin and Sandholm (2007b) and is also more closely aligned with the essence of the hand abstraction task in Texas Hold'em-style game.

6 Potential Outcome Isomorphism

In some games with ordered signals, the game rules establish equivalence among certain signal infosets. For example, consider Texas Hold'em poker, where a deck of cards comprises

[2] It's worth noting that while Waugh's experiments focus on imperfect-information games, Leduc Hold'em itself is a form of game with ordered signals. Moreover, the information set abstraction and action abstraction performed in their experiments correspond to the signal and action abstraction of games with ordered signals , respectively. Therefore, the concept of abstraction pathology also applies to the games with ordered signals.

[3] It's important to observe that although Waugh's conclusion is derived within the context of imperfect-information games, while games with ordered signals represent a particular subclass of imperfect-information games. Consequently, the same conclusion remains valid when applied to games with ordered signals.

four suits, and the strength of a hand remains unchanged when only the suits are rotated. The hand $[\spadesuit A, \heartsuit A | \clubsuit 3, \clubsuit 5, \diamondsuit Q, \diamondsuit K, \diamondsuit T]$ holds the same strength as $[\clubsuit A, \spadesuit A | \diamondsuit 3, \diamondsuit 5, \heartsuit Q, \heartsuit K, \heartsuit T]$; this property is known as **suit-insensitivity**. LI serves as a signal abstraction, aiming to eliminate such redundancy. However, in standard Texas Hold'em poker, the quantity of abstracted signal infosets in LI remains quite large, reaching the dimension of 10^9, and the differentiation between abstracted signal infosets cannot be quantified, thereby impeding further reduction of the scale of abstracted signal infosets through techniques such as clustering. These limitations constrain the application of LI in AI development.

In game theory, calculating the expected payoff at each node relies on the utility backpropagated from the terminal nodes. Consequently, mainstream signal abstraction algorithms cluster signals based on the showdown outcomes in the terminal stage, such as E[HS], PA, and PAEMD. These algorithms are referred to as **outcome-based imperfect-recall algorithms**. They possess the flexibility to adjust the scale of abstracted infosets. Our inquiry pertains to the resolution boundary, which refers to the quantity of signal infosets that the algorithm can recognize without exceeding, of such outcome-based imperfect-recall signal abstraction algorithms and whether they can achieve LI. The objective of this section is to develop a signal abstraction, termed as potential outcome isomorphism (POI), which aims to identify as many abstracted signal infosets as possible based on future showdown outcomes only. In the next section, we will elucidate why the POI can represent the resolution boundary of outcome-based imperfect-recall algorithms.

The construction of POI, as shown in Algorithm 1, is described as follows. The algorithm begins at the last stage, $r = |\mathfrak{r}|$, where $\mathfrak{r} = \{r = \gamma(\tilde{x}) \mid \tilde{x} \in \tilde{X}\}$, and proceeds bottom-up, computing each stage down to $r = 1$. For any player i and signal information set ϑ at stage r, we construct a potential outcome feature $f_i^{(r)}(\vartheta)$. These feature vectors consist of natural numbers and have varying dimensions across stages but remain consistent within the same stage.

After removing duplicates from the potential outcome features of all signal infosets for player i at stage r, we form a set denoted as $\mathcal{C}_i^{(r)}$. Upon sorting the potential outcome features in $\mathcal{C}_i^{(r)}$ in lexicographical order, a unique lexicographical identifier in the range of 0 to $|\mathcal{C}_i^{(r)}| - 1$ is assigned to each potential outcome feature $f_i^{(r)}$. This identifier is referred to as the lexicographical order identifier (*lexid*) for $f_i^{(r)}$. If two signal information sets, ϑ_1 and ϑ_2, in $\Theta_i^{(r)}$ share the same potential outcome feature $f_i^{(r)}$, they are categorized as the same abstracted signal infoset.

For player i at the last stage, $r = |\mathfrak{r}|$, the potential outcome feature for a signal infoset $\vartheta \in \Theta_i^{(r)}$ is defined as

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

where each component is calculated as follows

$$f_i^{(r),l}(\vartheta) = \begin{cases} \sum\limits_{\theta \in \vartheta} \mathbb{1}\left\{\exists j \neq i, \succeq(\theta, i, j) = \text{False and } \succeq(\theta, j, i) = \text{True}\right\} & \text{if } l = 0, \\ \sum\limits_{\theta \in \vartheta} \mathbb{1}\left\{\sum\limits_{j \neq i} \mathbb{1}\left\{\succeq(\theta, j, i) = \text{True}\right\} = l \text{ and } \forall j \neq i, \succeq(\theta, i, j) = \text{True}\right\} & \text{if } l \neq 0. \end{cases}$$

Where

Algorithm 1 Potential outcome isomorphism

```
Require:
    $r \in \{1, \dots, |\mathfrak{r}|\}$. Stages.
    $\mathit{\Theta}_i^{(r)}$. Signal infoset space.
    $Index_i(r, \cdot) : \mathit{\Theta}_i^{(r)} \mapsto \mathbb{N}$. Signal infoset index function.
    $\mathcal{D}_i^{(r+1)} : \mathbb{N} \mapsto \mathbb{N}$. Next stage potential outcome isomorphism map, if $r = |\mathfrak{r}|$, $\mathcal{D}_i^{(r+1)} = \emptyset$.
procedure StagePotentialOutcomeIsomorphism($r$, $\mathit{\Theta}_i^{(r)}$, $\mathcal{D}_i^{(r+1)}$)
    Initialize $\mathcal{C}_i^{(r)}$ vector as empty.
    Initialize $\mathcal{D}_i^{(r)}$ array arbitrarily with length $|\mathit{\Theta}_i^{(r)}|$.
    for $\vartheta \in \mathit{\Theta}_i^{(r)}$ do
        $feature \leftarrow f_i^{(r)}(\vartheta)$.
        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 $\vartheta \in \mathit{\Theta}_i^{(r)}$ do
        $feature \leftarrow f_i^{(r)}(\vartheta)$.
        $idx \leftarrow Index_i(r, \vartheta)$.
        Update $\mathcal{D}_i^{(r)}[idx]$ with $\mathcal{CI}_i^{(r)}[feature]$.
    end for
    return $\mathcal{D}_i^{(r)}$.
end procedure
```

- $f_i^{(r),0}(\vartheta)$ represents the number of signals in ϑ where player i is ranked lower than at least one other player.
- $f_i^{(r),l}(\vartheta)$ represents the number of signals in ϑ where player i is ranked no less than all other players and is ranked exactly l times higher than the other players.

Specifically, for a two-player game with ordered signals at the last stage, $r = |\mathfrak{r}|$, the feature vector has a dimension of 3. In this case, $f_i^{(r),0}(\vartheta)$ denotes the number of signals in ϑ where player i is ranked lower than the opponent, $f_i^{(r),1}(\vartheta)$ denotes the number of signals where player i is ranked equally with the opponent, and $f_i^{(r),2}(\vartheta)$ denotes the number of signals where player i is ranked higher than the opponent.

The potential outcome feature of player i for the signal infoset ϑ at stage $r < |\mathfrak{r}|$ represents the histogram distribution of possible abstracted signal infosets that it can transition to in the next stage. Due to the potentially large size of $\mathcal{C}^{(r+1)}$, a histogram distribution with a length of $|\mathcal{C}^{r+1}|$ can incur significant computational overhead. Fortunately, in games like Texas Hold'em, we can simplify this data structure. $D = \{0, \dots, |D| - 1\}$ represents a deck of cards. In each game stage, the dealer sequentially deals a certain number of community and private cards from this deck without replacement. We define $\mathcal{U} = (\mathcal{U}^1, \dots, \mathcal{U}^{|\mathfrak{r}|})$ as a vector indicating the number of community cards dealt in each stage of a single game, and $\mathcal{K} = (\mathcal{K}^1, \dots, \mathcal{K}^{|\mathfrak{r}|})$ as a vector indicating the number of private cards dealt to each player in each stage of a single game. Additionally, we use $\upsilon = (\upsilon^1, \dots, \upsilon^{|\mathfrak{r}|})$ to denote the vector of community cards dealt in a single game, where $|\upsilon^r| = \mathcal{U}^r$. The private cards received by player i in a single game are represented by $\kappa_i = (\kappa_i^1, \dots, \kappa_i^{|\mathfrak{r}|})$, where $|\kappa_i^r| = \mathcal{K}^r$. In the r-th stage, a signal θ can be described by the vector $(\upsilon^1, \dots, \upsilon^r, \kappa_1^1, \dots, \kappa_1^r, \dots, \kappa_N^1, \dots, \kappa_N^r)$, and the signal infoset $\vartheta_i(\theta)$ for player i to which signal θ belongs can be described by the vector $(\upsilon^1, \dots, \upsilon^r, \kappa_i^1, \dots, \kappa_i^r)$. Firstly, Texas Hold'em-style games have perfect-recall, which implys that in a given stage, all signals within a signal infoset share the same predecessor signal infoset in the earlier stages. For example, during the Turn stage in a game of Texas Hold'em, the preceding signal infoset for the signal infoset $[\spadesuit A, \heartsuit A | \clubsuit 3, \clubsuit 5, \diamondsuit Q, \diamondsuit K]$ can only be $[\spadesuit A, \heartsuit A | \clubsuit 3, \clubsuit 5, \diamondsuit Q]$ during the Flop stage, and $[\spadesuit A, \heartsuit A]$ during the Preflop stage. Secondly, in Texas Hold'em-style games, the transition of signal infosets between different stages is a **classical probability model**. In stage r, a signal infoset transition to $n^{(r+1)}$ signal infosets in stage $r+1$, all with equal probability, where $n^{(r+1)}$ is calculated as

