POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Marriage between Adversarial Team Games and 2-player Games: Enabling Abstractions, No-regret Learning, and Subgame Solving

Original
A Marriage between Adversarial Team Games and 2-player Games: Enabling Abstractions, No-regret Learning, and Subgame Solving / Carminati, L., Cacciamani, F., Ciccone, M., Gatti, N.. - 162:(2022), pp. 2638-2657. (39th International Conference on Machine Learning Baltimore, Maryland (USA) 17-23 July 2022).

Availability:
This version is available at: 11583/2971619 since: 2022-09-22T13:42:04Z

Publisher:
PMLR

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

28 November 2026

A Marriage between Adversarial Team Games and 2-player Games: Enabling Abstractions, No-regret Learning, and Subgame Solving

Luca Carminati [1] **Federico Cacciamani** [1] **Marco Ciccone** [2] **Nicola Gatti** [1]

Abstract

Ex ante correlation is becoming the mainstream approach for *sequential adversarial team games*, where a team of players faces another team in a zero-sum game. It is known that team members' asymmetric information makes both equilibrium computation APX-hard and team's strategies not directly representable on the game tree. This latter issue prevents the adoption of successful tools for huge 2-player zero-sum games such as, *e.g.*, abstractions, no-regret learning, and subgame solving. This work shows that we can recover from this weakness by bridging the gap between sequential adversarial team games and 2-player games. In particular, we propose a new, suitable game representation that we call *team-public-information*, in which a team is represented as a single coordinator who only knows information common to the whole team and prescribes to each member an action for any possible private state. The resulting representation is highly *explainable*, being a 2-player tree in which the team's strategies are behavioral with a direct interpretation and more expressive than the original extensive form when designing abstractions. Furthermore, we prove payoff equivalence of our representation, and we provide techniques that, starting directly from the extensive form, generate dramatically more compact representations without information loss. Finally, we experimentally evaluate our techniques when applied to a standard testbed, comparing their performance with the current state of the art.

[1]Politecnico di Milano [2]Politecnico di Torino. Correspondence to: Luca Carminati <luca.carminati@polimi.it>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

1. Introduction

Research efforts on imperfect-information games customarily focus on 2-player zero-sum games (*"2p0s"* games from here on), in which two players act receiving opposite payoffs. In this setting, superhuman performances have been achieved in real-world instances, such as *Poker Hold'em* (Brown & Sandholm, 2017b; 2019; Moravcík et al., 2017) and *Starcraft II* (Vinyals et al., 2019). The successful approach for 2p0s games is generally based on the generation of a game abstraction used *offline* to find a blueprint strategy which is refined *online* during the play.

In our work, we focus on sequential *adversarial team games* in which a team of 2 (or more) players cooperates against a common adversary or team of adversaries. In particular, we focus on *ex ante coordination*, in which the team members agree on a common strategy beforehand and commit to playing it during the game without communicating any further. The team members share the same payoffs and coordinate against an adversary having opposite payoffs, in face of private information given separately to each team member. Examples include collusion in poker games, bidding in the game of Bridge, and a team of drones acting against an intruder. Celli & Gatti (2018) show that the computation of a solution, called Team Maxmin Equilibrium with Correlation (TMEcor), is APX-hard. Furthermore, team members' asymmetric information makes a team equivalent to a single-player without *perfect recall* and therefore, as showed by Kuhn (1953), *behavioral* strategies defined on the game tree and *normal-form* strategies are not realization equivalent. In particular, normal-form strategies may lead to arbitrarily better outcomes than behavioral strategies. However, this comes at the cost of an exponential explosion of the strategy space and the impossibility to use tools for huge 2p0s games as normal-form strategies are not directly representable on game trees.

Related Work. To the best of our knowledge, Celli & Gatti (2018) are the first to compute the TMEcor of an adversarial team game by proposing the Hybrid Column Generation (HCG) algorithm. At each iteration, HCG exploits a Linear Program (LP) to compute a max-min solution and then an Integer LP (ILP) to find the team's best response to be added to the LP at the next iteration. Successively, Farina

et al. (2018) propose a variant of HCG, called Fictitious Team Play (FTP), in which the LP computing the max-min strategy is replaced by a step of the Fictitious Play algorithm (Brown, 1951). Later, Zhang et al. (2021), Zhang & An (2020b), Zhang & An (2020a), Farina et al. (2021) propose more efficient flavours of HCG and FTP algorithms. Among the above algorithms, the Faster Column Generation (FCG) algorithm (Farina et al., 2021) provides the best empirical performance. The rationale behind this class of approaches is to incrementally expand the LP strategy space to guess the actions in the equilibrium support without necessarily enumerating an excessively large portion of the space. The main weakness of this approach is the necessity to solve an ILP, which severely limits its scalability to large game instances even for the evaluation of the exploitability of a suboptimal solution. A recent alternative is proposed by Zhang & Sandholm (2022). The authors provide a generalization of the sequence form which, thanks to a suitable tree decomposition of the constraints, allows the description of a team's strategy space by a polytope. Thus, a TMEcor can be found by linear programming. This approach outperforms FCG with instances in which the degree of private information is limited. The idea to provide a convex representation of the strategy space adopted by Zhang & Sandholm (2022) is closely related to ours. The main differences reside in a better interpretability of our representation, together with the possibility to adopt abstractions Sandholm (2015); Gilpin et al. (2007), no-regret learning (Zinkevich et al., 2007; Celli et al., 2020), and subgame solving (Brown et al., 2018; Brown & Sandholm, 2017c).

We also mention Multi-Agent Reinforcement Learning (MARL) approaches proposed by Celli et al. (2019) and Cacciamani et al. (2021). These algorithms rely on implicit abstractions yielded by deep reinforcement learning to reduce the complexity of the problem. However, these approaches provide theoretical guarantees only in games in which team members have symmetric observability over other players' actions (chance included).

Original Contributions. As a preliminary step of our work, we first enrich the canonical extensive-form representation to capture information about public team members' observations. We call it extensive-form game *with visibility* (vEFG). Exploiting this representation, we provide an algorithmic procedure, called PUBLICTEAMCONVERSION, to convert an instance of adversarial team games into a 2p0s game, where a team is represented as a single coordinator who only knows information common to all team members and prescribes to each member an action for any possible private state. We formally prove that a Nash equilibrium of the converted game corresponds to a TMEcor in the original game and *vice versa*, thus enabling, for the first time, to the best of our knowledge, the adoption of techniques for 2p0s games to adversarial team games.

Differently from the representations previously proposed in the state of the art, *e.g.*, that by Zhang & Sandholm (2022), our representation is highly *explainable*, since the team's strategies are behavioral over the game tree with a direct interpretation. More precisely, *the coordination prescriptions sent to the team members in each public state can be interpreted as shared team conventions*. Remarkably, our representation also extends to adversarial settings the research line previously developed by Nayyar et al. (2013) and applied to cooperative games by Foerster et al. (2019) and Sokota et al. (2021), thus bridging the two approaches.

Furthermore, we show that our representation is more expressive than the extensive form as state/action abstractions applied to the extensive-form game can be captured by our representation, while the reverse does not hold. More importantly, the direct interpretability of our representation allows the design of techniques to prune and abstract the trees. In particular, we show that our techniques return a game representation with a size smaller than that generated by Zhang & Sandholm (2022), while guaranteeing explainability. Finally, we empirically evaluate the performance of no-regret algorithms applied to our representations.

2. Preliminaries

We introduce the basic concepts and definitions used throughout this work. For more details, we point an interested reader to Shoham & Leyton-Brown (2008).

Extensive-form Games and Adversarial Team Games The basic model for sequential interactions among a set $\mathcal{N}$ of N players with private information is the *Extensive-Form Game with imperfect information* (EFG). An EFG is a tuple $(\mathcal{N}, \mathcal{H}, \mathcal{Z}, \iota, \mathcal{A}, A, \mathcal{I}, \{u_p\}_{p\in\mathcal{N}})$ defining a tree where the set of nodes is denoted by $\mathcal{H}$ and the set of leaves (a.k.a. terminal nodes) is denoted by $\mathcal{Z} \subseteq \mathcal{H}$. The player acting at node $h \in \mathcal{H}$ is returned by function $\iota(h) \in \mathcal{N}$. Set $\mathcal{A} = \cup_{p\in\mathcal{N}}\mathcal{A}_p$ contains all the possible actions, where $\mathcal{A}_p$ is the set of actions available to player $p \in \mathcal{N}$. Given a node h, the set of available actions at h is $A(h)$. We also refer to a node h as a *history*, meaning the sequence of all the actions from the root to node h. Let $u_p : \mathcal{Z} \to \mathbb{R}$ be the *payoff function* of player p mapping every terminal node to a utility value. In order to account for imperfect information, we use *information sets* (for brevity, infosets). An infoset (also called private state) $I \subseteq \mathcal{H} \setminus \mathcal{Z}$ is a partition of the player p's nodes that are indistinguishable to p. We denote the set of player p's infosets as $\mathcal{I}_p$ and the set of all information sets as $\mathcal{I} = \cup_{p\in\mathcal{N}}\mathcal{I}_p$. With notation overload, we use $\iota(I)$ and $A(I)$ in place of $\iota(h)$ and $A(h)$ where $h \in I$ and we denote as $I(h)$ the infoset corresponding to node h, for any $h \in \mathcal{H}$.

We focus on Adversarial Team Games (ATGs). An ATG is an N-player EFG in which a *team* of players $\mathcal{T} \subseteq \mathcal{N}$ plays against an opponent o (or a team of players). If chance player c is present, we enrich the set of players with it. Thus, $\mathcal{N} = \mathcal{T} \cup \{o\} \cup \{c\}$. A team is a set of players sharing the same utility function. Formally, $\forall p \in \mathcal{T}$, $u_p = u_{\mathcal{T}}$ for some function $u_{\mathcal{T}}$. We restrict our analysis to zero-sum ATGs, in which $u_{\mathcal{T}} = -u_o$. Note that since chance c is a non-strategic player, its payoff is not defined. For an EFG, a *deterministic timing* is a labeling of the nodes in $\mathcal{H}$ with natural numbers such that the label of any node is strictly higher than the label of its parent. A deterministic timing is *exact* if all nodes in the same information set have the same label, and the game is called *timeable*. Furthermore, an EFG is *1-timeable*, when admitting an exact timing where the difference between the labels of the nodes and their parents is one. Furthermore, we focus on *perfect recall* games, in which no player forgets information. Exploiting the property of 1-timeability, we can define an ordering between different nodes. In particular, for two nodes $h, h' \in \mathcal{H}$ we say that h precedes h' (denoted as $h \preccurlyeq h'$) if the label assigned to h is smaller than the label assigned to h' and in the path from the root of the game tree to h', node h is encountered. With a slight abuse of notation, for two infosets $I, J \in \mathcal{I}$, we write that $I \preccurlyeq J$ if there exists $h \in I, h' \in J$ such that $h \preccurlyeq h'$. In addition, given an infoset I, the set of team members that will play in some infoset following I is denoted with $\mathcal{T}_I := \{p \in \mathcal{T} \mid \exists J \in \mathcal{I}_p \text{ s.t. } I \preccurlyeq J\}$.

