Final Presentation: Multi-Agent Coordination through Signal Mediated Strategies

Federico Cacciamani
federico.cacciamani@mail.polimi.it
CSE Track

POLITECNICO MILANO 1863

HONOURS PROGRAMME HP-SR in Information Technology

Outline

1. Introduction to the state of the art
2. Signal Mediated Strategies
3. Experimental Results

Outline

1. **Introduction to the state of the art**
2. Signal Mediated Strategies
3. Experimental Results

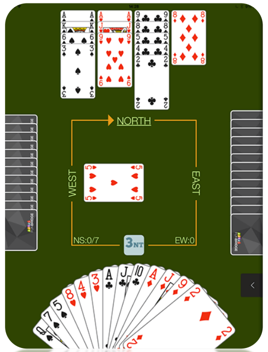
Introduction to the problem: what we study?

Introduction to the problem: what we study?

- We analyze multi-agent environments with mixed cooperative-competitive nature.

Introduction to the problem: what we study?

- We analyze multi-agent environments with mixed cooperative-competitive nature.
- Practical applications:
    - Recreative applications (e.g. contract Bridge).
    - Security.
    - Car racing.



Introduction to the problem: how we study it?

- Adopt an Algorithmic Game Theory approach.
- Mathematical formulation of the games and of the objectives: *equilibria.*
- Start by analyzing solutions proposed for two-player games with perfect recall.
- **Nash Equilibrium:** pair of strategies such that no player benefits from deviating

Introduction to the problem: state of the art

Two-players zero-sum games

Linear Programming

Solution of TPZSGs: Linear Programming

- Straightforward formulation: two-players zero-sum game as a maxmin problem.
- Example:

$$
\begin{array}{ll}
\max_x \min_y x^t P y & \text{s.t.} \\
\sum_i x_i = 1 & \\
\sum_i y_i = 1 & \\
x_i \geq 0 \quad \forall i & \\
y_i \geq 0 \quad \forall i &
\end{array}
$$

- LP needs the normal form representation of the game, that has a size that grows exponentially with the number of decision nodes

Introduction to the problem: state of the art

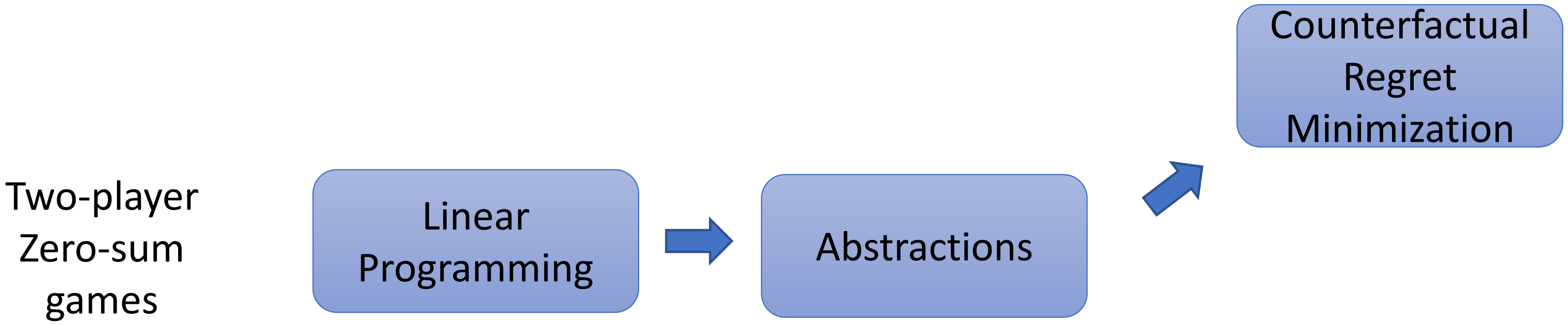
Two-player
Zero-sum
games

Linear Programming → Abstractions

Abstractions

- When considering large games, it can be useful to use an **abstract** (e.g. with lower complexity) version of the game.
- It is possible to build abstractions with three different approaches:
  - *Information abstraction*: some information sets of the original game are made indistinguishable in the abstract game.
  - *Action abstraction*: some actions in the original game are grouped in the abstract game, resulting in a smaller number of actions.
  - *Simulation-based abstraction*: the abstract version of the game is built starting from collected experiences.

Introduction to the problem: state of the art



Solution of TPZSGs: CFR

- (Zinkevich et al., 2008).
- Introduction of the concept of **regret**.
- CFR is an iterative algorithm that applies a regret minimizing scheme called **Regret Matching** locally at each information set.
- Average strategies converge to NE in a two-player zero-sum game with perfect recall, with a fast convergence rate ($\epsilon \sim O(\frac{1}{\sqrt{T}})$).
- Downsides:
  - It requires a full tree traversal at each iteration.
  - It requires to update regrets at each information set.

Introduction to the problem: state of the art

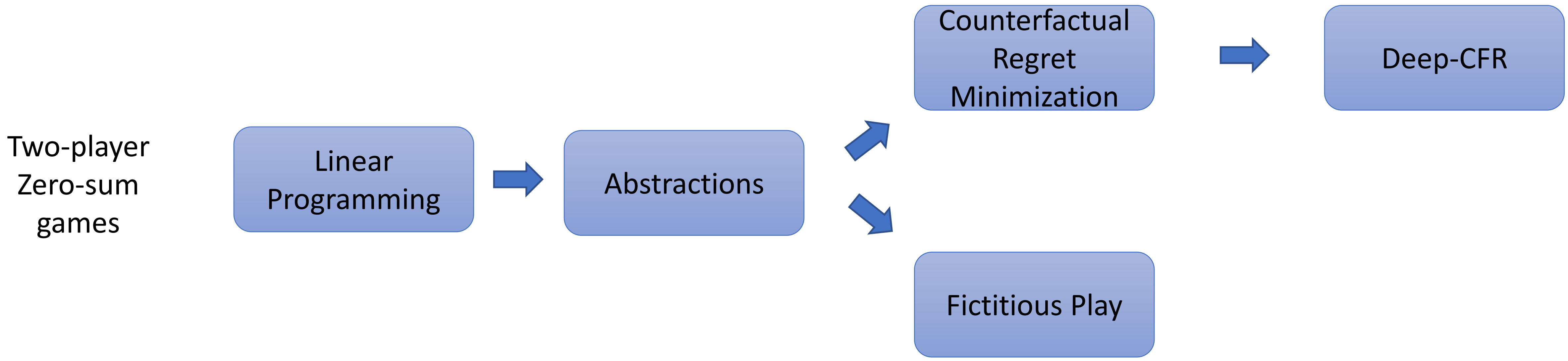
Two-player Zero-sum games

Linear Programming → Abstractions → Counterfactual Regret Minimization → Deep-CFR

Solution of TPZSGs: Deep-CFR

- (Brown et al., 2018).
- Reduces the complexity of tabular CFR by leveraging deep learning techniques.
- To avoid complete tree traversals, uses Monte Carlo sampling (also done by MCCFR).
- Uses two different neural networks for the players:
  - One used to simulate the behavior of tabular CFR (actions are chosen by regret matching on the output of this network).
  - The second used to keep track of the average strategy, the one that converges to the NE.
- Deep-CFR, with high probability, maintains the guarantees of CFR of converging to a NE.

Introduction to the problem: state of the art

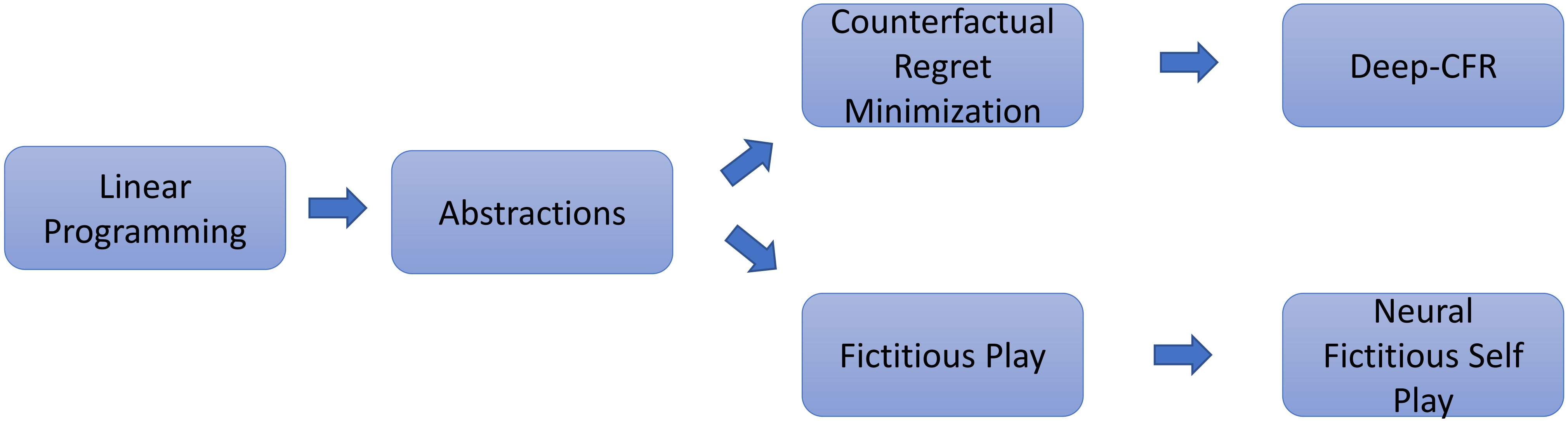


Solution of TPZSGs: Fictitious Play

- (Brown, 1951).
- At each iteration each player plays optimally (in best response) against the average strategy played by the opponent.
- In two-player zero-sum games with perfect recall, this learning dynamic average strategies are proved to converge to a NE, with slow learning rate ($\epsilon \sim O(T^{-\frac{1}{|S_i|}})$).

Introduction to the problem: state of the art

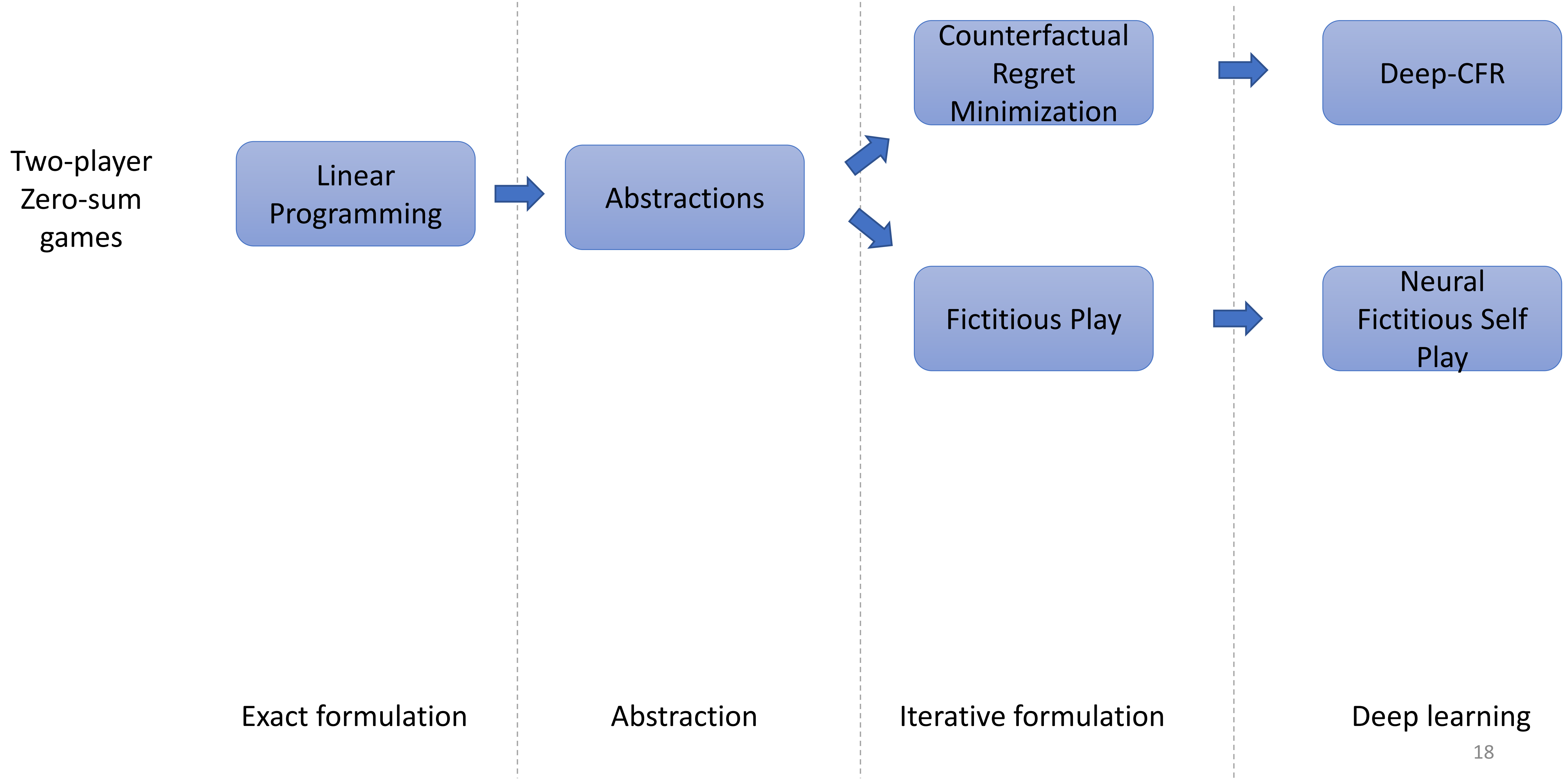
Two-player
Zero-sum
games



Solution of TPZSGs: Neural Fictitious Self Play

- (Heinrich and Silver, 2016).
- Combine FP with deep learning.
- Use two neural networks:
  - One that simulates the BR computation.
  - One that simulates the average strategy computation.
- Experiences are stored in two different buffers, one used for the DQN (replay buffer) and one used for the average strategy network (reservoir buffer).
- Average strategies converge to approximate NE in self-play.
- Maintains the convergence rate of FP.

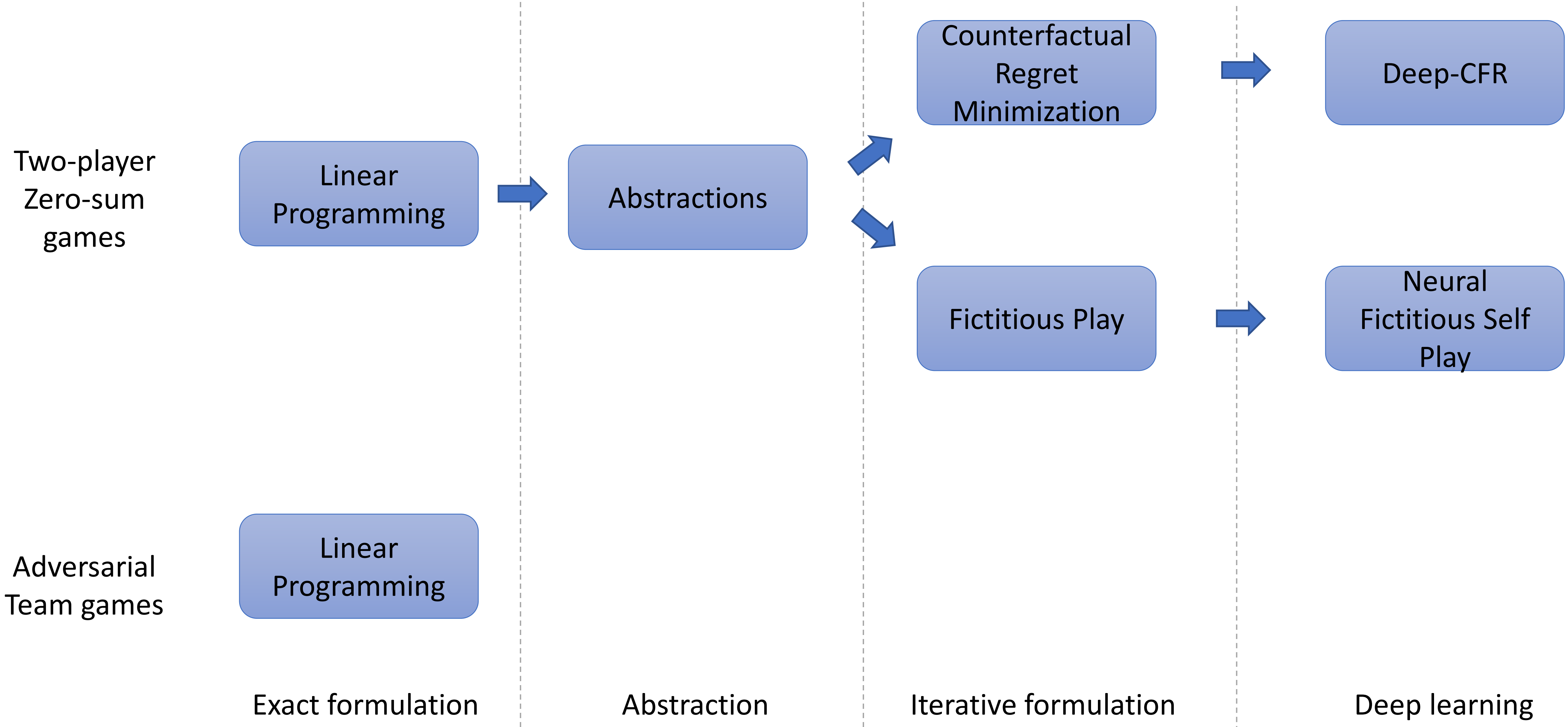
Introduction to the problem: state of the art



Considering multiple agents

- When the environment considered is multi-agent, additional complexity is added to the study of the problem.
- One successful solution is *Pluribus* (Brown and Sandholm, 2019), but the approach adopted is an heuristic one.
- Introducing mixed cooperative competitive nature: *Adversarial Team Games.*
- Considering ATGs brings several complications w.r.t. the two-players case:
  - **Inexpressivity** of decentralized behavioral strategies
  - In imperfect information games, team is an **imperfect-recall** player
- For ATGs coordination of strategies becomes fundamental, and the NE doesn't represent a ''good'' solution concept for the team. Hence the *TMEcor* is introduced.