$$n^{(r+1)} = \binom{|D| - \sum_{j=1}^{r}(\mathcal{U}^j + N\mathcal{K}^j)}{\mathcal{U}^{r+1}} \binom{|D| - \sum_{j=1}^{r}(\mathcal{U}^j + N\mathcal{K}^j) - \mathcal{U}^{r+1}}{\mathcal{K}^{r+1}}.$$

For example, in Texas Hold'em, the signal infosets from the Preflop, Flop, and Turn stages can transition to signal infosets in the Flop, Turn, and River stages, respectively, with $\binom{50}{3}$, $\binom{47}{1}$, and $\binom{46}{1}$ possible cases. We can simplify these histograms using **sparse representation**. Specifically, for a signal infoset ϑ in stage r that can transition to signal infosets $\vartheta_1, \dots, \vartheta_{n^{(r+1)}}$ in stage $r+1$, the potential outcome feature for ϑ is calculated as

$$f_i^{(r)}(\vartheta) = @\textsc{NonlastStageFeature}(r, \vartheta, \mathcal{D}_i^{(r+1)}).$$

Where the NonlastStageFearure operator is defined as shown in Algorithm 2.

Algorithm 1 and Algorithm 2 both involve an Index_i operator, which is responsible for mapping the signal infoset $\vartheta \in \Theta_i^{(r)}$ at stage r to a unique integer ranging from 0 to $|\Theta_i^{(r)}| - 1$. In games like Texas Hold'em, an effective mapping method is to use **colexicographic order** (Bollobás, 1986). As described by Waugh (2013), grouped colexicographic encoding can be used to encode grouped combinations. The order of the signal infoset $\vartheta_i = (\upsilon^1, \dots, \upsilon^r, \kappa_i^1, \dots, \kappa_i^r) \in \Theta_i^{(r)}$ can be represented as follows

$$\text{Index}_i(r, \vartheta_i) = \text{Indexgroup}^{|D|}_{\mathcal{U}^1, \mathcal{K}^1, \dots, \mathcal{U}^r, \mathcal{K}^r}(\upsilon^1, \kappa_i^1, \dots, \upsilon^r, \kappa_i^r).$$

Here, Indexgroup is an operator used to compute the order of grouped combinations. Let $l^1, \dots, l^k$ be grouped combinations extracted from the set $\{0, \dots, m-1\}$, where l^j has L^j elements. The index of this grouped combination can be defined as

$$\text{Indexgroup}^m_{L^1, \dots, L^k}(l^1, \dots, l^k) = \binom{m}{L^k} \text{Indexgroup}^{m-L^k}_{L^1, \dots, L^{k-1}}(l^1|_{l^k}, \dots, l^{k-1}|_{l^k}) + \text{colex}^m_{L^k}(l^k). \tag{2}$$

Here, colex represents the colexicographic order operator. Let $l = (c_1, \dots, c_n)$ be an n-dimensional vector, where $c_1, \dots, c_n$ are elements extracted from the set $\{0, \dots, m-1\}$

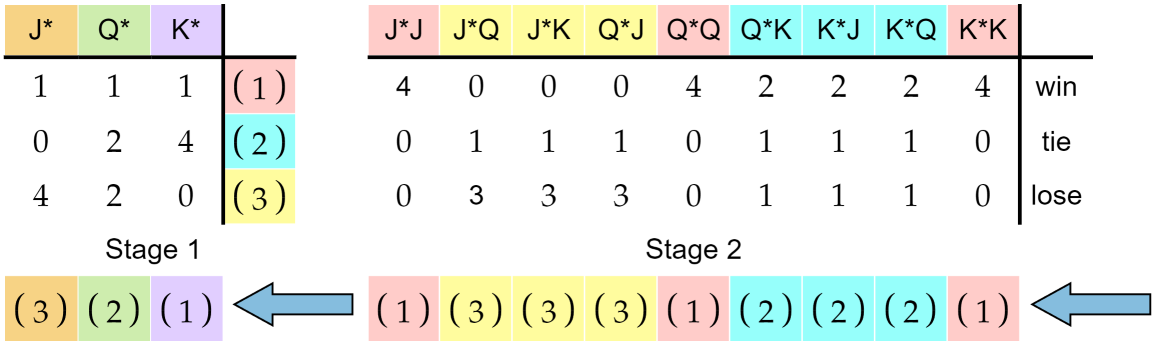
Algorithm 2 Non-last stage feature

```
Require:
    $r \in \{1, \dots, |\mathfrak{r}| - 1\}$. Stages.
    $\vartheta \in \Theta_i^{(r)}$. Signal infoset.
    $n^{(r+1)} \in \mathbb{N}$. The number of signal infosets that $\vartheta$ can transfer to at stage $r + 1$.
    $Index_i(r, \cdot) : \Theta_i^{(r)} \mapsto \mathbb{N}$. Signal infoset index function.
    $\mathcal{D}_i^{(r+1)} : \mathbb{N} \mapsto \mathbb{N}$. Next stage potential outcome isomorphism map.
procedure NonLastStageFearure($r$, $\vartheta$, $\mathcal{D}_i^{(r+1)}$)
    Initialize $feature$ array arbitrarily with length $n^{(r+1)}$.
    for $j = 0$ to $|n^{(r+1)}| - 1$ do
        $\vartheta_j \leftarrow$ the $j$-th possible signal infoset in next stage of $\vartheta$.
        $idx_j \leftarrow Index_i(r + 1, \vartheta_j)$.
        $feature[j] \leftarrow \mathcal{D}_i^{(r+1)}[idx_j]$.
    end for
    Sort the elements of $feature$ in ascending order.
    return $feature$.
end procedure
```

without replacement. Generally, we assume that $c_1 < \dots < c_n$. Additionally, we specify $\binom{n_1}{n_2} = 0$ when $n_1 < n_2$. Based on these definitions, the colexicographic order of vector l can be expressed as

$$\mathrm{colex}_n^m(l) = \mathrm{colex}_n^m(c_1, \dots, c_n) = \sum_{j=1}^{n} \binom{c_j}{j}.$$

In Equation (2), $l^j|_{l^k}$ is a shifting operation that maps the elements from l^j in the original space $\{0, \dots, m-1\}$ to a new space $\{0, \dots, m-1-L^k\}$. Specifically, if c is an element in l^j, and there are x elements in l^k that are smaller than c, then in $l^j|_{l^k}$, the element corresponding to c will become $c - x$.



J*	Q*	K*	
1	1	1	(1)
0	2	4	(2)
4	2	0	(3)

Stage 1

J*J	J*Q	J*K	Q*J	Q*Q	Q*K	K*J	K*Q	K*K	
4	0	0	0	4	2	2	2	4	win
0	1	1	1	0	1	1	1	0	tie
0	3	3	3	0	1	1	1	0	lose

Stage 2

(3)	(2)	(1)

(1)	(3)	(3)	(3)	(1)	(2)	(2)	(2)	(1)

Figure 3: The process of constructing potential outcome isomorphism in Leudc Hold'em

Figure 3 illustrates the process of building POI in Leduc Hold'em. In the second stage, each hand corresponds to four possible opponent hole combinations, resulting in four signals per signal infoset. We tally the win-tie-loss outcomes for each signal in each infoset after showdown and group signal infosets with identical distributions into the same abstracted signal infoset, assigning a label to each. In the first stage, each hand will form different second-stage abstracted signal infosets as five different flop cards are dealt. We tally these distributions and consider hands with the same distribution as belonging to the same abstracted signal infoset. It can be observed that in Leduc Hold'em, the quantity of abstracted signal infosets in the first stage is equal to the quantity of signal infosets, indicating good identification capability. However, in the second stage, POI can only identify three abstracted signal infosets, significantly fewer than the nine signal infosets. In Leduc Hold'em, the quantity of signal infosets equals the quantity of lossless abstracted signal infosets. This also indicates that in Leduc Hold'em, POI has much lower resolution than LI. We will analyze this phenomenon in the next section.