Strategies and Nash Equilibrium Game theory provides various strategy representations in EFGs. A *behavioral strategy* $\sigma_p : \mathcal{I}_p \to \Delta^{|A(I)|}$ is a function that maps each infoset h to a probability distribution over available actions $A(h)$. A *normal-form plan* (or *pure strategy*) $\pi_p \in \Pi_p := \bigtimes_{I \in \mathcal{I}_p} A(I)$ is a tuple specifying one action for each infoset, while a *normal-form strategy* $\mu_p \in \Delta^{|\Pi_p|}$ is a probability distribution over normal-form plans. Kuhn (1953) show that behavioral and normal-form strategies are equivalent in perfect-recall games, while this does not hold with imperfect recallness where normal-form strategies are (usually) more expressive than behavioral. A *reduced normal-form strategy* $\mu_p^\star$ is obtained from a normal-form strategy μ_p by aggregating plans distinguished by action played in unreachable nodes. With a slight abuse of notation, $\forall p \in \mathcal{N}$, we denote with $\sigma_p[z]$ (respectively $\mu_p[z]$) the probability of reaching terminal node $z \in \mathcal{Z}$ when following strategy σ_p (resp. μ_p). A strategy profile is a tuple associating a strategy to each player in the game. We denote normal-form strategy profiles with $\boldsymbol{\mu}$ and behavioral strategy profiles with $\boldsymbol{\sigma}$. Given a strategy profile $\boldsymbol{\mu}$, we denote with μ_p the strategy of player $p \in \mathcal{N}$ and with $\boldsymbol{\mu}_{-p}$ the strategies of all the other players. With an abuse of notation, the expected utility for player p when she plays strategy μ_p and all the other players play strategy $\boldsymbol{\mu}_{-p}$ is $u_p(\mu_p, \boldsymbol{\mu}_{-p})$. Furthermore, we define the *best response* of player p to strategy profile $\boldsymbol{\mu}_{-p}$ as the strategy that maximizes player p's utility against strategy $\boldsymbol{\mu}_{-p}$. Formally, $\mathsf{BR}_p(\boldsymbol{\mu}_{-p}) := \arg\max_\mu u_p(\mu, \boldsymbol{\mu}_{-p})$. A strategy profile $\boldsymbol{\mu}$ is a Nash Equilibrium (NE) if it is stable with respect to unilateral deviations of a single player. Formally, $\boldsymbol{\mu}$ is a NE if and only if $\forall p \in \mathcal{N}$, $\mu_p \in \mathsf{BR}_p(\boldsymbol{\mu}_{-p})$.

***Ex ante* Coordination in ATGs** Basilico et al. (2017) show that the team's expected payoff in a Nash equilibrium can be arbitrarily smaller than the payoff in a *Team Maxmin Equilibrium*, introduced by Von Stengel & Koller (1997), which in its turn can be arbitrarily smaller than the payoff in a Team Maxmin Equilibrium with Correlation strategies. The TMEcor can be computed through a LP formulated over the joint normal-form plans of the team players:

$$\begin{aligned} \max_{\mu_{\mathcal{T}}} \min_{\mu_o} \sum_{z \in \mathcal{Z}} \quad & \mu_{\mathcal{T}}[z]\, \mu_o[z]\, u_{\mathcal{T}}(z) \\ \text{s.t.} \quad & \mu_{\mathcal{T}} \in \Delta(\bigtimes_{p \in \mathcal{T}} \Pi_p) \\ & \mu_o \in \Delta(\Pi_o). \end{aligned} \tag{1}$$

The team strategy space $\bigtimes_{p \in \mathcal{T}} \Pi_p$ can grow exponentially in the size of the game tree, thus making Problem (1) unaffordable in practice except for toy games.

3. Extensive-Form Games with Visibility Representation

We introduce the concept of *Extensive-Form Game with visibility*. This representation allows us to explicitly capture the information common to a set of players (*e.g.*, team members) and to extend the notion of infoset accordingly.

Public Function. We first introduce a function $Pub_p : \mathcal{A} \to \{\text{obs}, \text{unobs}\}$, $\forall p \in \mathcal{N}$, specifying whether action $a \in \mathcal{A}$ is *observable* or *unobservable*, respectively, by a single player p when a is played by another player. Note that our definition of Pub_p does not depend on the nodes in which player p plays, and therefore it cannot capture potential imperfect recallness in which p forgets actions observed before. Trivially, the information structure of every perfect-recall game is induced by some $\{Pub_p\}_{p \in \mathcal{N}}$:

Proposition 3.1. *Any pair of histories h, h' of player p belong to the same infoset when the actions a in h observable by p and the actions a' in h' observable by p are the same, formally, when $(a)_{a \in h : Pub_p(a) = \text{obs}} = (a')_{a' \in h' : Pub_p(a') = \text{obs}}$.*

With notation overload, the definition of function Pub can be extended to a set of players $\mathcal{P}$ (*e.g.*, a team) as $Pub_{\mathcal{P}}$:

$\mathcal{A} \to \{\text{pub}, \text{priv}, \text{hidden}\}, \forall \mathcal{P} \subseteq \mathcal{N}$:

$Pub_{\mathcal{P}}(a) = \text{pub} \iff \forall p \in \mathcal{P} : Pub_p(a) = \text{obs};$

$Pub_{\mathcal{P}}(a) = \text{hidden} \iff \forall p \in \mathcal{P} : Pub_p(a) = \text{unobs};$

$Pub_{\mathcal{P}}(a) = \text{priv}$ otherwise.

Informally, action a is called pub for a set of players $\mathcal{P}$, when it is observable by all the players of that set; hidden, when it is not observable by all these players (notice that in this case a is played by a player not belonging to $\mathcal{P}$); and priv when some player(s) in $\mathcal{P}$ can observe it, while some other player(s) in $\mathcal{P}$ cannot. Finally, we can extend the standard definition of Extensive-Form game:

Definition 3.2 (Extensive-Form Game with Visibility)**.** An Extensive-Form Game with Visibility (vEFG) is a tuple defined as $(\mathcal{N}, \mathcal{H}, \mathcal{Z}, \iota, \mathcal{A}, A, \mathcal{I}, \{u_p\}_{p\in\mathcal{N}}, \{Pub_p\}_{p\in\mathcal{N}})$ where $\mathcal{I}$ is induced by $\{Pub_p\}_{p\in\mathcal{N}}$ as discussed above, and therefore every player is with perfect recall.

3.1. Beyond Infoset: Public State

By means of $Pub_{\mathcal{P}}$, we can introduce the notion of **public state** for a set of players $\mathcal{P} \subseteq \mathcal{N}$, which extends the notion of infoset to a set of players.

Definition 3.3 (Public State)**.** A public state S is a subset of nodes $\mathcal{H}$ such that any pair of histories h, h' of potentially different players in $\mathcal{P}$ belong to S when the actions a in h that are public for $\mathcal{P}$ and the actions a' in h' that are public for $\mathcal{P}$ are the same, formally, $(a)_{a\in h:Pub_{\mathcal{P}}(a)=\text{pub}} = (a')_{a'\in h':Pub_{\mathcal{P}}(a')=\text{pub}}$.

In other words, two histories belong to the same public state if they share the same public actions and differ only for their private actions. We call $\mathcal{S}$ the set of all public states. It can be easily seen that, if the node h of an infoset I belongs to a public state S, then S also contains all the other nodes of I, and the notion of public state reduces to the notion of infoset when $\mathcal{P}$ is composed of a single player. In principle, a public state can contain multiple infosets, that can be of the same player and/or of different players in $\mathcal{P}$. In the case in which $\mathcal{P}$ is a team of players, we call a public state for $\mathcal{P}$ as a *team-public infoset*. With an abuse of notation, for any set of players $\mathcal{P} \subseteq N$, we denote as $\mathcal{S}_{\mathcal{P}}(h)$ the set of all infosets belonging to players in $\mathcal{P}$ that are in the same public state as node h.

3.2. Public-turn-taking Games

We focus on a class of games, called **public-turn-taking**, in which every player knows, at every infoset she plays, the sequence of players acted from the root to that infoset. This property refines 1-timeability as it requires that, in addition to the length of the history, even the sequence of players is common knowledge. In public-turn-taking games, the public states have a specific structure that is central in our results, allowing the translation of an ATG as a 2p0s game. More precisely, every public state is composed of nodes of a single player whose histories have the same length.

Definition 3.4 (Public turn-taking property)**.** A vEFG is public turn-taking if:

$$\forall I \in \mathcal{I}, \forall h, h' \in I : (\iota(g))_{g\sqsubset h} = (\iota(g'))_{g'\sqsubset h'}.$$

Interestingly, we can show that, given an extensive-form game satisfying perfect recallness and timeability, we can generate a strategically equivalent game satisfying public-turn-taking property, whose size is polynomially upper bounded in the size of the original game (proofs omitted in the main paper are in Appendix A).

Theorem 3.5 (Transformation into a public-turn-taking game)**.** *Given any timeable vEFG with players $\mathcal{N}$ and nodes $\mathcal{H}$, there is a strategically equivalent (admitting the same reduced normal form) public-turn-taking vEFG whose size is $O(\,|\mathcal{N}|\;|\mathcal{H}|^2\,)$.*

3.3. Completely inflated games

Another important class of team games for the *ex ante coordination* scenario is called **completely inflated games**. In this class of games, every team member knows the exact action played by another team member at any information set. This property allows us to explicitly represent that teammates share their strategies before starting the game.

Definition 3.6 (Completely inflated vEFG (Kaneko & Kline, 1995))**.** A vEFG $\mathcal{G}$ is completely inflated with respect to a team of players $\mathcal{T}$ if:

$$Pub_{\mathcal{T}}(a) = \text{pub}\ \forall a \in \mathcal{A}_p \forall p \in \mathcal{T} \tag{2}$$

In the following, we focus on completely inflated vEFGs for the team $\mathcal{T}$. This can be ensured for a generic vEFG by modifying the function $Pub_t(a)$ in such a way that $Pub_t(a) = \text{obs}\ \forall t \in \mathcal{T}\ \forall a \in \mathcal{A}_{t'} \forall t' \in \mathcal{T}$.

4. Team-Public-Information Conversion Algorithm

4.1. Conversion Procedure

We present the algorithmic procedure to convert an ATG into a 2p0s game, denoted as *Team-Public-Information* (TPI) game, in which a coordinator player takes the strategic decision on behalf of the team. The pseudo-code is provided in Algorithm 1.

Definition 4.1 (Team-Public-Information game)**.** Given a completely inflated vEFG $\mathcal{G}$ that satisfies the public turn-taking property, the corresponding TPI game $\mathcal{G}'$[1] is defined as the output of the function ConvertGame($\mathcal{G}$) described in Algorithm 1.

The algorithm recursively traverses the extensive-form game tree in a depth-first post-order fashion: for each tra-

[1] Superscript $'$ denotes the elements of the converted game.

