Research Project Proposal:

Neural Function Approximation for Adversarial Team Games

Luca Carminati
luca5.carminati@mail.polimi.it
CSE Track

POLITECNICO MILANO 1863

HONOURS PROGRAMME HP-SR in Information Technology

Overview

1. Introduction to Algorithmic Game Theory
2. Main Questions
3. Preliminaries
4. State of the art
5. Project proposal

Overview

1. Introduction to Algorithmic Game Theory
2. Main Questions
3. Preliminaries
4. State of the art
5. Project proposal

Algorithmic Game Theory

*“**Game theory** is the name given to the methodology of using mathematical tools to model and analyse situations of interactive decision making. These are situations involving several decision makers (called players) with different goals, in which the decision of each affects the outcome for all the decision makers.”*

M. Maschler, E. Solan, S. Zamir. “Game Theory”. 2013

- **Algorithmic Game Theory** is the area at the intersection between Game Theory and Computer Science

Games

A **game** is a description of strategic interaction including:

- the *constraints* on the actions that the players can take
- players' interests, expressed as *utilities* for given outcomes of the game

A **solution** is a systematic description of the strategies that may emerge in a family of games, given a specific definition of rationality of the players.

Games

A **game** is a description of strategic interaction including:

- the *constraints* on the actions that the players can take
- players' interests, expressed as *utilities* for given outcomes of the game

A **solution** is a systematic description of the strategies that may emerge in a family of games, given a specific definition of rationality of the players.

Game theory suggests reasonable solutions for classes of games and examines their properties.

Games

A simple example: ***Rock Paper Scissors***

Characteristics of the game:

Games

A simple example: ***Rock Paper Scissors***

Characteristics of the game:

- 2 Players choosing 1 action concurrently

Games

A simple example: ***Rock Paper Scissors***

Characteristics of the game:

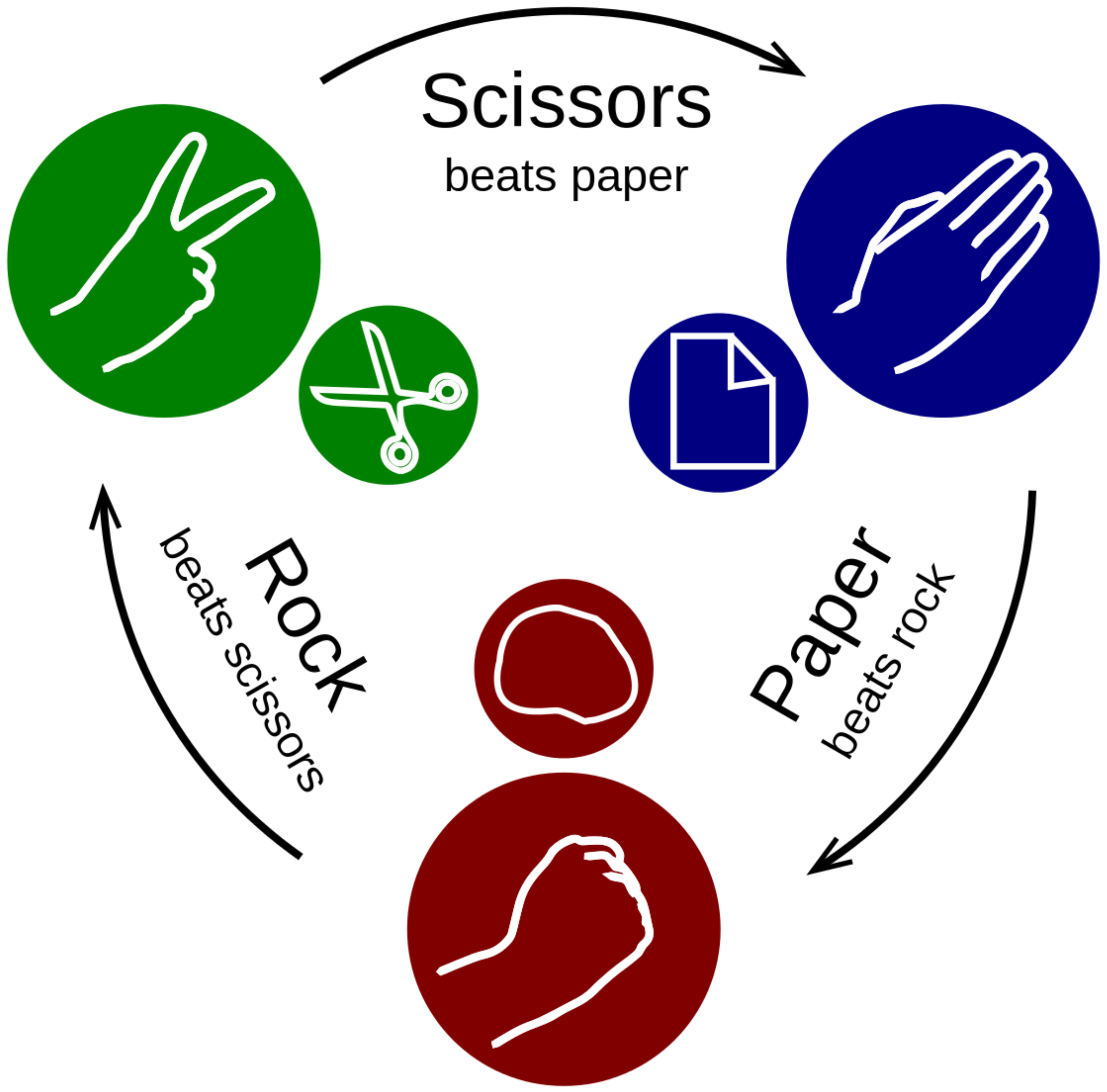
- 2 Players choosing 1 action concurrently
- Utility: *zero-sum game*
  - +1 for winning player
  - 0 in case of draw
  - -1 for losing player

Games

A simple example: ***<u>Rock Paper Scissors</u>***

Characteristics of the game:

- 2 Players choosing 1 action concurrently
- Utility: *zero-sum game*
  - +1 for winning player
  - 0 in case of draw
  - -1 for losing player
- Winning player determined by those rules:

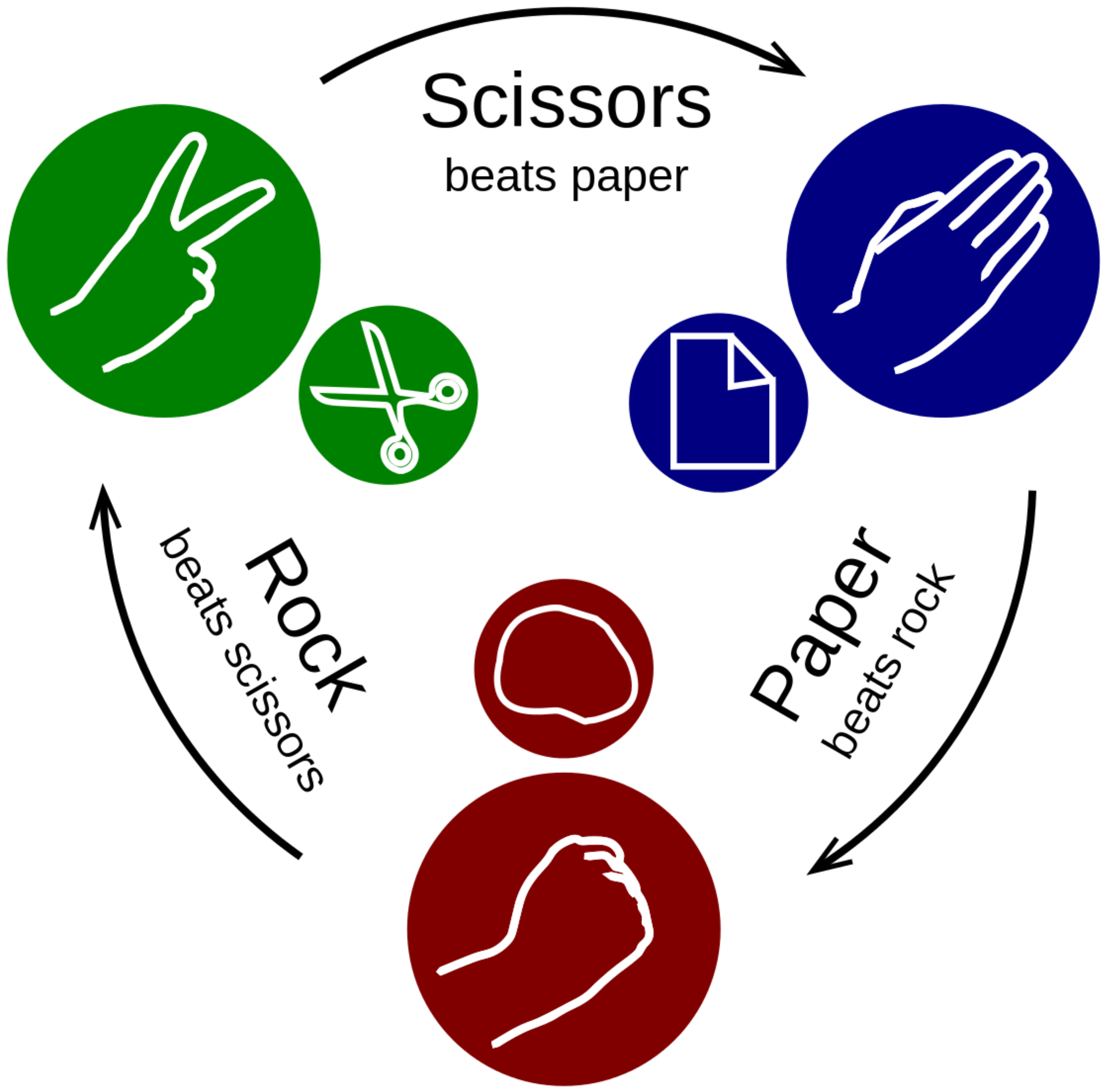


Games

A simple example: ***Rock Paper Scissors***

Characteristics of the game:

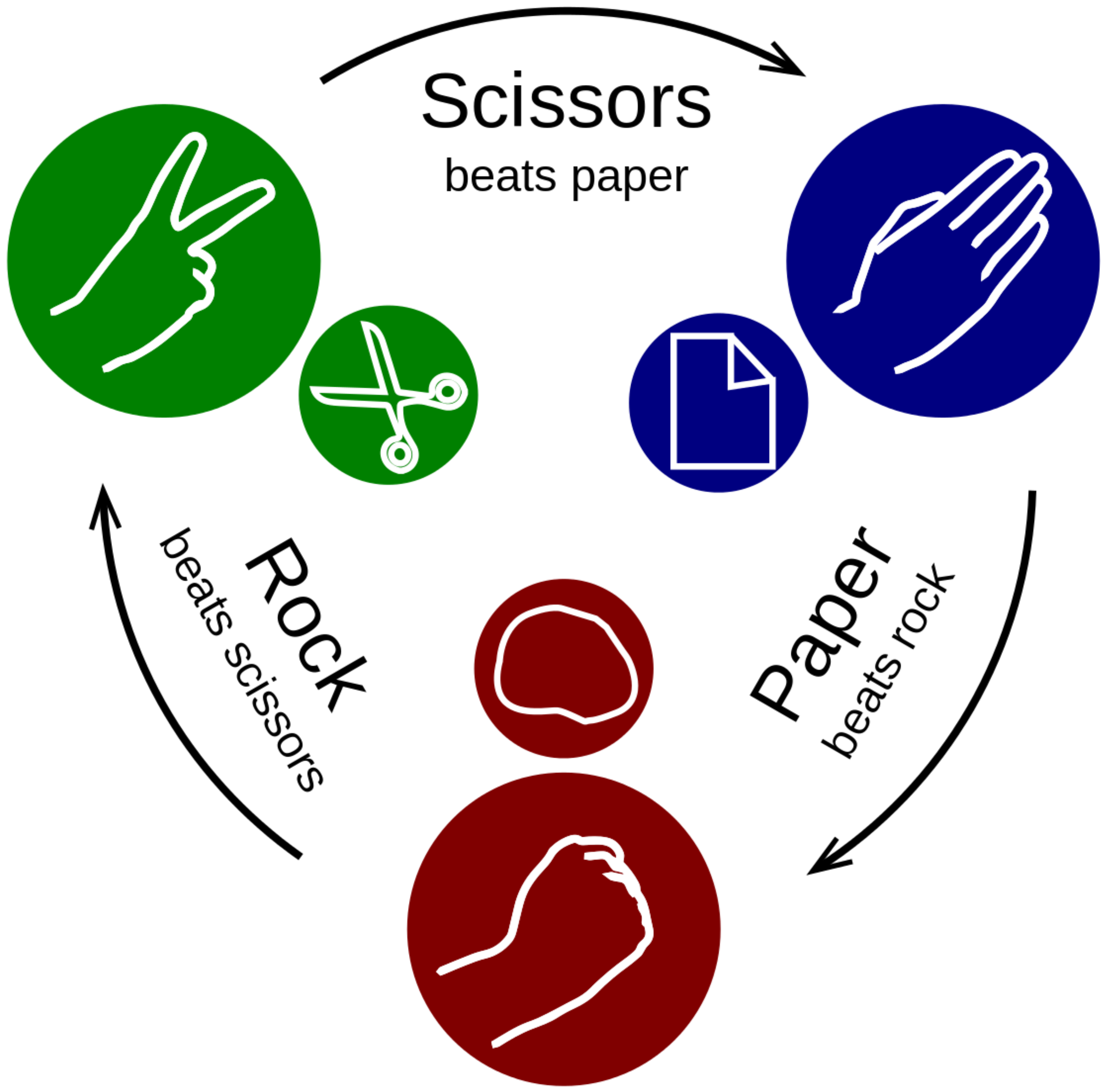
- 2 Players choosing 1 action concurrently
- Utility: *zero-sum game*
  - +1 for winning player
  - 0 in case of draw
  - -1 for losing player
- Winning player determined by those rules:



GOAL: Get the most possible expected utility, with no possibility of being exploited by a learning player

Games

A simple example: ***Rock Paper Scissors***



Characteristics of the game:

- 2 Players choosing 1 action concurrently
- Utility: *zero-sum game*
  - +1 for winning player
  - 0 in case of draw
  - -1 for losing player
- Winning player determined by those rules:

STRATEGY: play a uniform random action $\frac{1}{3}\ \boldsymbol{Rock},\ \frac{1}{3}\ \boldsymbol{Paper},\frac{1}{3}\ \boldsymbol{Scissor}$

Benchmark Games

Games used as benchmarks of algorithms in the past years:

CHESS

Benchmark Games

Games used as benchmarks of algorithms in the past years:

CHESS

Perfect Information setting

All the information characterizing a game state is available to all players

Benchmark Games

Games used as benchmarks of algorithms in the past years:

CHESS

Perfect Information setting

All the information characterizing a game state is available to all players

⇒ no need to remember past history of action to characterize the strategic situation

Benchmark Games

Games used as benchmarks of algorithms in the past years:

CHESS

Perfect Information setting
All the information characterizing a game state is available to all players

$\Rightarrow$ no need to remember past history of action to characterize the strategic situation

$< 10^{50}$ possible **discrete states**

Benchmark Games

Games used as benchmarks of algorithms in the past years:

CHESS

Perfect Information setting
All the information characterizing a game state is available to all players

$\Rightarrow$ no need to remember past history of action to characterize the strategic situation