7 The Resolution Boundary of Mainstream Signal Abstraction Algorithms

In this section, we aim to assess the resolution boundary of mainstream signal abstraction algorithms such as E[HS] and PA&PAEMD. To achieve this, we introduce a novel technique called common refinement, which is utilized to delineate the resolution boundaries of signal abstraction algorithms. Subsequently, we demonstrate that POI serves as a common refinement of the prevailing outcome-based imperfect-recall algorithms.

7.1 Common Refinement

We first consider the common refinement of several signal abstractions:

Definition 4 *In games with ordered signals, we refer to the common refinement of several signal abstractions $\tilde{\alpha}_i^{\Theta,1}, \dots, \tilde{\alpha}_i^{\Theta,n}$ for player i as $\tilde{\alpha}_i^{\Theta}$, if $\tilde{\alpha}_i^{\Theta} \sqsupseteq \tilde{\alpha}_i^{\Theta,j}$ holds for all $j \in \{1, \dots, n\}$.*

If $\tilde{\alpha}_i^{\Theta}$ serves as the common refinement of $\tilde{\alpha}_i^{\Theta,1}, \dots, \tilde{\alpha}_i^{\Theta,n}$, then any signal infoset identifiable by $\tilde{\alpha}_i^{\Theta,1}, \dots, \tilde{\alpha}_i^{\Theta,n}$ can all be identified by $\tilde{\alpha}_i^{\Theta}$. This implies that for $\vartheta^1, \vartheta^2 \in \Theta_i$, if there exists $j \in \{1, \dots, n\}$ such that $\tilde{\alpha}_i^{\Theta,j}$ assigns ϑ^1, ϑ^2 to different abstracted signal infosets, then $\tilde{\alpha}_i^{\Theta}$ will definitely assign ϑ^1, ϑ^2 to different abstracted signal infosets; if $\tilde{\alpha}_i^{\Theta}$ considers ϑ^1, ϑ^2 to belong to the same abstracted signal infoset, then $\tilde{\alpha}_i^{\Theta,1}, \dots, \tilde{\alpha}_i^{\Theta,n}$ will all consider ϑ^1, ϑ^2 to belong to the same abstracted signal infoset. From the one-player-abstraction-perspective, let $(\tilde{\alpha}_i^{\Theta}, \Theta_{-i})$ denote the signal abstraction profile where player i adopts $\tilde{\alpha}_i^{\Theta}$ while others maintain their original signal infosets. In this context, the exploitability of the strategy transition from the Nash equilibrium of the abstracted game $\tilde{\Gamma}^{(\tilde{\alpha}_i^{\Theta}, \Theta_{-i})}$ is lower than that of any $\tilde{\Gamma}^{(\tilde{\alpha}_i^{\Theta,1}, \Theta_{-i})}, \dots, \tilde{\Gamma}^{(\tilde{\alpha}_i^{\Theta,n}, \Theta_{-i})}$.

The signal abstraction algorithm, denoted as Alg, yields a signal abstraction given a set of parameters ψ. Hereafter, we provide the definition of common refinement for a signal abstraction algorithm:

Definition 5 *In a game with ordered signals, given a signal abstraction algorithm Alg, if there exists a signal abstraction $\tilde{\alpha}_i^{\Theta}$ such that, for any set of parameters ψ, the signal abstraction $\tilde{\beta}_i^{\Theta}$ generated by algorithm Alg can be refined by $\tilde{\alpha}_i^{\Theta}$, then we refer to $\tilde{\alpha}_i^{\Theta}$ as the common refinement of algorithm Alg in the game, and simply as the common refinement of algorithm Alg.*

The common refinement of a signal abstraction algorithm reflects, on one hand, the extent to which the algorithm retains information across the parameter space, and, on the other hand, the algorithm's ultimate performance in generating abstractions under the optimal set of parameters. Having a high-quality common refinement for an algorithm doesn't necessarily imply that the algorithm itself is excellent. However, the presence of a low-quality common refinement can indicate certain deficiencies in the algorithm.

7.2 Potential Outcome Isomorphism Serves as the Common Refinement of Mainstream Signal Abstraction Algorithms

The current mainstream signal abstraction algorithms, E[HS] and PA&PAEMD, are based on showdown outcomes clustering for lossy signal abstraction with imperfect-recall. These bottom-up clustering relies on the distance between cluster centroids and signal infosets (hand combinations). As shown in Equation (3), Algdis is the function within algorithm Alg used to calculate the distance between cluster centroids and signals, $CT_j^t(\text{Alg}, \psi, \Theta)$ represents the j-th cluster centroids generated by algorithm Alg with parameter ψ in iteration t, and $v(\text{Alg}, \vartheta)$ is a computational metric of ϑ involved in the clustering process.

$$d_j^t(\vartheta) = \text{Algdis}(CT_j^t(\text{Alg}, \psi, \Theta), v(\text{Alg}, \vartheta)). \tag{3}$$

We will demonstrate that POI serves as a common refinement for E[HS], PA&PAEMD algorithms. It is essential to emphasize that the key to this argument lies in showing that within any potential outcome isomorphism class, two signal infosets in the same class belong to the same abstracted signal infoset generated by the above-mentioned algorithms.

In the following discussion, we employ a skill where, if we can establish that within the same potential outcome isomorphism class, different signal infosets ϑ^1 and ϑ^2 share identical computational metrics involved in the calculation of Equation (3), i.e., $v(\text{Alg}, \vartheta^1) = v(\text{Alg}, \vartheta^2)$, we need not consider the specific implementation details of Algdis. As long as Algdis is a deterministic algorithm devoid of random factors, and for any j and t, we have $d_j^t(\vartheta^1) = d_j^t(\vartheta^2)$, it ensures that ϑ^1 and ϑ^2 will certainly be categorized into the same abstracted signal infoset in iteration t.

7.2.1 E[HS] Algorithm

The E[HS] algorithm is a signal abstraction algorithm designed for 2-player Texas Hold'em. For a player i's signal infoset ϑ, the E[HS] algorithm defines a metric for measuring the distance between a signal and cluster centroids, called **equity**. The formula for calculating this metric is as follows

$$e = 0 \cdot l + \frac{1}{2} \cdot t + w. \tag{4}$$

Here, l, t, w represent the probabilities of signals in the infoset ϑ for player i to lose, tie, and win in future showdown stages, respectively.

The E[HS] algorithm evenly assigns signal infosets to different classes based on their equity values. For instance, if the total number of classes is n, signals with equity values in the range 0 to $\frac{1}{n}$ are assigned to class 1, signals in the range $\left(\frac{1}{n}, \frac{2}{n}\right]$ are assigned to class 2, and so on.

Theorem 6 *The POI serves as a common refinement of algorithm E[HS].*

Proof We employ mathematical induction to prove that within the same potential outcome isomorphism class, different signal infosets will be assigned to the same abstracted signal infoset in the E[HS] algorithm for a set of parameters. To do this, it is sufficient to demonstrate that each signal infoset within the same potential outcome isomorphism class has the same equity value.

First, consider the last stage $r = |\mathfrak{r}|$. According to the expression in Equation (1), in this final stage of a 2-player scenario, the outcome isomorphism vector $(f_i^{(r),0}(\vartheta), f_i^{(r),1}(\vartheta), f_i^{(r),2}(\vartheta))$ for a signal infoset ϑ is simply the signal's losing rate, tying rate, and winning rate for player i, each multiplied by the number of signals in ϑ. This is in accordance with Equation (4), confirming our assumption holds.