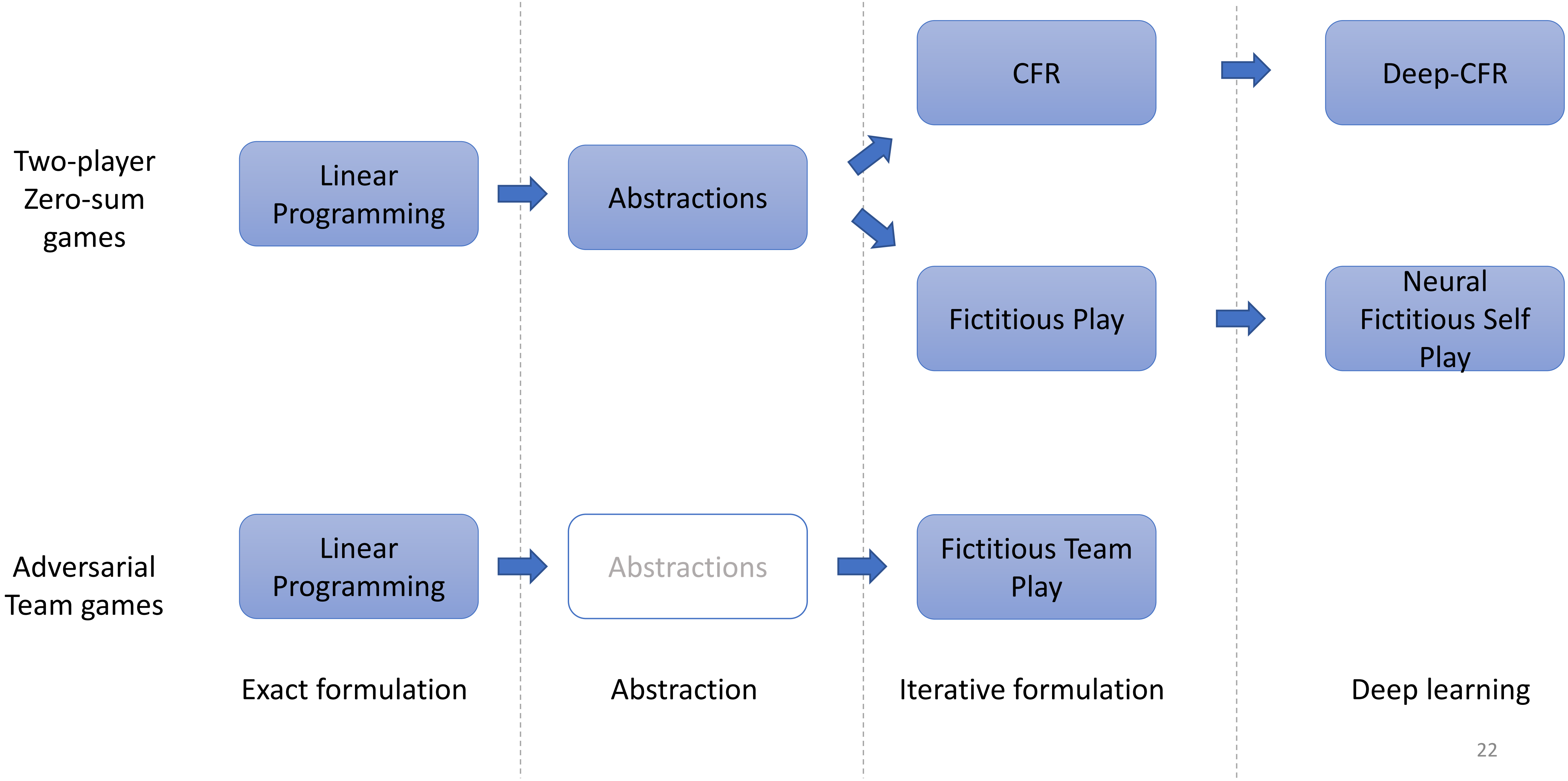
Introduction to the problem: state of the art



Solution of ATGs: Hybrid Column Generation

- (Celli and Gatti, 2017).
- Exploits an hybrid representation of the game in which the opponent strategy and the team strategy are represented in different forms.
- Works by iteratively solving three Integer Linear Programs.
- Computes the TMEcor.
- The adoption of ILPs (NP-hard) brings high computational complexity and low scalability.

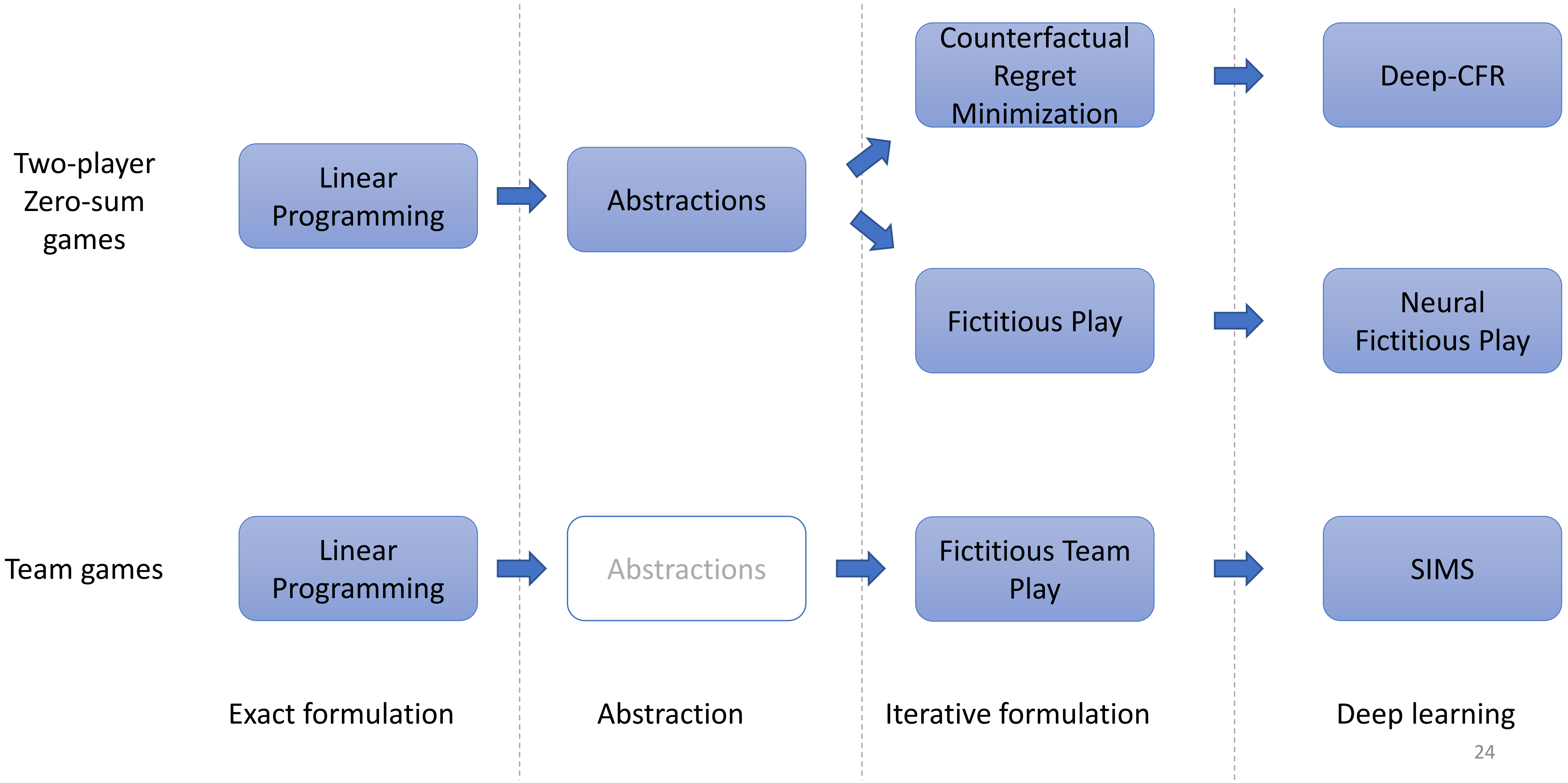
Introduction to the problem: state of the art



Solution of ATGs: Fictitious Team Play

- (Farina et al., 2018).
- Adaptation of Fictitious Play to the case of Adversarial Team Games.
- Adopts an auxiliary representation of the game tree obtained by fixing the strategy of one of the team players at the root of the tree.
- Works by iteratively solving two Mixed Integer Linear Programs defined in the auxiliary game
- Converges to the TMEcor.
- Adoption of MILPs (NP-hard) causes high computational complexity and low scalability.

Introduction to the problem: state of the art



Outline

1. Introduction to the state of the art
2. **Signal Mediated Strategies**
3. Experimental Results

Decoupling the problem

Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:

Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
    - Trajectory sampling

Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling

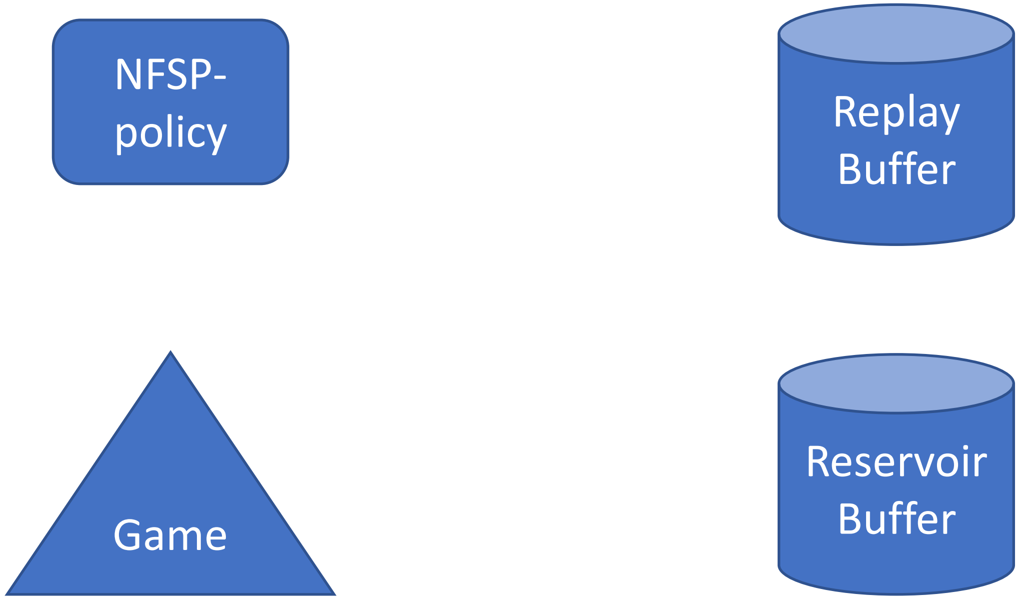
NFSP

Deep-CFR

Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling

NFSP

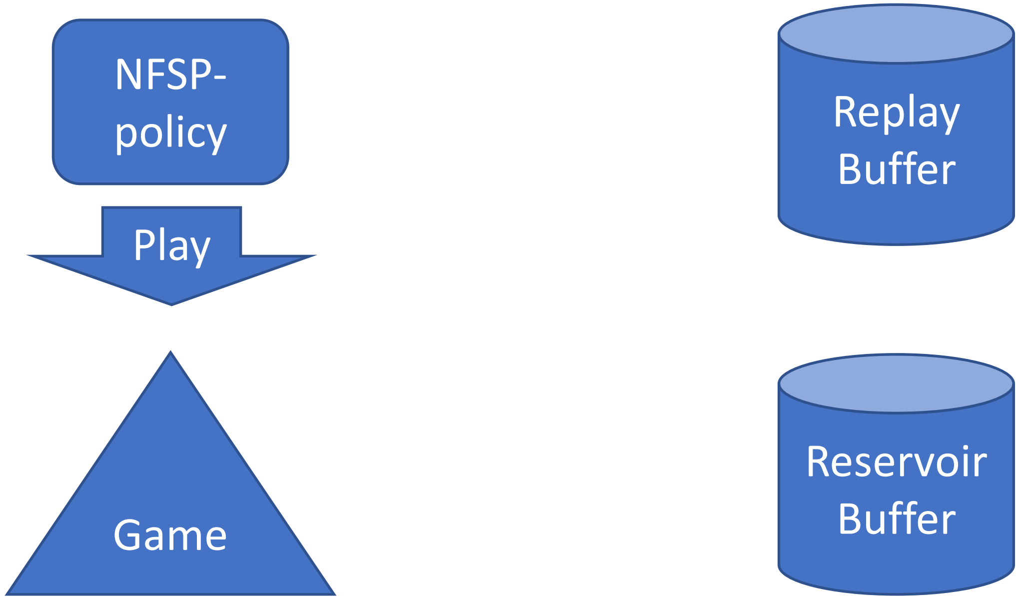


Deep-CFR

Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling

NFSP

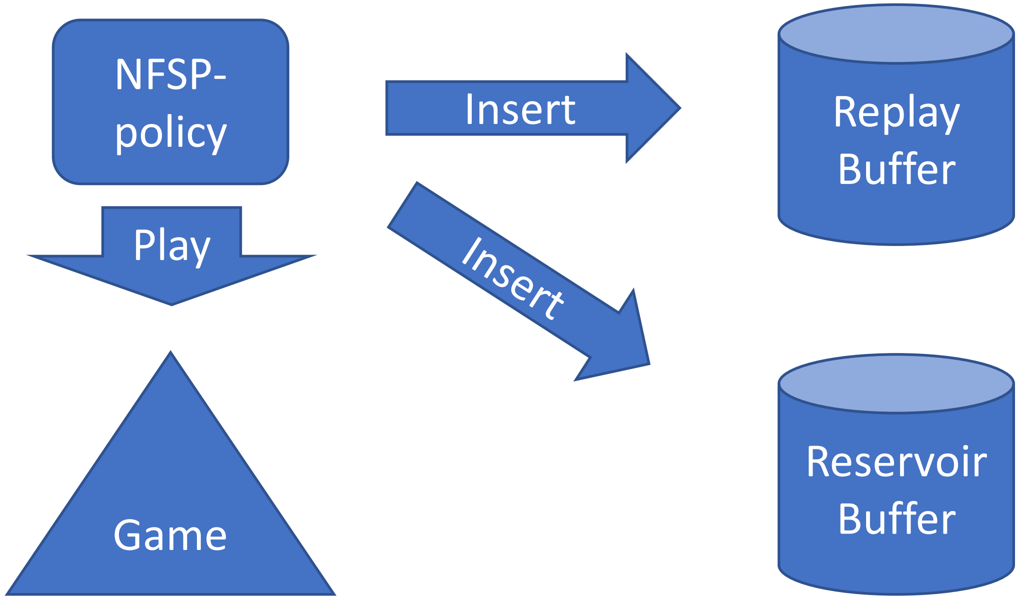


Deep-CFR

Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling

NFSP

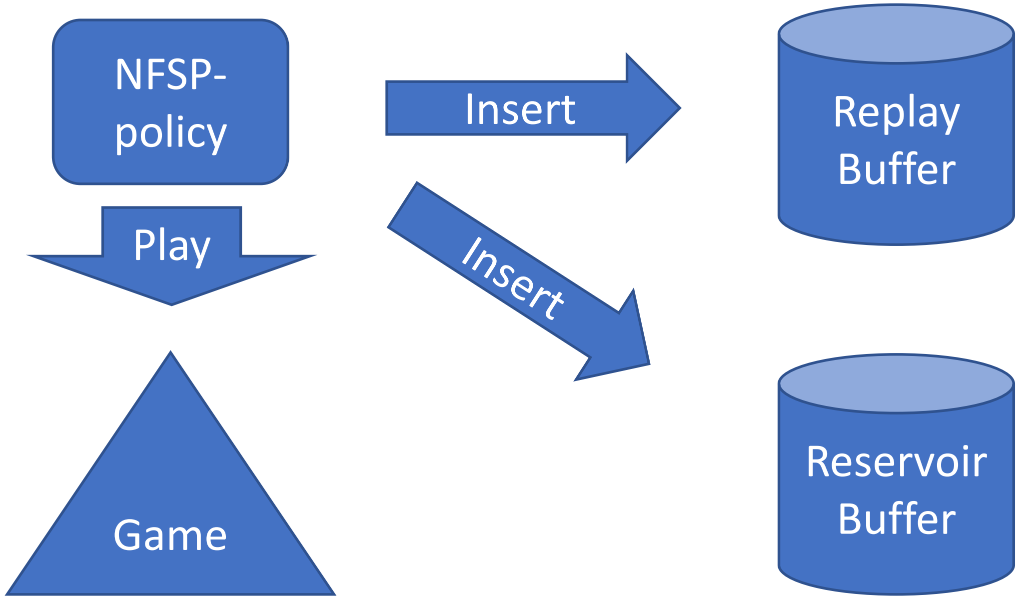


Deep-CFR

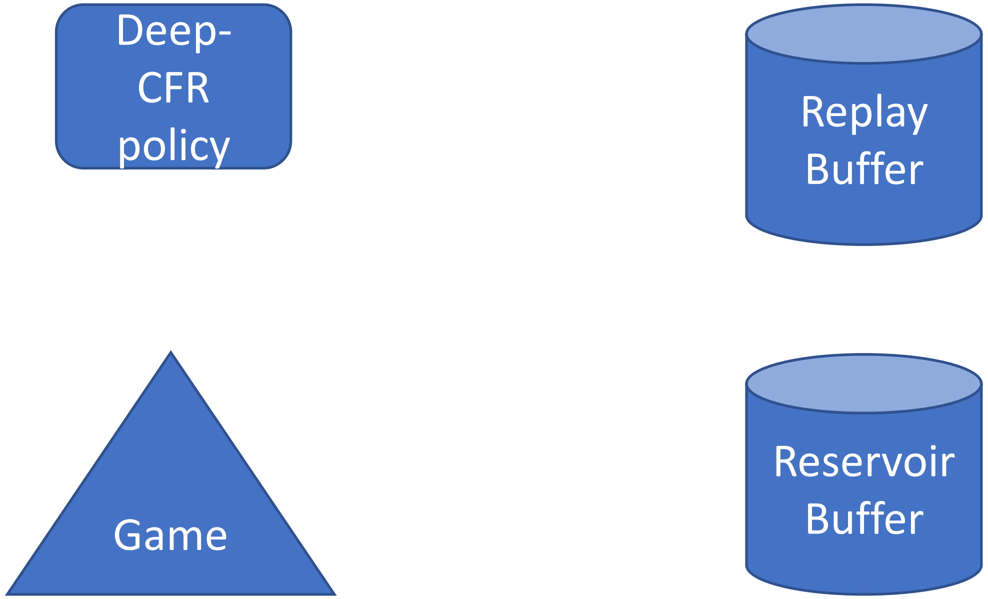
Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling

NFSP



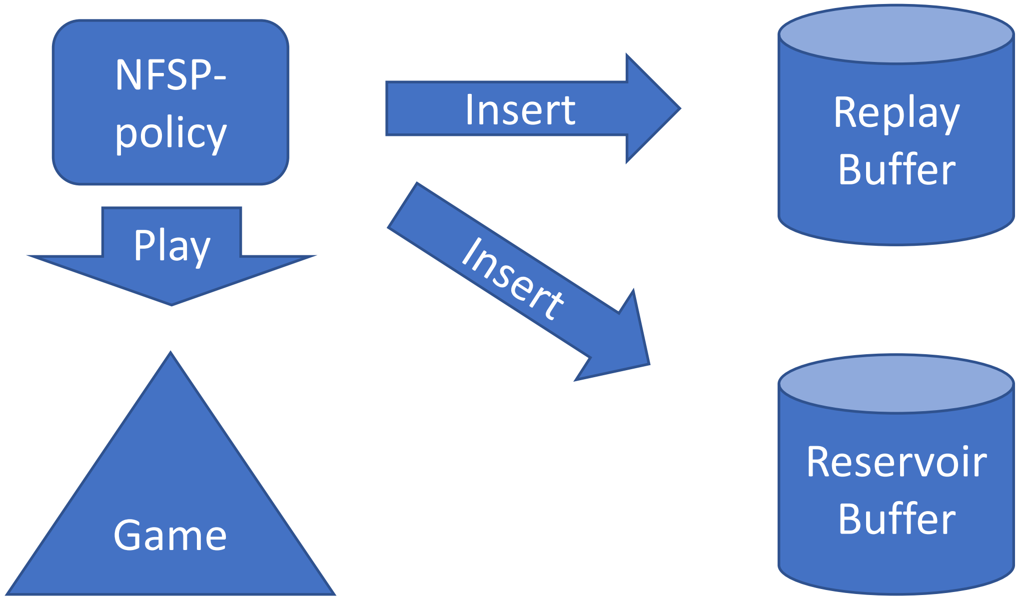
Deep-CFR



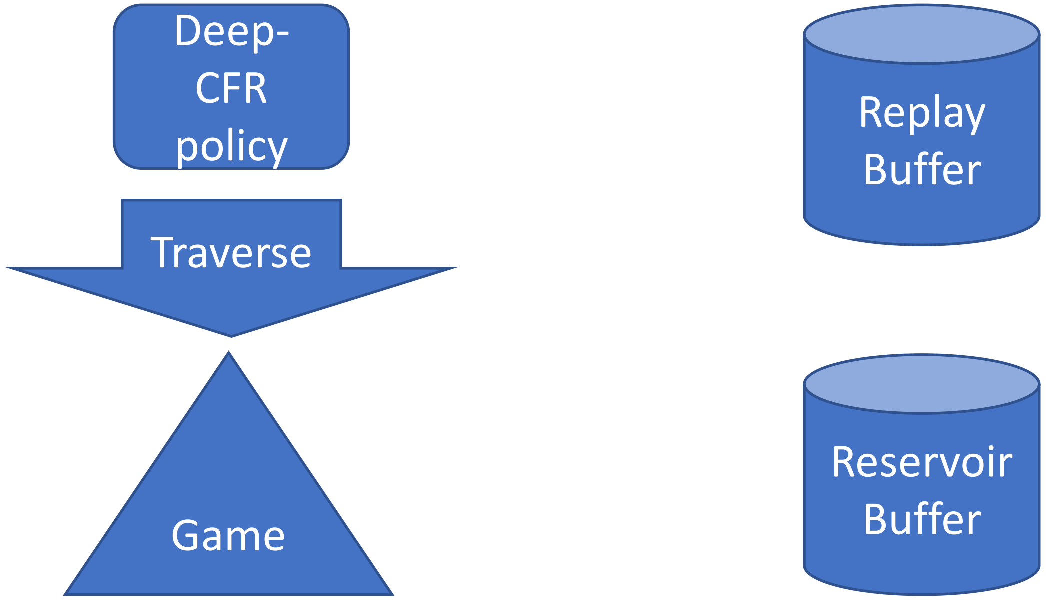
Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling

NFSP



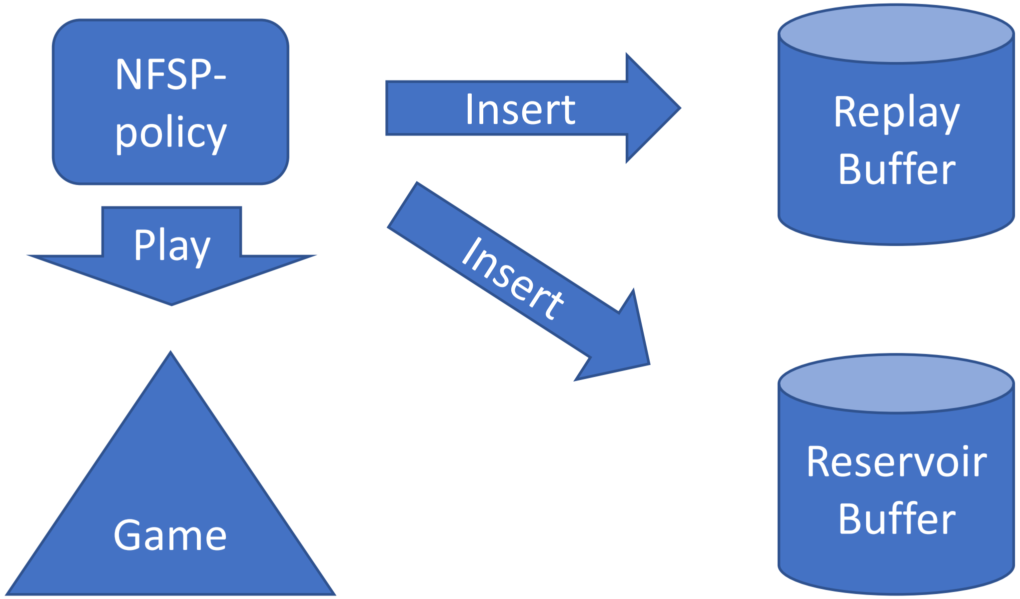
Deep-CFR



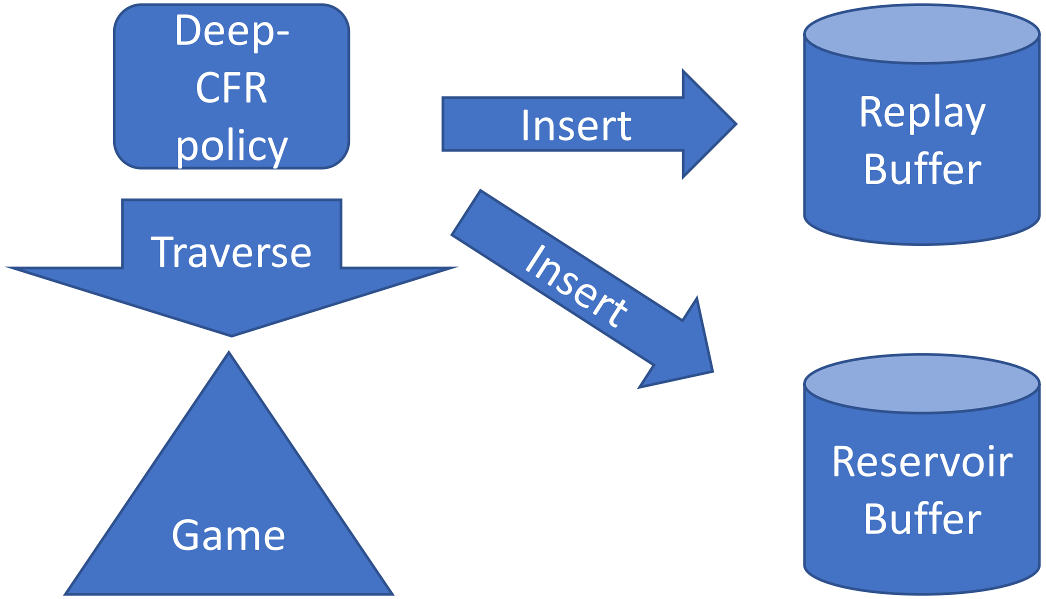
Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling

NFSP



Deep-CFR



Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling
  - Average strategy computation

Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling
  - Average strategy computation

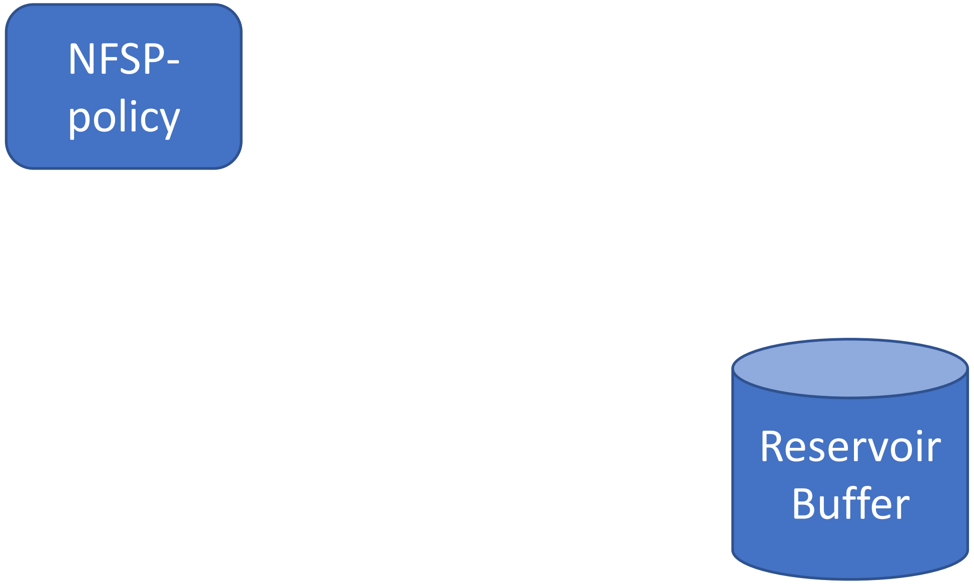
NFSP

Deep-CFR

Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling
  - Average strategy computation

NFSP

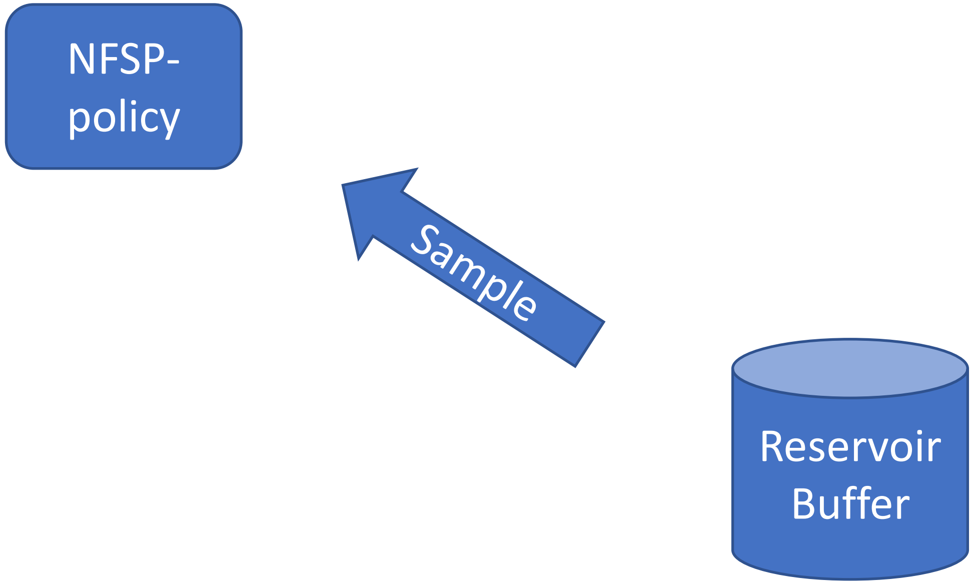


Deep-CFR

Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling
  - Average strategy computation

NFSP

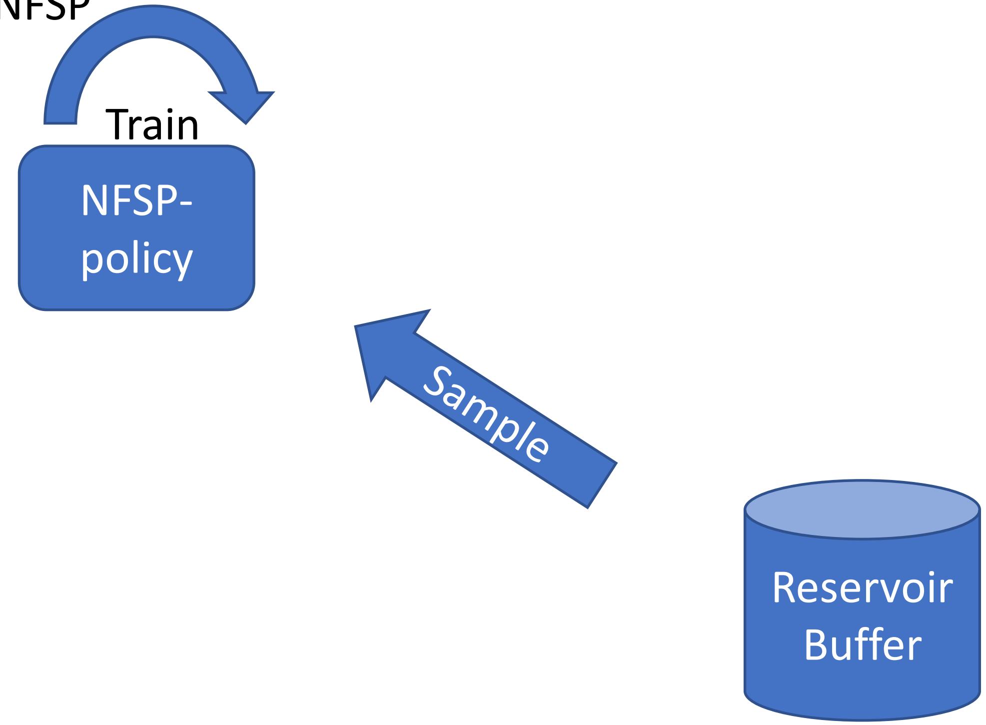


Deep-CFR

Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling
  - Average strategy computation