Algorithm 1 Team-Public-Information Conversion

function CONVERTGAME($\mathcal{G}$)
  ▷ $\mathcal{G} = (\mathcal{N}, \mathcal{H}, \mathcal{Z}, \iota, \mathcal{A}, A, \mathcal{I}, \{u_p\}_{p\in\mathcal{N}}, \{Pub_p\}_{p\in\mathcal{N}})$
  initialize $\mathcal{G}'$ new game
  ▷ $\mathcal{G}' = (\mathcal{N}', \mathcal{H}', \mathcal{Z}', \iota', \mathcal{A}', A', \mathcal{I}', \{u'_p\}_{p\in\mathcal{N}'}, \{Pub'_p\}_{p\in\mathcal{N}'})$
  $\mathcal{N}' \leftarrow \{t, o, c\}$
  $h'_\varnothing \leftarrow$ PUBTEAMCONV($h_\varnothing, \mathcal{G}, \mathcal{G}'$) ▷ new game root
  return $\mathcal{G}'$

function PUBTEAMCONV($h, \mathcal{G}, \mathcal{G}'$)
  initialize $h' \in \mathcal{H}'$
  if $h \in \mathcal{Z}$ **then** ▷ terminal node
    $\mathcal{Z}' \leftarrow \mathcal{Z}' \cup \{h'\}$
    $u'_p(h') \leftarrow u_p(h) \quad \forall p \in \mathcal{N}$
    $u'_t(h') \leftarrow \sum_{p\in\mathcal{T}} u_p(h)$
    $u'_o(h') \leftarrow -u'_t(h')$
  else if $\iota(h) \in \{o, c\}$ **then** ▷ opponent or chance
    $\iota'(h') \leftarrow \iota(h)$
    $A'(h') \leftarrow A(h)$
    if $\iota(h) = c$ **then**
      $\sigma'_c(h') = \sigma_c(h)$
    for $a' \in A'(h')$ **do**
      $Pub'_t(a') \leftarrow \text{obs}$ **if** $Pub_\mathcal{T}(a') = \text{pub}$ **else** unobs
      $Pub'_o(a') \leftarrow Pub_o(a')$
      $h'a' \leftarrow$ PUBTEAMCONV($ha', \mathcal{G}, \mathcal{G}'$)
  else ▷ team member
    $\iota'(h') = t$
    $I \leftarrow I(h)$
    $A'(h') \leftarrow \bigtimes_{J\in\mathcal{S}_{\mathcal{T}_I}(h)} A(J)$ ▷ prescriptions
    for $\Gamma' \in A'(h')$ **do**
      $Pub'_t(\Gamma') \leftarrow \text{seen}, Pub'_o(\Gamma') \leftarrow \text{unseen}$
      $a' \leftarrow \Gamma'[I(h)]$ ▷ extract chosen action
      initialize $h'' \in \mathcal{H}'$
      $A'(h'') \leftarrow \{a'\}$
      $\iota(h'') = c$
      $Pub'_t(a') \leftarrow \text{seen}$
      $Pub'_o(a') = Pub_o(a')$
      $\sigma'_c(h'') = $ play a' with probability 1
      $h''a' \leftarrow$ PUBTEAMCONV($ha', \mathcal{G}, \mathcal{G}'$)
      $h'\Gamma \leftarrow h''$
  return h'

versed node, some corresponding nodes are instantiated in the converted game as follows. The chance, terminal, and adversary nodes are copied unaltered as the coordinator player t has only access to the public information observable to the team members. Each team member node of the extensive form is instead mapped to a new coordinator node, in which she plays a prescription Γ among all the combinations of possible actions for each information state I belonging to the public team state. In other words, given a public state S, the coordinator issues to the players different recommendations for every possible information set belonging to S. For example, in Fig. 1(a), players 1 and 2 are team members, and the decision nodes compose a unique public state, therefore there is a single information set for the coordinator player in the converted game depicted in Fig. 1(b). In particular, the actions available to the coordinator are prescriptions specifying an action per information set of the extensive form (equivalently, an action per private state). See Appendix B on how private information affects the construction of our conversion, and Appendix D for a richer conversion example.

4.2. Strategic Equivalence

The central result of the present paper is the proof that the transformed Team Public Information game is strategically equivalent to the extensive form. In particular, we show the equivalence between a Nash Equilibrium in the converted game and the TMEcor in the extensive form. Before proving such a result, we introduce the following instrumental lemmas. We also remark that, while we make use of reduced normal-form plans, for simplicity, we refer to them as plans and pure strategies, dropping the superscript "$\star$".

Lemma 4.2. *Given a public-turn-taking vEFG $\mathcal{G}$, and the corresponding TPI game $\mathcal{G}' =$ CONVERTGAME($\mathcal{G}$), each joint pure strategy $\pi_\mathcal{T}$ in $\mathcal{G}$ can be mapped to a strategy π_t in $\mathcal{G}'$, such that the traversed histories have been mapped by* PUBTEAMCONV. *Formally, $\forall\pi_\mathcal{T}$, there is a π_t such that $\forall\pi_o, \pi_c$ the following holds:*

$$(\text{PUBTEAMCONV}(h))_{h \text{ reached by playing } (\pi_\mathcal{T},\pi_o,\pi_c) \text{ in } \mathcal{G}} \equiv (h')_{h' \text{ reached by playing } (\pi_t,\pi_o,\pi_c) \text{ in } \mathcal{G}'}.$$

Lemma 4.3. *Given a public-turn-taking vEFG $\mathcal{G}$, and the corresponding TPI game $\mathcal{G}' =$ CONVERTGAME($\mathcal{G}$), each coordinator pure strategy π_t in $\mathcal{G}'$ can be mapped to a strategy $\pi_\mathcal{T}$ in $\mathcal{G}$, such that the traversed histories have been mapped by* PUBTEAMCONV. *Formally, $\forall\pi_t$, there is $\pi_\mathcal{T}$ such that $\forall\pi_o, \pi_c$ the following holds:*

$$(\text{PUBTEAMCONV}(h))_{h \text{ reached by playing } (\pi_\mathcal{T},\pi_o,\pi_c) \text{ in } \mathcal{G}} \equiv (h')_{h' \text{ reached by playing } (\pi_t,\pi_o,\pi_c) \text{ in } \mathcal{G}'}.$$

We can now define the following functions to map strategies from the extensive form game to the converted game.

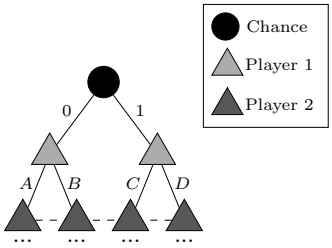
Definition 4.4 (Mapping functions)**.** We define:

- $\rho : \Pi_\mathcal{T} \to \Pi_t$ maps each $\pi_\mathcal{T}$ to the π_t specified by the procedure described in the proof of Lemma 4.2;
- $\sigma : \Pi_t \to \Pi_\mathcal{T}$ maps each π_t to the $\pi_\mathcal{T}$ specified by the procedure described in the proof of Lemma 4.3.

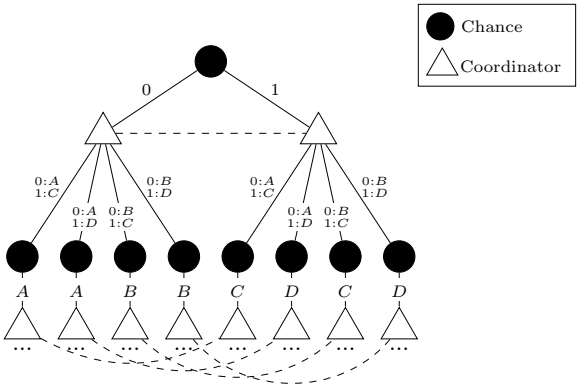
Those two functions can also be extended to mixed strategies, by converting each pure plan and summing the probability masses of the converted plans. Formally, we have:

$$\forall\mu_\mathcal{T} \in \Delta^{\Pi_\mathcal{T}} : \rho(\mu_\mathcal{T})[\pi_t] = \sum_{\pi_\mathcal{T}:\rho(\pi_\mathcal{T})=\pi_t} \mu_\mathcal{T}(\pi_\mathcal{T}),$$

$$\forall\mu_t \in \Delta^{\Pi_t} : \sigma(\mu_t)[\pi_\mathcal{T}] = \sum_{\pi_t:\sigma(\pi_t)=\pi_\mathcal{T}} \mu_t(\pi_t).$$



(a) Example of extensive form of an adversarial team game (only the first three levels are depicted; the symbol "…" denotes that the game continues below). Player 1 and Player 2 are in the same team and have different visibility over the chance actions.



(b) Team-public-information representation of the extensive-form game depicted in (a).

Figure 1: Example of game conversion: from extensive form to team-public-information representation.

We can now state the payoff-equivalence between a game $\mathcal{G}$ and the corresponding TPI game $\mathcal{G}'$ as follows.

Theorem 4.5. *A public-turn-taking vEFG $\mathcal{G}$ and its TPI game $\mathcal{G}' = \text{CONVERTGAME}(\mathcal{G})$ are payoff-equivalent, i.e.*

$$\forall \pi_{\mathcal{T}}\ \forall \pi_o, \pi_c : u_{\mathcal{T}}(\pi_{\mathcal{T}}, \pi_o, \pi_c) = u_t(\rho(\pi_{\mathcal{T}}), \pi_o, \pi_c),$$
$$\forall \pi_t\ \forall \pi_o, \pi_c : u_{\mathcal{T}}(\sigma(\pi_t), \pi_o, \pi_c) = u_t(\pi_t, \pi_o, \pi_c).$$

The correspondence between the strategies of the two representations is used to derive the main result of this work that shows the equivalence between a NE of the converted 2p0s game and a TMEcor of the original ATG.

Theorem 4.6. *Given a public-turn-taking vEFG $\mathcal{G}$, and the corresponding TPI $\mathcal{G}' = \text{CONVERTGAME}(\mathcal{G})$, a Nash Equilibrium μ_t^* in $\mathcal{G}'$ is realization equivalent to a TMEcor $\mu_{\mathcal{T}}^* = \sigma(\mu_t^*)$ in $\mathcal{G}$.*

4.3. Games with Compact TPI

The procedure to convert an extensive-form game into the equivalent TPI game exploits the information structure of the team to prescribe to the team players an action for every possible private state. In general, this makes the size of the TPI to grow exponentially with the number of possible private states belonging to a public state. However, we can find a class of games in which their information structure allows the generation of a TPI game with a size upper bounded by a polynomial in the size of the extensive form:

Definition 4.7 (Games with common external information)**.** A vEFG $\mathcal{G}$ has *common external information* for a set of players $\mathcal{T} \subseteq \mathcal{N}$ if all the actions performed by the other players (chance included) have the same visibility for all players in $\mathcal{T}$, formally, $\forall p \in \mathcal{N} \setminus \mathcal{T},\ \forall a \in \mathcal{A}_p$:

$$Pub_{\mathcal{T}}(a) \neq \text{priv}.$$

Theorem 4.8. *Given a public-turn-taking vEFG $\mathcal{G}$ with common external information for the team $\mathcal{T}$, the tree of corresponding TPI game $\mathcal{G}'$ has a number of nodes linear in the nodes of $\mathcal{G}$.*

Intuitively, Theorem 4.8 states that if the game has common external information for the team, then it is possible to find the TMEcor in polynomial time. This result matches what was previously known in literature. When common external information is satisfied, one can, indeed, resort to Kaneko & Kline (1995) to find a polynomial-time algorithm to find an equilibrium. This is the case, *e.g.*, of Goofspiel game (Ross, 1971) which admits a compact TPI.

4.4. TPI Expressivity and Abstractions

Abstractions demonstrated to be a successful tool to tackle real-world 2p0s game (Sandholm, 2015). Generally, these are obtained by merging different infosets of the same player (*state* abstractions) and/or different actions of the same player at the same infoset (*action* abstractions). However, despite their importance, the use of abstractions in ATGs has remained unexplored so far. By defining the team's strategies as behavioral, the Team-Public-Information representation provides a suitable and direct tool for designing abstractions for ATGs, while we can show that the extensive-form is not sufficiently expressive.

Proposition 4.9. *Any action or state abstraction that, once applied to an extensive-form game $\mathcal{G}$, returns a perfect-recall timeable game can be mapped specularly in the team-public-information representation $\mathcal{G}' = \text{CONVERTGAME}(\mathcal{G})$. The reverse is not true.*

It can be observed that the properties required by the above proposition are satisfied by most of the abstractions, *e.g.*, by Gilpin & Sandholm (2007) and Gilpin et al. (2007).