Next, assuming the conclusion holds at stage $r + 1$, let's consider stage r. According to Algorithm 1, we can conclude that signal infosets within the same potential outcome isomorphism class $\tilde{\vartheta}^{(r)}$ share the same potential outcome feature. As stage r is not the final stage, the potential outcome feature of a signal infoset ϑ at this stage can be represented as $(\tilde{\vartheta}_1^{(r+1)}, \dots, \tilde{\vartheta}_{n^{(r+1)}}^{(r+1)})$, where $\tilde{\vartheta}_1^{(r+1)} \leq \dots \leq \tilde{\vartheta}_{n^{(r+1)}}^{(r+1)}$ are the indices of potential outcome isomorphism classes to which the signal infoset in $\tilde{\vartheta}^{(r)}$ can transition at stage $r + 1$, sorted in ascending order. According to our assumption, at stage $r + 1$, signal infosets within the same potential outcome isomorphism class have the same equity values. Therefore, for any potential outcome isomorphism class $\tilde{\vartheta}^{(r)}$ within stage r, the equity value vectors of signal infosets ϑ are the same and can be represented as

$$e = \sum_{i=1}^{n^{(r+1)}} \frac{1}{n^{(r+1)}} e_i.$$

Here, e_i represents the equity value of $\tilde{\vartheta}_i^{(r+1)}$ in stage $r + 1$. This completes the proof by mathematical induction, demonstrating that in all stages, the signal infosets within the same potential outcome isomorphism class have the same equity values. ■

7.2.2 PA&PAEMD Algorithm

The E[HS] algorithm, while proficient in reflecting the expected strength of hand rankings, lacks the ability to characterize the changes in hand strength across different stages of the game. In response to this limitation, Gilpin et al. (2007) introduced the PA algorithm. This algorithm employs a bottom-up approach, starting from the final stage, to perform k-means clustering on hands. Clustering in the final stage is based on the distances between cluster centroids and hand equities. For other stages, histograms are constructed for each cluster centroid to the next stage's centroids, as well as for each hand to the next stage's centroids.

The clustering process utilizes the L_2 distance between the histograms of cluster centroids and hand histograms, ensuring a more nuanced understanding of hand strength dynamics throughout the game. Building upon the PA algorithm, Ganzfried and Sandholm (2014) further enhanced it with the PAEMD algorithm. This iteration incorporates the EMD between histograms of cluster centroids and hand histograms in non-final stages, alongside an efficient approximate method for calculating this distance.

It's crucial to note that these algorithms involve techniques such as approximating equity expectations using sampling and approximating the EMD using heuristics. While effective, these techniques inherently entail certain randomness and distance asymmetry characteristics. Therefore, the subsequent proof will be applicable only under conditions where accurate equity values and standard EMD distances are maintained.

Theorem 7 *The POI serves as a common refinement of algorithm PA and PAEMD.*

Proof We first consider the last stage, $r = |\mathfrak{r}|$. It suffices to demonstrate that every signal infoset within the same potential outcome isomorphism class holds the same equity value, akin to the E[HS] algorithm's case, i.e., the conclusion holds.

Next, assuming that the conclusion holds at stage $r + 1$, it suffices at stage r to prove that every distinct signal infoset within the same potential outcome isomorphism class has the same histogram of PA algorithm's next-stage cluster centroids. For any potential outcome isomorphism class $\tilde{\vartheta}^{(r+1)}$ at stage r, any pair of signal infosets ϑ and ϑ' from $\tilde{\vartheta}^{(r+1)}$ share an identical potential outcome feature. Since r is not the final stage, this potential outcome feature is represented as a sparse histogram over potential outcome isomorphism classes from stage $r + 1$, denoted as $(\tilde{\vartheta}_1^{(r+1)}, \ldots, \tilde{\vartheta}_{n^{(r+1)}}^{(r+1)})$. Consequently, there exist $\vartheta_1, \vartheta'_1 \in \tilde{\vartheta}_1^{(r+1)}, \ldots, \vartheta_{n^{(r+1)}}, \vartheta'_{n^{(r+1)}} \in \tilde{\vartheta}_{n^{(r+1)}}^{(r+1)}$, where $\vartheta_1, \ldots, \vartheta_{n^{(r+1)}}$ are signal infosets that ϑ can transition to in stage $r+1$, and $\vartheta'_1, \ldots, \vartheta'_{n^{(r+1)}}$ are signal infosets that ϑ' can transition to in stage $r+1$. According to the induction hypothesis, ϑ_j and ϑ'_j are also assigned to the same cluster centroid C_j under the PA algorithm with the given parameters ψ. Therefore, ϑ and ϑ' share the same next-stage cluster centroid histogram in the PA algorithm with the given parameters ψ, and their sparse representations are both $(C_1, \ldots, C_{n^{(r+1)}})$. We have now completed the proof by mathematical induction, demonstrating that across all stages, distinct signal infosets within the same potential outcome isomorphism class in the outcome isomorphism will be assigned to the same abstracted signal infoset in the PA algorithm with a set of parameters ψ. ■

7.3 Limitations of Potential Outcome Isomorphism

Through the above discussion, we have learned that POI is a common refinement for both the E[HS] algorithm and the PA&PAEMD algorithm. In other words, the strategies trained on the abstracted signal infosets generate by these two algorithms will not surpass those trained using POI, at least in one-player-abstraction-perspective. However, POI has its clear limitations. Table 1 displays the quantity of original signal infosets at each stage of Texas Hold'em, along with the quantity of abstracted signal infosets that both the LI and POI can identify at each stage. It can be observed that as the game progresses, the

Stage	Null Abstraction	LI	POI
Preflop	$\binom{52}{2}$=1326	169	169
Flop	$\binom{52}{2,3}$=25989600	1286792	1137132
Turn	$\binom{52}{2,3,1}$=1195521600	55190538	2337912
River	$\binom{52}{2,3,1,1}$=5379847200	2428287420	20687

Table 1: Quantity of abstracted signal Infosets for various signal abstractions in HUNL&HULHE

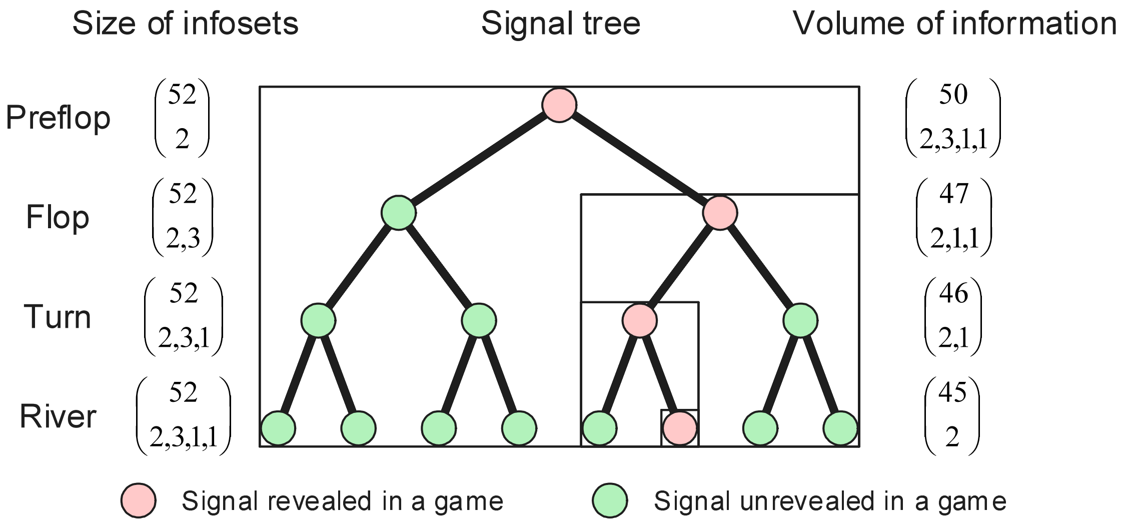
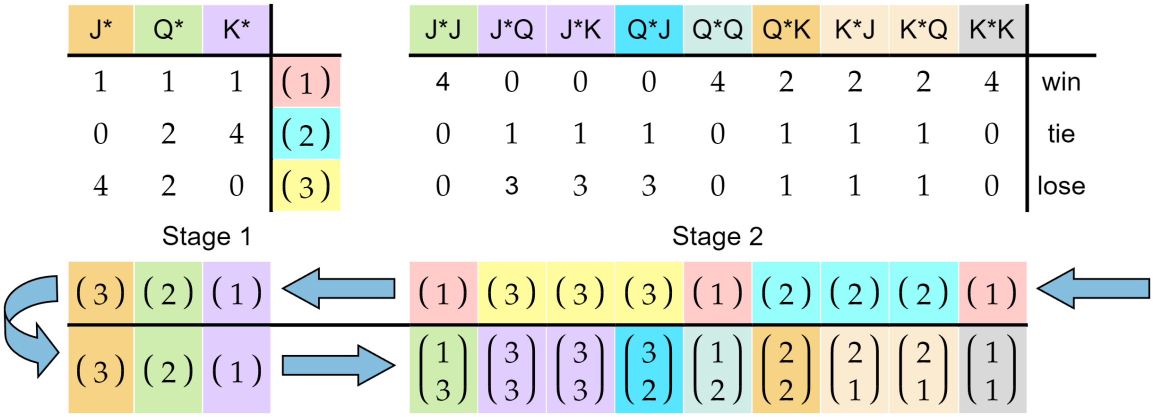


Figure 4: Two factors influencing the quantity of abstracted signal infosets

quantity of original signal infosets gradually increases, forming a triangular pattern. The LI algorithm can capture this pattern while simplifying the data. However, POI fails to reflect this pattern, and its quantity of abstracted signal infosets exhibits a spindle-shaped pattern, i.e., fewer in the early and late stages of the game and more in the middle stages. This indicates that POI excessively simplifies the data in the later stages of the game.