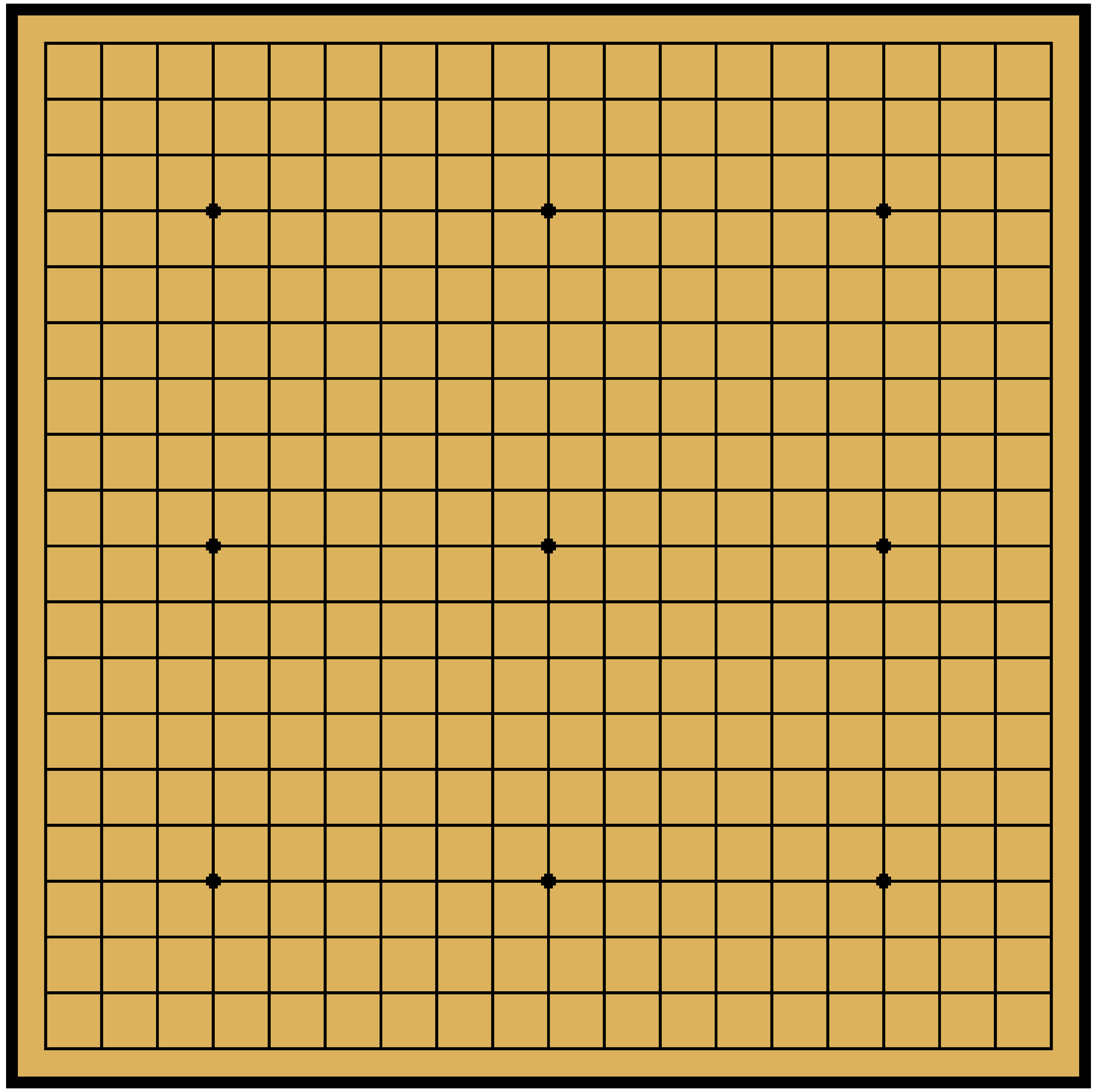
$< 10^{50}$ possible **discrete states**

IBM

Deep Blue, 1996

Benchmark Games

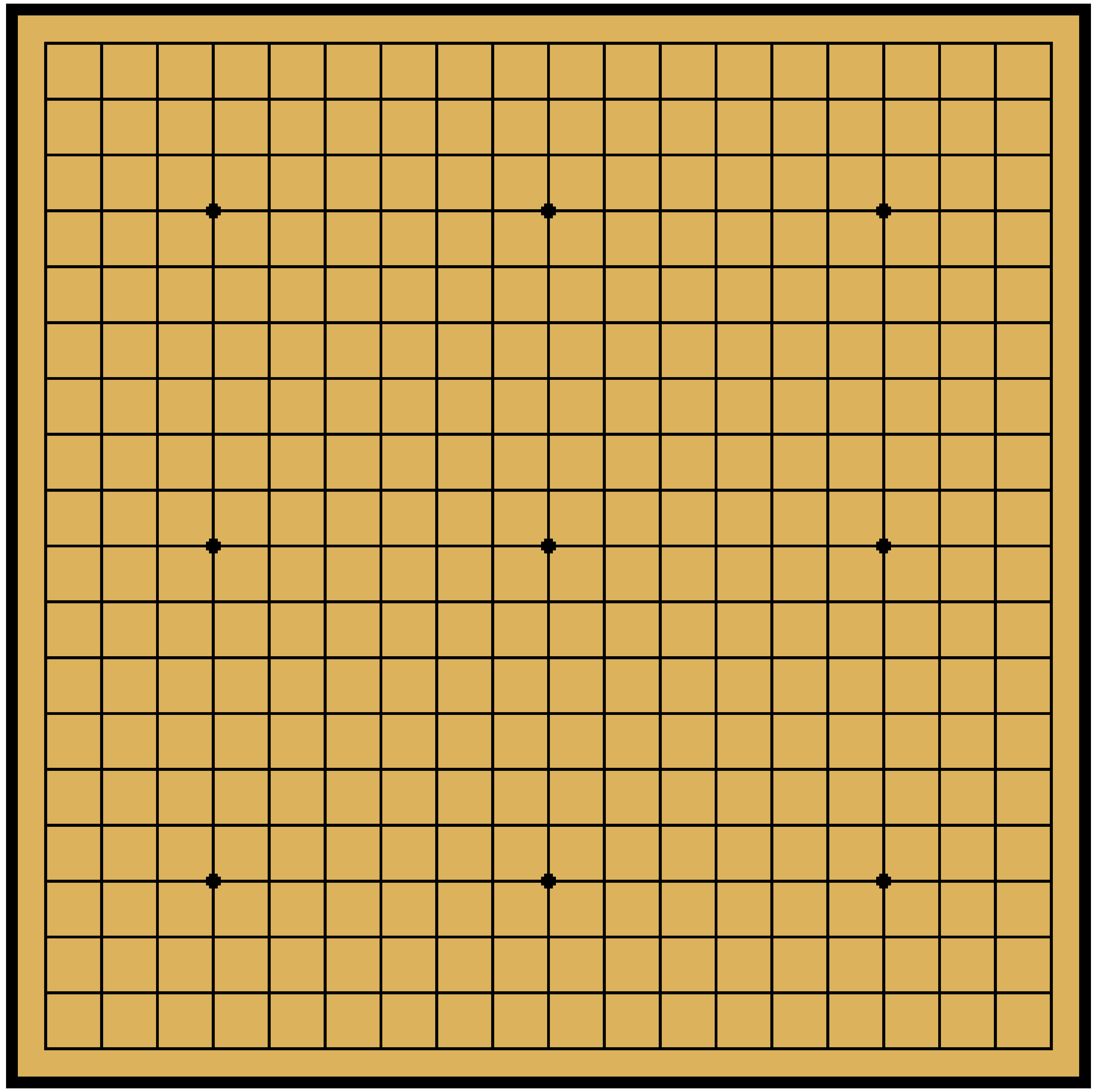
Games used as benchmarks of algorithms in the past years:



GO

Benchmark Games

Games used as benchmarks of algorithms in the past years:

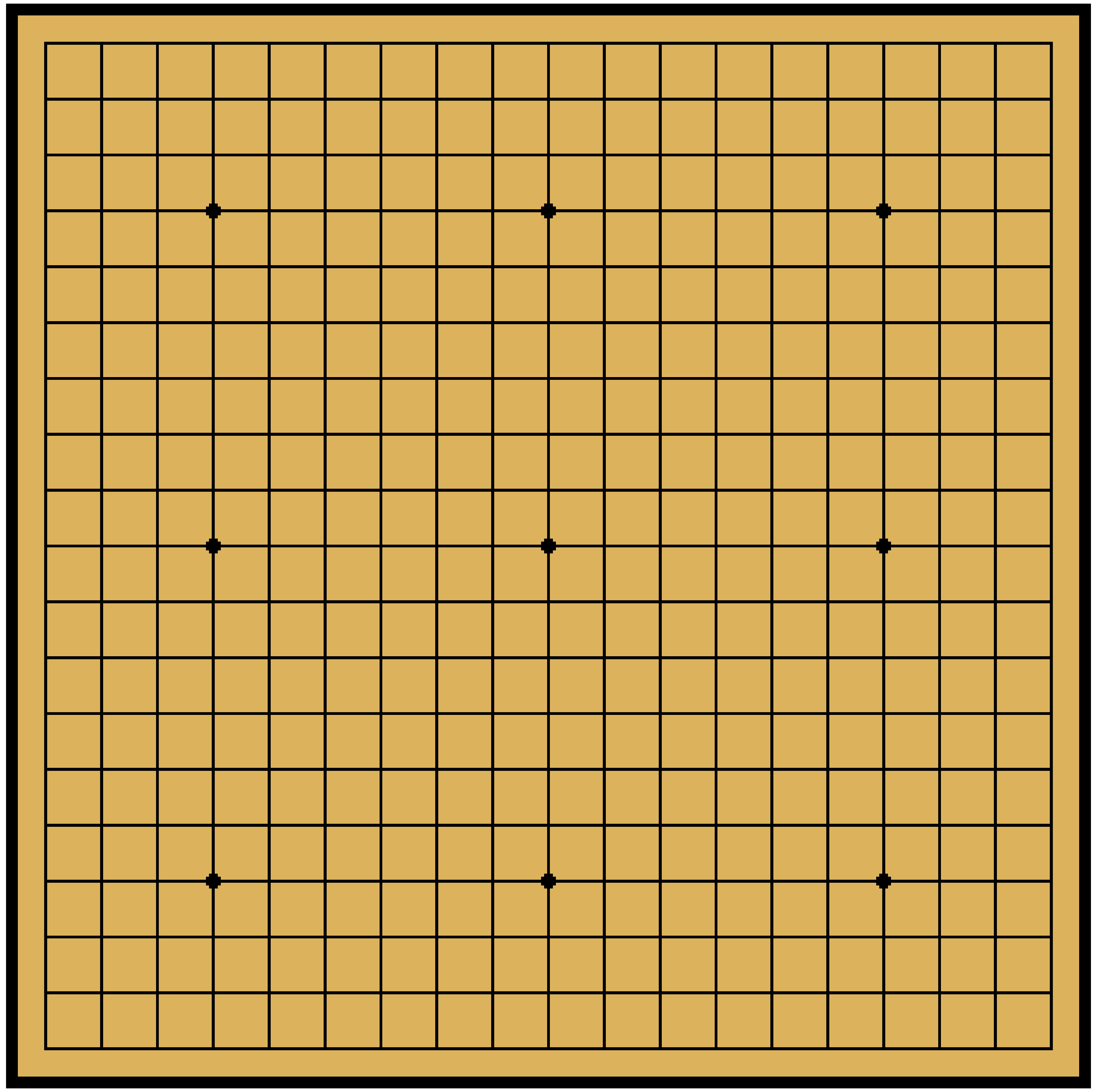


GO

Perfect Information setting
All the information characterizing a game state is available to all players

Benchmark Games

Games used as benchmarks of algorithms in the past years:



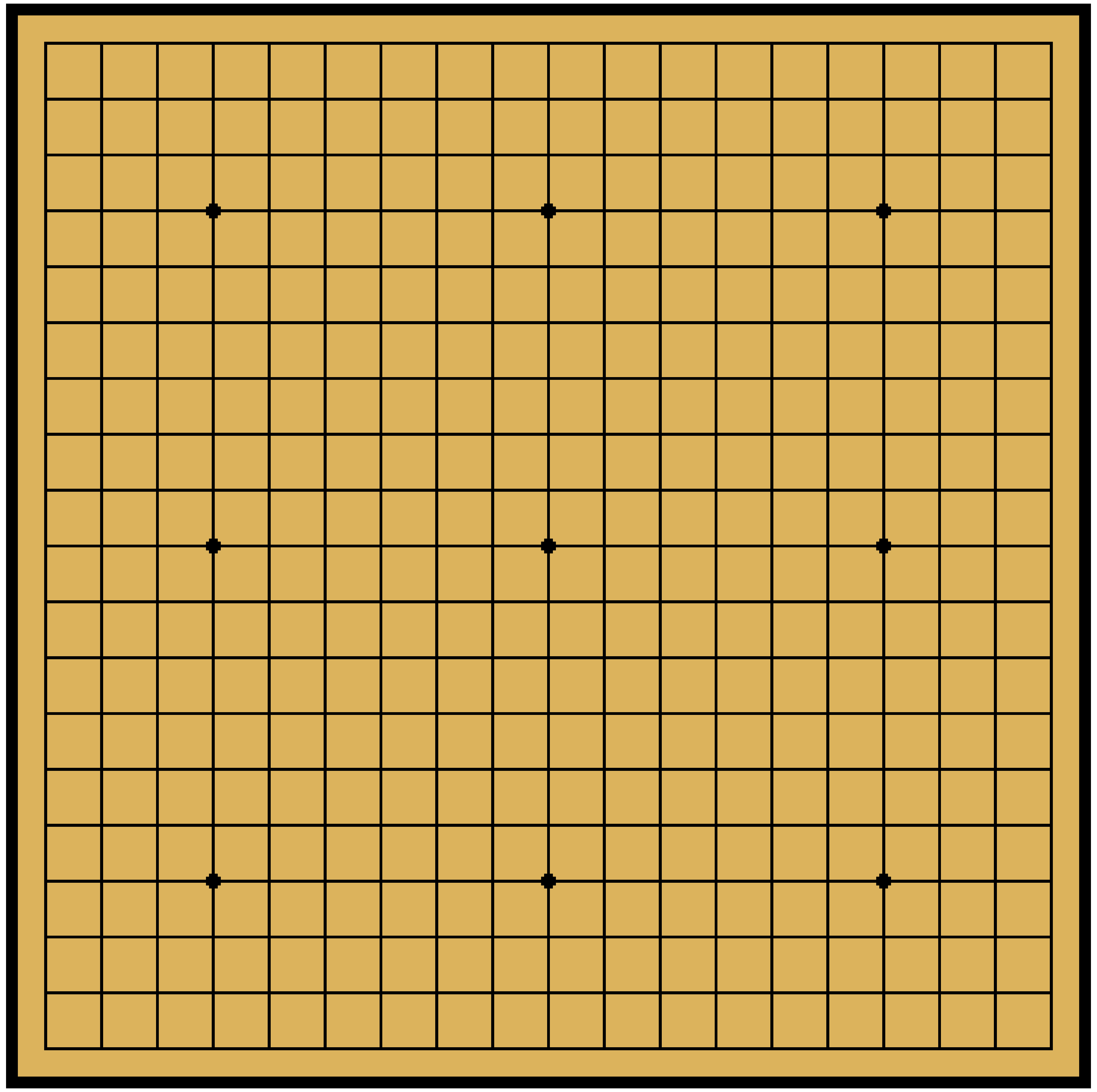
GO

Perfect Information setting

All the information characterizing a game state is available to all players

Benchmark Games

Games used as benchmarks of algorithms in the past years:



<u>GO</u>

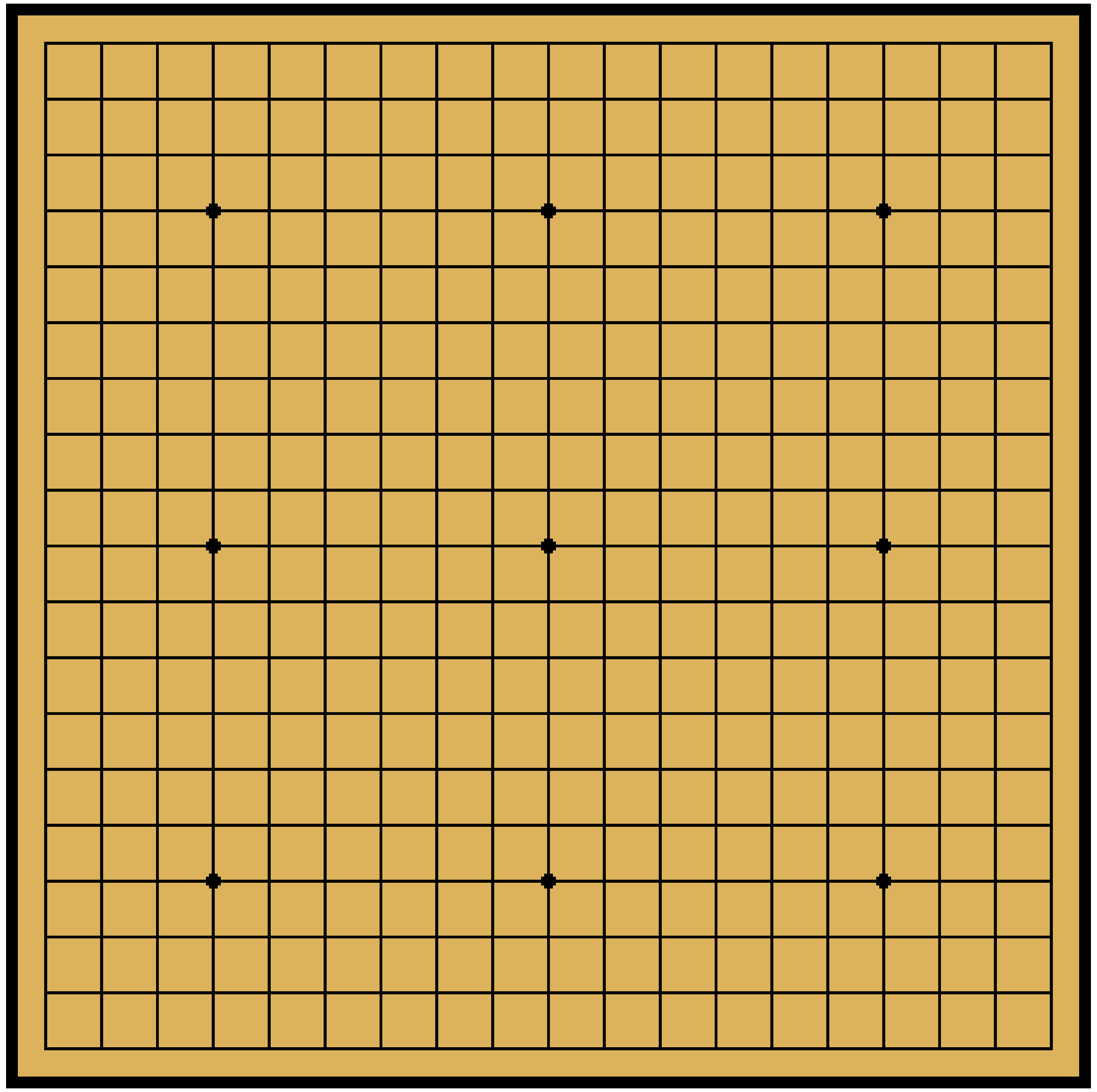
Perfect Information setting

All the information characterizing a game state is available to all players

$\Rightarrow$ no need to remember past history of action to characterize the strategic situation

Benchmark Games

Games used as benchmarks of algorithms in the past years:



<u>GO</u>

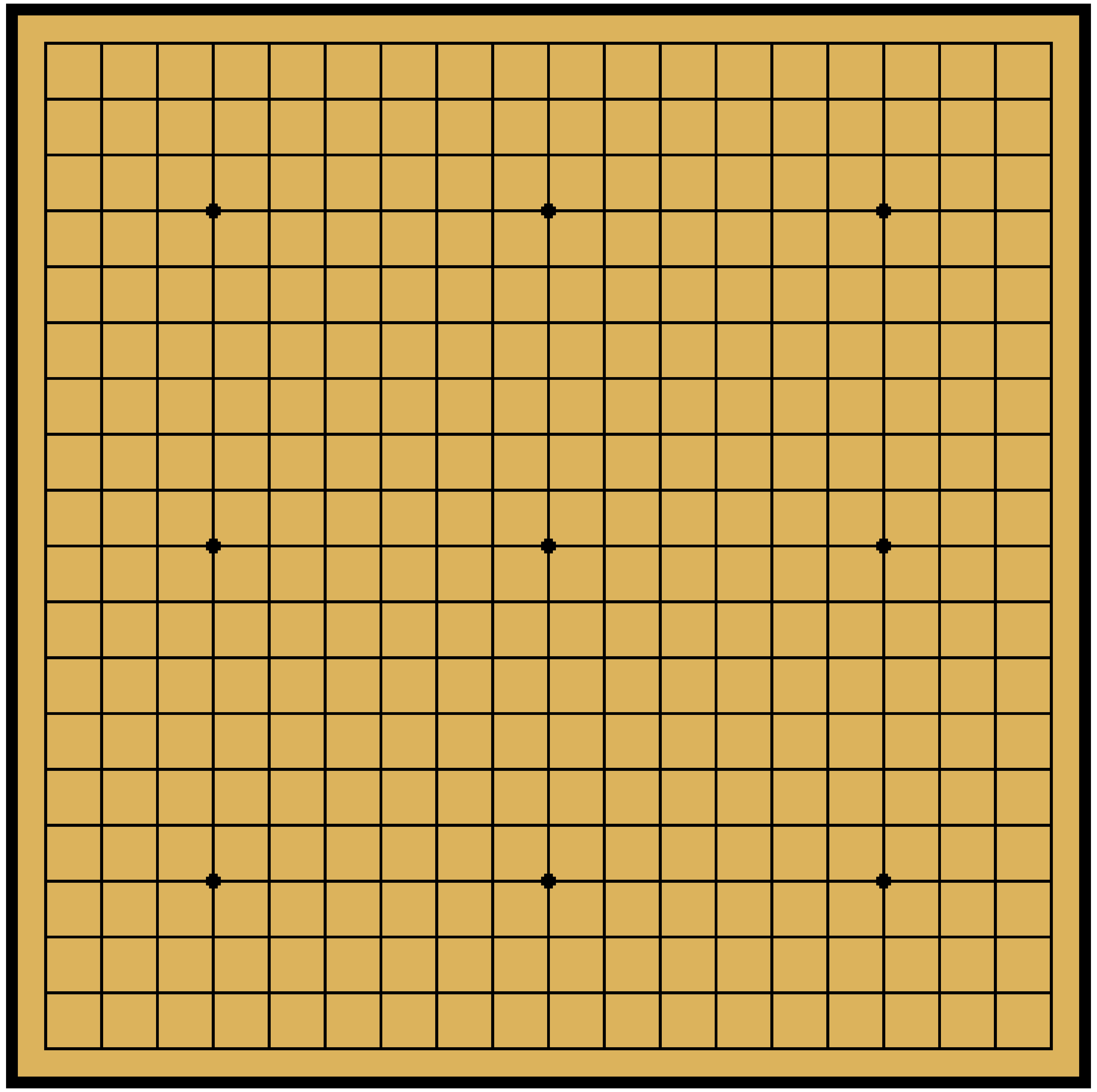
Perfect Information setting
All the information characterizing a game state is available to all players

$\Rightarrow$ no need to remember past history of action to characterize the strategic situation

$< 10^{170}$ possible **discrete states**

Benchmark Games

Games used as benchmarks of algorithms in the past years:



GO

Perfect Information setting
All the information characterizing a game state is available to all players

$\Rightarrow$ no need to remember past history of action to characterize the strategic situation

$< 10^{170}$ possible **discrete states**

DeepMind Alpha Go 2015

Benchmark Games

Games used as benchmarks of algorithms in the past years:

HOLD’EM POKER

Benchmark Games

Games used as benchmarks of algorithms in the past years:

<u>HOLD'EM POKER</u>

Imperfect Information setting

Cards held by opponents are hidden

Benchmark Games

Games used as benchmarks of algorithms in the past years:

HOLD'EM POKER

Imperfect Information setting

Cards held by opponents are hidden

⇒ history of play is needed to characterize the strength of their hands

⇒ randomized action are needed at each state to not disclose too much information

Benchmark Games

Games used as benchmarks of algorithms in the past years:

<u>HOLD'EM POKER</u>

Imperfect Information setting

Cards held by opponents are hidden

$\Rightarrow$ history of play is needed to characterize the strength of their hands

$\Rightarrow$ randomized action are needed at each state to not disclose too much information

$< 6 \cdot 10^{165}$ possible **discrete states** **mainly due to betting**

Benchmark Games

Games used as benchmarks of algorithms in the past years:

HOLD'EM POKER

Imperfect Information setting

Cards held by opponents are hidden

$\Rightarrow$ history of play is needed to characterize the strength of their hands

$\Rightarrow$ randomized action are needed at each state to not disclose too much information

$< 6 \cdot 10^{165}$ possible **discrete states** **mainly due to betting**

Carnegie Mellon University

Libratus 2017

Benchmark Games

Games used as benchmarks of algorithms in the past years:

<u>STARCRAFT 2</u>

Benchmark Games

Games used as benchmarks of algorithms in the past years:

STARCRAFT 2

Imperfect Information setting

Opponent’s resources hidden, map visibility

Benchmark Games

Games used as benchmarks of algorithms in the past years:

STARCRAFT 2

Imperfect Information setting

Opponent’s resources hidden, map visibility

⇒History and contextual information needed to describe current state

Benchmark Games

Games used as benchmarks of algorithms in the past years:

STARCRAFT 2

Imperfect Information setting

Opponent's resources hidden, map visibility

⇒ History and contextual information needed to describe current state

∼ **continuous states** **since physical interaction in a 3D world**

Benchmark Games

Games used as benchmarks of algorithms in the past years:

STARCRAFT 2

Imperfect Information setting

Opponent's resources hidden, map visibility

$\Rightarrow$ History and contextual information needed to describe current state

$\sim$ **continuous states** **since physical interaction in a 3D world**

DeepMind AlphaStar 2019

Benchmark Games

Games used as benchmarks of algorithms in the past years:

<u>HANABI</u>

Benchmark Games

Games used as benchmarks of algorithms in the past years:

HANABI

Imperfect Information setting

Own cards are hidden

Benchmark Games

Games used as benchmarks of algorithms in the past years:

<u>HANABI</u>

Imperfect Information setting

Own cards are hidden

Collaborative Game

Benchmark Games

Games used as benchmarks of algorithms in the past years:

HANABI

Imperfect Information setting
Own cards are hidden

Collaborative Game

2 to 5 players Game

Benchmark Games

Games used as benchmarks of algorithms in the past years:

<u>HANABI</u>

Imperfect Information setting
Own cards are hidden

Collaborative Game

2 to 5 players Game

OPEN CHALLENGE!

Real World Applications

Real World applications of algorithmic game theory:

AUTONOMOUS CAR RACING

Real World Applications

Real World applications of algorithmic game theory:

AUTONOMOUS CAR RACING

Imperfect Information setting

Concurrent execution of actions

Real World Applications

Real World applications of algorithmic game theory:

AUTONOMOUS CAR RACING

Imperfect Information setting
Concurrent execution of actions

Adversarial Setting

Real World Applications

Real World applications of algorithmic game theory:

<u>AUTONOMOUS CAR RACING</u>

Imperfect Information setting

Concurrent execution of actions

Adversarial Setting

$\Rightarrow$ need of a strategy to manage interaction with adversary’s trajectory

Real World Applications

Real World applications of algorithmic game theory:

AUTONOMOUS CAR RACING

Imperfect Information setting
Concurrent execution of actions

Adversarial Setting

$\Rightarrow$ need of a strategy to manage interaction with adversary's trajectory

Multiple Goals:

- Stay inside
- Do not collide
- Stay ahead
- Increment progess vs adversary

Real World Applications

Real World applications of algorithmic game theory:

MICRO-LEVEL FIGHTS CONTROL

Coordinate Control of individual units

Real World Applications

Real World applications of algorithmic game theory:

<u>MICRO-LEVEL FIGHTS CONTROL</u>

Coordinate Control of individual units

Imperfect Information setting

Concurrent execution of actions

Real World Applications

Real World applications of algorithmic game theory:

<u>MICRO-LEVEL FIGHTS CONTROL</u>

Coordinate Control of individual units

Imperfect Information setting

Concurrent execution of actions

Adversarial Setting

Real World Applications

Real World applications of algorithmic game theory:

<u>MICRO-LEVEL FIGHTS CONTROL</u>

Coordinate Control of individual units

Imperfect Information setting

Concurrent execution of actions

Adversarial Setting

⇒ need of a strategy to manage fight with adversary

Real World Applications

Real World applications of algorithmic game theory:

<u>MICRO-LEVEL FIGHTS CONTROL</u>

Coordinate Control of individual units

Imperfect Information setting

Concurrent execution of actions

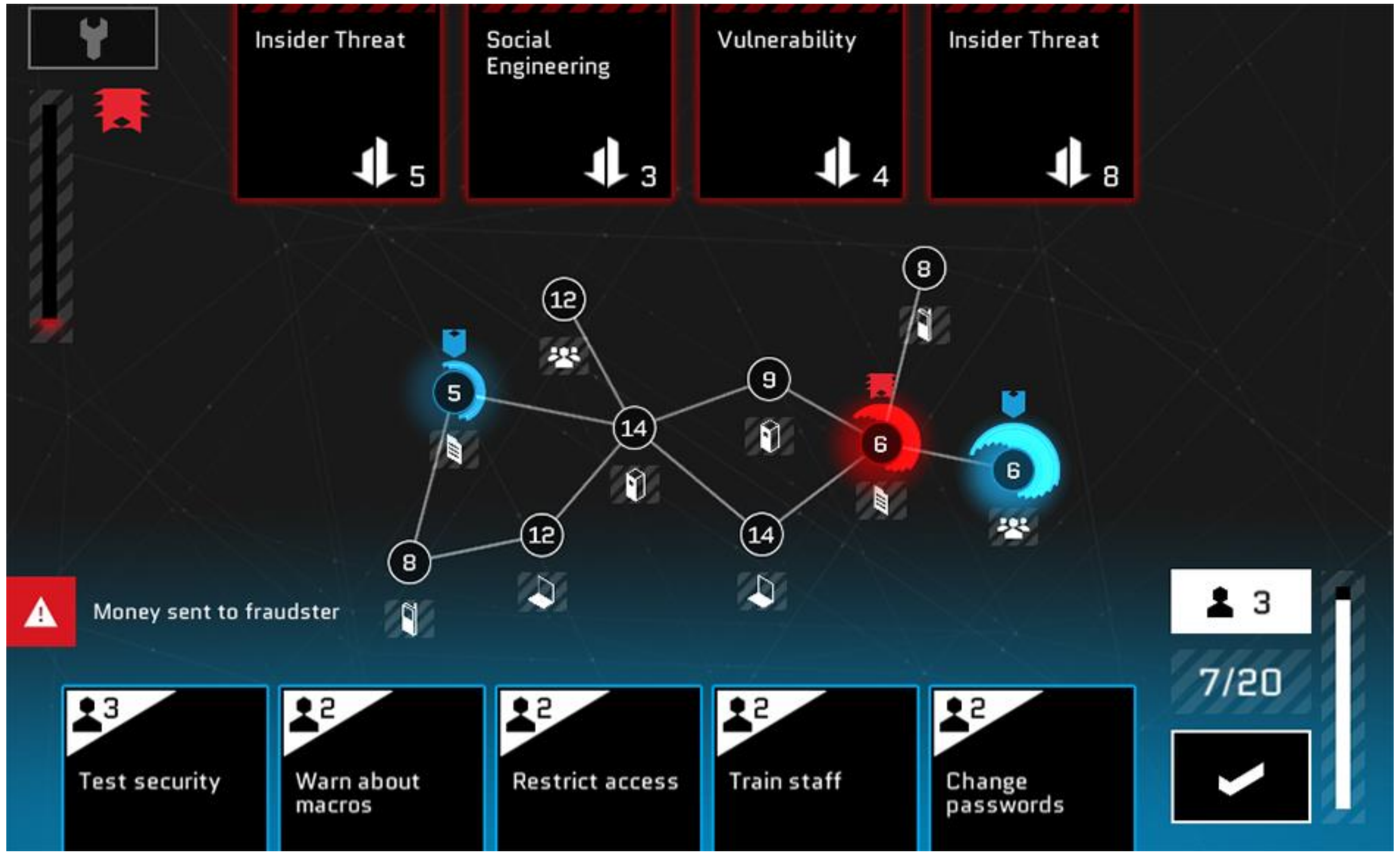
Adversarial Setting

$\Rightarrow$ need of a strategy to manage fight with adversary

Coordination required across a <u>team</u> of agents

Real World Applications

Real World applications of algorithmic game theory:

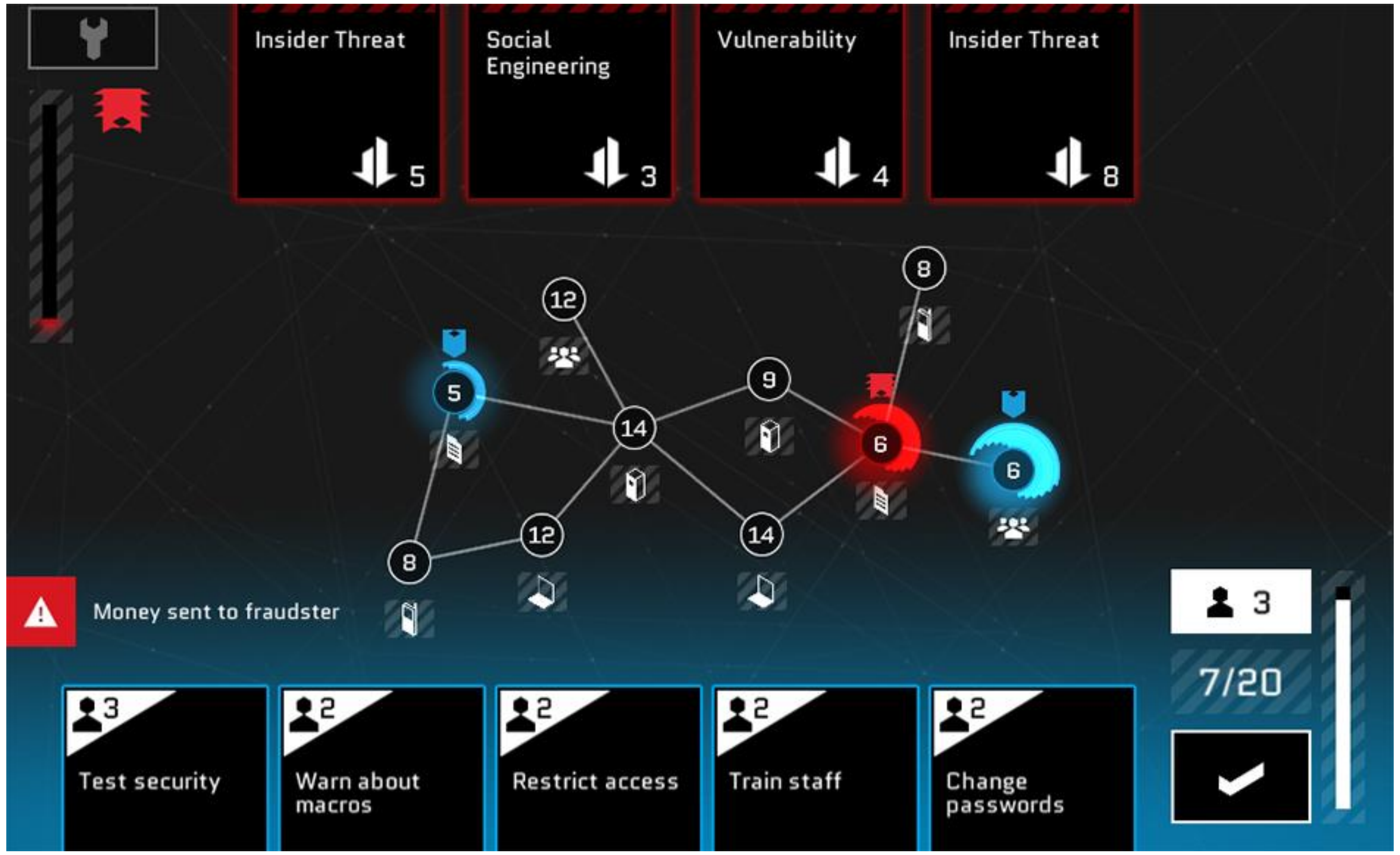


__MACRO-LEVEL SECURITY GAMES__

Strategic decision of Attack and Defence resource allocation

Real World Applications

Real World applications of algorithmic game theory:



<u>MACRO-LEVEL SECURITY GAMES</u>

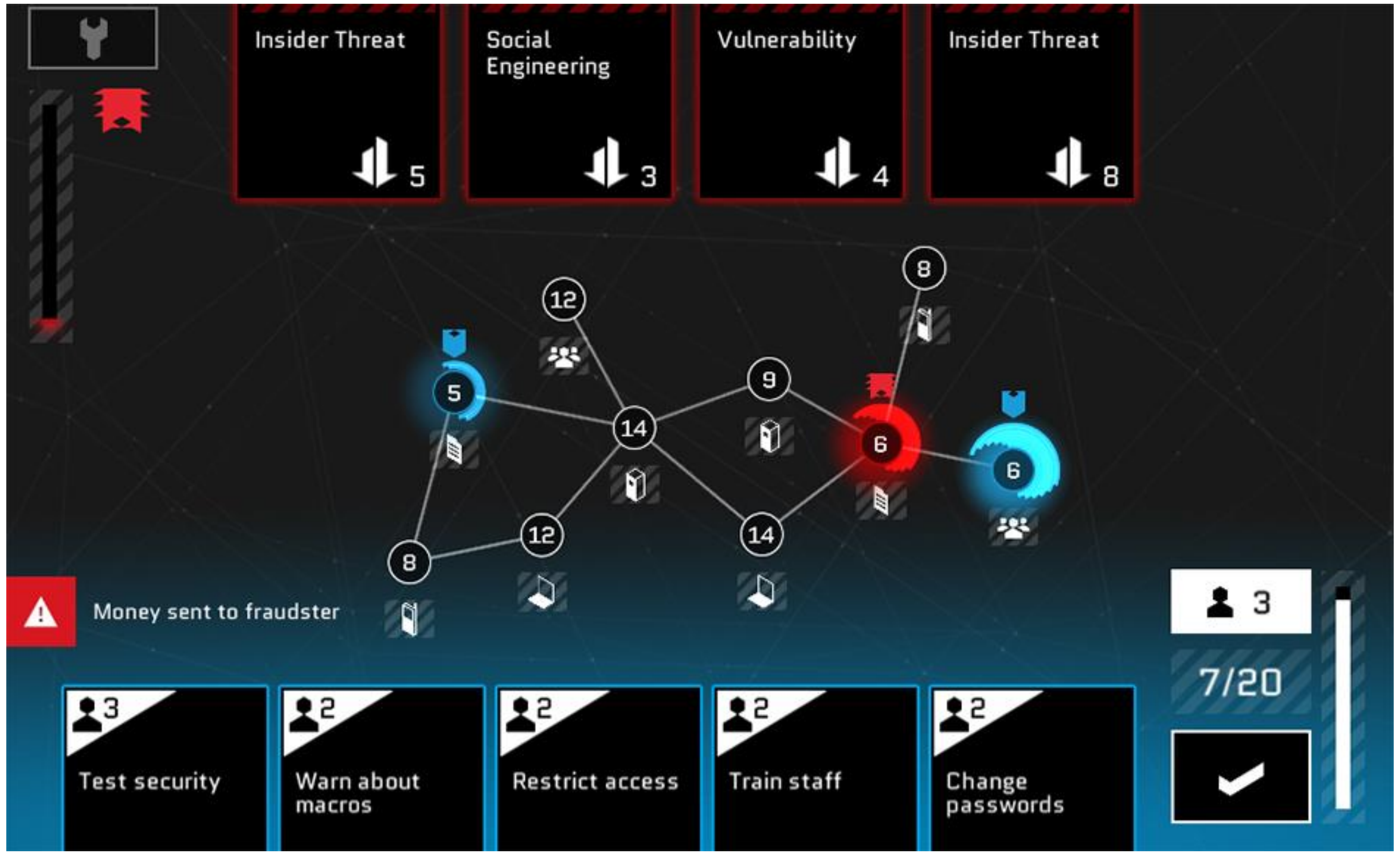
Strategic decision of Attack and Defence resource allocation

Imperfect Information setting

Concurrent execution of actions

Real World Applications

Real World applications of algorithmic game theory:



<u>MACRO-LEVEL SECURITY GAMES</u>

Strategic decision of Attack and Defence resource allocation

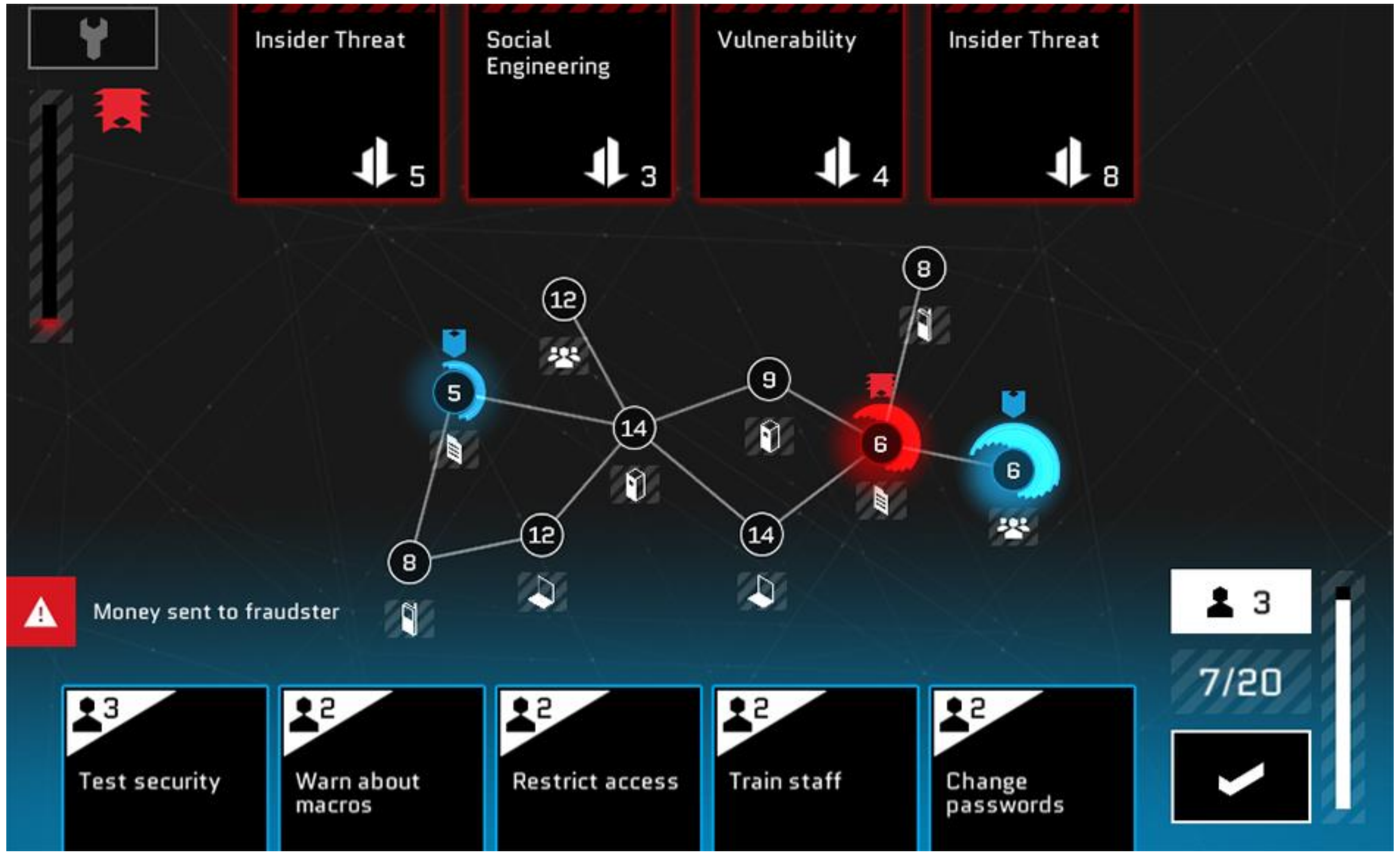
Imperfect Information setting

Concurrent execution of actions

Adversarial Setting

Real World Applications

Real World applications of algorithmic game theory:



<u>MACRO-LEVEL SECURITY GAMES</u>

Strategic decision of Attack and Defence resource allocation

Imperfect Information setting

Concurrent execution of actions

Adversarial Setting

$\Rightarrow$ need of a strategy to manage interaction with adversary

Overview

1. Introduction to Algorithmic Game Theory
2. Main Questions
3. Preliminaries
4. State of the art
5. Project proposal

Main Questions

Main Questions

How can we compactly describe Games and Strategies of players?

Main Questions

How can we compactly describe Games and Strategies of players?

How can we characterize stable strategic outcomes of players' interaction?

Main Questions

How can we compactly describe Games and Strategies of players?

How can we characterize stable strategic outcomes of players' interaction?

Given a game, how can we find the strategies played at one such equilibrium?