4.5. TPI and Subgame Solving

Subgame solving (Moravcík et al., 2016; Brown & Sandholm, 2017a) demonstrated to be a central technique to face huge imperfect-information 2p0s games, such as, *e.g.*, poker games (Moravcík et al., 2017; Brown & Sandholm, 2017b). More precisely, subgame solving takes as input a strategy (usually called *blueprint*) computed with a coarse abstraction of the game and refines it in the neighborhoods of the currently reached information set while playing (intuitively, subgame solving algorithms perform a sequence of local reoptimizations). The basic idea is to extract a portion (called *subgame*) of the original game and generate on-the-fly an auxiliary game to solve just in time. The solving algorithm is initialized with the blueprint mapped to the auxiliary game and then it refines such a strategy. In particular, in every information set a player moves, the strategy refinement algorithm is performed. A notable example of subgame solving technique is *depth-limited subgame solving* (Brown et al., 2018). In this algorithm, the auxiliary game is built starting from the subgame rooted at the public state corresponding to the infoset in which the player is playing. The subgame is truncated at a given depth, after which the players are assumed to play according to the blueprint. As widely shown in real-world applications (Brown & Sandholm, 2017b), depth-limited subgame solving can dramatically reduce the players' exploitability.

Since our TPI conversion generates a 2p0s game preserving the public structure of the original game, subgame solving techniques, including, *e.g.*, depth-limited solving, can be applied directly. The only caveat concerns the size of the auxiliary game, which is exponentially larger than the size of the subgame in the extensive form. Developing efficient subgame-solving techniques for the TPI game is an interesting line of research and is left as future work.

5. Experimental Evaluation

5.1. Experimental Setting

Game Instances. We conduct our experimental activity with a subset of instances customarily adopted as testbed for adversarial team games, *e.g.*, by Zhang & Sandholm (2022). More precisely, we use multi-player parametric versions of Kuhn (Kuhn, 1950) and Leduc (Southey et al., 2005) poker where one player is the adversary and the remaining players collude against him. We use the following values for the parameters. In *Kuhn* poker, team members are from 2 to 3, ranks are from 3 to 6. In *Leduc* poker, team members are from 2 to 3, the maximum number of bets allowed in each betting round is from 1 to 5, ranks are from 2 to 5, suits are 3. Details are provided in Appendix E.

Representations. By exploiting the interpretability of our representation, we design pruning and/or abstraction techniques reducing the tree size. In our experiments, we focus on the following reduced representations (more details on the conversions are in Appendix C, while Appendix D provides a conversion example per representation).

Basic: it is the game returned by Algorithm 1.

Pruned: The play of a public action by a team member allows to prune, in the following part of the tree, the private states with a different recommendation. Thus, we safely discard a subset of the private states reducing the number of possible prescriptions in subsequent nodes. The pseudocode is in Algorithm 2 in Appendix C.

Folded: while pruned representation allows to safely reduce the number of possible private states, it does not address the large number of nodes in the converted game. This is due to the fact that Algorithm 1 preserves the chance sampling as in the original game. However, we can avoid to sample a private state and instead keep a belief over the private states of the team members.

Imperfect-recall abstraction of folded: the folded representation may include multiple replicas of the same subgames reachable from different histories. We connect the corresponding infosets in the subgames over all the replicas, thus leading to an imperfect-recall game that is *well-formed* in the sense of Lanctot et al. (2012b).

Lossy imperfect-recall abstraction of folded: we discard all coordinator's prescriptions recommending the same action (Fold or Raise or Call) to every private state. The resulting game keeps to be well-formed.

Algorithms. We test our representations with state-of-the-art no-regret algorithms for 2p0s games as *Counter Factual Regret plus* (CFR+) (Tammelin, 2014) and *Outcome Sampling Monte Carlo Counter Factual Regret* (OS-MCCFR) (Lanctot et al., 2009). We recall that, as showed by Lanctot et al. (2012b), CFR-based algorithms converge to the equilibrium even with imperfect-recall games satisfying well-formed properties as for the case of our representations. To abstract from the specific implementation details, we use OpenSpiel (Lanctot et al., 2019a).

5.2. Experimental Results

Representation Size and Game Value. In Tab. 1, we report the size of the game instances obtained by our conversions, and we compare them with the size of the representation used by Zhang & Sandholm (2022). Although it is not based on a tree, there is a strict connection between their representation and ours. In particular, their *locally feasible sets* are strictly related to our actions, as they are two different approaches to describe the Cartesian product of the team members' actions given their possible private states. Both locally feasible sets and actions determine the size of the two representations and are helpful to analyze how their

Table 1: Size of the game trees returned by the different favours of our conversion (basic, pruned, folded, imperfect-recall abstraction of folded, and lossy imperfect-recall abstraction of folded) and by the tree decomposition by Zhang & Sandholm (2022); size of the reduced normal form. We use the following notation: mnKr is Kuhn poker with a team of m players facing a team of n player and r ranks; mnLbrc is Leduc poker with a team of m players facing a team of n players, a maximum number b of bets allowed in each betting round, a number of ranks r, and a number of indistinguishable suits c). Game values are provided both for the exact case (in white) and when using our abstraction (in red). The empty cells are due to instances with more than $2 \cdot 10^9$ nodes or out-of-memory.

				game instances											
				21K3	21K4	21K5	21K6	21K8	31K5	21L133	21L143	21L153	21L223	21L523	31L133
normal form	plans	team		$\sim 10^6$	$\sim 10^8$	$\sim 10^9$	$\sim 10^{11}$	$\sim 10^{15}$	$\sim 10^{20}$	$\sim 10^{70}$	$\sim 10^{126}$	$\sim 10^{197}$	$\sim 10^{252}$	$\sim 10^{3200}$	$\sim 10^{283}$
	plans	adversary		$\sim 10^3$	$\sim 10^4$	$\sim 10^6$	$\sim 10^7$	$\sim 10^9$	$\sim 10^{10}$	$\sim 10^{54}$	$\sim 10^{96}$	$\sim 10^{150}$	$\sim 10^{134}$	$\sim 10^{3900}$	$\sim 10^{152}$
basic	nodes			7336	200,681	3,714,326				35,140,264			6,140,623		
	infosets	team		888	10,661	117,938				1,625,647			427,984		
	infosets	adversary		12	16	20				228			630		
	actions	team		2,101	24,641	265,517				4,135,497			1,287,852		
	actions	adversary		25	33	41				457			1,443		
pruned	nodes			4,360	95,225	324,766	15,007,117			35,140,264			724,009		
	infosets	team		495	4,505	35,943	267,229			101,389			45,440		
	infosets	adversary		12	16	20	24			228			630		
	actions	team		1,087	9,849	77,947	574,709			339,243			127,352		
	actions	adversary		25	33	41	49			457			1,443		
folded	nodes			4,108	66,349	740,406	7,002,763	488,157,721	202,660,366	1,691,158	61,983,093	1,973,610,366	538,111	222,239,487	277,714,570
	infosets	team		495	4,505	35,943	267,229	13,194,833	11,783,620	96,115	2,625,209	67,400,747	44,252	18,308,851	17,403,080
	infosets	adversary		12	16	20	24	32	40	228	400	620	630	49,584	816
	actions	team		1,086	9,849	77,947	574,709	27,978,929	25,689,691	208,136	5,736,593	147,671,105	106,963	45,969,475	37,743,473
	actions	adversary		24	32	41	49	65	81	457	801	1,241	1,443	123,153	1,633
imperfect-recall abstraction of folded	nodes			4,108	66,349	740,406	7,002,763	488,157,721	202,660,366	1,691,158	61,983,093	1,973,610,366	538,111	222,239,487	277,714,570
	infosets	team		81	321	1,213	4,585	68,321	108,480	23,071	4,600	105,742	4,522	361,969	184,394
	infosets	adversary		12	16	20	24	32	40	228	400	620	630	49,584	816
	actions	team		253	1,433	8,237	48,341	1,710,449	886,591	13,659	97,577	682,095	13,646	1,261,733	568,211
	actions	adversary		25	32	41	49	65	81	800	457	1,241	1,443	123,153	1,633
tree decomposition	sequences	team		91	177	not available	433	801	2,611	2,725	6,377	12,361	5,765	492,605	42,361
	sequences	adversary		25	33	not available	49	65	81	457	801	1,241	1,433	123,143	1,633
	loc. feas. sets	team		351	1,749	not available	52,669	1,777,061	974,470	17,718	115,281	757,884	21,729	2,042,641	703,390
	loc. feas. sets	adversary		25	33	not available	49	65	81	703	1,225	1,891	3,123	305,835	2,479
exact	**game value**			**0.000**	**-0.0416**	**-0.0251**	**-0.0236**		**-0.0392**	**0.2148**	**0.1072**	**0.0240**	**0.5155**	**0.9520**	**0.1894**
lossy imperfect-recall abstraction of folded	**game value**			**-0.166**	**-0.0450**	**-0.0271**	**-0.0262**		**-0.0392**	**0.0888**	**0.0623**	**0.0004**	**0.3642**	**0.5858**	**0.1894**
	nodes			1,480	36,429	512,766	5,574,547	445,611,353	92,309,616	184,729	7,502,765	298,052,671	36,269	3,073,197	7,203,775
	infosets	team		64	287	1,146	4,453	67,803	91,021	1,930	11,981	70,636	2.513	198,329	37,435
	infosets	adversary		12	16	20	24	32	40	228	400	620	630	49,584	816
	actions	team		145	899	5,721	37,231	1,517,163	518,591	3,913	30,263	281,981	5,759	492,599	75,499
	actions	adversary		25	33	41	49	65	81	457	801	1,241	1,443	123,153	1,633

sizes grow as the size of the extensive form increases.

Interestingly, our basic representation is exponentially smaller than the reduced normal form. Furthermore, our information-lossless general-purpose techniques allow a dramatic reduction of the size of the tree up to 3 orders of magnitude. Furthermore, by using the imperfect-recall abstraction of the folded representation, we obtain a number of actions smaller than the number of locally feasible sets, suggesting that our representation is more efficient than that by Zhang & Sandholm (2022), while guaranteeing explainability and the possibility of designing abstractions. In particular, in some instances (*e.g.*, 21L523), the number of actions in our representation is almost the half than the locally feasible sets. We also observe that our lossy imperfect-recall abstraction of the folded representation dramatically reduces the game size suffering from a small loss in terms of game value, averagely, 0.086.

Exploitability vs. Iterations/Running Time. We show in Fig. 2 the dependency of the exploitability with CFR+ and OS-MC-CFR on the iterations and time for instance 21L133. Considering the number of iterations, except for a negligible term, the exploitability with CFR+ is the same for all the information-lossless representations, while the convergence of the lossy abstraction is slightly faster. However, considering the execution time, we can fully appreciate the importance of developing techniques to reduce the representation size. Indeed, CFR+ applied to our lossy abstraction is more than one order of magnitude faster than the other representations, and even three order of magnitude faster than the basic one. This is due to the need for performing full traversals of the tree at every iteration. At the same time, a trajectory sampling algorithm like OS-MC-CFR benefits when reducing the number of infosets, as the variance of the estimates on the regret reduces. Remarkably, the adoption of abstractions unlocks a significant scale-up the algorithms in practice.