NFSP

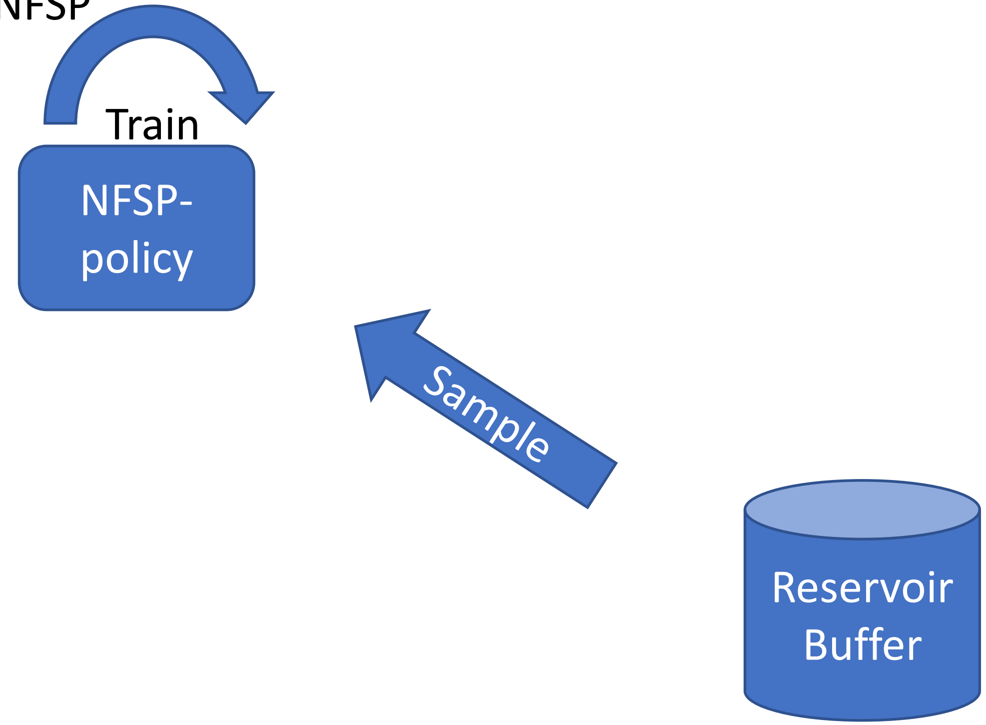


Deep-CFR

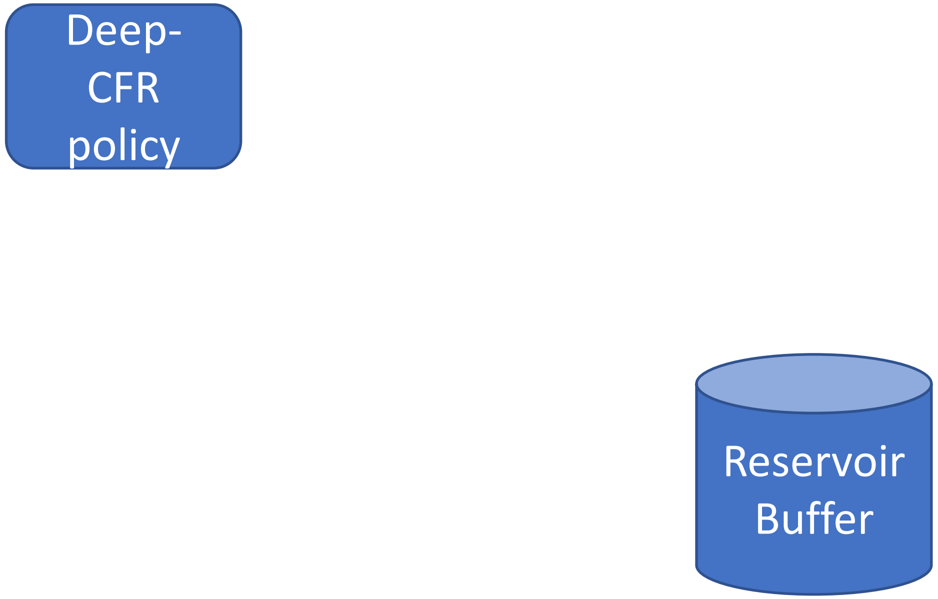
Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling
  - Average strategy computation

NFSP



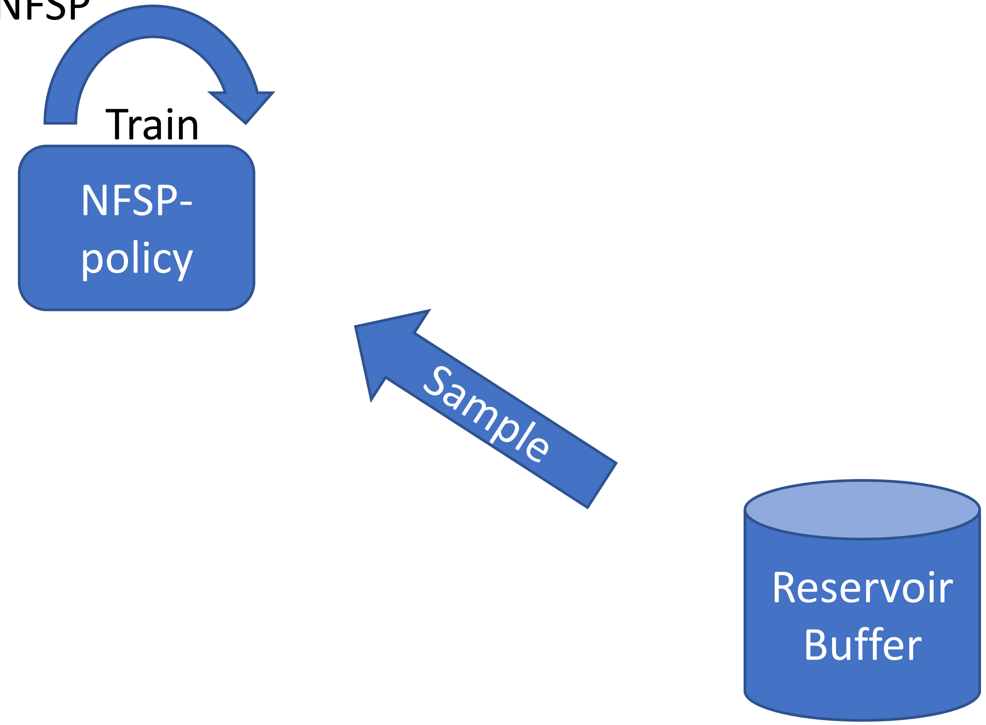
Deep-CFR



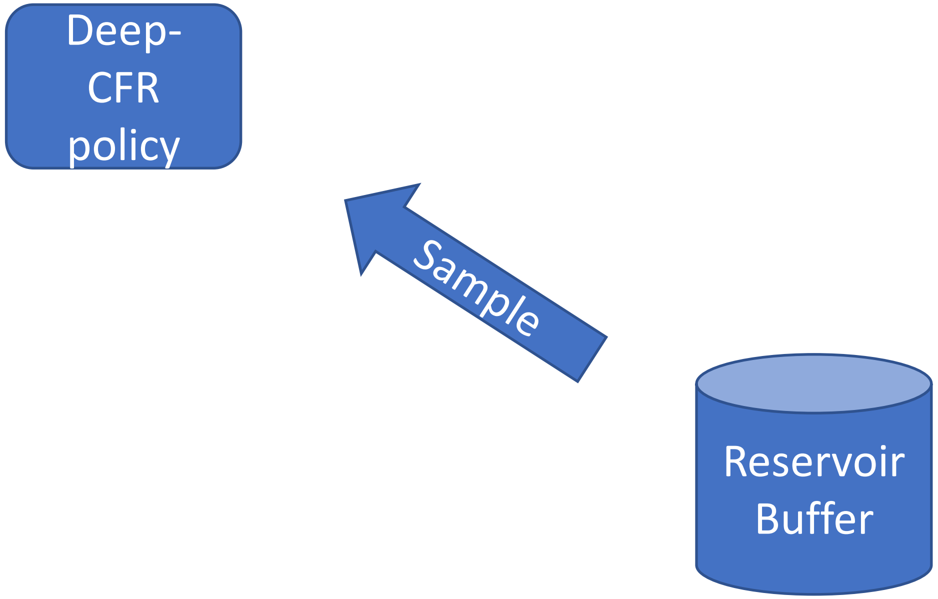
Decoupling the problem

- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling
  - Average strategy computation

NFSP

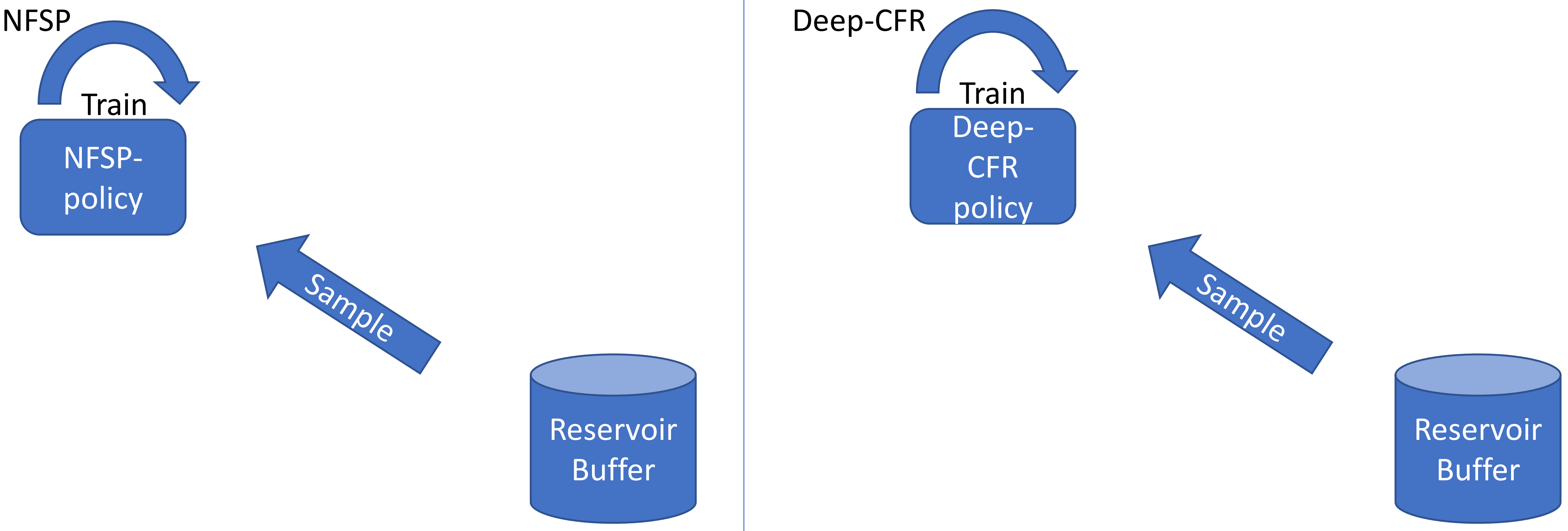


Deep-CFR



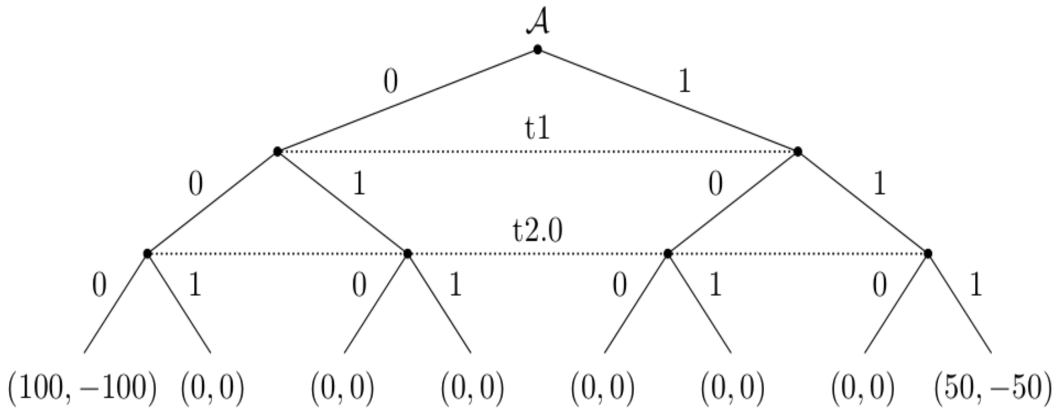
Decoupling the problem

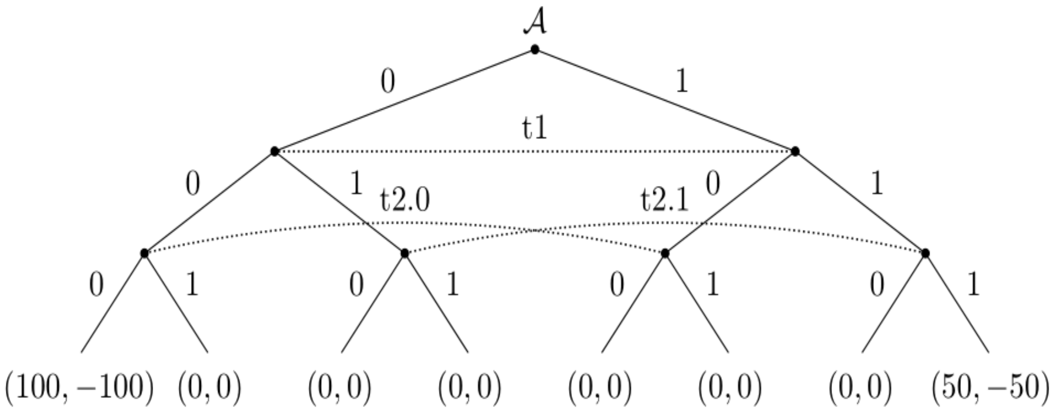
- Consider NFSP and Deep-CFR: we can highlight two main aspects of the algorithm:
  - Trajectory sampling
  - Average strategy computation



Trajectory sampling (1)

- **Perfect-recall refinement**: given a generic game, obtain the equivalent formulation of the game in which player $\mathcal{T}$ (team) has perfect recall.
- Example (coordination game):





Trajectory sampling (2)

Trajectory sampling (2)

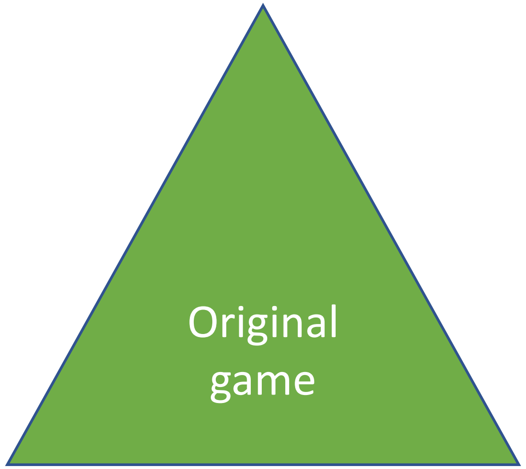
- Mix the ML paradigm of **centralized training with decentralized execution** with the notion of perfect **recall refinement:**

Trajectory sampling (2)

- Mix the ML paradigm of **centralized training with decentralized execution** with the notion of perfect **recall refinement:**
  1. Obtain the relaxed (refined) version of the game,

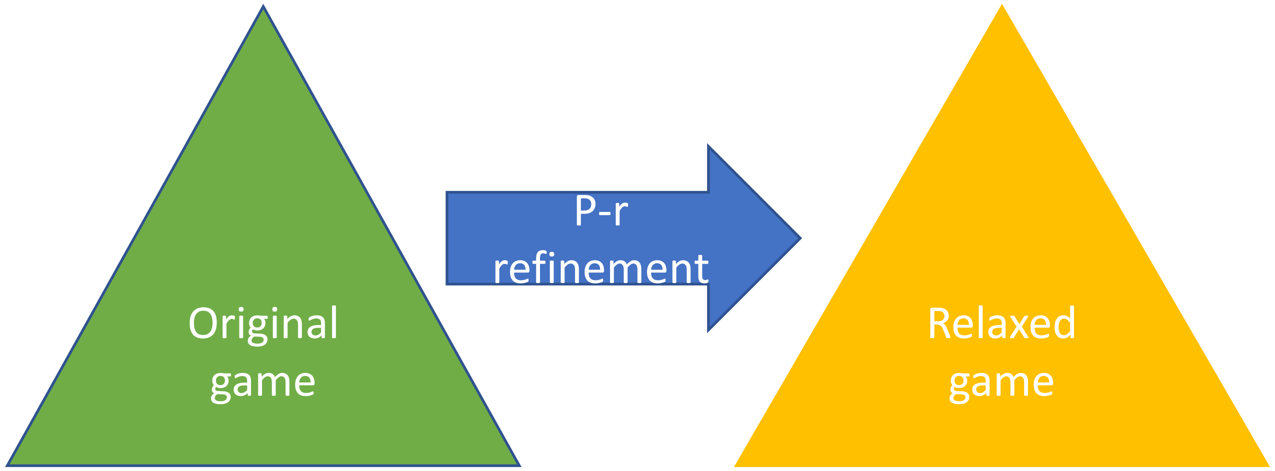
Trajectory sampling (2)

- Mix the ML paradigm of **centralized training with decentralized execution** with the notion of perfect **recall refinement:**
    1. Obtain the relaxed (refined) version of the game,



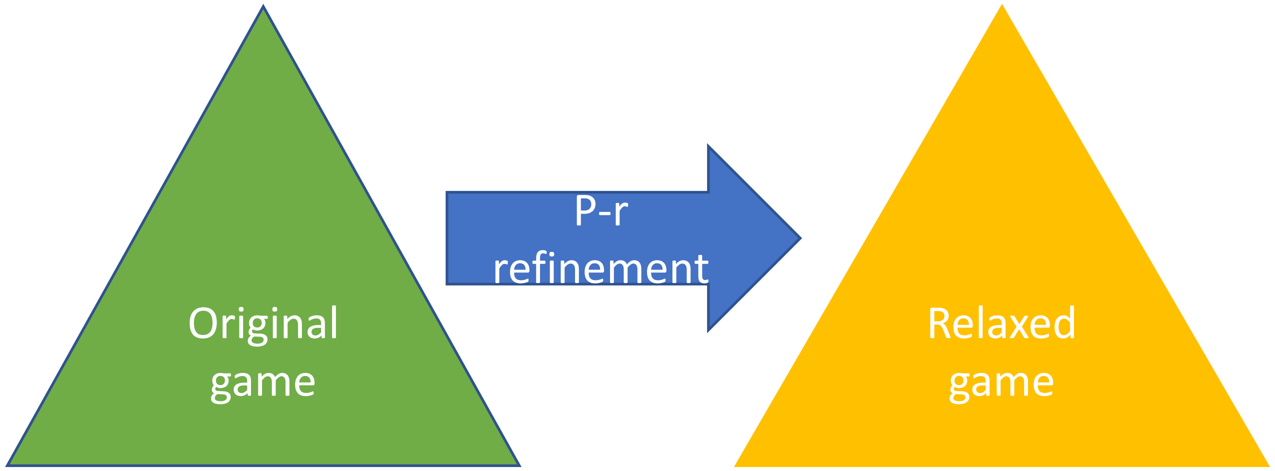
Trajectory sampling (2)