Let's delve into the reasons behind this phenomenon. As illustrated in Figure 4, the factors influencing the scale of potential outcome isomorphism classes mainly revolve around two aspects. Firstly, the quantity of signal infosets to be identified grows as the game progresses. The greater the quantity of signal infosets, the more distinct potential outcome isomorphism classes can be identified. Secondly, the volume of information contained within the potential outcome features plays a pivotal role. Take the River stage as an example, the potential outcome features consider all possible showdown outcomes with opponent hands when drawing two private cards from the remaining 45 cards. At the Preflop stage, the potential outcome features encompass scenarios where, after removing one's own two private cards from the remaining 52 cards, two private cards of the opponent and five community cards, resulting in all possible showdown outcomes. Due to POI being the type



J*	Q*	K*	
1	1	1	(1)
0	2	4	(2)
4	2	0	(3)

Stage 1

J*J	J*Q	J*K	Q*J	Q*Q	Q*K	K*J	K*Q	K*K	
4	0	0	0	4	2	2	2	4	win
0	1	1	1	0	1	1	1	0	tie
0	3	3	3	0	1	1	1	0	lose

Stage 2

(3) (2) (1)
(3) (2) (1)
(1) (3) (3) (3) (1) (2) (2) (2) (1)
$\binom{1}{3}$ $\binom{3}{3}$ $\binom{3}{3}$ $\binom{3}{2}$ $\binom{1}{2}$ $\binom{2}{2}$ $\binom{2}{1}$ $\binom{2}{1}$ $\binom{1}{1}$

Figure 5: The process of constructing 1-recall outcome isomorphism in Leudc Hold'em

of imperfect-recall signal abstraction that demands players to forget all past information, the volume of information gradually diminishes as the game progresses. Consequently, the potential outcome features exhibit an increased probability of repetition due to reduced volume of information, thereby leading to a reduction in the scale of distinct potential outcome isomorphism classes identified. The combined influence of these two factors results in a trend where the quantity of potential outcome isomorphism classes follows a spindle-shaped distribution, and the phenomenon of excessive abstraction becomes more evident in the later stages.

8 K-Recall Outcome Isomorphism

We have identified the underlying reasons for the phenomenon of excessive abstraction in POI. This arises from the gradual reduction of information within potential outcome features, as discussed in the preceding section. During the identification of a signal infoset, only the information spanning from the current stage to the end of the game is employed, i.e., imperfect-recall that demands players to forget all past information. This, in turn, increases the probability of potential outcome feature repetition. In this section, we will introduce the concept of k-recall outcome isomorphism (KROI), which addresses this issue by supplementing information from past stages to mitigate the problem of excessive abstraction.

As illustrated in Algorithm 3, for a signal infoset at stage r, we can compute its k-recall outcome isomorphism, where $k = 0, \ldots, r-1$. The primary procedure of the algorithm closely mirrors that of the POI algorithm, with the only distinction being the replacement of the original potential outcome features with k-recall outcome features in lines 5 and 11. The k-recall outcome features for a signal infoset ϑ constitute a vector of length $k+1$, where the component at index j stores the index of potential outcome isomorphism class for the predecessor signal infoset of ϑ at stage $r-j$. Since POI incorporates information from the current stage to the end of the game, k-recall outcome features encompass the information of ϑ spanning the preceding $k+1$ stages, including stage r.

Algorithm 3 K-Recall Outcome Isomorphism

```
Require:
    $r \in \{1, \ldots, |\mathfrak{r}|\}$. Stages.
    $\mathit{\Theta}_i^{(r)}$. Signal infoset space.
    $Index_i(r, \cdot) : \mathit{\Theta}_i^{(r)} \mapsto \mathbb{N}$. Signal infoset index function.
    $\mathcal{D}_i^{(1)}, \ldots, \mathcal{D}_i^{(r)}$. Potential outcome isomorphism maps.
procedure KRecallOutcomeIsomorphism($r$,$k$, $\mathit{\Theta}_i^{(r)}$,$\mathcal{D}_i^{(1)}$,..., $\mathcal{D}_i^{(r)}$)
    Initialize $\mathcal{KC}_i^{(r)}$ vector as empty.
    Initialize $\mathcal{KD}_i^{(r)}$ array arbitrarily with length $|\mathit{\Theta}_i^{(r)}|$.
    for $\vartheta \in \mathit{\Theta}_i^{(r)}$ do
        $feature \leftarrow$ KRecallOutcomeFeature($r$, $k$, $\vartheta$,$\mathcal{D}_i^{(1)}$,..., $\mathcal{D}_i^{(r)}$).
        Append $feature$ to $\mathcal{KC}_i^{(r)}$.
    end for
    Eliminate duplicates from $\mathcal{KC}_i^{(r)}$.
    Sort the elements of $\mathcal{KC}_i^{(r)}$ in lexicographical order.
    Construct hash table $\mathcal{KCI}_i^{(r)}$ from $\mathcal{KC}_i^{(r)}$. Store the index $lexid$ and value $feature$ of $\mathcal{KC}_i^{(r)}$ in $\mathcal{KCI}_i^{(r)}$ as key-value pairs $(feature, lexid)$.
    for $\vartheta \in \mathit{\Theta}_i^{(r)}$ do
        $feature \leftarrow$ KRecallOutcomeFeature($r$, $k$, $\vartheta$,$\mathcal{D}_i^{(1)}$,..., $\mathcal{D}_i^{(r)}$).
        $idx \leftarrow Index_i(r, \vartheta)$.
        Update $\mathcal{KD}_i^{(r)}[idx]$ with $\mathcal{KCI}_i^{(r)}[feature]$.
    end for
    return $\mathcal{KD}_i^{(r)}$.
end procedure
procedure KRecallOutcomeFeature($r$,$k$, $\vartheta$,$\mathcal{D}_i^{(1)}$,..., $\mathcal{D}_i^{(r)}$)
    Initialize $feature$ array arbitrarily with length $k + 1$.
    for $j = r$ to $r - k$ do
        $\vartheta_j \leftarrow$ the signal infoset predecessor in the $j$-th stage of $\vartheta$.
        $idx_j \leftarrow Index_i(j, \vartheta_j)$.
        $feature[r - j] \leftarrow \mathcal{D}_i^{(j)}[idx_j]$.
    end for
    return $feature$.
end procedure
```

Figure 5 illustrates the process of constructing 1-ROI in Leudc Hold'em. The bottom-up part follows the same procedure as constructing POI. In the first stage, only 0-ROI construction is possible, which aligns with the result of POI. The change occurs in the second stage, where each signal infoset is assigned the current stage's POI class label, as well as the POI class label where its predecessor in the previous stage is located. These labels form a vector, known as the 1-recall outcome feature. We group signal infosets with the same 1-recall outcome feature into the same abstracted signal infoset. It can be observed

Stage	0-ROI	1-ROI	2-ROI	3-ROI
Preflop	169	169[1]	169[2]	169[3]
Flop	1137132	1241210	1241210[2]	1241210[3]
Turn	2337912	38938975	42040233	42040233[3]
River	20687	39792212	586622784	638585633

Table 2: Quantity of abstracted signal Infosets for k-recall outcome isomorphism in HUNL&HULHE

that in Leduc Hold'em, the quantity of 1-ROI classes in the second stage is 7, higher than the 3 abstracted signal infosets of POI. This represents an expansion of the resolution of the outcome-based imperfect-recall algorithm.