Overview

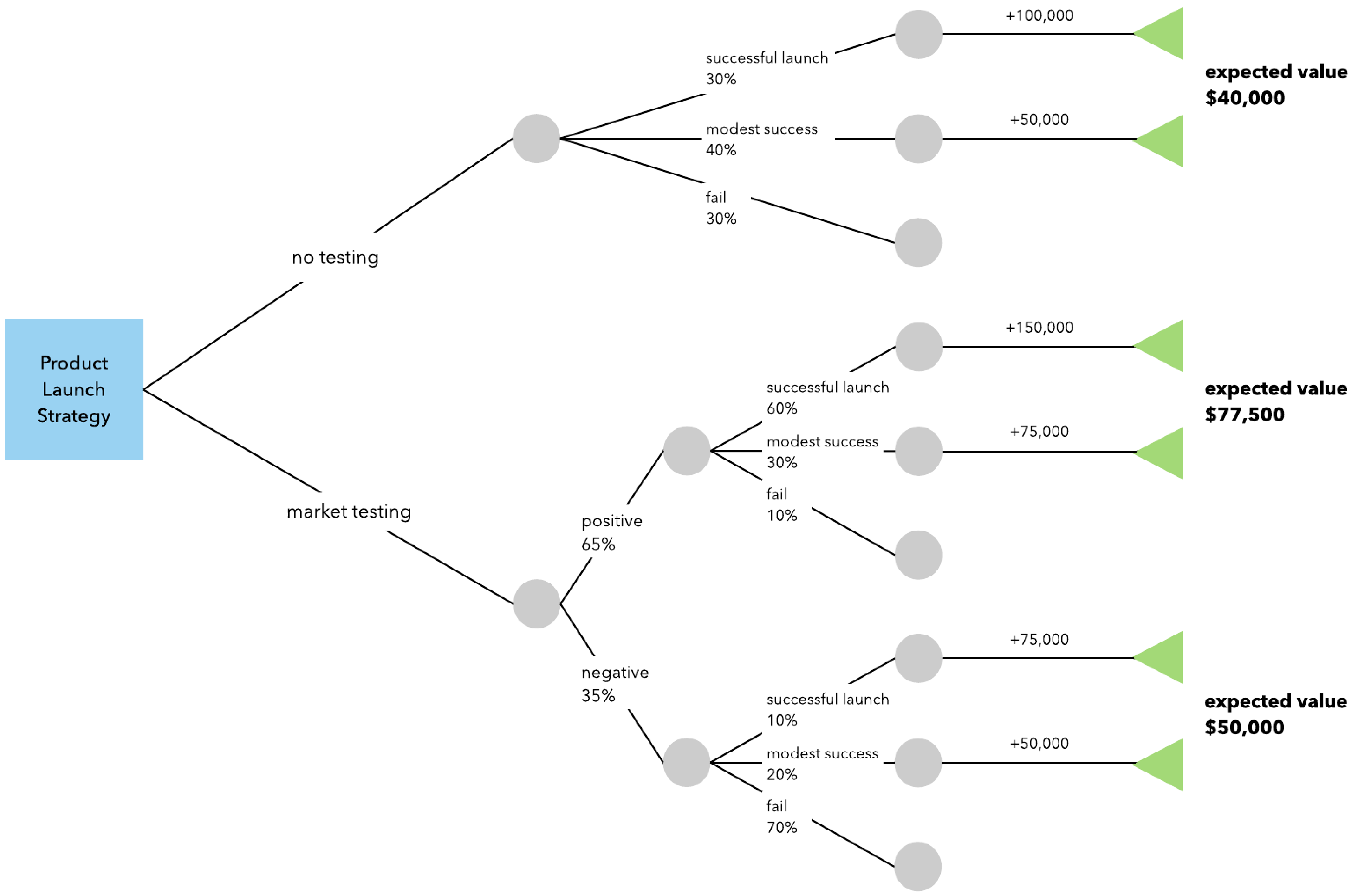
1. Introduction to Algorithmic Game Theory
2. Main Questions
3. Preliminaries
4. State of the art
5. Project proposal

Game Representation

Games can be represented as ***Decision Trees***

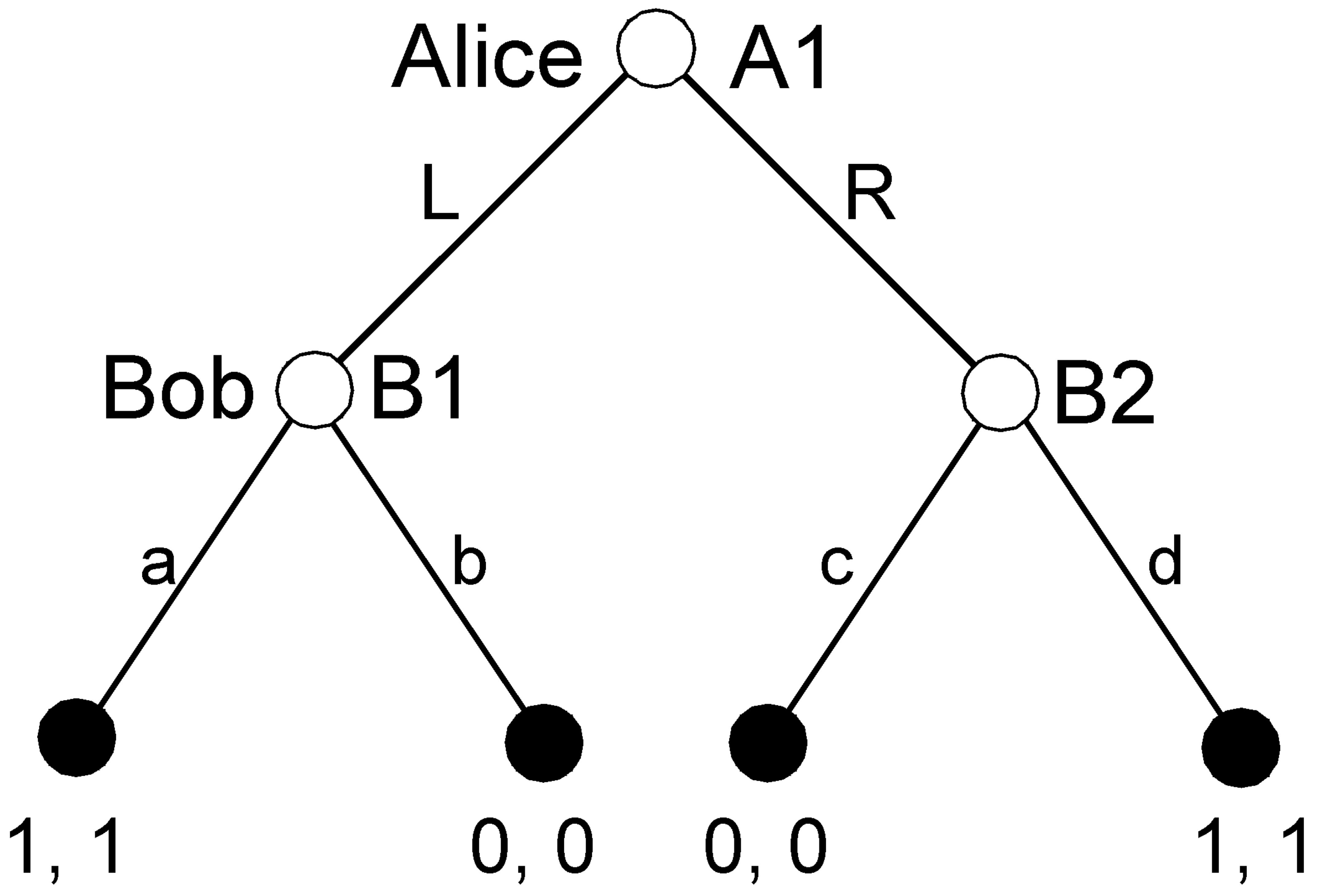
Each player in the game has to ***choose one possible action*** *from the available ones* ***whenever it is their turn***

The ***payoff*** *is determined at the end* ***depending on the sequence of actions*** *taken*



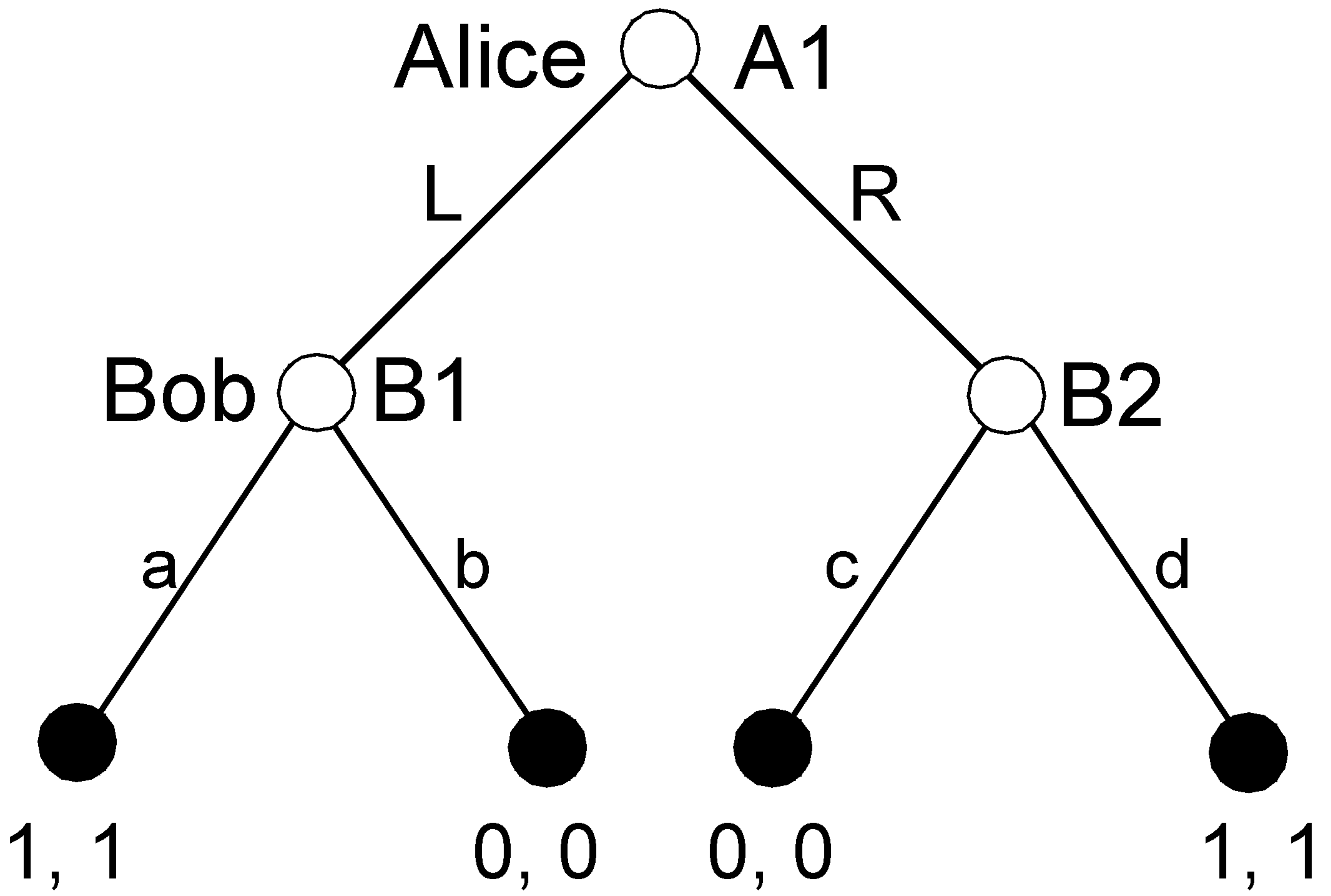
Game Representation

Extensive Form Representation = game represented as a Tree *(states are nodes)*



Game Representation

***Extensive Form Representation* =** game represented as a Tree *(states are nodes)*

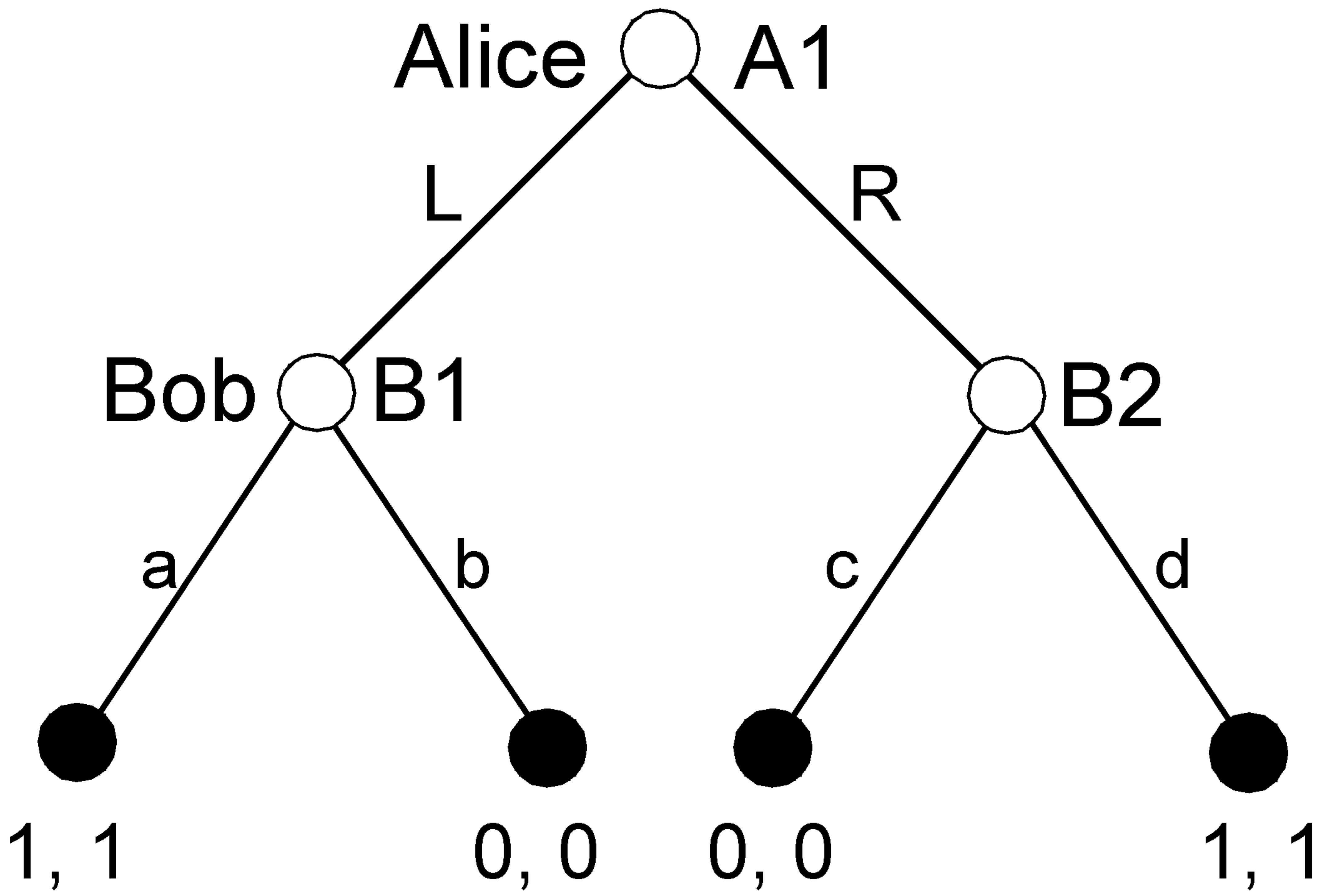


Perfect Information setting
Each node in the game can be uniquely identified by the player