- Mix the ML paradigm of **centralized training with decentralized execution** with the notion of perfect **recall refinement:**
    1. Obtain the relaxed (refined) version of the game,



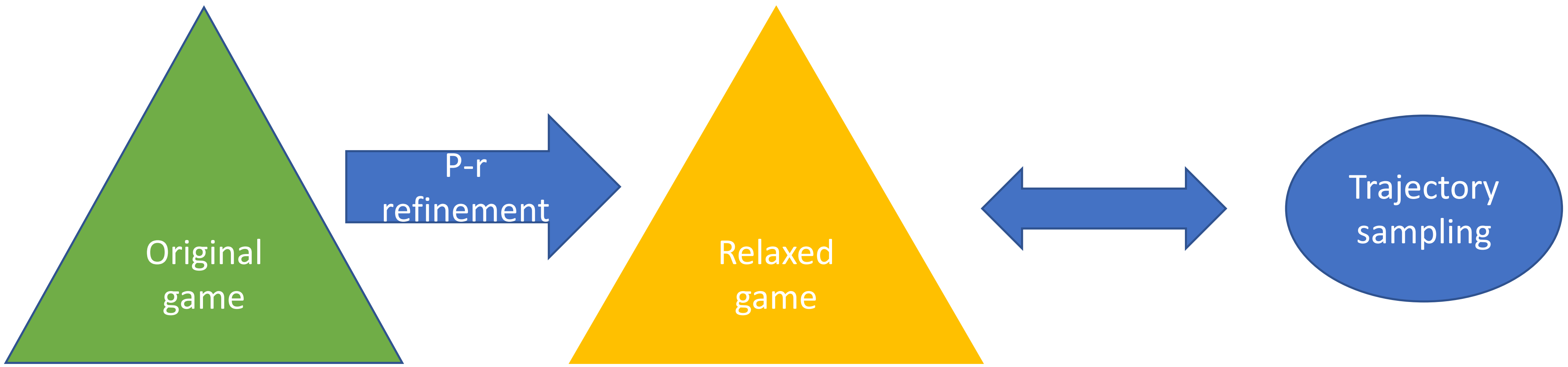
Trajectory sampling (2)

- Mix the ML paradigm of **centralized training with decentralized execution** with the notion of perfect **recall refinement:**
  1. Obtain the relaxed (refined) version of the game,
  2. Do trajectory sampling on the relaxed version of the game,



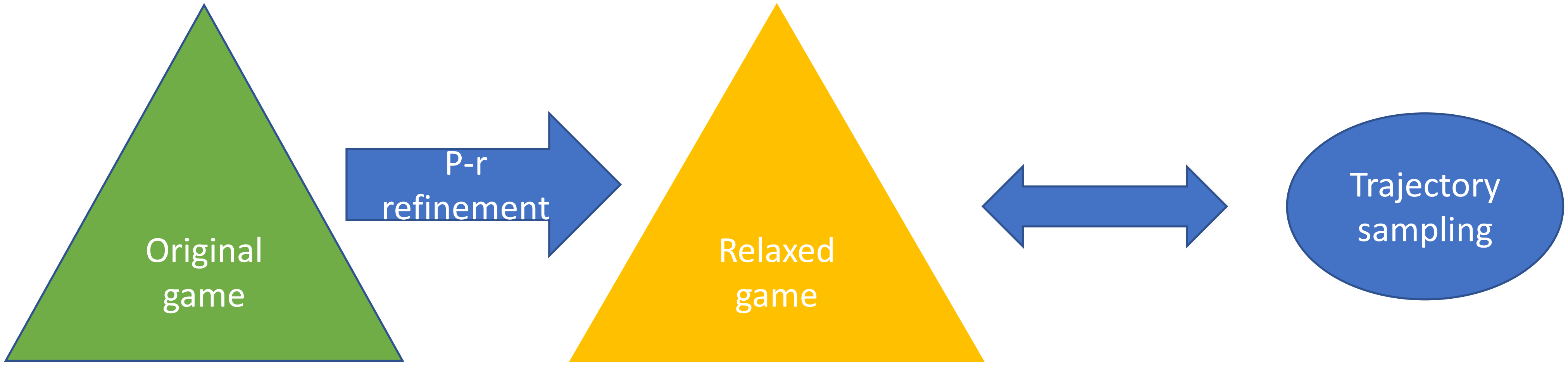
Trajectory sampling (2)

- Mix the ML paradigm of **centralized training with decentralized execution** with the notion of perfect **recall refinement:**
    1. Obtain the relaxed (refined) version of the game,
    2. Do trajectory sampling on the relaxed version of the game,



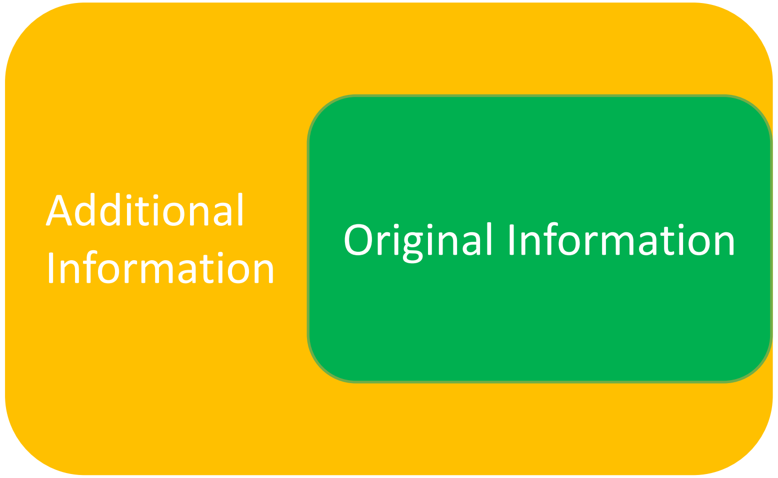
Trajectory sampling (2)

- Mix the ML paradigm of **centralized training with decentralized execution** with the notion of perfect **recall refinement:**
  1. Obtain the relaxed (refined) version of the game,
  2. Do trajectory sampling on the relaxed version of the game,
  3. Purge from additional information when inserting in reservoir buffer



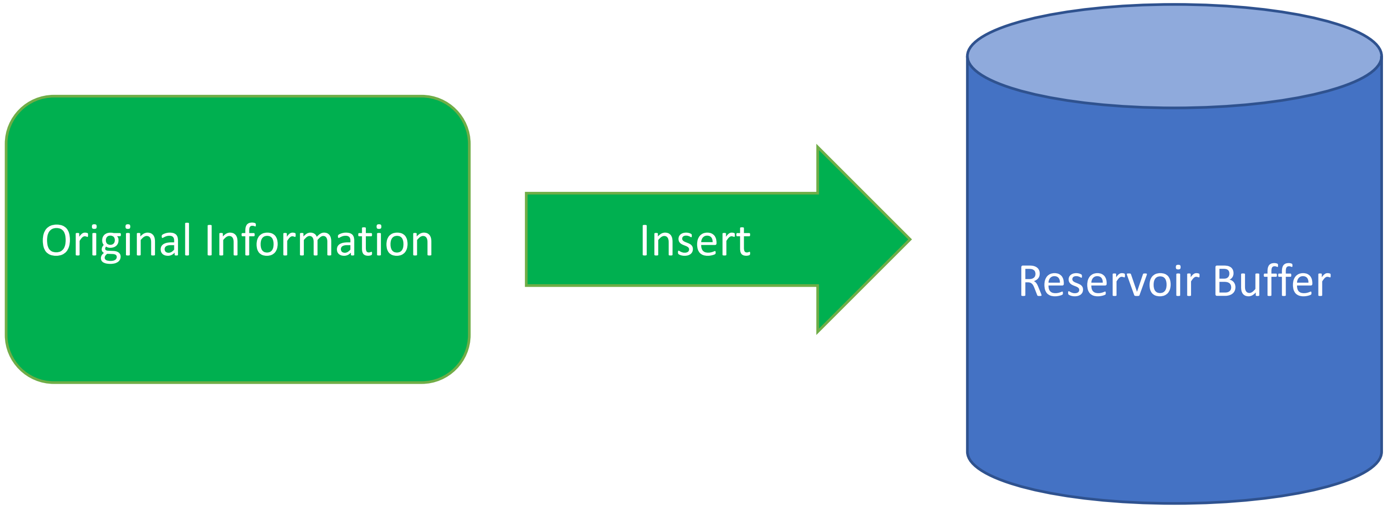
Trajectory sampling (2)

- Mix the ML paradigm of **centralized training with decentralized execution** with the notion of perfect **recall refinement:**
  1. Obtain the relaxed (refined) version of the game,
  2. Do trajectory sampling on the relaxed version of the game,
  3. Purge from additional information when inserting in reservoir buffer



Trajectory sampling (2)

- Mix the ML paradigm of **centralized training with decentralized execution** with the notion of perfect **recall refinement:**
  1. Obtain the relaxed (refined) version of the game,
  2. Do trajectory sampling on the relaxed version of the game,
  3. Purge from additional information when inserting in reservoir buffer

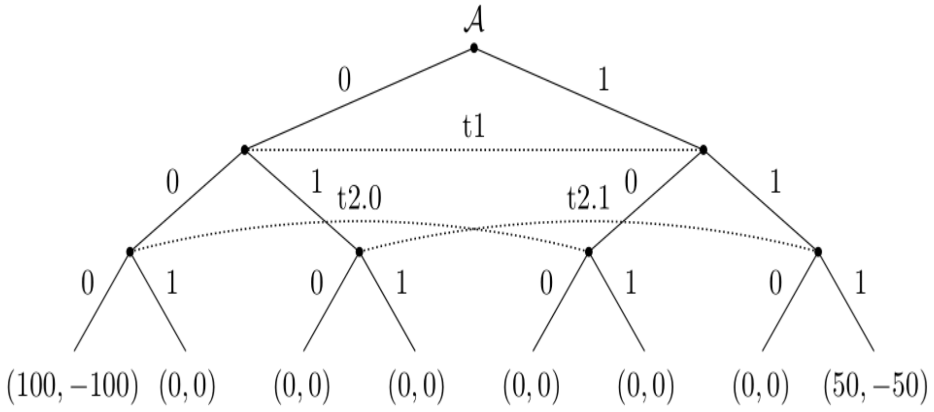


Trajectory sampling (3)

- Running example (coordination game):

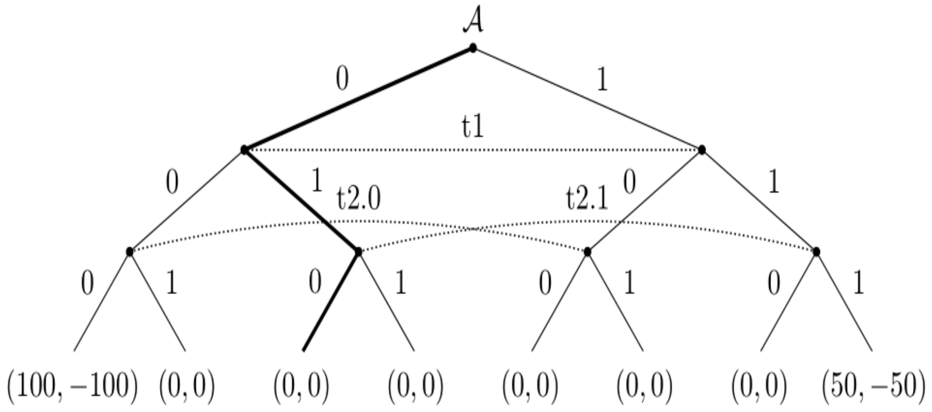
Trajectory sampling (3)

- Running example (coordination game):



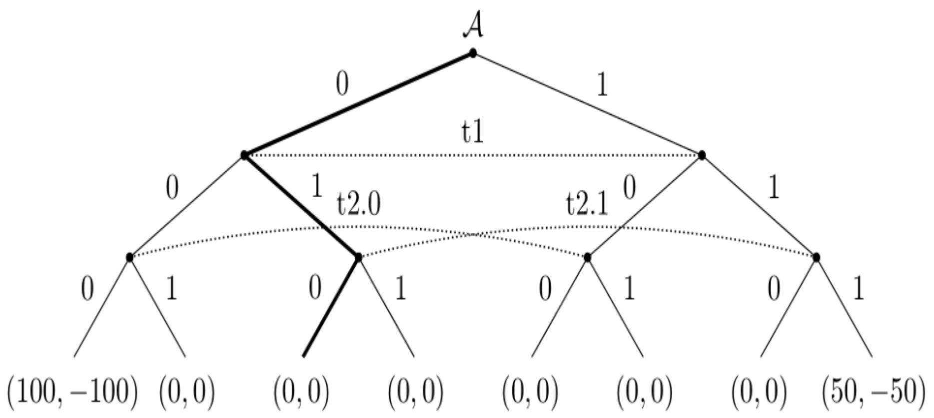
Trajectory sampling (3)

- Running example (coordination game):



Trajectory sampling (3)

- Running example (coordination game):

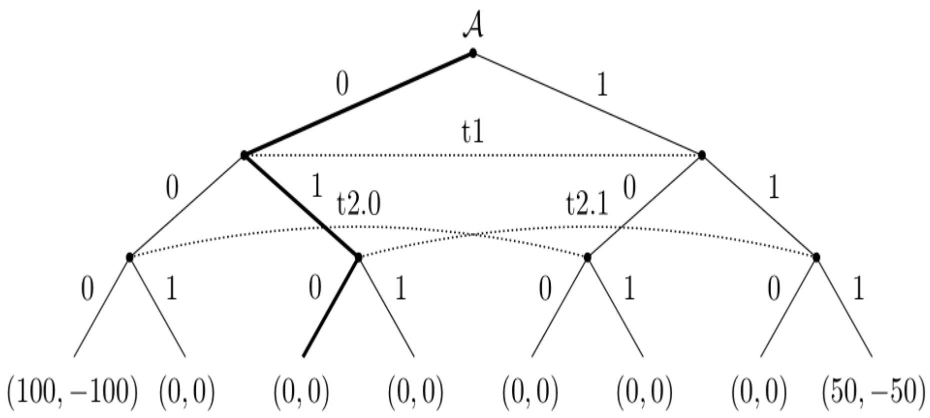


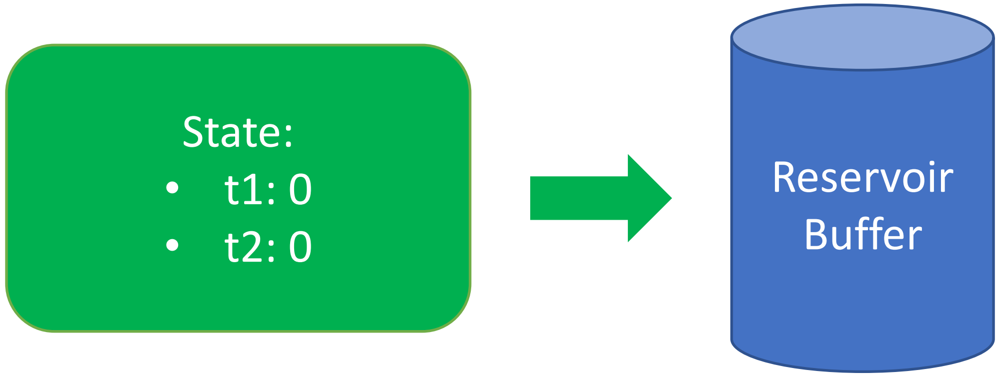
Observed actions:
- t1: None
- t2: 1

State:
- t1: 0
- t2: 0

Trajectory sampling (3)

- Running example (coordination game):





Average strategy computation (1)

- Strategy representation:
  - **Problem**: the space of joint strategies grows exponentially with the number of team players (high spatial complexity).
  - **Solution**: compute average strategies in a decentralized manner.
- Expressiveness of strategy space:
  - **Problem**: recall that decentralized behavioral policies do not have enough expressiveness to capture correlation among agents.
  - **Solution**: employ a signaling scheme to extend the expressive power of the set of policies.