Table 2 presents the quantity of abstracted signal infosets identified by k-recall outcome isomorphism at each stage in HUNL&HULHE, varying with k. When $k = 0$, k-recall outcome isomorphism is essentially equivalent to POI. For stages where $r \leq k$, the quantity of abstracted signal infosets is directly inherited from $(k-1)$-ROI. Furthermore, when incorporating information from all preceding stages (e.g., 3-ROI in HUNL&HULHE), the quantity of abstracted signal infosets follows a triangular distribution, mirroring that observed in the LI and the original game's signal infosets.

It should be noted that for games with a maximum stage of r, r-ROI serves as a perfect-recall signal abstraction. However, KROI can lead to more generalized outcome-based imperfect-recall signal abstraction algorithms, allowing players to retain partial memory instead of forgetting everything. For instance, when designing abstraction algorithms, we only cluster $\mathcal{D}_i^{(r)}$ at each stage r, without considering the connections between stages. Such an abstraction algorithm remains imperfect-recall, as it does not require consistent paths between abstracted signal infosets. Thus, we obtain a signal abstraction framework with higher resolution, with KROI serving as their resolution boundary.

9 Experiment and Result

Figure 5 and Table 2 illustrate that KROI expands the resolution boundary of the outcome-based imperfect-recall algorithm. However, it also indicates that there is still a gap between the resolution of KROI and LI. We are interested in understanding to what extent this gap can impact its performance in games, as well as how much the resolution improvement of KROI relative to POI can enhance its performance in games.

Experiments were conducted on Numeral211 Hold'em, a simplified version of two-player limit Texas Hold'em, which is divided into three stages; specific rules are detailed in Appendix D.2. In Numeral211 Hold'em, the quantity of abstracted signal infosets identified by each method at different stages is presented in Table 3. The performance of 0-Recall Outcome Isomorphism (i.e., Potential Outcome Isomorphism, POI), 2-Recall Outcome Isomorphism (2-ROI), and Lossless Isomorphism (LI) was evaluated in following experiments.

[1] Inherited from 0-ROI

[2] Inherited from 1-ROI

[3] Inherited from 2-ROI

Stage	0-ROI	1-ROI	2-ROI	LI
Preflop	100	100[1]	100[2]	100
Flop	2250	2260	2260[2]	2260
Turn	3957	51176	51228	62020

Table 3: Quantity of abstracted signal Infosets for k-recall outcome isomorphism and lossless isomorphism in Numeral211 Hold'em

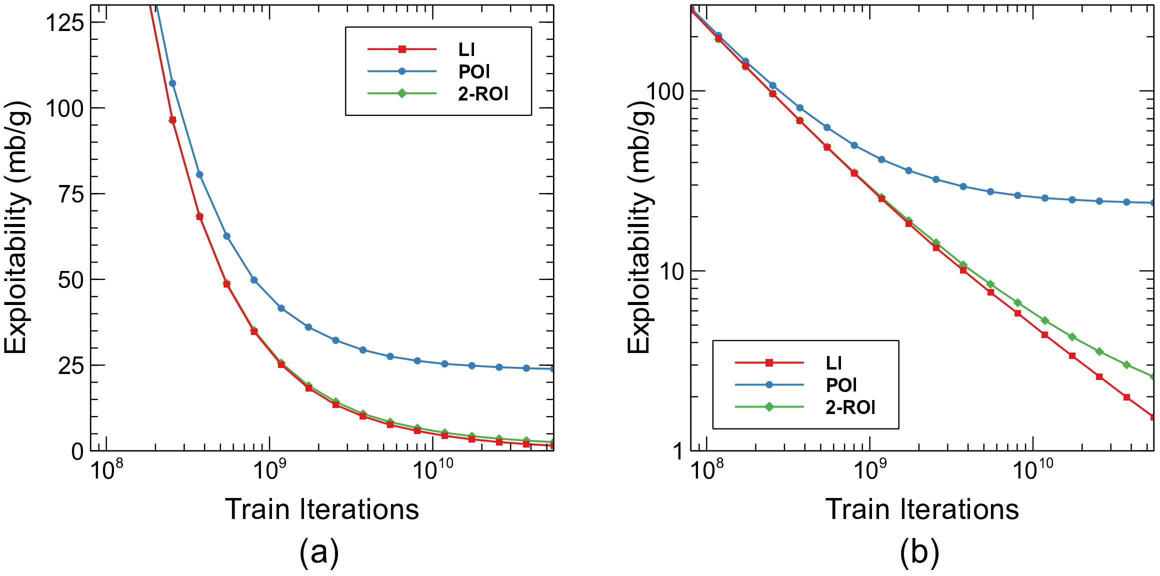


Figure 6: Convergence rate of exploitability within Symmetric abstraction settings

9.1 Asymmetric Abstraction in CFR Training

Our fisrt investigation focuses on the one-player-abstraction-perspective. Given a signal abstraction $\tilde{\alpha}^{\Theta} = (\tilde{\alpha}_1^{\Theta}, \tilde{\alpha}_2^{\Theta})$, we construct scenarios where player 1 employs the signal abstraction, while player 2 does not, resulting in the signal abstracted game $\tilde{\Gamma}^1 = \tilde{\Gamma}^{(\tilde{\alpha}_1^{\Theta}, \Theta_2)}$. Conversely, we adjust the setup such that player 1 does not use abstraction, whereas player 2 does, thereby creating another signal abstracted game $\tilde{\Gamma}^2 = \tilde{\Gamma}^{(\Theta_1, \tilde{\alpha}_2^{\Theta})}$. Subsequently, by applying the chance-sampled counterfactual regret minimization (CSMCCFR) (Zinkevich et al., 2007; Lanctot et al., 2009), we iteratively solve for ϵ-Nash equilibrium and map the resulting strategies back to the original game $\tilde{\Gamma}$ to obtain $\sigma^{1,*} = (\sigma_1^{1,*}, \sigma_2^{1,*})$ and $\sigma^{2,*} = (\sigma_1^{2,*}, \sigma_2^{2,*})$. Ultimately, by amalgamating these strategies into $\sigma^* = (\sigma_1^{1,*}, \sigma_2^{2,*})$, which we called **asymmetric abstraction Nash strategy**, we adopt the σ^* strategy in real-game scenarios as a means to counter strategies available in a non-abstracted setting from an abstracted viewpoint. The exploitability of σ is then analyzed in terms of milli blinds per game (mb/g). The experimental results are depicted in Figure 6, where (a)

[1] Inherited from 0-ROI

[2] Inherited from 1-ROI

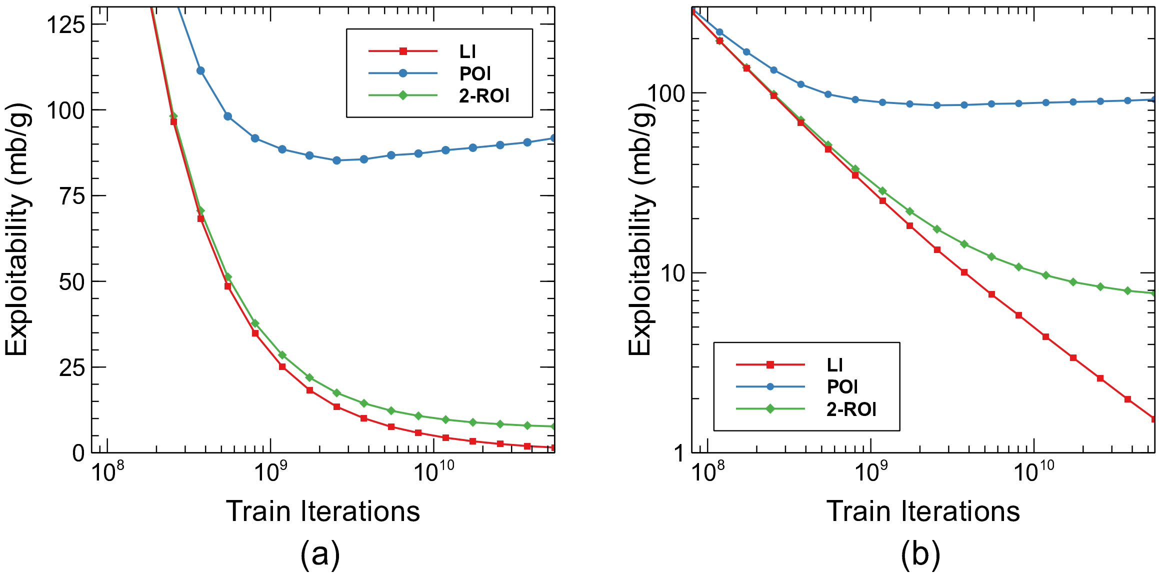


Figure 7: Symmetric abstraction experiment

showcases the results plotted on a linear scale of exploitability, while (b) presents the same results on a logarithmic scale.