Finally, we remark that we cannot directly compare the running time of our algorithms with that by Zhang & Sand-

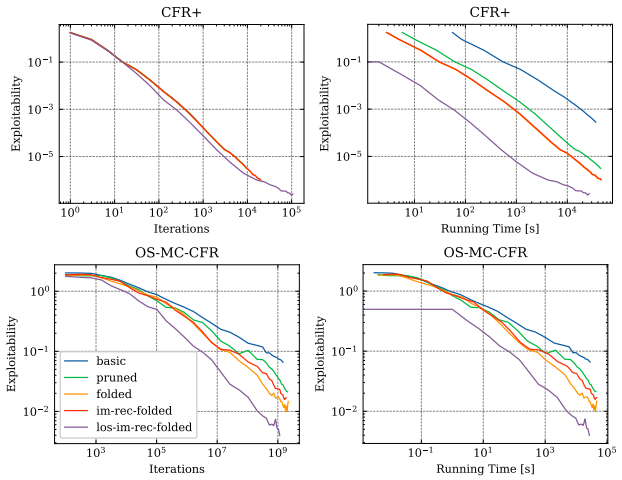


Figure 2: Exploitability of CFR+ and OS-MC-CFR with 21L133 game in the number of iterations and time (seconds).

holm (2022) due to the use of different technologies and implementation details. Notably, our approach and that by Zhang & Sandholm (2022) take in input representations whose size increases with the same dependency in the size of the extensive form, suggesting that, abstracting from implementation details, the relative perfomances of these two approaches are similar to those of no-regret learning and LP with 2p0s games, see, *e.g.*, Zhang & Sandholm (2020). We point the reader to Appendix E.3 for a detailed discussion.

6. Conclusions and Future Work

We bridge the realm of sequential 2-player zero-sum games with that of adversarial team games. In particular, we show that any sequential adversarial team game satisfying mild assumptions can be converted into a suitable sequential 2-player zero-sum game such that a Nash Equilibrium in the converted game is strategically equivalent to a TMEcor in the original game. This equivalence enables the adoption of successful tools for solving huge 2-player zero-sum games to adversarial team games. Furthermore, thanks to the high explainability of our representation, pruning and abstraction techniques can be easily designed to dramatically reduce the size of the tree. In particular, we empirically show that we can produce a game representation smaller than that provided by the current state of the art without any loss of information while guaranteeing explainability. Furthermore, we provide, to the best of our knowledge, the first example of abstractions for adversarial team games, showing that it allows a remarkable reduction of the tree size suffering from a small loss, and the first attempt to use no-regret learning with this class of games. Open challenges include the design of *ad hoc* algorithms for abstractions, no-regret learning, and subgame solving (whose potential impact needs to be evaluated) capable of exploiting the structure of these games to scale up to huge instances.

References

Basilico, N., Celli, A., De Nittis, G., and Gatti, N. Team-maxmin equilibrium: efficiency bounds and algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Brown, G. Iterative solution of games by fictitious play. *Activity Analysis of Production and Allocation*, 13, 01 1951.

Brown, N. and Sandholm, T. Safe and nested subgame solving for imperfect-information games. In *NIPS*, 2017a.

Brown, N. and Sandholm, T. Libratus: The superhuman AI for no-limit poker. In Sierra, C. (ed.), *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pp. 5226–5228. ijcai.org, 2017b. doi: 10.24963/ijcai.2017/772. URL https://doi.org/10.24963/ijcai.2017/772.

Brown, N. and Sandholm, T. Safe and nested subgame solving for imperfect-information games. *Advances in neural information processing systems*, 30, 2017c.

Brown, N. and Sandholm, T. Superhuman ai for multi-player poker. *Science*, 365(6456):885–890, 2019. ISSN 0036-8075. doi: 10.1126/science.aay2400. URL https://science.sciencemag.org/content/365/6456/885.

Brown, N., Sandholm, T., and Amos, B. Depth-limited solving for imperfect-information games. In *NeurIPS*, 2018.

Cacciamani, F., Celli, A., Ciccone, M., and Gatti, N. Multi-agent coordination in adversarial environments through signal mediated strategies. In *AAMAS*, 2021.

Celli, A. and Gatti, N. Computational results for extensive-form adversarial team games. In *AAAI*, 2018.

Celli, A., Ciccone, M., Bongo, R., and Gatti, N. Coordination in adversarial sequential team games via multi-agent deep reinforcement learning. *ArXiv*, abs/1912.07712, 2019.

Celli, A., Marchesi, A., Farina, G., and Gatti, N. No-regret learning dynamics for extensive-form correlated equilibrium. *Advances in Neural Information Processing Systems*, 33:7722–7732, 2020.

Farina, G., Celli, A., Gatti, N., and Sandholm, T. Ex ante coordination and collusion in zero-sum multi-player extensive-form games. In *NeurIPS*, 2018.

Farina, G., Celli, A., Gatti, N., and Sandholm, T. Connecting optimal ex-ante collusion in teams to extensive-form correlation: Faster algorithms and positive complexity results. In *ICML*, 2021.

Foerster, J., Song, F., Hughes, E., Burch, N., Dunning, I., Whiteson, S., Botvinick, M., and Bowling, M. Bayesian action decoder for deep multi-agent reinforcement learning. pp. 1942–1951, 2019.

Gilpin, A. and Sandholm, T. Lossless abstraction of imperfect information games. *J. ACM*, 54(5):25–es, 2007.

Gilpin, A., Sandholm, T., and Sørensen, T. B. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of texas hold'em poker. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pp. 50–57. AAAI Press, 2007.

Kaneko, M. and Kline, J. Behavior strategies, mixed strategies and perfect recall. *International Journal of Game Theory*, 24:127–145, 1995.

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

Kuhn, H. W. *Extensive games and the problem of information*. Princeton University Press, Princeton, NJ, 1953.

Lanctot, M., Waugh, K., Zinkevich, M. A., and Bowling, M. Monte carlo sampling for regret minimization in extensive games. In *NIPS*, 2009.

Lanctot, M., Gibson, R., Burch, N., Zinkevich, M., and Bowling, M. No-regret learning in extensive-form games with imperfect recall. *arXiv:1205.0622 [cs]*, May 2012a. URL http://arxiv.org/abs/1205.0622. arXiv: 1205.0622.

Lanctot, M., Gibson, R. G., Burch, N., and Bowling, M. No-regret learning in extensive-form games with imperfect recall. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012b.

Lanctot, M., Lockhart, E., Lespiau, J., Zambaldi, V. F., Upadhyay, S., Pérolat, J., Srinivasan, S., Timbers, F., Tuyls, K., Omidshafiei, S., Hennes, D., Morrill, D., Muller, P., Ewalds, T., Faulkner, R., Kramár, J., Vylder, B. D., Saeta, B., Bradbury, J., Ding, D., Borgeaud, S., Lai, M., Schrittwieser, J., Anthony, T. W., Hughes, E., Danihelka, I., and Ryan-Davis, J. Openspiel: A framework for reinforcement learning in games. *CoRR*, abs/1908.09453, 2019a. URL http://arxiv.org/abs/1908.09453.

Lanctot, M., Lockhart, E., Lespiau, J.-B., Zambaldi, V., Upadhyay, S., Pérolat, J., Srinivasan, S., Timbers, F., Tuyls, K., Omidshafiei, S., et al. Openspiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453*, 2019b.

Moravcík, M., Schmid, M., Ha, K., Hladík, M., and Gaukrodger, S. Refining subgames in large imperfect information games. In *AAAI*, 2016.

Moravcík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M. B., and Bowling, M. H. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356: 508 – 513, 2017.

Nayyar, A., Mahajan, A., and Teneketzis, D. Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Transactions on Automatic Control*, 58:1644–1658, 2013.

Ross, S. M. Goofspiel—the game of pure strategy. *Journal of Applied Probability*, 8(3):621–625, 1971.

Sandholm, T. Abstraction for solving large incomplete-information games. In *AAAI*, 2015.

Shoham, Y. and Leyton-Brown, K. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, USA, 2008. ISBN 0521899435.

Sokota, S., Lockhart, E., Timbers, F., Davoodi, E., D'Orazio, R., Burch, N., Schmid, M., Bowling, M. H., and Lanctot, M. Solving common-payoff games with approximate policy iteration. In *AAAI*, 2021.

Southey, F., Bowling, M., Larson, B., Piccione, C., Burch, N., Billings, D., and Rayner, C. Bayes' bluff: opponent modelling in poker. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pp. 550–558, 2005.

Tammelin, O. Solving large imperfect information games using cfr+. *ArXiv*, abs/1407.5042, 2014.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J., Jaderberg, M., Vezhnevets, A., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pp. 1–5, 2019.

Von Stengel, B. and Koller, D. Team-maxmin equilibria. *Games and Economic Behavior*, 21(1):309 – 321, 1997.

Zhang, B. H. and Sandholm, T. Sparsified linear programming for zero-sum equilibrium finding. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pp. 11256–11267, 2020.

Zhang, B. H. and Sandholm, T. Team correlated equilibria in zero-sum extensive-form games via tree decompositions. 2022.

Zhang, Y. and An, B. Computing team-maxmin equilibria in zero-sum multiplayer extensive-form games. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 2318–2325. AAAI Press, 2020a.

Zhang, Y. and An, B. Converging to team-maxmin equilibria in zero-sum multiplayer games. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11033–11043. PMLR, 2020b.

Zhang, Y., An, B., and Cerný, J. Computing ex ante coordinated team-maxmin equilibria in zero-sum multiplayer extensive-form games. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 5813–5821. AAAI Press, 2021.

Zinkevich, M. A., Johanson, M. B., Bowling, M., and Piccione, C. Regret minimization in games with incomplete information. In *NIPS*, 2007.

A. Proofs Omitted from the Main Paper

Theorem 3.5 (Transformation into a public-turn-taking game)**.** *Given any timeable vEFG with players $\mathcal{N}$ and nodes $\mathcal{H}$, there is a strategically equivalent (admitting the same reduced normal form) public-turn-taking vEFG whose size is $O(\,|\mathcal{N}|\;|\mathcal{H}|^2\,)$.*

Proof. We provide the following procedure which returns in output a public-turn-taking game. This is achieved by assigning each level of the converted game to a player, alternating between them (chance included). Then, we add all the histories of the original game one by one, while forcing that at each level only the player corresponding to that level can play. If the history has no action assigned to the level's player, then we can add a dummy player node, with only a single action, and try to prosecute with the actions of the original history in the next node. The visibility of the added action is "unseen" for all players except the one playing it.

This procedure guarantees to get a strategically equivalent game by adding at most $\mathcal{O}((|\mathcal{N}|+1)|\mathcal{H}|)$ for any of the $|\mathcal{H}|$ histories in the original game. This proves that the number of histories in the converted game is $\mathcal{O}((|\mathcal{N}|+1)|\mathcal{H}|^2)$. □

Lemma 4.2. *Given a public-turn-taking vEFG $\mathcal{G}$, and the corresponding TPI game $\mathcal{G}' = \textsc{ConvertGame}(\mathcal{G})$, each joint pure strategy $\pi_{\mathcal{T}}$ in $\mathcal{G}$ can be mapped to a strategy π_t in $\mathcal{G}'$, such that the traversed histories have been mapped by* PubTeamConv*. Formally, $\forall \pi_{\mathcal{T}}$, there is a π_t such that $\forall \pi_o, \pi_c$ the following holds:*

$$(\textsc{PubTeamConv}(h))_{h \text{ reached by playing } (\pi_{\mathcal{T}},\pi_o,\pi_c) \text{ in } \mathcal{G}} \equiv (h')_{h' \text{ reached by playing } (\pi_t,\pi_o,\pi_c) \text{ in } \mathcal{G}'}.$$

Proof. To show that 4.2 holds, we show how for any pure joint strategy $\pi_{\mathcal{T}}$ for the team in $\mathcal{G}$ it is possible to construct an equivalent pure strategy π_t in $\mathcal{G}'$. Such goal can be achieved by recursively by traversing both $\mathcal{G}$ and $\mathcal{G}'$ while constructing π_t.