⇒perfect knowledge of opponent's and own past

Game Representation

Extensive Form Representation = game represented as a Tree *(states are nodes)*



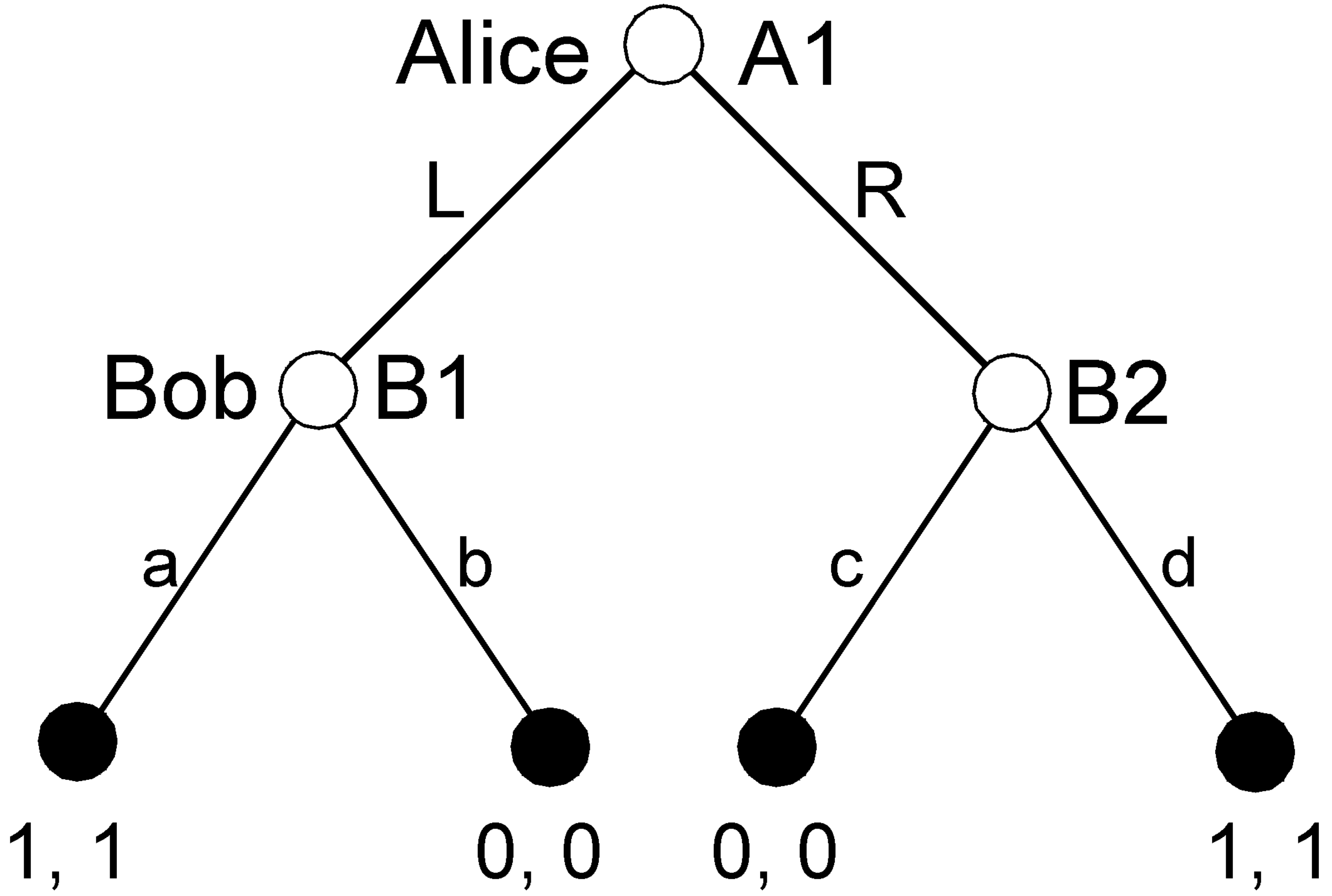
Perfect Information setting
Each node in the game can be uniquely identified by the player

⇒ perfect knowledge of opponent's and own past

A *Strategy* is a representation of probability distribution of actions at each node

Game Representation

Extensive Form Representation **=** game represented as a Tree *(states are nodes)*



Perfect Information setting
Each node in the game can be uniquely identified by the player

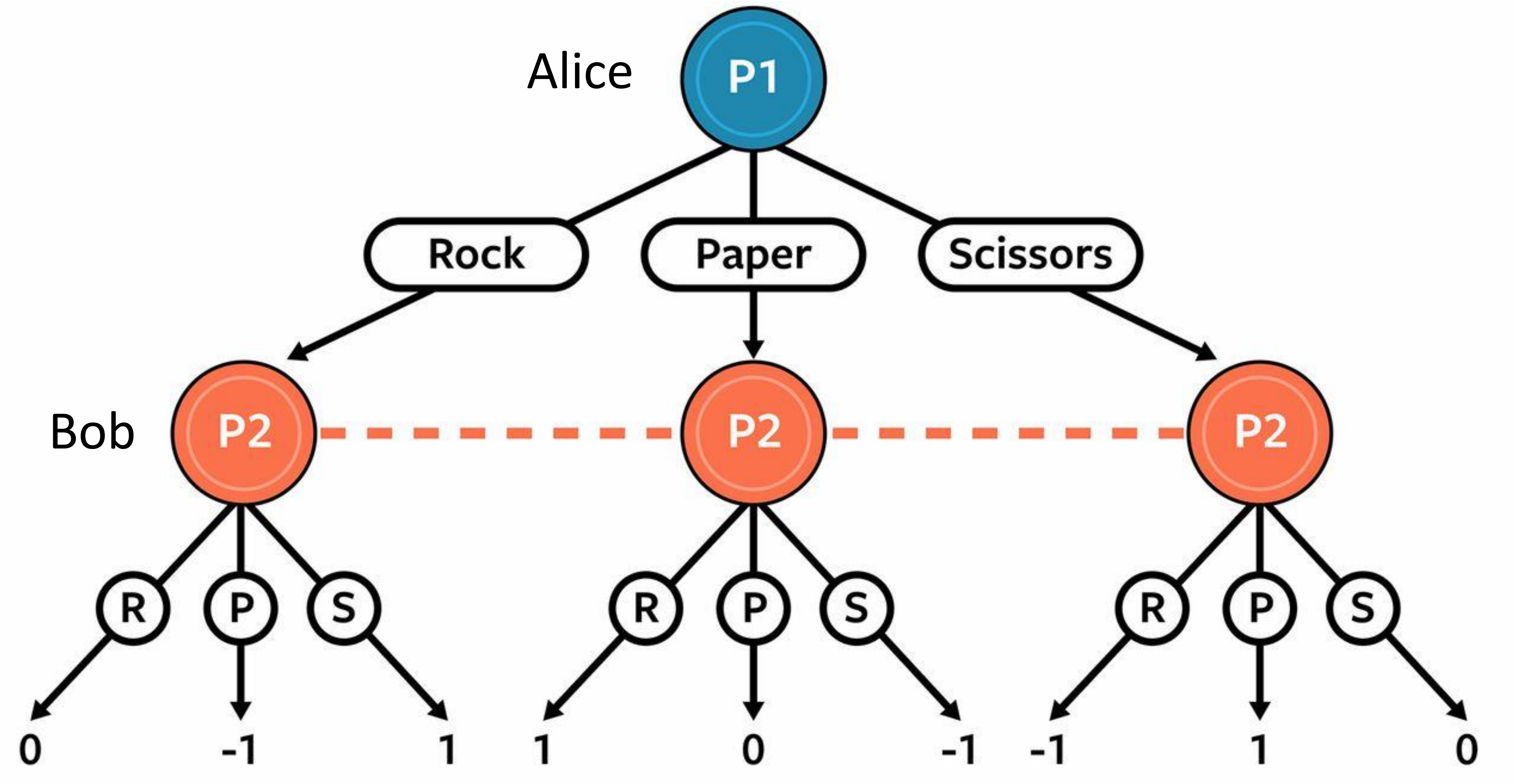
⇒ perfect knowledge of opponent's and own past

A *Strategy* is a representation of probability distribution of actions at each node

Alice's strategy: A1 - L: 0.5, R:0.5 *mixed strategy*
Bob's strategy: B1 – a:1.0, b:0.0
B2 – c:0.0, d:1.0 *pure strategy*

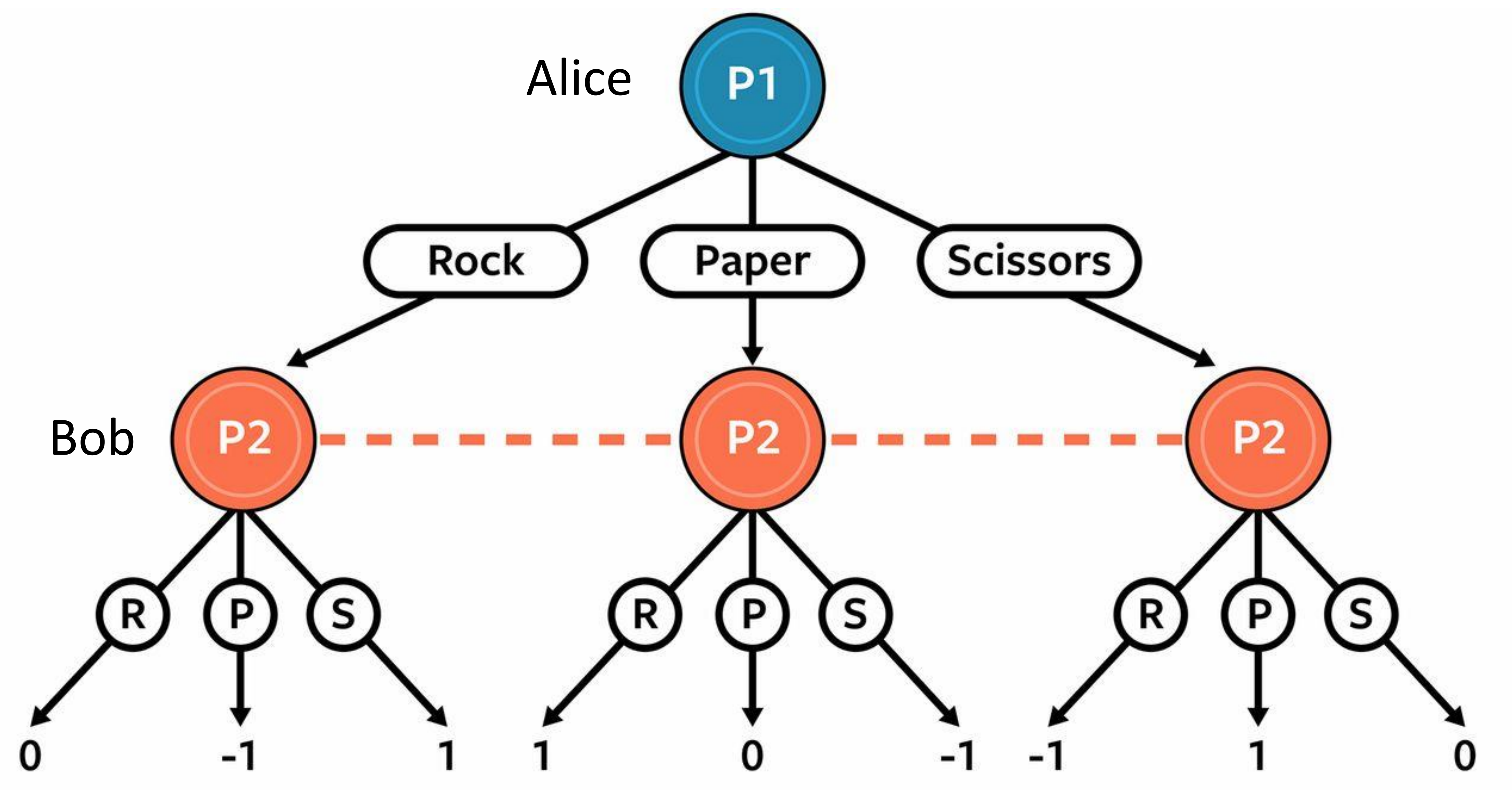
Game Representation

Extensive Form Representation = game represented as a Tree *(states are nodes)*



Game Representation

***Extensive Form Representation* =** game represented as a Tree *(states are nodes)*

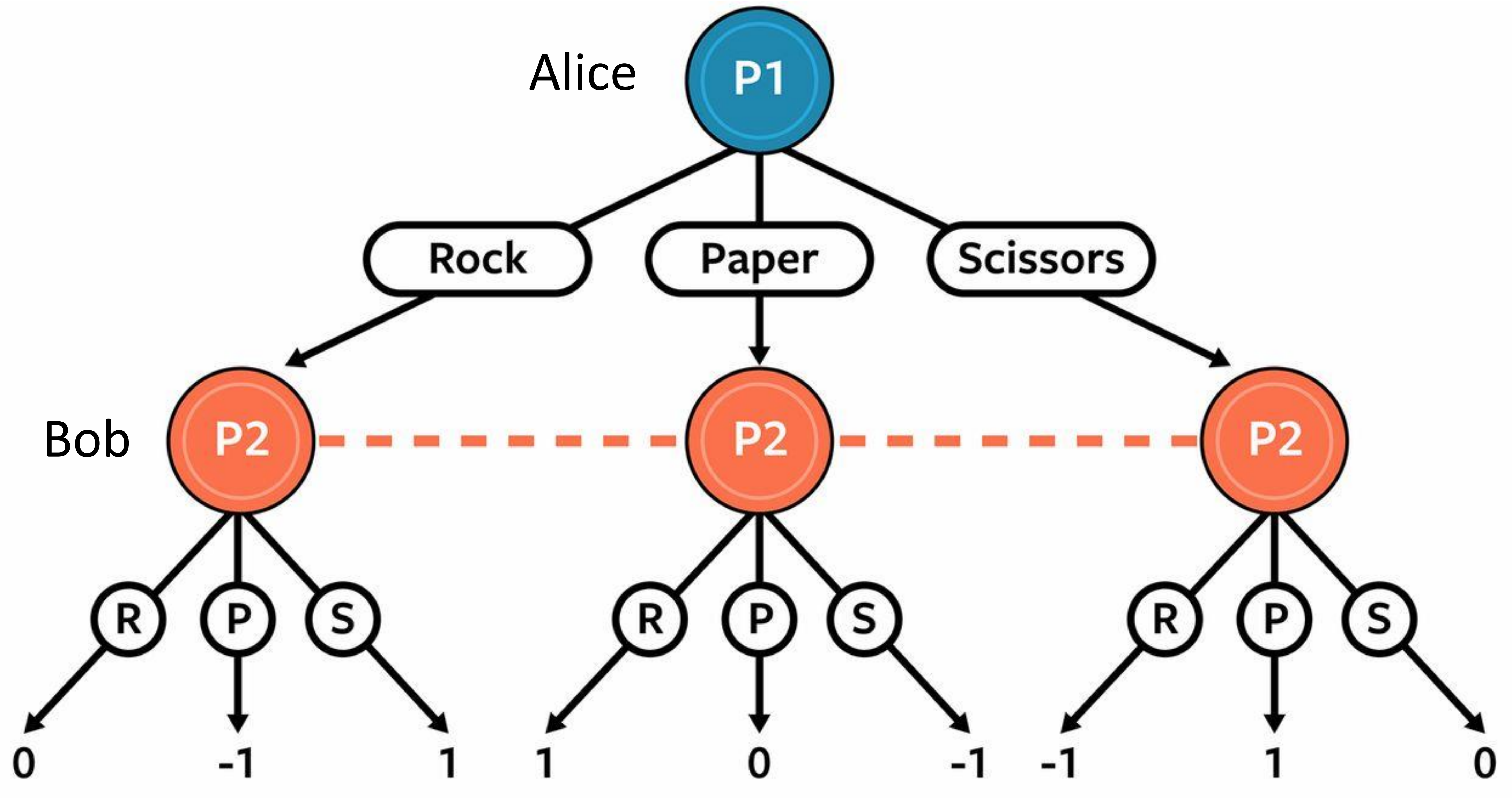


Imperfect Information setting
Information set constraint for Bob: he must play the same strategy at each node in the same infoset

$\Rightarrow$ some nodes are indistinguishable for him, since he is not expected to know the action played by Alice

Game Representation

Extensive Form Representation **=** game represented as a Tree *(states are nodes)*



Imperfect Information setting

Information set constraint for Bob: he must play the same strategy at each node in the same infoset

⇒ some nodes are indistinguishable for him, since he is not expected to know the action played by Alice

Alice's strategy: P1 - R: 0.2, P:0.3, S:0.5

Bob's strategy: P2 – r:0.33, p:0.33, s:0.33

Nash Equilibrium

A ***Nash Equilibrium*** is a joint combination of strategies stable with respect to unilateral deviations of a single player

Nash Equilibrium

A ***Nash Equilibrium*** is a joint combination of strategies stable with respect to unilateral deviations of a single player

[i.e. no player can gain more utility by changing part of his/her strategy, given the opponent plays a fixed strategy]

~ local optimum

Nash Equilibrium

A ***Nash Equilibrium*** is a joint combination of strategies stable with respect to unilateral deviations of a single player