Average strategy computation (2)

- Signal Mediated Strategies (SIMS):

Average strategy computation (2)

- Signal Mediated Strategies (SIMS):

$\pi_{1,\phi_1}(a_1|o_1, z)$

$\pi_{2,\phi_2}(a_2|o_2, z)$

Average strategy computation (2)

- Signal Mediated Strategies (SIMS):

Buffer
$(\boldsymbol{o}, t)$

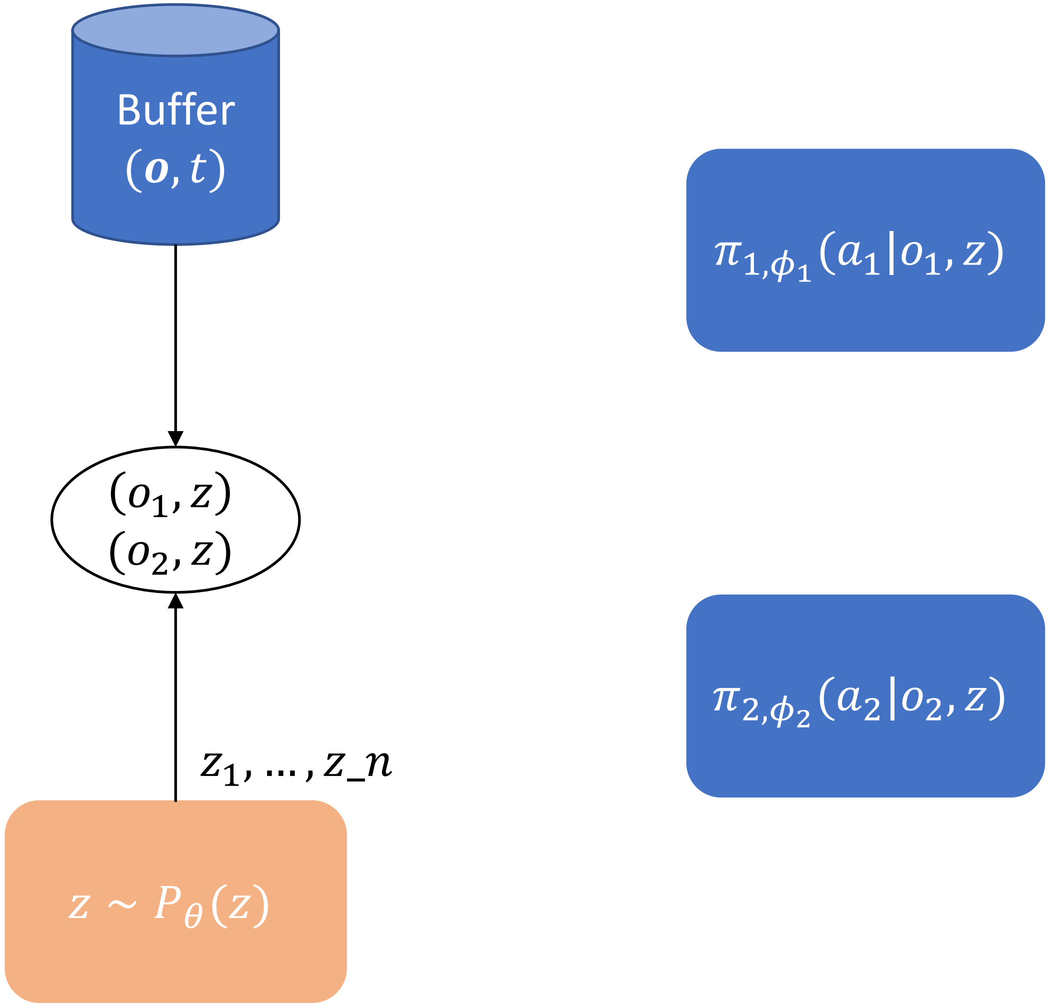
$\pi_{1,\phi_1}(a_1|o_1, z)$

$\pi_{2,\phi_2}(a_2|o_2, z)$

$z \sim P_\theta(z)$

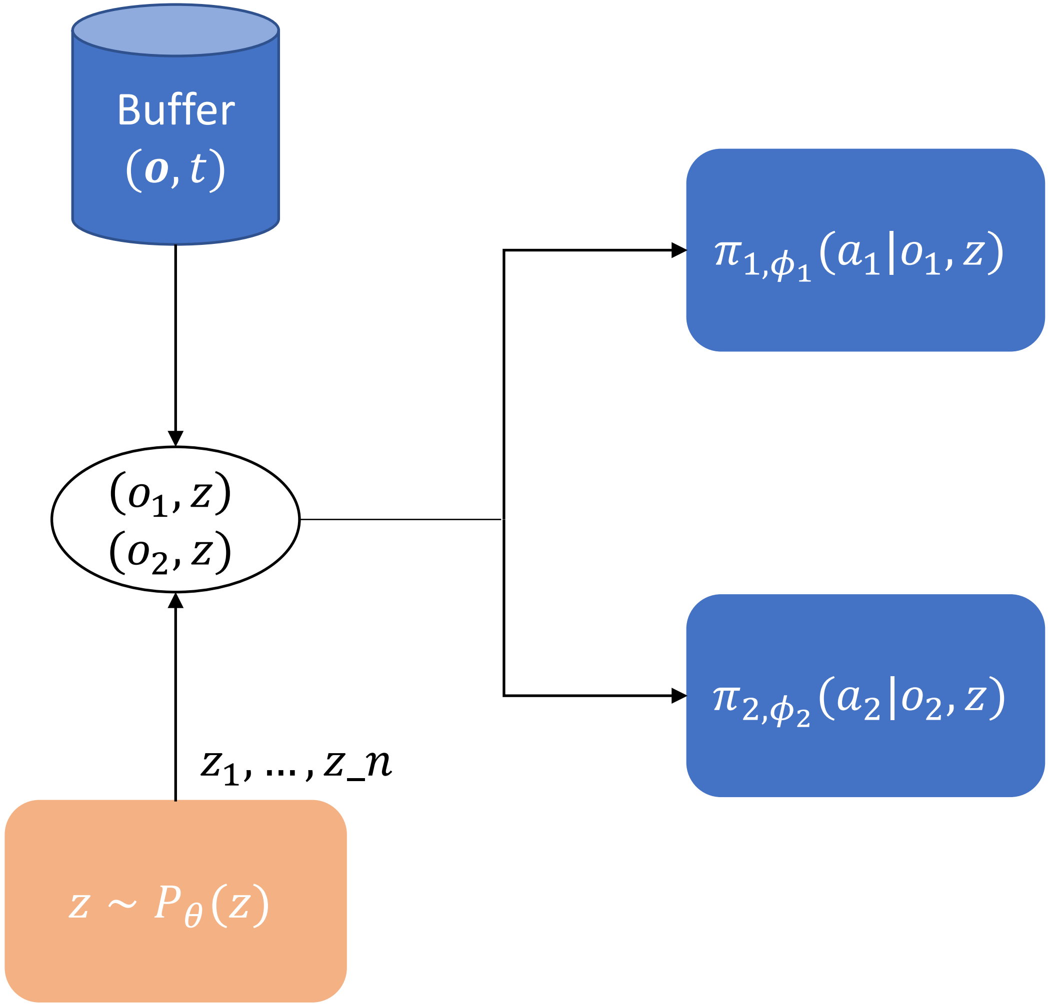
Average strategy computation (2)

- Signal Mediated Strategies (SIMS):



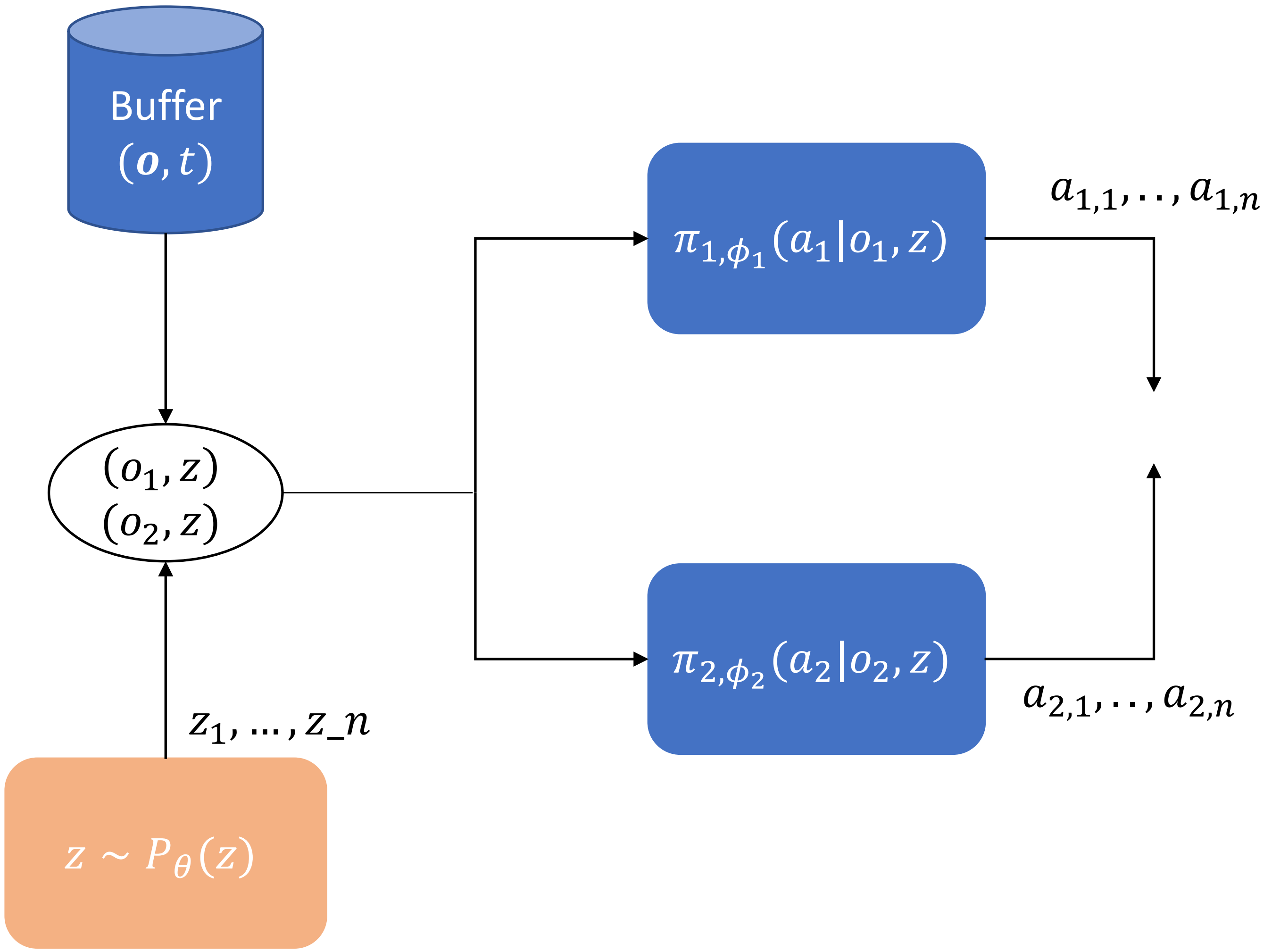
Average strategy computation (2)

- Signal Mediated Strategies (SIMS):



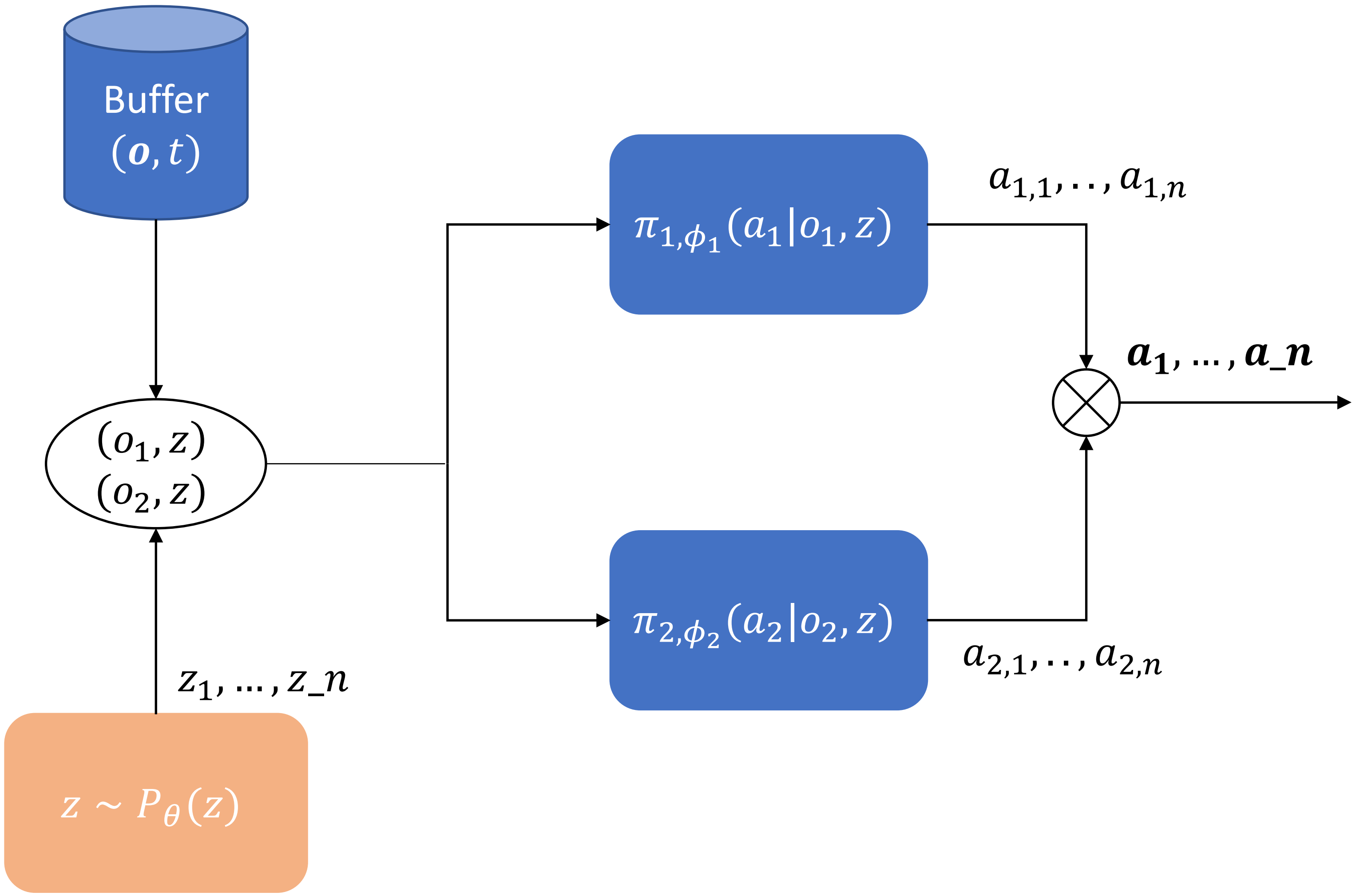
Average strategy computation (2)

- Signal Mediated Strategies (SIMS):



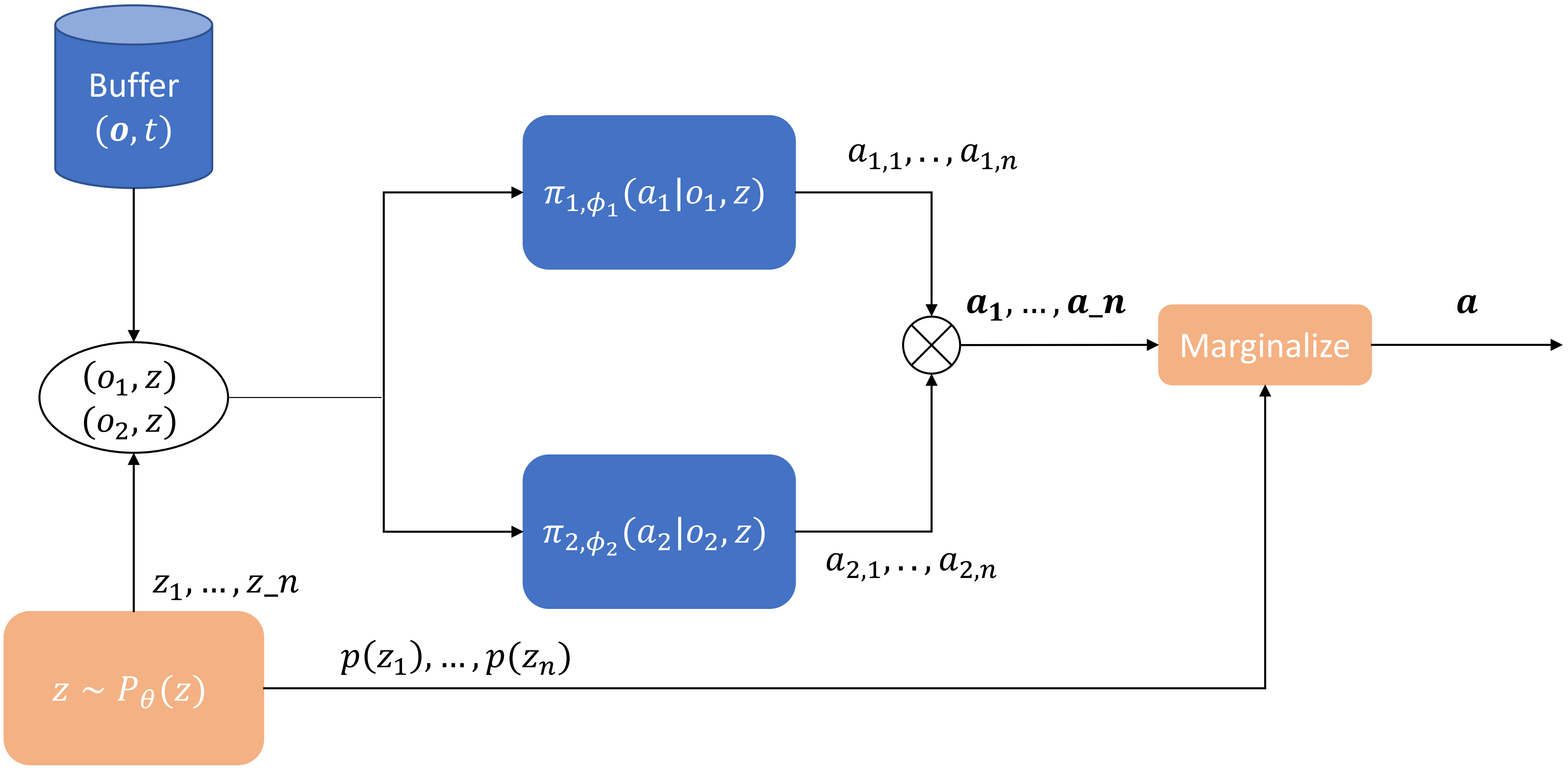
Average strategy computation (2)