First, consider the empty histories $h_\varnothing$ and $h'_\varnothing$ for which it trivially holds that $h'_\varnothing = \textsc{PubTeamConv}(h_\varnothing)$.

Let h and $h' = \text{PubTeamConv}(h, \mathcal{G}, \mathcal{G}')$ be the nodes currently reached by the algorithm PubTeamConv respectively in $\mathcal{G}$ and $\mathcal{G}'$. We thus have the guarantee that h and h' are both terminal or both share the same player (thanks to public turn taking). Therefore, we can differentiate between the following cases:

- Case **team member node**
  Let $a = \pi_{\mathcal{T}}[I(h)]$ be the action specified by $\pi_{\mathcal{T}}$ to be taken at $I(h)$. We can construct a prescription $\Gamma = (\pi_{\mathcal{T}}[I])_{I \in \mathcal{S}[h]}$ equivalent to the pure strategy $\pi_{\mathcal{T}}$ in this public state. We set $\pi_t[I'(h')] = \Gamma$, and prosecute our proof from the two reached nodes $h'\Gamma$ and ha. The construction procedure PubTeamConv guarantees in fact that $h'\Gamma a = \textsc{PubTeamConv}(ha)$.
- Case **chance or opponent node**
  π_o and π_c are common to both the traversals. This guarantees that the action a suggested by the policy is equal, and by construction of the conversion procedure $h'a' = \textsc{PubTeamConv}(ha)$. We can thus proceed considering $h'a$ and ha.
- Case **terminal node**
  By construction, they have the same value for all players.

This concludes the proof. □

Lemma 4.3. *Given a public-turn-taking vEFG $\mathcal{G}$, and the corresponding TPI game $\mathcal{G}' = \textsc{ConvertGame}(\mathcal{G})$, each coordinator pure strategy π_t in $\mathcal{G}'$ can be mapped to a strategy $\pi_{\mathcal{T}}$ in $\mathcal{G}$, such that the traversed histories have been mapped by* PubTeamConv*. Formally, $\forall \pi_t$, there is $\pi_{\mathcal{T}}$ such that $\forall \pi_o, \pi_c$ the following holds:*

$$(\textsc{PubTeamConv}(h))_{h \text{ reached by playing } (\pi_{\mathcal{T}},\pi_o,\pi_c) \text{ in } \mathcal{G}} \equiv (h')_{h' \text{ reached by playing } (\pi_t,\pi_o,\pi_c) \text{ in } \mathcal{G}'}.$$

Proof. We can prove Lemma 4.3 recursively by traversing both $\mathcal{G}'$ and $\mathcal{G}$ while constructing the equivalent pure strategy in the original game. We start by $h'_\varnothing$ and $h_\varnothing$. We know that $h'_\varnothing = \textsc{PubTeamConv}(h_\varnothing)$.

As in the proof of Lemma 4.2 let h and $h' = \text{PubTeamConv}(h, \mathcal{G}, \mathcal{G}')$ be the nodes currently reached by the algorithm PubTeamConv respectively in $\mathcal{G}$ and $\mathcal{G}'$. We thus have the guarantee that h and h' are both terminal or both share the

same player (thanks to public turn taking). Hence, we can differentiate between the following cases:

- Case **team member node**
  Let $\Gamma = \pi_t[I'(h')]]$ be the prescription specified by π_t to be taken at $I'(h')$. We can extract the prescribed action $a = \Gamma[I]$ to be played in history h. We set $\pi_{\mathcal{T}}[I(h)] = a$, and prosecute our proof from the two reached nodes $h'\Gamma$ and ha. The PUBTEAMCONV procedure guarantees, indeed, that $h'\Gamma = \text{PUBTEAMCONV}(ha)$.
- Case **chance or opponent node**
  π_o and π_c are common to both the traversals. This guarantees that the action a suggested by the policy is equal, and by construction of the conversion procedure $h'a' = \text{PUBTEAMCONV}(ha)$. We can thus proceed with the proof considering $h'a$ and ha.
- Case **terminal node**
  By construction, they have the same value for all players.

This concludes the proof. □

Theorem 4.5. *A public-turn-taking vEFG $\mathcal{G}$ and its TPI game $\mathcal{G}' = \text{CONVERTGAME}(\mathcal{G})$ are payoff-equivalent, i.e.*

$$\forall \pi_{\mathcal{T}}\ \forall \pi_o, \pi_c : u_{\mathcal{T}}(\pi_{\mathcal{T}}, \pi_o, \pi_c) = u_t(\rho(\pi_{\mathcal{T}}), \pi_o, \pi_c),$$
$$\forall \pi_t\ \forall \pi_o, \pi_c : u_{\mathcal{T}}(\sigma(\pi_t), \pi_o, \pi_c) = u_t(\pi_t, \pi_o, \pi_c).$$

Proof. The proof follows trivially from Lemmas 4.2 and 4.3. Indeed, one can resort to the proof of Lemma 4.2 to obtain, for each strategy $\pi_{\mathcal{T}}$ in $\mathcal{G}$, a payoff-equivalent strategy π_t in $\mathcal{G}'$. The other direction can be obtained by following the proof of Lemma 4.3. □

Theorem 4.6. *Given a public-turn-taking vEFG $\mathcal{G}$, and the corresponding TPI $\mathcal{G}' = \text{CONVERTGAME}(\mathcal{G})$, a Nash Equilibrium μ_t^* in $\mathcal{G}'$ is realization equivalent to a TMEcor $\mu_{\mathcal{T}}^* = \sigma(\mu_t^*)$ in $\mathcal{G}$.*

Proof. By hypothesis that μ_t^* is a NE, we have that:

$$\mu_t^* \in \arg \max_{\mu_t \in \Delta^{\Pi_t}} \min_{\mu_o \in \Delta^{\Pi_o}} \sum_{\substack{\pi_t \in \Pi_t \\ \pi_o \in \Pi_o \\ \pi_c \in \Pi_c}} \mu_t(\pi_t)\mu_o(\pi_o)\mu_c(\pi_c)u_t(\pi_t, \pi_o, \pi_c).$$

We need to prove:

$$\sigma(\mu_t^*) \in \arg \max_{\mu_{\mathcal{T}} \in \Delta^{\Pi_{\mathcal{T}}}} \min_{\mu_o \in \Delta^{\Pi_o}} \sum_{\substack{\pi_{\mathcal{T}} \in \Pi_{\mathcal{T}} \\ \pi_o \in \Pi_o \\ \pi_c \in \Pi_c}} \mu_{\mathcal{T}}(\pi_{\mathcal{T}})\mu_o(\pi_o)\mu_c(\pi_c)u_{\mathcal{T}}(\pi_{\mathcal{T}}, \pi_o, \pi_c)$$

Let $\min_{TMEcor}(\mu_{\mathcal{T}})$ and $\min_{NE}(\mu_t)$ be the inner minimization problem in the TMECor and NE definition respectively.

Absurd. Suppose $\exists\ \bar{\mu}_{\mathcal{T}}$ with a greater value than $\sigma(\mu_t^*)$. Formally:

$$\min_{TMEcor}(\bar{\mu}_{\mathcal{T}}) > \min_{TMEcor}(\mu_t^*).$$

In such a case, we could define $\bar{\mu}_t = \rho(\bar{\mu}_{\mathcal{T}})$ having value:

$$\min_{NE}(\bar{\mu}_t) = \min_{TMEcor}(\bar{\mu}_{\mathcal{T}}) > \min_{TMEcor}(\sigma(\mu_t^*)) = \min_{NE}(\mu_t^*),$$

where the equalities are due to the payoff equivalence. However this is absurd since by hypothesis μ_t^* is a maximum. Therefore necessarily:

$$\sigma(\mu_t^*) \in \arg \max_{\mu_{\mathcal{T}} \in \Delta^{\Pi_{\mathcal{T}}}} \min_{NE}(\mu_{\mathcal{T}}).$$

This concludes the proof. □

Theorem 4.8. *Given a public-turn-taking vEFG $\mathcal{G}$ with common external information for the team $\mathcal{T}$, the tree of corresponding TPI game $\mathcal{G}'$ has a number of nodes linear in the nodes of $\mathcal{G}$.*

Proof. Consider first the opponent and chance nodes. Such nodes are copied unaltered, hence this operation does not increase the total number of nodes. Now, let us focus on team players' nodes. In order to prove the Theorem we have to show that, for any $h' \in \mathcal{H}'$, only one infoset of the original game can be mapped to the public state $\mathcal{S}_t(h')$. This ensures

that node h' has the same number of actions in output as infoset to which it is mapped, hence the overall number of nodes does not increase.

Fix a node $h \in \mathcal{H}$ and let $h' = \text{PubTeamConv}(h, \mathcal{G}, \mathcal{G}')$. The public state is characterized by all the actions publicly observed by the team. Formally, the set of such actions in $\mathcal{G}$ at history h is:

$$\Lambda = \{a \in h \mid Pub_{\mathcal{T}}(a) = \text{pub}\}.$$

Assume now, by absurd, that the set $\mathcal{S}_{\mathcal{T}}(h)$ contains two distinct information sets $I, J \in \mathcal{I}$. This would mean that there exists $a_I \in h_I \setminus \Lambda, a_J \in h_J \setminus \Lambda$ for $h_I \in I$, $h_J \in J$ such that:

$$Pub_p(a_I) \neq Pub_p(a_J), \tag{3}$$

where $p = \iota(h) \in \mathcal{T}$. Intuitively, the condition expressed by Equation (3) states that the two infosets are distinct.

However, this is impossible as the condition violates the assumption of A-loss refinement and common external information. This results in generating a node h' with the same number of actions as h, hence the dimension of the TPI $\mathcal{G}'$ does not increase with respect to the dimension of $\mathcal{G}$. □

Proposition 4.9. *Any action or state abstraction that, once applied to an extensive-form game $\mathcal{G}$, returns a perfect-recall timeable game can be mapped specularly in the team-public-information representation $\mathcal{G}' = \text{CONVERTGAME}(\mathcal{G})$. The reverse is not true.*

Proof. Trivially, any aggregation of states or actions defined in the extensive form leads to a game that can be converted in the corresponding team-public-information representation by using Algorithm 1. On the other hand, not all abstractions in the public information game can be reflected in the original one. As an example, consider Figure 1. If we perform *action abstraction* in the converted game, by collapsing action "0:A, 1:C" onto action "0:A, 1:D", this abstraction cannot be remapped onto the original game. This happens because such abstraction corresponds to a constraint on the possible strategies that Player 1 can choose since we are forbidding him to play any pure strategy that requires to play action A at infoset 0 and action C at infoset 1. Such an abstraction does not modify the original game structure, since all A, B, C, D may be played for some specific prescription. □

B. Information Structure in Team Games

The core problem of finding a TMEcor in adversarial team games resides in *asymmetric visibility* since team members have a private state that does not allow creating a perfect recall joint coordination player by trivially merging the players without any modification of their information structure.

In the following, we characterize the possible types of asymmetric visibility that may cause imperfect recall for the joint player, and singularly address them.