[i.e. no player can gain more utility by changing part of his/her strategy, given the opponent plays a fixed strategy]
~ local optimum

	r	p	s
R	0,0	-1,1	1,-1
P	1,-1	0,0	-1,1
S	-1,1	1,-1	0,0

Nash Equilibrium

A ***Nash Equilibrium*** is a joint combination of strategies stable with respect to unilateral deviations of a single player

[i.e. no player can gain more utility by changing part of his/her strategy, given the opponent plays a fixed strategy]
~ local optimum

	r	p	s
R	0,0	-1,1	1,-1
P	1,-1	0,0	-1,1
S	-1,1	1,-1	0,0

<u>Strategy Profile 1:</u>

Alice - R:1.0, P:0.0, S:0.0

Bob - r:0.5, p:0.5, s:0.0

Not a Nash Equilibrium → Bob would deviate to P:1.0

<u>Strategy Profile 2:</u>

Alice - R:0.33, P:0.33, S:0.33

Bob - r:0.33, p:0.33, s:0.33

Nash Equilibrium → No one can increase payoff

Nash Equilibrium

A ***Nash Equilibrium*** is a joint combination of strategies stable with respect to unilateral deviations of a single player

[i.e. no player can gain more utility by changing part of his/her strategy, given the opponent plays a fixed strategy]
~ local optimum

	r	p	s
R	0,0	-1,1	1,-1
P	1,-1	0,0	-1,1
S	-1,1	1,-1	0,0

<u>Strategy Profile 1:</u>
Alice - R:1.0, P:0.0, S:0.0
Bob - r:0.5, p:0.5, s:0.0
Not a Nash Equilibrium → Bob would deviate to P:1.0

<u>Strategy Profile 2:</u>
Alice - R:0.33, P:0.33, S:0.33
Bob - r:0.33, p:0.33, s:0.33
Nash Equilibrium → No one can increase payoff

By finding Nash Equilibria, we can find stable strategies that express rationally stable situations

Overview

1. Introduction to Algorithmic Game Theory
2. Main Questions
3. Preliminaries
4. State of the art
5. Project proposal

State of the Art

General categorization of state of the art:

Solution Concept Definition

Iterative Algorithms to find Equilibria

State of the Art

General categorization of state of the art:

Solution Concept Definition	**How can we characterize stable strategic outcomes of players' interaction?**
Iterative Algorithms to find Equilibria	

State of the Art

General categorization of state of the art:

Solution Concept Definition	**How can we characterize stable strategic outcomes of players' interaction?**
Iterative Algorithms to find Equilibria	**Given a game, how can we find the strategies played at one such equilibrium?**

State of the Art

General categorization of state of the art:

	2 Players Zero Sum Games	General Sum Games	Team Games
Solution Concept Definition			
Iterative Algorithms to find Equilibria			

State of the Art

General categorization of state of the art:

	2 Players Zero Sum Games	General Sum Games	Team Games
Solution Concept Definition	**Nash Equilibria** [Nash, 1954]		
Iterative Algorithms to find Equilibria			

State of the Art

General categorization of state of the art:

	2 Players Zero Sum Games	General Sum Games	Team Games
Solution Concept Definition	**Nash Equilibria** [Nash, 1954]		
Iterative Algorithms to find Equilibria	*Tabular algorithms:* **LP techniques** **CFR** [Zinkevich et al, 2008] **Expansions: Abstraction, Sampling, Discounting, Search** **XFP** [Heinrich et al, 2015] *Neural Approximating algorithms:* **NFSP** [Heinrich et al, 2016] **DEEP CFR** [Brown et al, 2018] **DREAM** [Steinberger et al, 2020] **REBEL** [Brown et al, 2020]		

State of the Art

General categorization of state of the art:

	2 Players Zero Sum Games	General Sum Games	Team Games
Solution Concept Definition	**Nash Equilibria** [Nash, 1954]	**Correlated Equilibria** [Aumann, 1974] **EF(C)CE, AF(C)CE variants** **Hindsight Rationality** [Morril et al, 2020]	
Iterative Algorithms to find Equilibria	***Tabular algorithms:*** **LP techniques** **CFR** [Zinkevich et al, 2008] **Expansions: Abstraction, Sampling, Discounting, Search** **XFP** [Heinrich et al, 2015] ***Neural Approximating algorithms:*** **NFSP** [Heinrich et al, 2016] **DEEP CFR** [Brown et al, 2018] **DREAM** [Steinberger et al, 2020] **REBEL** [Brown et al, 2020]		

State of the Art

General categorization of state of the art:

	2 Players Zero Sum Games	General Sum Games	Team Games
Solution Concept Definition	**Nash Equilibria** [Nash, 1954]	**Correlated Equilibria** [Aumann, 1974] **EF(C)CE, AF(C)CE variants** **Hindsight Rationality** [Morril et al, 2020]	
Iterative Algorithms to find Equilibria	***Tabular algorithms:*** **LP techniques** **CFR** [Zinkevich et al, 2008] **Expansions: Abstraction, Sampling, Discounting, Search** **XFP** [Heinrich et al, 2015] ***Neural Approximating algorithms:*** **NFSP** [Heinrich et al, 2016] **DEEP CFR** [Brown et al, 2018] **DREAM** [Steinberger et al, 2020] **REBEL** [Brown et al, 2020]	**Internal Regret minimizers** [Hart, MasColell, 2000] **CFR-Jr** [Celli et al, 2019] **ICFR** [Celli et al 2020]	

State of the Art

General categorization of state of the art:

	2 Players Zero Sum Games	General Sum Games	Team Games
Solution Concept Definition	**Nash Equilibria** [Nash, 1954]	**Correlated Equilibria** [Aumann, 1974] **EF(C)CE, AF(C)CE variants** **Hindsight Rationality** [Morril et al, 2020]	**Team Maxmin Equilibria** [VonStengel and Koller, 1997]
Iterative Algorithms to find Equilibria	***Tabular algorithms:*** **LP techniques** **CFR** [Zinkevich et al, 2008] **Expansions: Abstraction, Sampling, Discounting, Search** **XFP** [Heinrich et al, 2015] ***Neural Approximating algorithms:*** **NFSP** [Heinrich et al, 2016] **DEEP CFR** [Brown et al, 2018] **DREAM** [Steinberger et al, 2020] **REBEL** [Brown et al, 2020]	**Internal Regret minimizers** [Hart, MasColell, 2000] **CFR-Jr** [Celli et al, 2019] **ICFR** [Celli et al 2020]	

State of the Art

General categorization of state of the art:

	2 Players Zero Sum Games	General Sum Games	Team Games
Solution Concept Definition	**Nash Equilibria** [Nash, 1954]	**Correlated Equilibria** [Aumann, 1974] **EF(C)CE, AF(C)CE variants** **Hindsight Rationality** [Morril et al, 2020]	**Team Maxmin Equilibria** [VonStengel and Koller, 1997]
Iterative Algorithms to find Equilibria	***Tabular algorithms:*** **LP techniques** **CFR** [Zinkevich et al, 2008] **Expansions: Abstraction, Sampling, Discounting, Search** **XFP** [Heinrich et al, 2015] ***Neural Approximating algorithms:*** **NFSP** [Heinrich et al, 2016] **DEEP CFR** [Brown et al, 2018] **DREAM** [Steinberger et al, 2020] **REBEL** [Brown et al, 2020]	**Internal Regret minimizers** [Hart, MasColell, 2000] **CFR-Jr** [Celli et al, 2019] **ICFR** [Celli et al 2020]	**Normal-Form algorithms, efficiency bounds** [Basilico et al, 2016] **Extensive-form definition, efficiency bounds , HCG algorithm** [Celli and Gatti 2017] **FTP** [Farina and Celli, 2018] **STAC** [Celli et al, 2019] **SIMS** [Cacciamani et al, 2020]

State of the Art

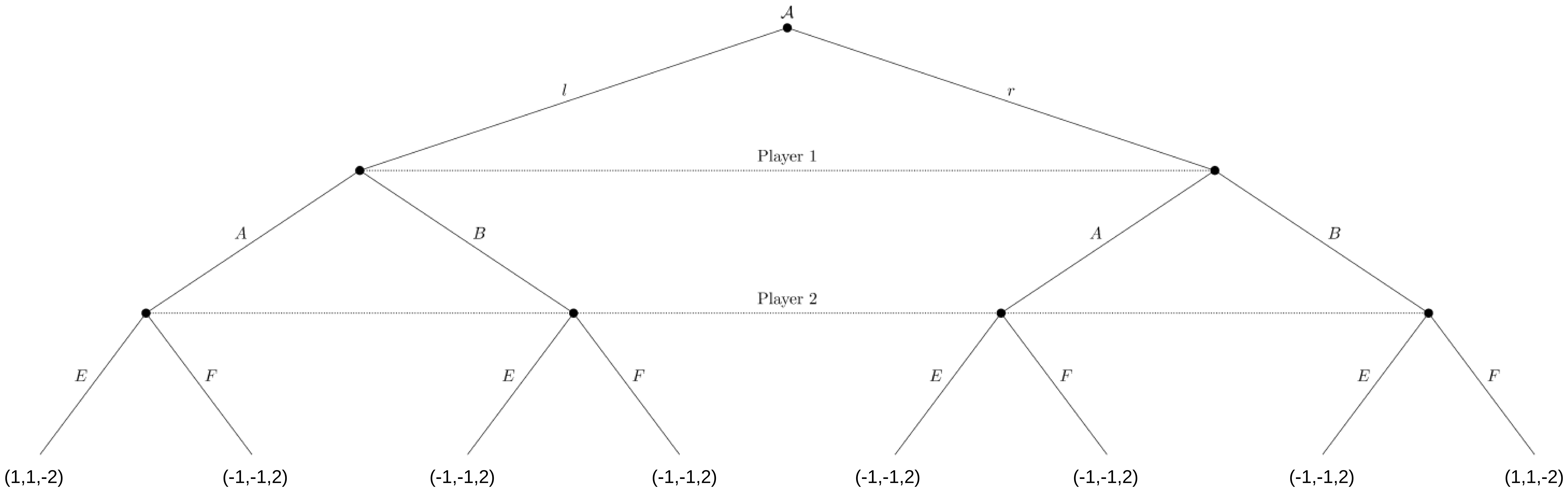
General categorization of state of the art:

	2 Players Zero Sum Games	General Sum Games	Team Games
Solution Concept Definition	**Nash Equilibria** [Nash, 1954]	**Correlated Equilibria** [Aumann, 1974] EF(C)CE, AF(C)CE variants **Hindsight Rationality** [Morril et al, 2020]	**Team Maxmin Equilibria** [VonStengel and Koller, 1997]
Iterative Algorithms to find Equilibria	*Tabular algorithms:* **CFR** Expansi **XFP** [Heinrich et al, 2015] *Neural Approximating algorithms:* **NFSP** [Heinrich et al, 2016] **DEEP CFR** [Brown et al, 2018] **DREAM** [Steinberger et al, 2020] **REBEL** [Brown et al, 2020]	**Internal Regret minimizers** [Hart, MasColell, 2000] **CFR-Jr** [Celli et al, 2019] **ICFR** [Celli et al 2020]	**Normal-Form algorithms, efficiency bounds** [Basilico et al, 2016] **Extensive-form definition, efficiency bounds , HCG algorithm** [Celli and Gatti 2017] **FTP** [Farina and Celli, 2018] **STAC** [Celli et al, 2019] **SIMS** [Cacciamani et al, 2020]

Our Focus will be on Team Games:

- **2 players Zero sum games already refined solutions and bigger research groups working**
- **General sum games are a possibility**
- **Team Games are a topic of great expertise of the research team, and an already established environment is available**

Team Games



Team Games can be characterized as *N vs M players zero-sum games*, in which all members of the same team have *identical payoffs*

In our research we will focus on 2 vs 1 games

Team Maxmin Equilibrium

Possible Solution Concepts:

TME = *Team Players maximize their value against a minimizing adversary.*
Team Members cannot communicate if not prescribed by the game

Team Maxmin Equilibrium

Possible Solution Concepts:

TME = *Team Players maximize their value against a minimizing adversary.*
Team Members cannot communicate if not prescribed by the game

⇒ Base case, no communication infrastructure available among players

Team Maxmin Equilibrium

Possible Solution Concepts:

TME = *Team Players maximize their value against a minimizing adversary.*
Team Members cannot communicate if not prescribed by the game

$\Rightarrow$ Base case, no communication infrastructure available among players

$\Rightarrow$ NP-Hard to find

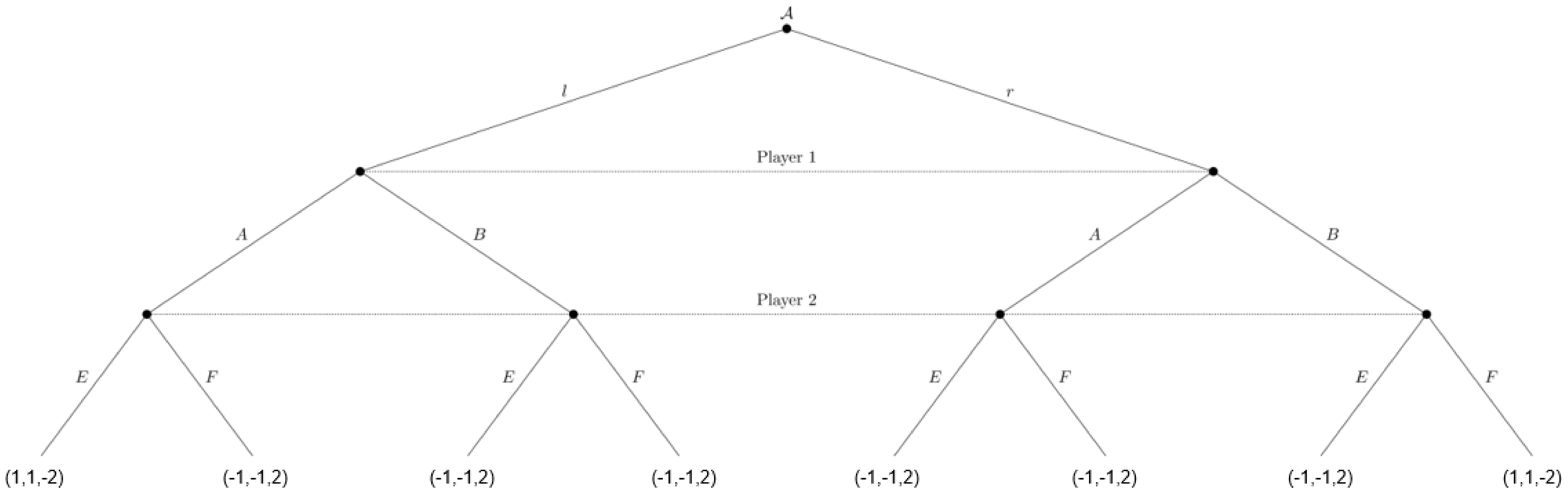
Team Maxmin Equilibrium

Possible Solution Concepts:

TME = *Team Players maximize their value against a minimizing adversary.*
Team Members cannot communicate if not prescribed by the game

⇒ Base case, no communication infrastructure available among players

⇒ NP-Hard to find



TME:
A - l:0.5 r:0.5
P1 - A:0.5 B:0.5
P2 - E:0.5 F:0.5

Team Maxmin Equilibrium

Possible Solution Concepts:

TMECor = *Team Players maximize their value against a minimizing adversary.*
Team Members can communicate before the game starts

Team Maxmin Equilibrium

Possible Solution Concepts:

TMECor = *Team Players maximize their value against a minimizing adversary.*
Team Members can communicate before the game starts

$\Rightarrow$ Coordination in randomization of actions can be achieved

Team Maxmin Equilibrium

Possible Solution Concepts:

TMECor = *Team Players maximize their value against a minimizing adversary.*
Team Members can communicate before the game starts