Within the asymmetric abstraction setting, the issue of abstraction pathology does not arise. All asymmetric abstraction Nash strategy derived from abstraction refined by $\tilde{\alpha}^{\Theta}$ exhibit exploitability that is at least as those asymmetric abstraction Nash strategies derived from $\tilde{\alpha}^{\Theta}$. We note a significant discrepancy between the POI method and the ground truth benchmark of the LI method, indicating that the abstraction refined by the POI, including E[HS] and PA&PAEMD, diverge substantially from the ground truth. In contrast, the performance gap between the 2-ROI method and the ground truth is minimal, demonstrating a high degree of approximation to the ground truth even under logarithmic scale as shown in Figure 6b.

9.2 Symmetric Abstraction in CFR Self-Play Training

Subsequently, we shift our focus to the setting of symmetric abstraction, which more closely aligns with the practical scenarios of AI training. Given a specific signal abstraction $\tilde{\alpha}^{\Theta}$, we construct the corresponding signal abstracted game $\tilde{\Gamma}^{\tilde{\alpha}^{\Theta}}$. Utilizing the CSMCCFR algorithm, we iteratively solve for the ϵ-Nash equilibrium and map the resulting strategy back to the original game $\tilde{\Gamma}$, obtaining σ^{*}, and then calculate its exploitability. The experimental results are presented in Figure 7.

The results are consistent with the experiments presented in Section 9.1, where the POI method exhibits a significant gap from the ground truth, while the 2-ROI method shows results that are closer to the ground truth. Furthermore, within the setting of symmetric abstraction, while both POI and 2-ROI methods reveal a greater disparity from the ground truth due to the extensive assumptions regarding the opponent's strategy space, it is specifically POI that displays overfitting phenomena. In a larger game, abstraction pathology does not significantly affect such overall trends, indicating that future abstraction algorithms utilizing KROI as a common refinement may further enhance AI's competitiveness.

10 Discussion and Future Work

The main purpose of this study is to identify a low-resolution boundary present in current outcome-based imperfect-recall algorithms, which we refer to as the POI resolution boundary. This boundary affects the competitiveness of AI built using signal abstraction algorithms constrained by the POI resolution boundary, especially when scaling up computational resources, leading to a limited performance. The root of this problem lies in the algorithms neglecting historical information in gameplay. Under the outcome-based imperfect-recall framework, we introduce KROI, which incorporates historical information and exhibits a finer resolution boundary compared to POI. We refer to this as the KROI resolution boundary. Experimental results demonstrate that these oversimplified resolution boundaries indeed impact AI performance.

In future research, we will develop signal abstraction algorithms aiming to surpass the POI resolution boundary and target the KROI resolution boundary. Although KROI, as a signal abstraction, has perfect-recall, signal abstraction algorithms targeting the KROI resolution boundary do not necessarily require perfect-recall.

Our experimental results do not conclusively demonstrate that signal abstraction targeting the KROI resolution boundary outperforms signal abstraction targeting the POI resolution boundary with the same quantity of abstract signal infosets. We will verify this proposition in future research.

Appendix A. Imperfect-Information Games

Definition 8 *An extensive-form game $\Gamma = \langle \mathcal{N}_c, H, Z, \rho, A, \chi, \tau, P, u \rangle$ is a structured tuple, where $\mathcal{N}_c$ is the set of extended players, H is the set of non-terminal nodes, Z is the set of terminal nodes, ρ is the player function, A is the action set, χ is the action mapping function, τ is the successor function, P is the random event function, and u is the vector of payoff functions. More specifically:*

- *$\mathcal{N}_c = \mathcal{N} \cup c$, where $\mathcal{N} = \{1, ..., N\}$ is the set of players, and c is a special player, called nature, whose actions represent random events in the game.*
- *$\rho : H \mapsto \mathcal{N}_c$ maps each non-terminal node to the unique extended player who can act at that node.*
- *$X = H \cup Z$ is the set of all nodes. Where:*
  - *$x^o \in X$ is the initial node of the game.*
  - *$H_i = \{h \in H \mid \rho(h) = i\}$ represents the set of non-terminal nodes where the acting player is i, and nodes where the acting player is nature are called chance nodes.*
- *$\chi : H \mapsto 2^A$ maps each non-terminal node to the set of actions that can be taken.*
- *$\tau : H \times A \mapsto X$. Specifically, $\tau(h, a)$ represents the new node reached after taking action $a \in \chi(h)$ at non-terminal node $h \in H$. In extensive-form games, each node has a unique path of arrival. In other words, if $\tau(h_1, a_1) = \tau(h_2, a_2)$, then it must be that $h_1 = h_2$ and $a_1 = a_2$.*
- *$P(h) \in \Delta(\chi(h))$ represents the probability distribution of random events occurring at chance node $h \in H_c$, for convenience, $P(a \mid h)$ can be used to represent the probability of event $a \in \chi(h)$ occurring at h.*
- *$u = (u_1, ..., u_N)$, where $u_i : Z \mapsto \mathbb{R}$, $u_i(z)$ represents the payoff obtained by player $i \in \mathcal{N}$ at terminal node $z \in Z$.*

Definition 9 *An imperfect-information game $\Gamma = (\mathcal{N}_c, H, Z, \rho, A, \chi, \tau, P, u, \mathcal{I})$ is a structured tuple, where $(\mathcal{N}_c, H, Z, \rho, A, \chi, \tau, P, u)$ constitutes an extensive-form game, and $\mathcal{I}$ is the set of infosets. More specifically:*

- *$\mathcal{I} = \bigcup_{i \in \mathcal{N}} \mathcal{I}_i$:*
  - *$\mathcal{I}_i$ is a partition of H_i.*
  - *$I \in \mathcal{I}_i$ is an infoset for player i, consisting of nodes indistinguishable to player i. For any $h, h' \in H_i$, a necessary condition for player i being unable to distinguish between h and h' is $\chi(h) = \chi(h')$, and $\chi(I)$ can be used to represent the set of actions for the infoset I.*

Appendix B. Abstraction in Imperfect-Information Games

Definition 10 *In imperfect-information games, $\alpha = (\alpha_1, ..., \alpha_N)$ is referred to as a abstraction profile, where for player i, a abstraction $\alpha_i = \langle \alpha_i^{\mathcal{I}}, \alpha_i^{\chi} \rangle$ is a tuple consisting of the following components:*

- *$\alpha_i^{\mathcal{I}}$ is a partition of H_i. For any $\hat{I} \in \alpha_i^{\mathcal{I}}$, it is possible to identify some infosets within $\mathcal{I}_i$ that share the same action spaces, such that the collection of these infosets forms a partition of $\hat{I}$. $\hat{I}$ is referred to as a abstraction infoset.*
- *α_i^{χ} is a function defined on H_i. $\alpha_i^{\chi}(h) \subseteq \chi(h)$ represents the abstracted action set for non-terminal node $h \in H_i$. For all non-terminal nodes h and h' within the same abstracted infoset $\hat{I} \in \alpha_i^{\mathcal{I}}$, it holds that $\alpha_i^{\chi}(h) = \alpha_i^{\chi}(h')$.*

Appendix C. Perfect-Recall Games

Definition 11 *In an extensive-form imperfect-information game Γ, player i is said to have perfect-recall if the following two conditions are satisfied:*

1. *In any path of Γ, no two nodes within the same information set $I \in \mathcal{I}_i$ of player i can occur simultaneously. In this case, for any node x satisfying $I \sqsubseteq x$, $I[x]$ denotes the node in the path from x^o to x that belongs to I.*
2. *For any path in Γ where only the head x and the tail x' have player i as the acting player, if there exists a node $\hat{x}'$ in $\mathcal{I}(x')$—the infoset of x'—that is different from x', then there must exist another path $\langle \hat{x}, \ldots, \hat{x}' \rangle$ where $\hat{x} \in \mathcal{I}(x)$ and only the head and the tail have player i as the acting player, and the two paths have the same action chosen at the heads x and $\hat{x}$.*

If each player in an imperfect-information game has perfect-recall, then the game is said to have perfect-recall.

Appendix D. Some Simplified Texas Hold'em-style Games

In essence, Leduc Hold'em provides a strategic platform for two-player poker, balancing simplicity with depth, making it an ideal candidate for computational analysis and the exploration of advanced gameplay strategies.

D.1 Leduc Hold'em

Leduc Hold'em is a poker variant designed for two players, conceived primarily as a platform for computer game-playing research and introduced by Waugh et al. (2009a). It was engineered to possess a strategic depth akin to Texas Hold'em while maintaining a manageable scale suitable for the development of intelligent gameplay strategies.