- Signal Mediated Strategies (SIMS):



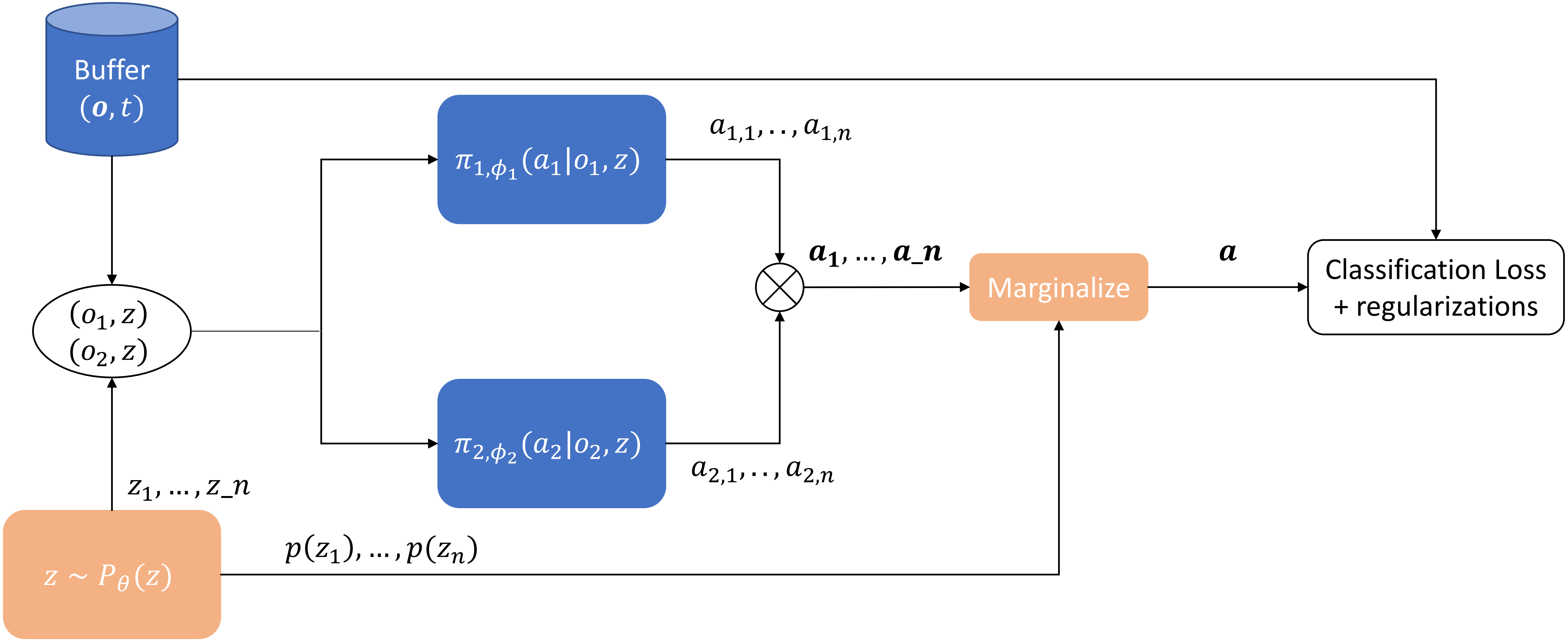
Average strategy computation (2)

- Signal Mediated Strategies (SIMS):



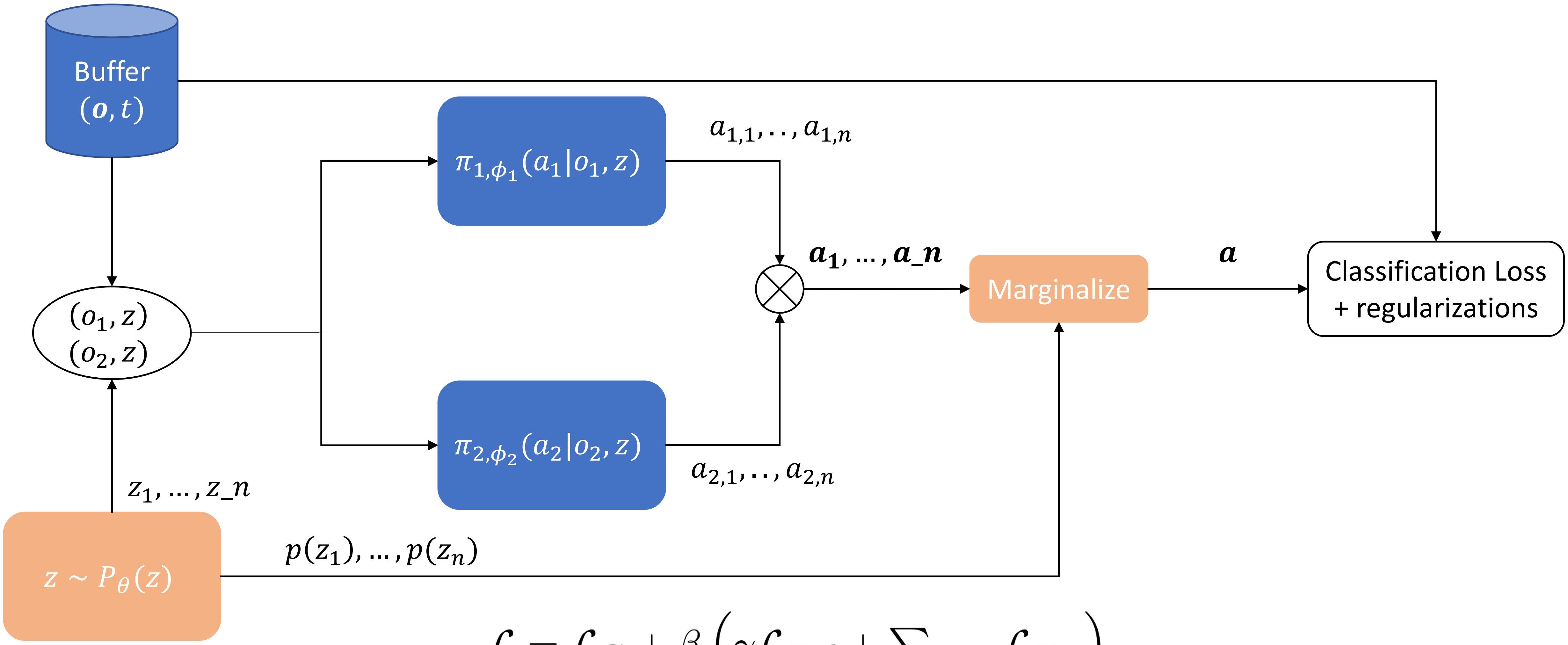
Average strategy computation (2)

- Signal Mediated Strategies (SIMS):



Average strategy computation (2)

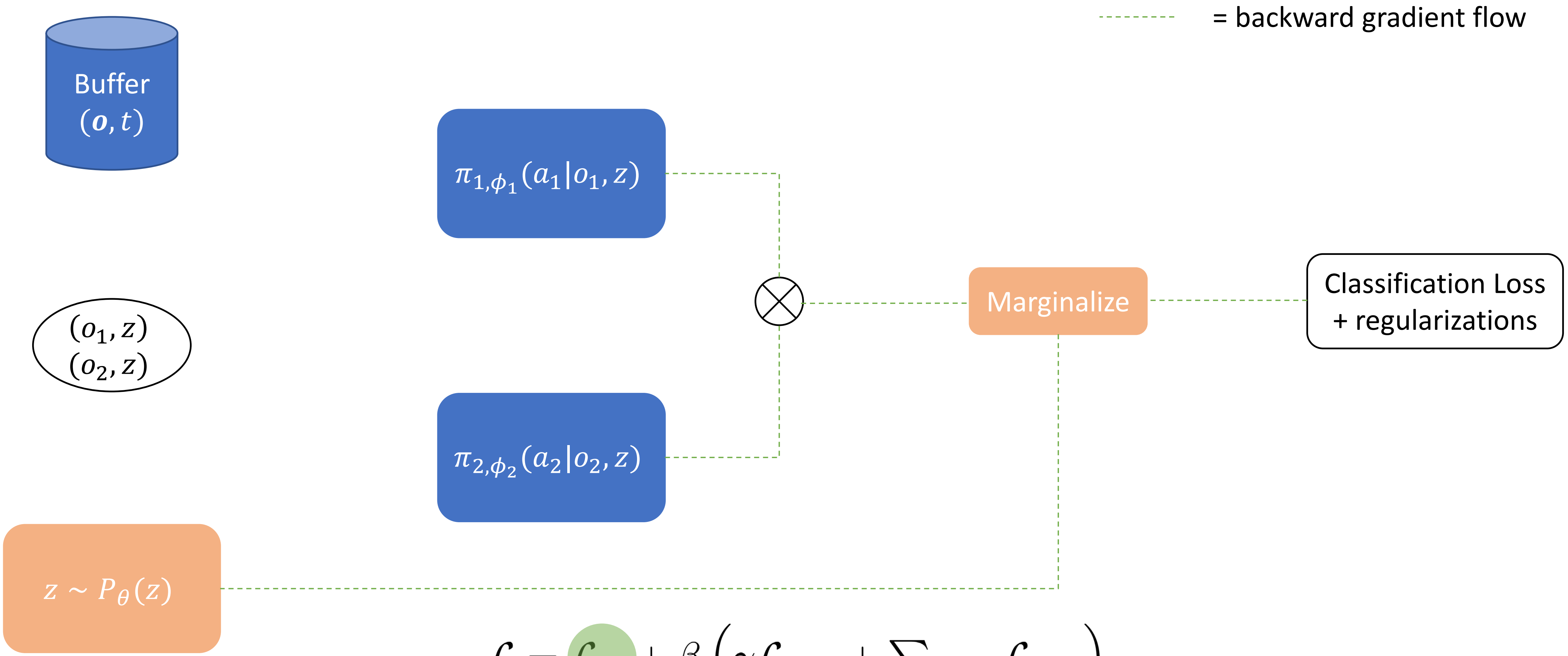
- Signal Mediated Strategies (SIMS):



$$\mathcal{L} = \mathcal{L}_{\mathbf{C}} + \beta\left(\gamma\mathcal{L}_{E,d} + \sum_{k\leq n}\mathcal{L}_{E,s}\right)$$

Average strategy computation (2)

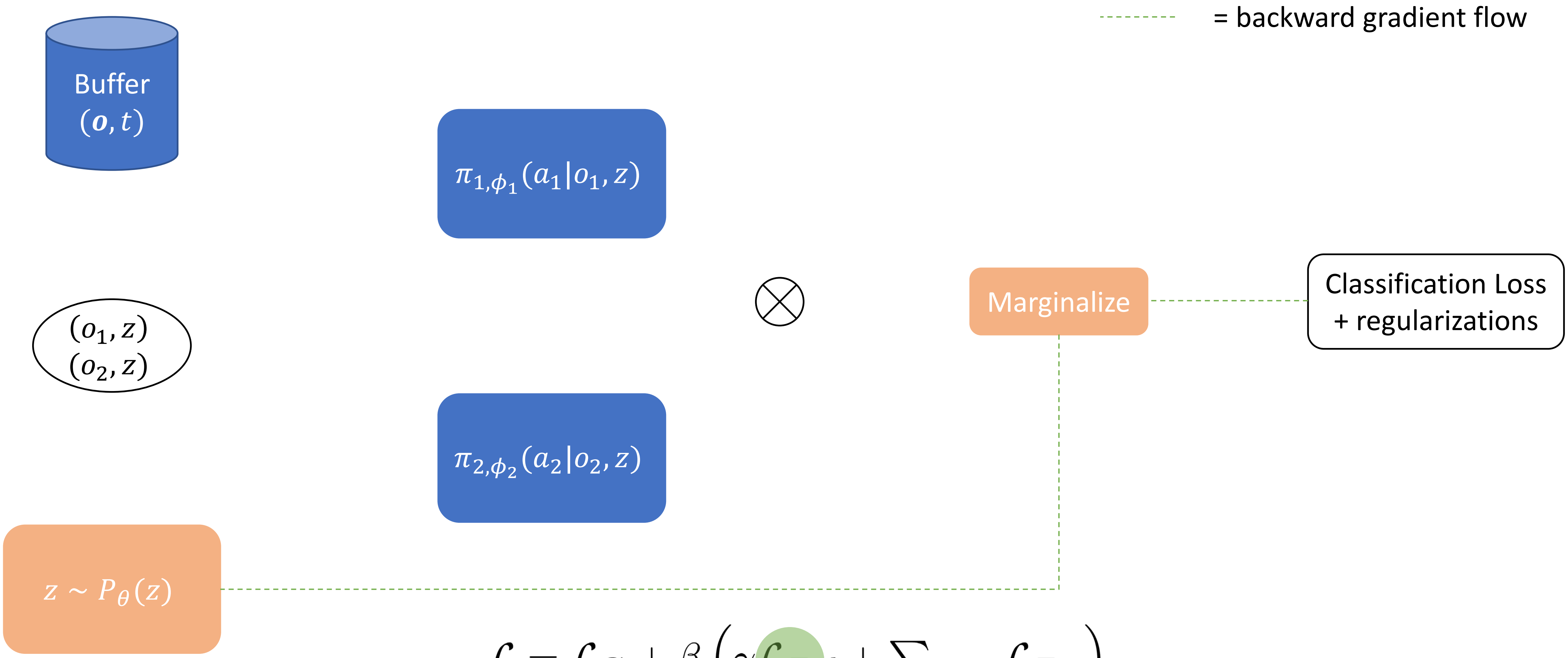
- Signal Mediated Strategies (SIMS):



$$\mathcal{L} = \mathcal{L}_{\mathrm{C}} + \beta\left(\gamma \mathcal{L}_{E,d} + \sum_{k \le n} \mathcal{L}_{E,s}\right)$$

Average strategy computation (2)

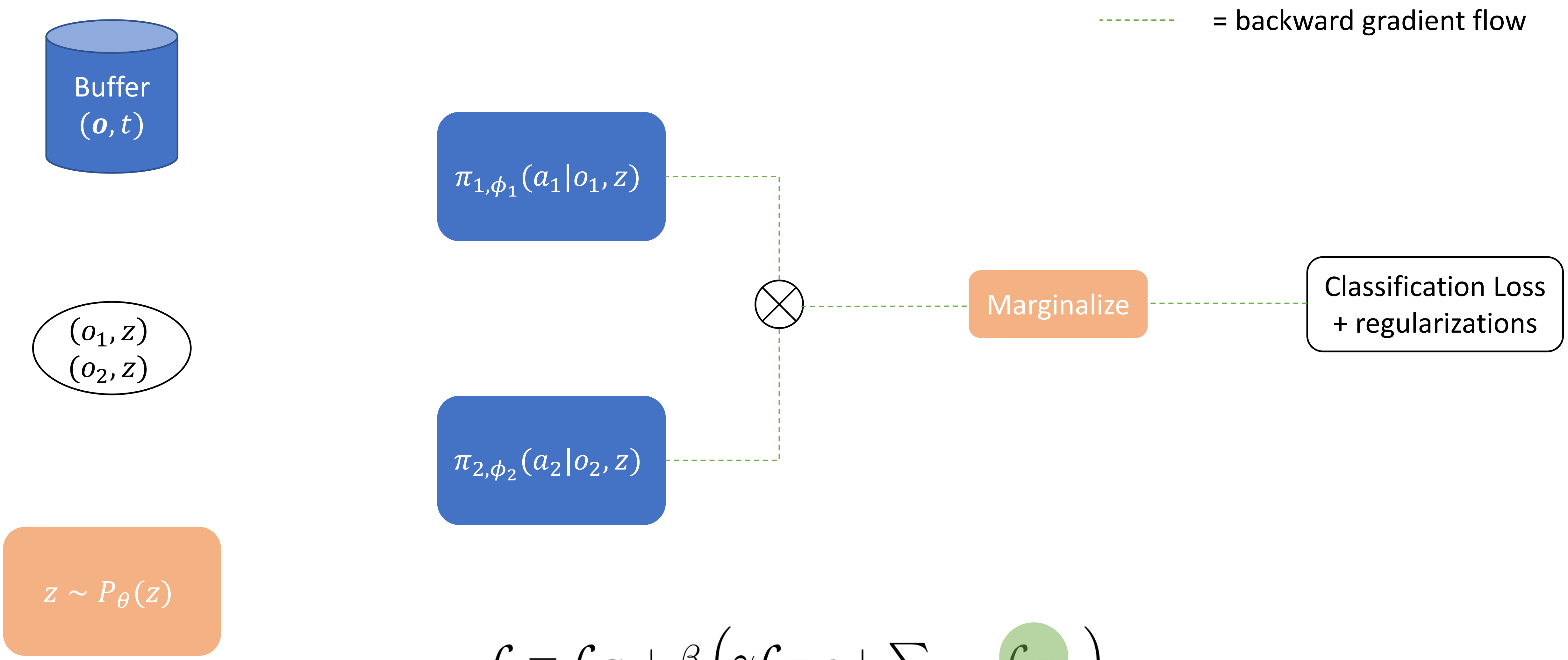
- Signal Mediated Strategies (SIMS):



$$\mathcal{L} = \mathcal{L}_{\mathbf{C}} + \beta \left(\gamma \mathcal{L}_{E,d} + \sum_{k \le n} \mathcal{L}_{E,s} \right)$$

Average strategy computation (2)

- Signal Mediated Strategies (SIMS):



$$\mathcal{L} = \mathcal{L}_{\mathbf{C}} + \beta\left(\gamma\mathcal{L}_{E,d} + \sum_{k\leq n}\mathcal{L}_{E,s}\right)$$

Outline

Games considered

- For simplicity focus on multi-stage games.
- Coordination games: variations of the game used as examples during the presentation:
  - Various payoffs.
  - Various lengths of the game tree:
- Goofspiel:
  - Various ranks (number of cards in each suit).