- **Non-visibility over a team member's action**. If a team member plays an action hidden from another team member, the joint team player would have imperfect recall due to the forgetting of his own played actions. This source of imperfect recallness can be avoided in a TMEcor by considering the shared deterministic strategies before the game starts, thanks to *ex-ante coordination*. This allows us to know a priori the exact actions played by team members in each node. Thus it is safe to apply a perfect recall refinement in the original game, which corresponds to always considering the chosen action of a team member as obs by other team members.
- **Non-visible game structure**. Consider two nodes in the same information set for a player before which the other team member may have played a variable number of times, due to a chance outcome non-visible to the team member of these nodes. In this case, a perfect recall refinement is not applicable to distinguish the nodes, because it would give the joint coordinator information that is private of the current player. To solve this edge case, we require the property of public turn-taking.
- **Private information disclosed by chance/adversary to specific team members.** It is the most complex type of non-visibility, since in a TMEcor we have no explicit communication channels through which to share information, and therefore this type of joint imperfect recall can only be addressed by considering a strategically equivalent representation of the game in which at most one of the team players has private information.

Algorithm 2 Pruned Public-Team Conversion

function CONVERTGAME($\mathcal{G}$)
  initialize $\mathcal{G}'$ new game
  $\mathcal{N}' \leftarrow \{t, o\}$
  $h'_{\varnothing} \leftarrow$ PUBTEAMCONV($h_{\varnothing}, \mathcal{G}, \mathcal{G}'$) ▷ new game root
  return $\mathcal{G}'$

function PUBTEAMCONV(h, $\mathcal{G}$, $\mathcal{G}'$, $\boldsymbol{\mathcal{X}}$)
  initialize $h' \in \mathcal{H}'$
  if $h \in \mathcal{Z}$ **then** ▷ terminal node
    $h' \leftarrow h' \in \mathcal{Z}'$
    $u'_p(h') \leftarrow u_p(h) \quad \forall p \in \mathcal{N}$
  else if $\mathcal{P}(h) \in \{o, c\}$ **then** ▷ opponent or chance
    $\mathcal{P}'(h') \leftarrow \mathcal{P}(h)$
    $\mathcal{A}'(h') \leftarrow \mathcal{A}(h)$
    if h is chance node **then**
      $\sigma'_c(h') = \sigma_c(h)$
    for $a' \in \mathcal{A}'(h')$ **do**
      $Pub'_t(a') \leftarrow \text{obs}$ **if** $Pub_{\mathcal{T}}(a') = \text{pub}$ **else** unobs
      $Pub'_o(a') \leftarrow Pub_o(a')$
      $h'a' \leftarrow$ PUBTEAMCONV(ha', $\mathcal{G}$, $\mathcal{G}'$, $\boldsymbol{\mathcal{X}}$)
  else ▷ team member
    $\mathcal{P}'(h') = t$
    $I \leftarrow I(h)$
    $\mathcal{A}'(h') \leftarrow \bigtimes_{J \in \mathcal{S}_{\mathcal{T}_I}(h) : \nexists J' \in \boldsymbol{\mathcal{X}} \text{ matching } J} \mathcal{A}(I)$ ▷ prescriptions
    for $\Gamma' \in \mathcal{A}'(h')$ **do**
      $Pub'_t(\Gamma') \leftarrow \text{seen}, Pub'_o(\Gamma') \leftarrow \text{unseen}$
      $a' \leftarrow \Gamma'[I(h)]$ ▷ extract chosen action
      $\boldsymbol{\mathcal{X}} \leftarrow \boldsymbol{\mathcal{X}} \cup \{\boldsymbol{J} : \boldsymbol{\Gamma'(J)} \neq \boldsymbol{a'}\}$ ▷ update $\mathcal{X}$ removing incompatible private states
      initialize $h'' \in \mathcal{H}'$
      $\mathcal{A}'(h'') \leftarrow \{a'\}$
      $\mathcal{P}(h'') = c$
      $Pub'_t(a') \leftarrow \text{seen}$
      $Pub'_o(a') = Pub_o(a')$
      $\sigma'_c(h'') = \text{play } a' \text{ with probability 1}$
      $h''a' \leftarrow$ PUBTEAMCONV(ha', $\mathcal{G}$, $\mathcal{G}'$, $\boldsymbol{\mathcal{X}}$)
      $h'\Gamma \leftarrow h''$
  return h'

C. Pruning and Abstraction Techniques to Generate More Concise Representations

As aforementioned, in the worst case, our representation cannot have a size upper bounded by a polynomial in the size of the extensive form unless $\mathsf{P} = \mathsf{NP}$. Nevertheless, in many cases, the game tree generated by our conversion may contain redundant information, and thus it can be compressed without any loss of information. In the following, we provide different procedures to generate a much more concise team-public-information representation of an adversarial team game.

Pruned Representation. Whenever the coordinator prescribes a team member to play an action a such that $Pub_{\mathcal{T}}(a) = \text{pub}$, where $\mathcal{T}$ is the team, the possible private states in which the player may be can be reduced after observing the action chosen from the given prescription, and this may also impact on the possible private states of other team members. Since the number of prescriptions depends on the number of private states, a dramatic reduction of the number of prescriptions is achieved without any loss of information.

The pseudocode of the procedure to directly generate a TPI in its pruned representation is provided in Algorithm 2. It takes as input the same vEFG as Algorithm 1. In particular, the procedure is obtained by a simple modification of Algorithm 1,

adding a parameter $\mathcal{X}$ in PUBTEAMCONV which is used to store the private states that can be excluded in the following part of the tree once played a public action. To ease the visualization, modifications to Algorithm 1 are highlighted in bold. By excluding every information set in $\mathcal{X}$ when building the prescription in Line 23, we can effectively prune the number of private states to be considered by the coordinator. An example of pruned representation is provided in Figure 5 in Appendix D.

Folded Representation. In the basic TPI game produced by Algorithm 1, chance outcomes are explicitly represented in the game tree independently of the visibility of the outcomes, thus branching the game tree into different subgames according to the specific outcome. Consider the case of a chance action that can be observed by a team member and not observed by the adversary. In the converted game, such an action is not observable to any player, and therefore it can be safely postponed as long as no specific action depends on it. The folding representation takes advantage of this property to avoid sampling these types of private states. Instead, it samples an action from the prescription depending on the probability that a specific private state is present at a given point in the game, given the previous actions of all players and their current strategies. The dummy chance nodes h'' instantiated in Algorithm 1 therefore may present different actions, each with a probability given by the sum of the probabilities of the private states for which that action has been prescribed.

This approach can be considered as a hybrid game-specific representation between the public tree of the team and the original tree of the adversary, allowing a dramatic reduction of the size of games with private signals such as Poker. To apply the folded representation to Kuhn and Leduc Poker, we maintain a belief over the possible joint cards assigned to the team members and perform a Bayesian update whenever new information is disclosed. In Poker, this happens by choosing a public action after a prescription and drawing a public card. This belief can then be integrated with full history information (adversary and public card) to determine the probability of picking specific actions from a given prescription, and to evaluate the payoffs at the terminal nodes. The information state of the coordinator is described by the full sequence of prescriptions given and public information for the team. This type of belief and reach probability are not novel as they have been introduced by Foerster et al. (2019) and Sokota et al. (2021) in cooperative multiagent RL settings.

The name *Folded Representation* is inspired by the fact that trajectories with the same public actions but different private states are folded one over the other in the converted game. An example of folded representation is provided in Figure 6 in Appendix D.

Imperfect-Recall Abstraction of the Folded Representation. This representation takes advantage of the fact that subgames rooted in information states, whose current belief and public actions are the same, correspond to the same state of the original game. Therefore those subgames share the same structure and the same payoffs.

Thus, we can avoid including the full sequence of prescriptions in the information set of each player. This does not directly reduce the number of nodes, but it reduces the number of information sets, simplifying the information structure of the game. This also reduces the space requirements to represent the strategies and simplifies the information structure of the coordinator. This abstraction technique is theoretically sound and leads to a *well-formed* game in the sense by Lanctot et al. (2012a). Therefore, in these settings, as showed by Lanctot et al. (2012a), no-regret algorithms converge to the equilibrium. In our experiments, we employ this information state refinement technique on top of the folded representation. An example of imperfect-recall abstraction of the folded representation is provided in Figure 7 in Appendix D.

Lossy Imperfect-Recall Abstraction of the Folded Representation. The compression techniques used for generating the pruned and folded representations have a high impact whenever the coordinator's prescription includes different actions to different private states. These actions are observable to the team members. Since different actions are played at different private states, observing an action reveals the private state, thus simplifying the part of the games following such a prescription. On the other hand, whenever the coordinator prescribes the same action to every private state, playing an action does not reveal any information. Therefore, the public state keeps having a combinatorial size. Intuitively, the higher the degree of signaling (communication), the smaller the size of the tree.

The main idea behind our lossy abstraction is to discard all the uninformative prescriptions recommending to play the same card at every private state. More precisely, in our Poker instance, we discard from the game tree all the prescriptions recommending to play Fold at every private state, and we do the same for the cases of Call and Raise. Notice that such discarding is equivalent to forcing the coordinator to play those prescriptions with zero probability. Interestingly, this abstraction cannot be defined on the extensive-form game, while it can be defined on our representation.

In particular, we discard the above coordinator's actions from the folded representation and apply the imperfect-recall abstraction described above, thus obtaining a *well-formed game* as defined in Lanctot et al. (2012a). An example of imperfect-recall abstraction of the folded representation is provided in Figure 8 in Appendix D.

D. Comparison among the Representations

We provide an example of extensive-form game and of the three conversions described in the paper in Figs. 3–6. To ease the visualization, we focus on a cooperative game with no adversary.

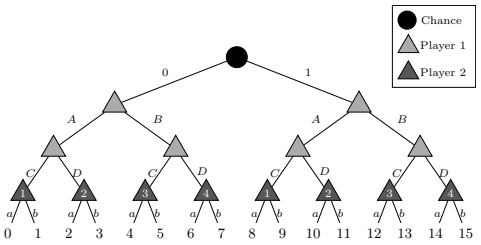


Figure 3: Extensive form of a 2-player team game with chance and without adversary, where Player 2 observes all actions except those of chance. Nodes of a player with same number are in the same infoset.

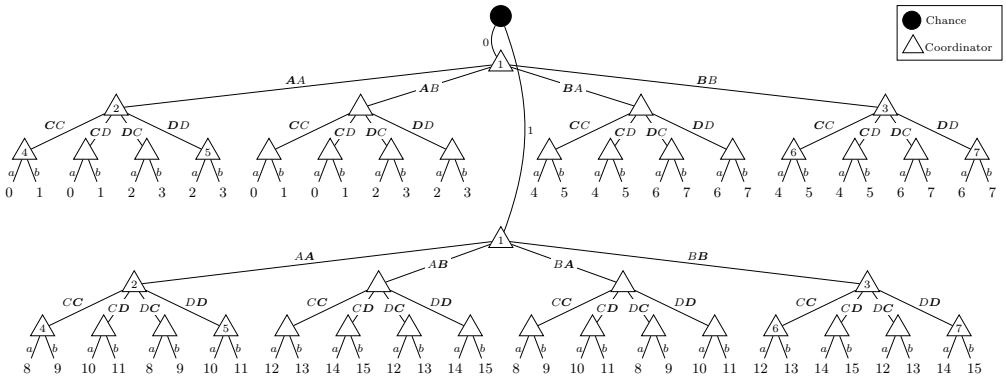


Figure 4: Team-public-information representation of the game depicted in Figure 3. Nodes of a player with same number are in the same infoset. For the sake of notation, dummy chance nodes are not represented, prescriptions include the action to take for private state 0 and 1, the action taken afterward is in bold in the prescription.