The game unfolds as follows:

1. **Ante:** Each player antes one chip into the pot at the start of the hand.

Rank	Hand Equivalence Classes	Hand Poker	Prob.
1	Pair of Kings	KK	1/15
2	Pair of Queens	QQ	1/15
3	Pair of Jacks	JJ	1/15
4	King-high with Queen	KQ or QK	4/15
5	King-high with Jack	KJ or JK	4/15
6	Queen-high with Jack	QJ or JQ	4/15

Table 4: Hand ranks of Leduc Hold'em

2. **Deck:** The deck consists of only six cards: two Jacks (J), two Queens (Q), and two Kings (K).

3. **Hole Card:** Both players are dealt one private card face down, known as the hole card.

4. **First Betting Stage:** Following the distribution of hole cards, a stage of betting occurs. Players can choose to check or bet, with the bet size set at two chips.

5. **Flop:** After the initial betting stage, a single community card, termed the flop, is revealed from the deck.

6. **Second Betting Stage:** Another stage of betting takes place after the flop, with the bet size increasing to four chips.

7. **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 4 show the hand ranks of Leduc Hold'em.

8. **Betting Options:** Throughout the game, players have options to fold, call, or raise. Each betting stage permits only one bet and one raise per player, with fixed bet sizes of two chips in the first stage and four chips in the second.

D.2 Numerall211 Hold'em

The Numeral211 Hold'em introduced in this paper is a poker variant more complex than Leduc and Rhode Island Hold'em, yet significantly simpler than HULHE. With ample diversity in hand possibilities, it serves as an ideal testbed for studying hand abstraction tasks.

The game unfolds as follows:

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 Stage:** Following the distribution of hole cards, a stage of betting occurs. Players can choose to check or bet, with the bet size set at 10 chips.

5. **Flop:** After the initial betting stage, a single community card, termed the flop, is revealed from the deck.

6. **Second Betting Stage:** Another stage of betting takes place after the flop, with the bet size increasing to 20 chips.

7. **Turn:** After the Second betting stage, another community card, termed the turn, is revealed from the deck.

8. **Third Betting Stage:** Another stage of betting takes place after the turn, with the bet size still set at 20 chips.

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 5 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 stage, the total sum of bets and raises is limited to a maximum of 4, with fixed bet sizes of 10 chips in the first stage and 20 chips in the last two betting stages.

Appendix E. A Transition from Games with Ordered Signals to Imperfect-Information Games

The games with ordered signals are a special type of imperfect-information game. We can transform an ordered signal game into an imperfect-information game, and we will demonstrate this process. Given a game with ordered signals $\Gamma = (\tilde{\mathcal{N}}, \tilde{H}, \tilde{Z}, \tilde{\rho}, \tilde{A}, \tilde{\chi}, \tilde{\tau}, \gamma, \Theta, \varsigma, O, \omega, \succeq, \tilde{u})$, we make some auxiliary definitions. Let $\mathfrak{r} = \{r = \gamma(\tilde{x}) \mid \tilde{x} \in \tilde{X}\}$ be the set of reachable stages in this ordered signal game. We can partition the public nodes by stage, such that $\tilde{H}^{(r)} = \{h \in \tilde{H} \mid \gamma(h) = r\}$, $\tilde{Z}^{(r)} = \{z \in \tilde{Z} \mid \gamma(z) = r\}$, $\tilde{X}^{(r)} = \tilde{H}^{(r)} \cup \tilde{Z}^{(r)}$.

Construct $\Gamma' = (\mathcal{N}_c, H, Z, \rho, A, \chi, \tau, P, u, \mathcal{I})$, where:

- $\mathcal{N}_c = \tilde{\mathcal{N}} \setminus \{pub\}$.
- $H = \bigcup_{r \in \mathfrak{r}} (\Theta^{(r)} \times \tilde{H}^{(r)})$, $Z = \bigcup_{r \in \mathfrak{r}} (\Theta^{(r)} \times \tilde{Z}^{(r)})$, $X = H \cup Z$.
- ρ is a function on X, such that for $x = (\theta, \tilde{x}) \in X$,

$$\rho(x) := \tilde{\rho}(\tilde{x}).$$

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 5: Hand ranks of Numeral211 Hold'em

- $A = \tilde{A} \cup \{\mathit{Reveal}(\theta) \mid \theta \in \Theta\} \setminus \{\mathit{Reveal}\}$, $\mathit{Reveal}(\theta)$ represents the chance player unveil a signal θ.
- χ is a function on H, such that for $h = (\theta, \tilde{h}) \in H$,

$$\chi(h) := \begin{cases} \tilde{\chi}(\tilde{h}) & \text{if } \tilde{h} \notin \tilde{H}_c, \\ \{\mathit{Reveal}(\theta') \mid \theta' \in \Theta, \varsigma(\theta' \mid \theta) > 0\} & \text{if } \tilde{h} \in \tilde{H}_c. \end{cases}$$

- τ is a function on $H \times A$, such that for $h = (\theta, \tilde{h}) \in H$ and $a \in A$,

$$\tau(h, a) := \begin{cases} (\theta', \tilde{\tau}(\tilde{h}, \mathit{Reveal})) & \text{if } a = \mathit{Reveal}(\theta'), \\ (\theta, \tilde{\tau}(\tilde{h}, a)) & \text{otherwise.} \end{cases}$$

- P is a function on H_c regarding the probability of random events, such that for $h = (\theta, \tilde{h}) \in H_c$,

$$P(\mathit{Reveal}(\theta') \mid h) := \varsigma(\theta' \mid \theta).$$

- u is a function on Z, such that for $z = (\theta, \tilde{z}) \in Z$,

$$u(z) := \tilde{u}(\theta, \tilde{z}).$$

- $\mathcal{I} = \bigcup_{i \in N_c} \mathcal{I}_i$, where:
  - $I(h) = \{(\theta', \tilde{h}) \mid \theta' \in \vartheta_{\rho(h)}(\theta)\}$, for $h = (\theta, \tilde{h})$.
  - $\mathcal{I}_i = \{I(h) \mid h \in H_i\}$.

It can be verified that Γ' is an imperfect-information game.

References

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.

Béla Bollobás. *Combinatorics: Set Systems, Hypergraphs, Families of Vectors, and Combinatorial Probability.* Cambridge University Press, USA, 1986. ISBN 0521330599.

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

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

GH Freeman. The tactics of liar dice. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 38(3):507–516, 1989.

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.

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.

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.

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, 2007a.

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

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.

John C Harsanyi. Games with incomplete information. *The American Economic Review*, 85(3):291–303, 1995.

Michael Johanson. Robust strategies and counter-strategies: Building a champion level computer poker player. Master's thesis, University of Alberta, 2007.

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.

Christian Kroer and Tuomas Sandholm. Extensive-form game abstraction with bounds. In *ACM Conference on Economics and Computation (EC)*, pages 621–638, 2014.

Christian Kroer and Tuomas Sandholm. Imperfect-recall abstractions with bounds in games. In *ACM Conference on Economics and Computation (EC)*, pages 459–476, 2016.

Christian Kroer and Tuomas Sandholm. A unified framework for extensive-form game abstraction with bounds. *International Conference on Neural Information Processing Systems (NeurIPS)*, 31, 2018.

Harold W Kuhn. A simplified two-person poker. *Contributions to the Theory of Games*, 1: 97–103, 1950.

Harold W Kuhn. Extensive games and the problem of information. *Contributions to the Theory of Games*, 2(28):193–216, 1953.

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.

Marc Lanctot, Richard Gibson, Neil Burch, Martin Zinkevich, and Michael Bowling. No-regret learning in extensive-form games with imperfect recall. In *International Coference on International Conference on Machine Learning (ICML)*, pages 1035–1042, 2012.

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.

John Nash. Non-cooperative games. *Annals of Mathematics*, pages 286–295, 1951.

Tuomas Sandholm. The state of solving large incomplete-information games, and application to poker. *AI Magazine*, 31(4):13–32, 2010.

Tuomas Sandholm and Satinder Singh. Lossy stochastic game abstraction with bounds. In *ACM Conference on Electronic Commerce (ICEC)*, pages 880–897, 2012.

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.

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

Kevin Waugh, Nolan Bard, and Michael Bowling. Strategy grafting in extensive games. *International Conference on Neural Information Processing Systems (NeurIPS)*, 22, 2009a.

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, 2009b.

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 2009c.

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.