Notation

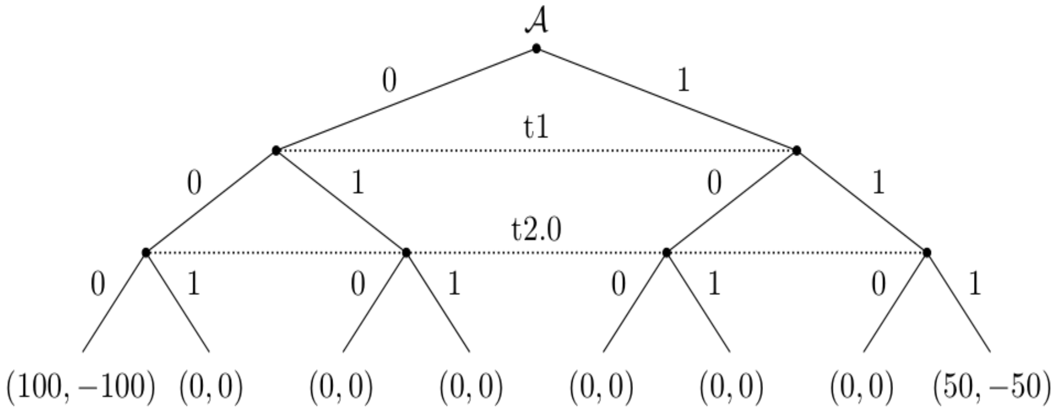
- We will use both original versions of the games and perfect recall refinements.
- For clarity we will denote the perfect recall refinements of the games with the prefix *i.*
- Also the algorithms that run on the perfect recall refinements (e.g. for trajectory sampling will be denoted with the prefix *i*).

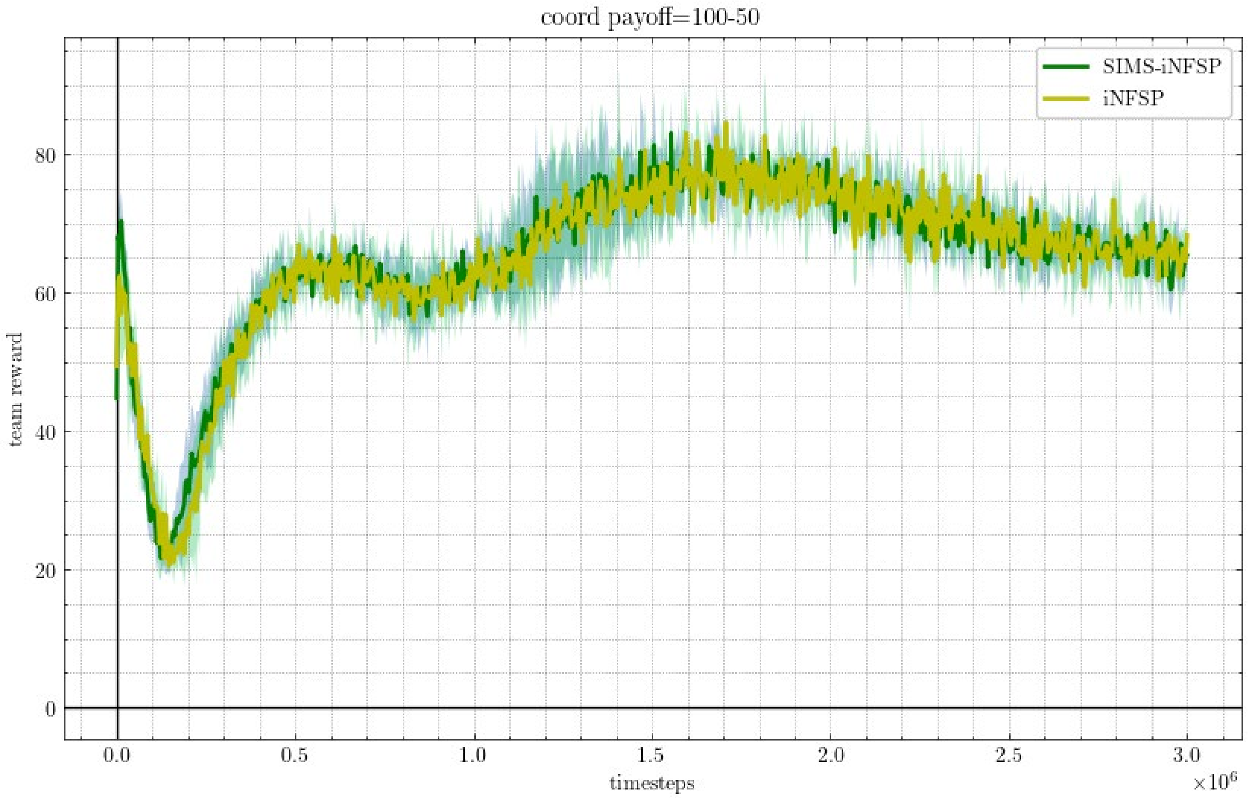
Algorithms tested

- We tested different state-of-the-art RL frameworks:
  - MADDPG, (Lowe et al., 2017),
  - SIC-MADDPG, (Chen et al., 2019),
  - QMIX, (Rashid et al., 2018).
- In order to test SIMS, we test two different algorithms for traectory sampling:
  - *i*-NFSP,
  - *i*-QMIX.

Test 1: Goodness of strategy computation (1)

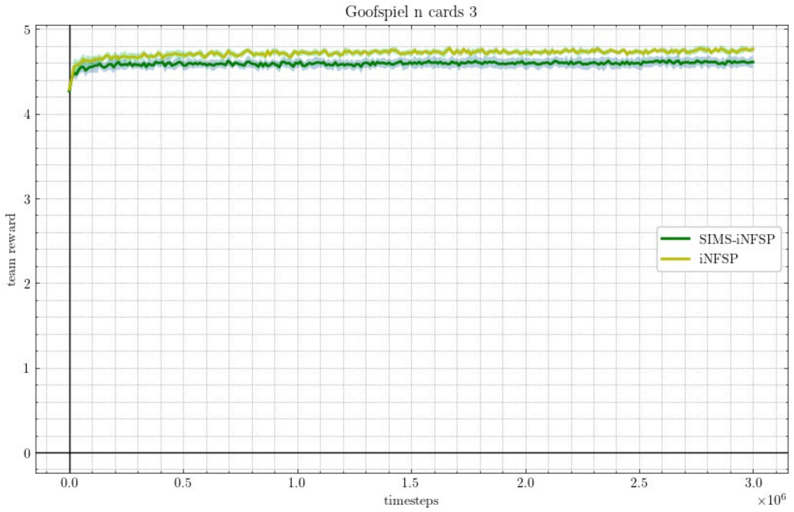
- Coordination game:

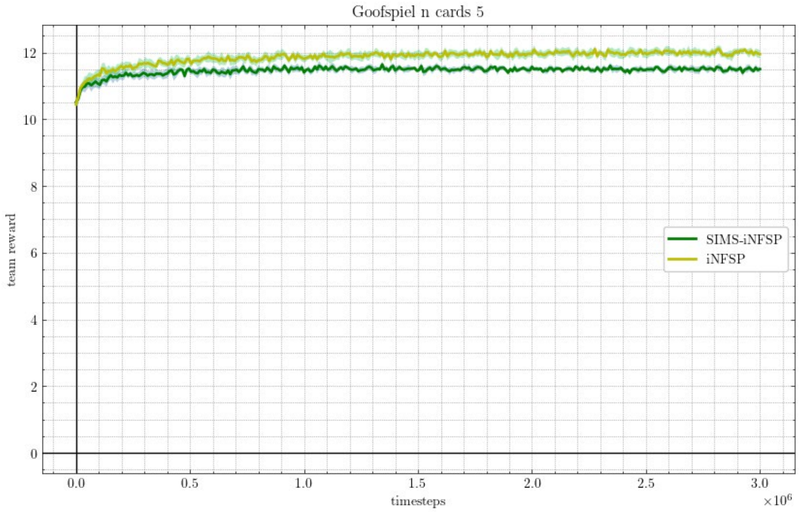


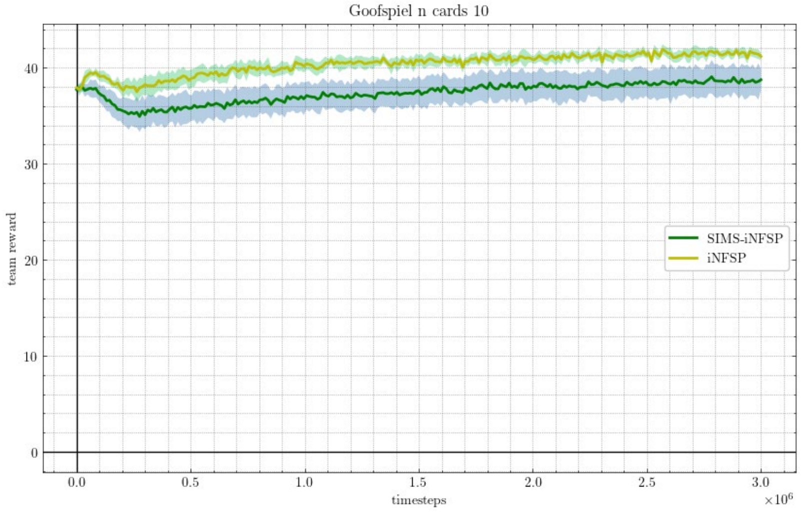


Test 1: Goodness of strategy computation (2)

- Goofspiel:

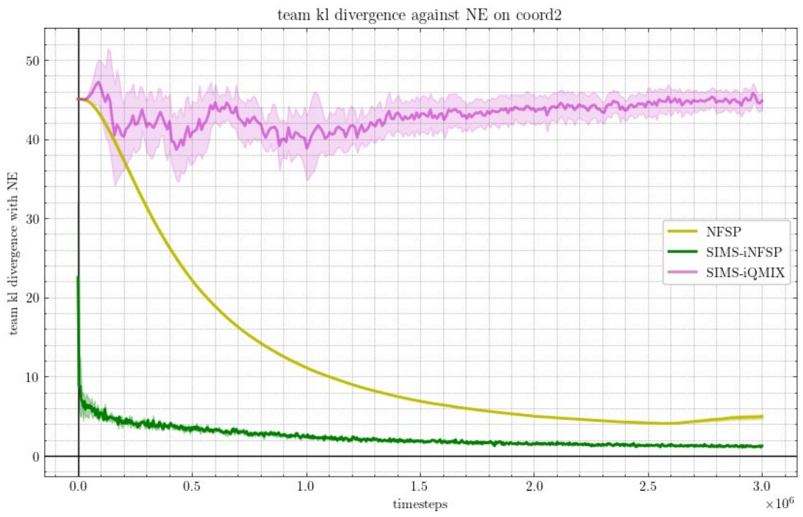


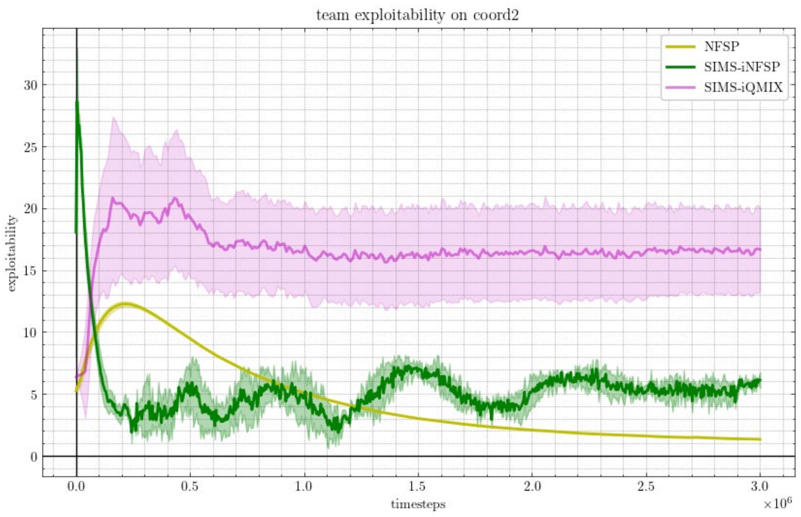




Test 2: Comparison with SOTA frameworks (1)

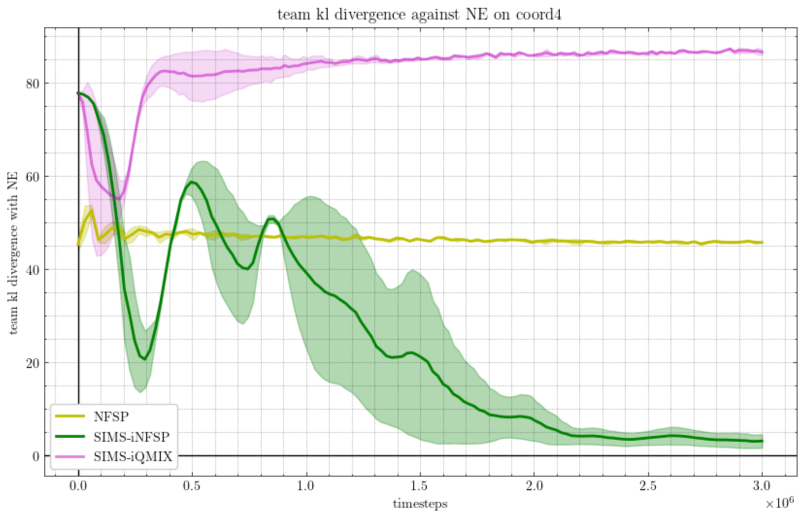
- Coordination game horizon 2:

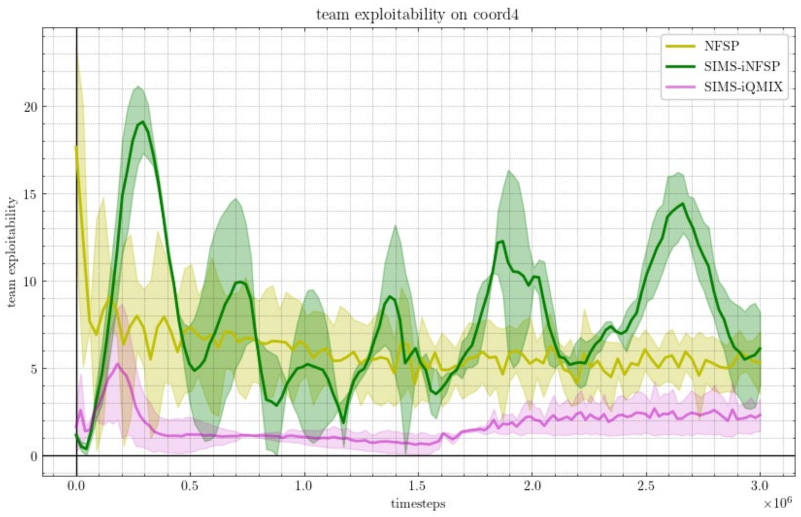


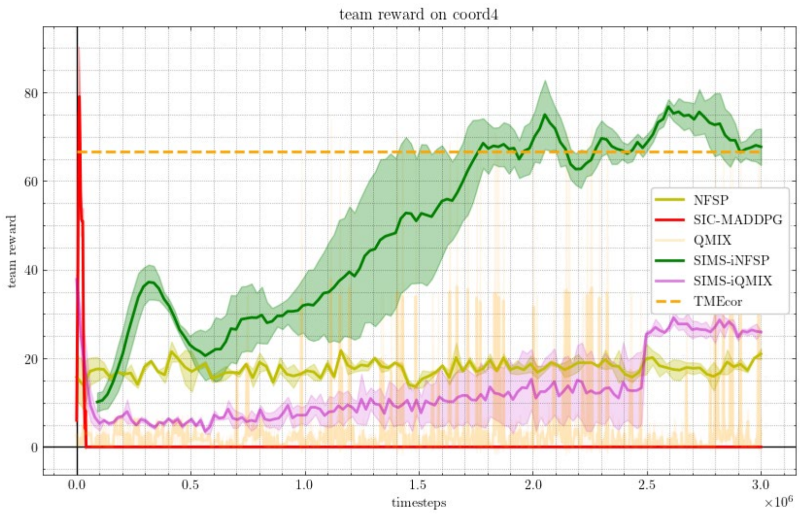


Test 2: Comparison with SOTA frameworks (2)

- Coordination game horizon 4:







Future work

- Study different possibilities for trajectory sampling (e.g. Deep-CFR).
- Analyze the case of general Adversarial Team Games.
- Investigate what happens in cases when the asymmetry of information between team members increases.