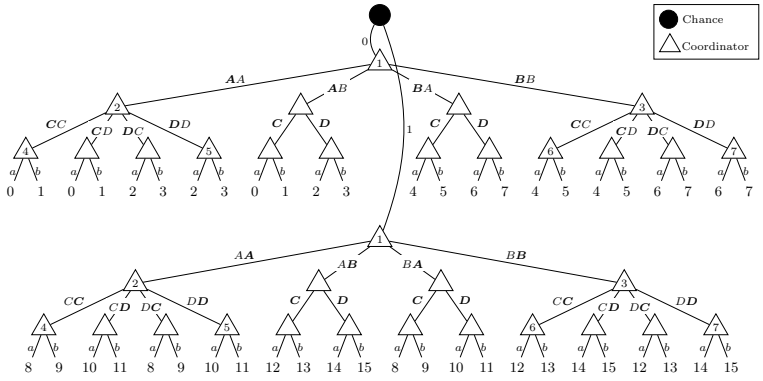


Figure 5: Pruned team-public-information representation of the game depicted in Figure 3. Nodes of a player with the same number are in the same infoset. For the sake of notation, dummy chance nodes are not represented, prescriptions include the action to take for private state 0 and 1, the action taken afterward is in bold in the prescription.

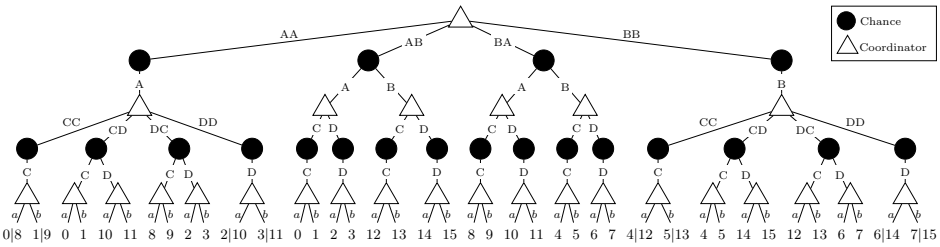


Figure 6: Folded team-public-information representation of the game depicted in Figure 3. For the sake of notation, prescriptions include the action to take for private state 0 and 1. Terminal nodes in the form $x|y$ represent a terminal node which has a weighted average value with respect to the outcomes x and y.

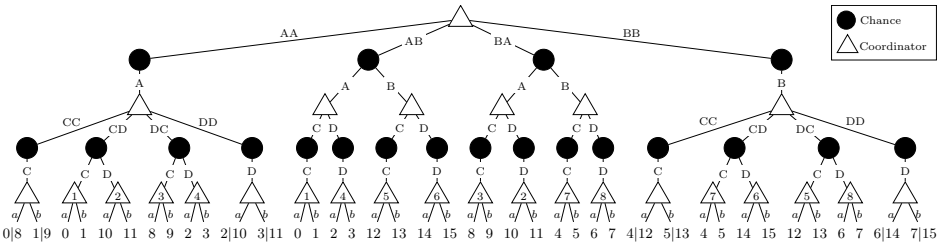


Figure 7: Imperfect-recall abstraction of the folded team-public-information representation of the game depicted in Figure 3. For the sake of notation, prescriptions include the action to take for private state 0 and 1. Terminal nodes in the form $x|y$ represent a terminal node which has a weighted average value with respect to the outcomes x and y. Note that the coordinator has imperfect recall on the nodes characterized by the knowledge of a specific private state and sharing the same public history of played actions.

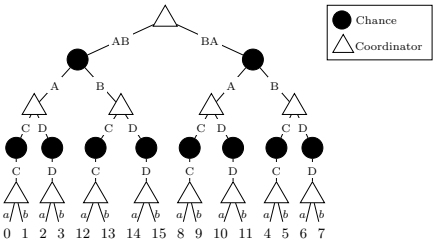


Figure 8: Lossy imperfect-recall abstraction of the folded team-public-information representation of the game depicted in Figure 3. For the sake of notation, prescriptions include the action to take for private state 0 and 1. Terminal nodes in the form $x|y$ represent a terminal node which has a weighted average value with respect to the outcomes x and y. In this case, the imperfect recall abstraction does not coarsen the information structure of the coordinator, since all the nodes at the last level are characterized by a different private state-public history combination.

E. Experimental settings

E.1. Poker instances

We refer to the three-player generalizations of Kuhn and Leduc poker proposed by (Farina et al., 2018).

Like all poker games, at the start of the game each player antes one to the pot, and receives a private card. Then players play sequentially in turn. Each player may check by adding to the pot the difference between the higher bet made by other players and their current bet (i.e. by matching the maximum bet made by others). Each player may fold whenever a check requires putting more money into the pot and the player instead decides to withdraw. Each player may raise whenever the maximum number of raises allowed by the game is not reached, by adding to the pot the amount required by a check plus an extra amount called raise amount. A betting round ends when all non-folded players except the last raising player have checked.

In **Kuhn poker**, there are three players and k possible ranks with k different ranks. The maximum number of raises is one, and the raising amount is 1. At the end of the first round, the showdown happens. The player having the highest card takes all the pot as payoff.

In **Leduc poker**, there are three players, k possible ranks having 3 cards in the deck each, and 1 or 2 raises. The raise amount is 2 for the first raise and 4 for the second raise. At the end of the first round, a public card is shown, and a new round of betting starts from the same player starting in the first round. In the end, the showdown happens. Winning players are having a private card matching the rank of the public card. If no player forms a pair, then the winning player is the one with the card with the highest rank. In the case of multiple winners, the pot is split equally.

E.2. Implementation details

We implemented the folded representation of both Kuhn and Leduc taking advantage of the OpenSpiel (Lanctot et al., 2019b) framework. The framework allowed us to specify the game as an evolving state object and provided the standard resolution algorithms for the computation of a Nash Equilibrium in the converted game.

The experiments have been performed on a machine running Ubuntu 20.04 with a Intel Xeon Platinum 8358 (128) @ 3.300GHz CPU with 503 GB of memory. The implementation is single-threaded.

E.3. Design choices

Customarily, researchers developed *ad hoc* codes with different programming languages, each exploiting various programming optimization. This approach makes the comparison among the different algorithms difficult, hiding their actual scalability and sometimes emphasizing ancillary, non-central issues (*e.g.*, adopting different versions of GUROBI or CPLEX). For this reason, we opted to adopt a tool publicly available to represent and solve the transformed games (*i.e.*, OpenSpiel framework). While such a framework is general and readily available, some implementation choices for memory allocation and game representation slow down the performance with respect to the custom implementation by Zhang & Sandholm (2021). The only metric allowing us to have a comparison not depending on the specific technology is the size of the optimization problem. This is the reason why we directly compare the number of variables and constraints of the linear program used by Zhang & Sandholm (2021) with the number of infosets and actions of our game tree. Interestingly, there is a strict connection between the variables in Zhang & Sandholm (2021) and our actions, and the number of constraints in Zhang & Sandholm (2021) and our infosets. The interesting point is that the size of the problem by Zhang & Sandholm (2021) and the size of problem (produced thanks to abstractions) are asymptotically the same as the size of the instance increases. This suggests that, asymptotically, the relative performance of solving our tree and the problem by Zhang & Sandholm (2021) depend only on the two algorithms (as the size of the instances is the same). In particular, the relative performance between no-regret and linear programming is known (Zhang & Sandholm, 2020).

F. Plots

We report a larger version of the exploitability plots provided in the main body of the paper.

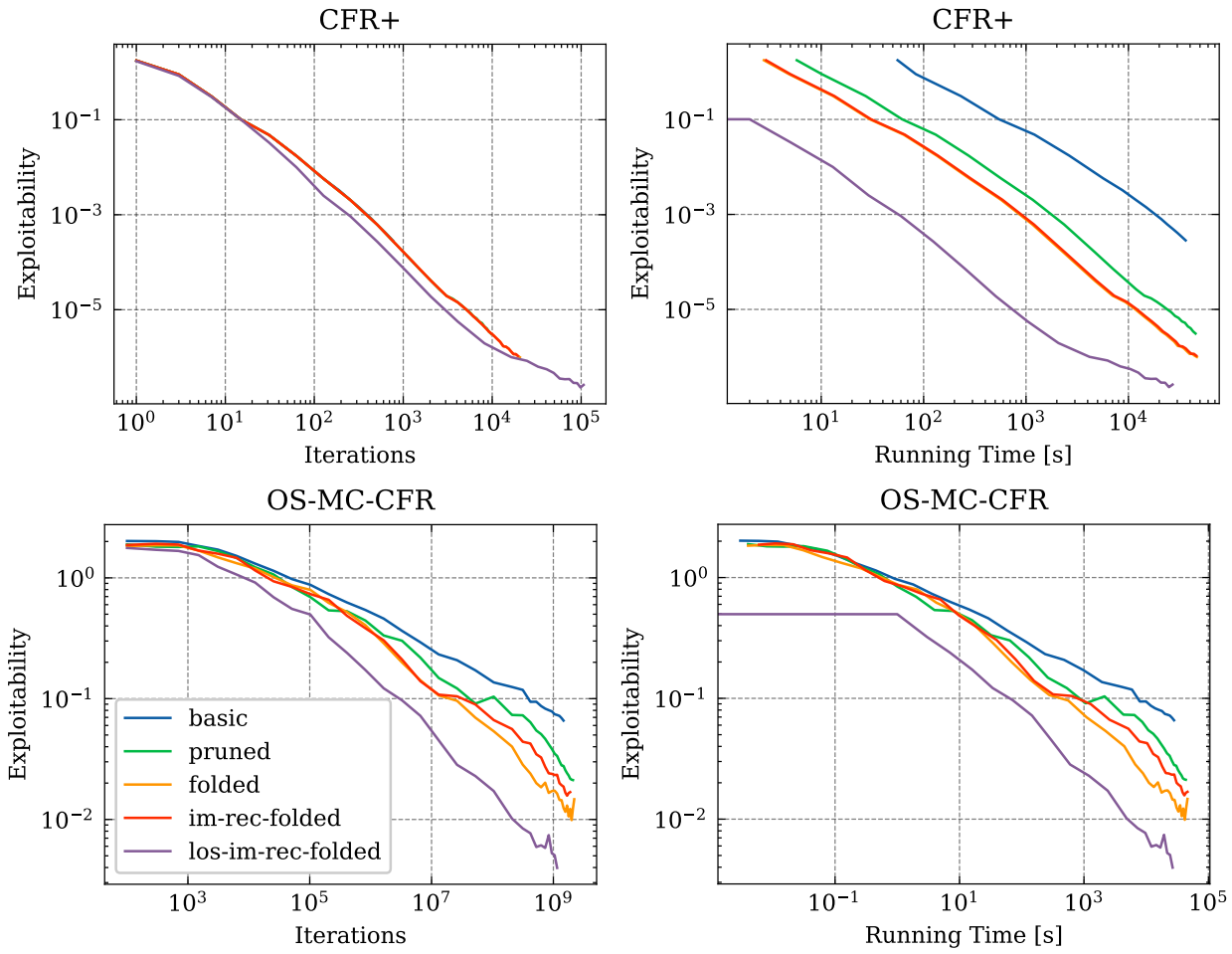


Figure 9: Exploitability of CFR+ and OS-MC-CFR with 21L133 game in the number of iterations and time (seconds).