$\Rightarrow$ Coordination in randomization of actions can be achieved

$\Rightarrow$ Modeled as a common signal extracted at start of the game

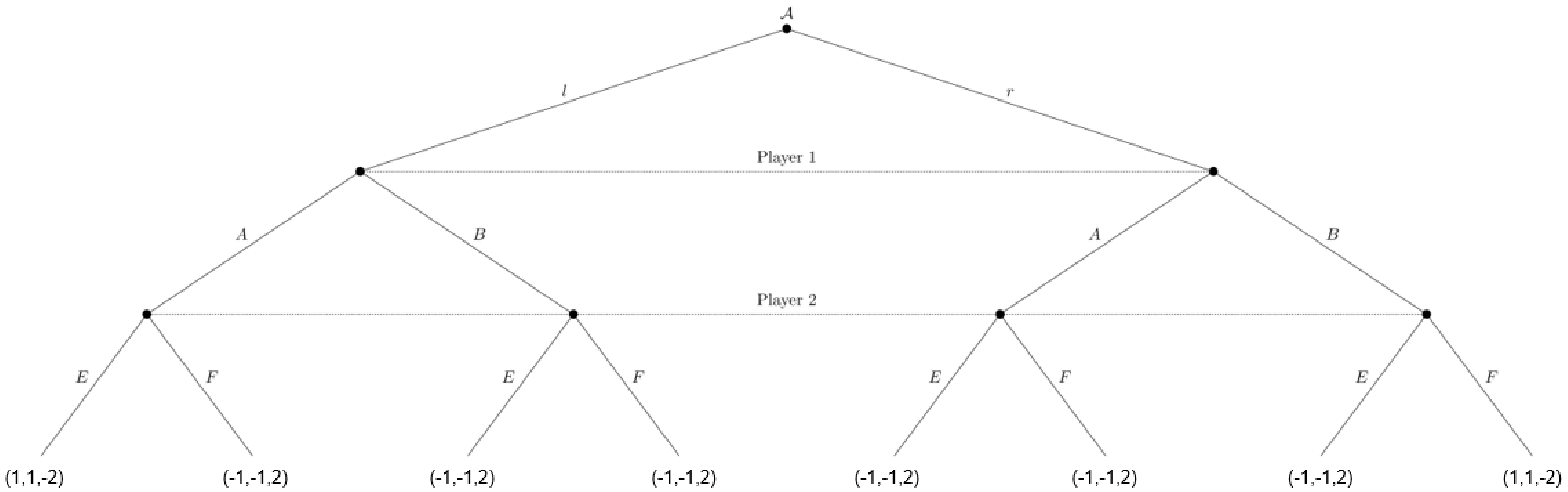
Team Maxmin Equilibrium

Possible Solution Concepts:

TMECor = *Team Players maximize their value against a minimizing adversary.*
Team Members can communicate before the game starts

⇒ Coordination in randomization of actions can be achieved

⇒ Modeled as a common signal extracted at start of the game



TMECor:
A - l:0.5 r:0.5
S - s1: 0.5 s2:0.5

P1 - A if s1, B if s2
P2 – E if s1, F if s2

Team Maxmin Equilibrium

Possible Solution Concepts:

TMEComm = *Team Players maximize their value against a minimizing adversary.*
Team Members can communicate at any moment

Team Maxmin Equilibrium

Possible Solution Concepts:

TMEComm = *Team Players maximize their value against a minimizing adversary.*
Team Members can communicate at any moment

$\Rightarrow$ Observations are completely shared

Team Maxmin Equilibrium

Possible Solution Concepts:

TMEComm = *Team Players maximize their value against a minimizing adversary.*
Team Members can communicate at any moment

$\Rightarrow$ Observations are completely shared

$\Rightarrow$ Team Players can effectively operate as a single player

Team Maxmin Equilibrium

Possible Solution Concepts:

TMEComm = *Team Players maximize their value against a minimizing adversary.*
Team Members can communicate at any moment

⇒ Observations are completely shared

⇒ Team Players can effectively operate as a single player

⇒ Algorithms for 2 players zero sum games can be employed

Team Maxmin Equilibrium

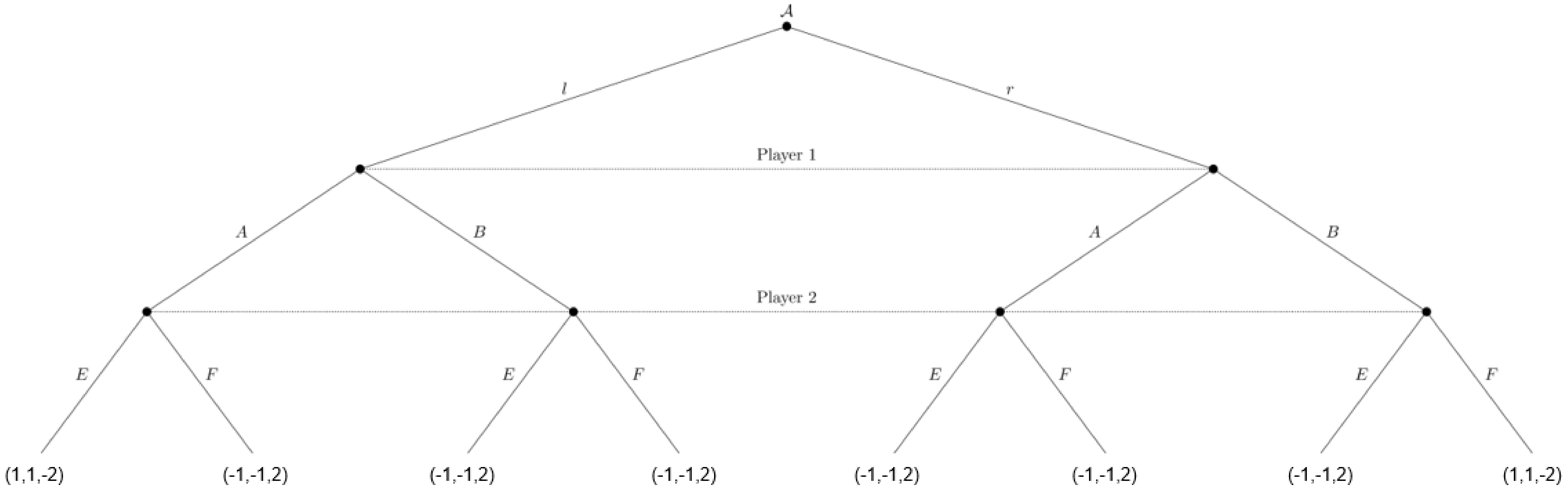
Possible Solution Concepts:

TMEComm *= Team Players maximize their value against a minimizing adversary.*
Team Members can communicate at any moment

⇒ Observations are completely shared

⇒ Team Players can effectively operate as a single player

⇒ Algorithms for 2 players zero sum games can be employed



TMEComm:
A - l:0.5 r:0.5
P1+P2 - AE:0.5 BF:0.5

Team Maxmin Equilibrium

Considerations:

Team Maxmin Equilibrium

Considerations:

Communication can be really useful in team games
From [Celli and Gatti, 2017]:
Value provided by a TMECor can be arbitrarily larger with respect to a TME
Value provided by a TMEComm can be arbitrarily larger with respect to a TMECor

Team Maxmin Equilibrium

Considerations:

Communication can be really useful in team games

From [Celli and Gatti, 2017]:

Value provided by a TMECor can be arbitrarily larger with respect to a TME

Value provided by a TMEComm can be arbitrarily larger with respect to a TMECor

Pre Game coordination across team members is easily implementable in many contexts

Team Maxmin Equilibrium

Considerations:

Communication can be really useful in team games
From [Celli and Gatti, 2017]:
Value provided by a TMECor can be arbitrarily larger with respect to a TME
Value provided by a TMEComm can be arbitrarily larger with respect to a TMECor

Pre Game coordination across team members is easily implementable in many contexts

In Game communication across team members cannot be implemented in many contexts

Team Maxmin Equilibrium

Considerations:

Communication can be really useful in team games

From [Celli and Gatti, 2017]:

Value provided by a TMECor can be arbitrarily larger with respect to a TME

Value provided by a TMEComm can be arbitrarily larger with respect to a TMECor

Pre Game coordination across team members is easily implementable in many contexts

In Game communication across team members cannot be implemented in many contexts

Examples:

Card Games, Security Scenarios

Scopone scientifico, Bridge, Briscola

Coordinated Micromanagement of agents

Coordinated Macromanagement of resources

Team Maxmin Equilibrium

Considerations:

Communication can be really useful in team games

From [Celli and Gatti, 2017]:

Value provided by a TMECor can be arbitrarily larger with respect to a TME

Value provided by a TMEComm can be arbitrarily larger with respect to a TMECor

Pre Game coordination across team members is easily implementable in many contexts

In Game communication across team members cannot be implemented in many contexts

Examples:

Card Games, Security Scenarios

Scopone scientifico, Bridge, Briscola
Coordinated Micromanagement of agents
Coordinated Macromanagement of resources

Can we find a TMECor for a given Team Game?

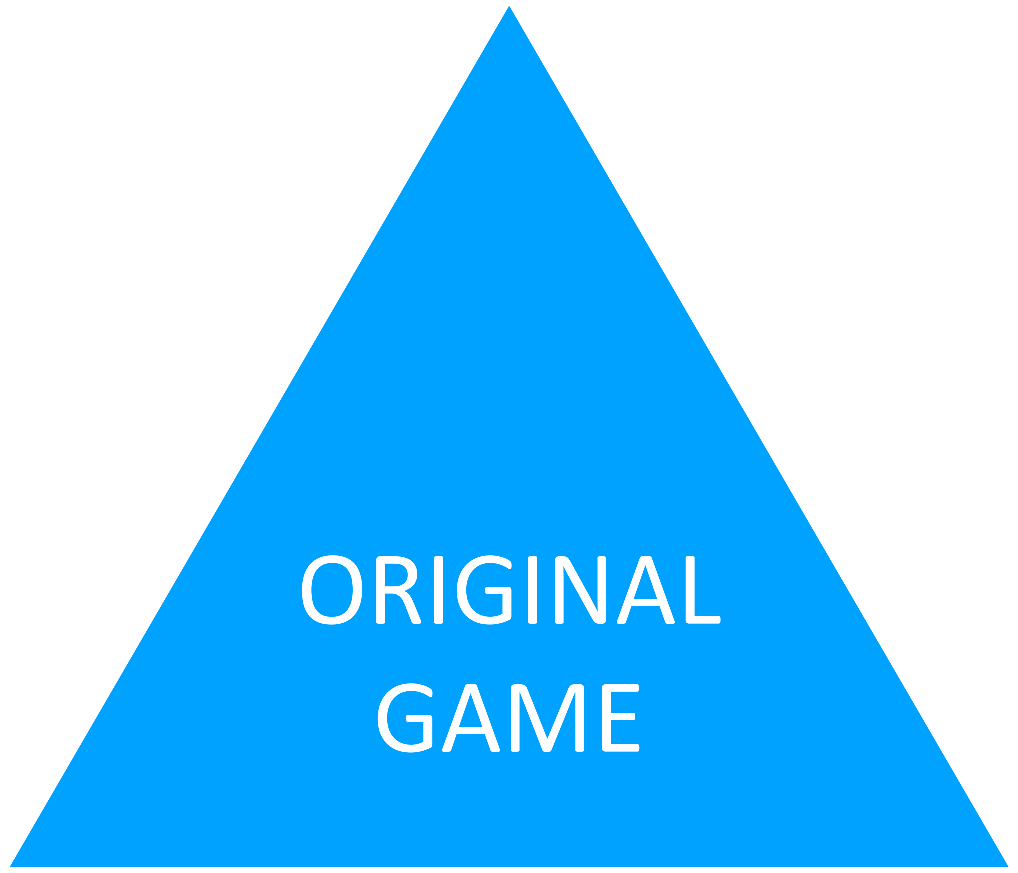
Algorithms for TME

Algorithms for TME

Hybrid Column Generation [Celli and Gatti, 2017]
= Two LPs formulated on a progressively larger hybrid formulation of the game
= Integer LP oracle to find the next Joint Strategy to add

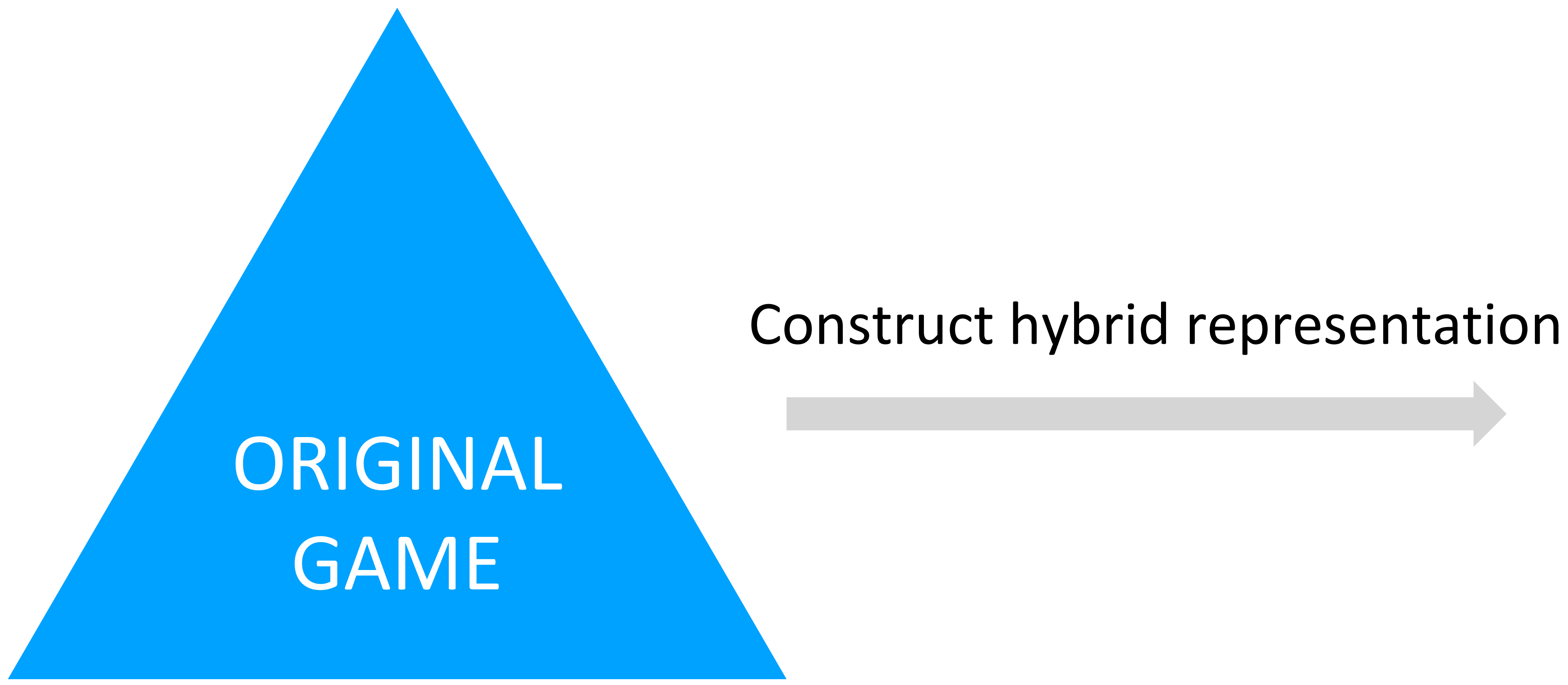
Algorithms for TME

Hybrid Column Generation [Celli and Gatti, 2017]
= *Two LPs formulated on a progressively larger hybrid formulation of the game*
= *Integer LP oracle to find the next Joint Strategy to add*



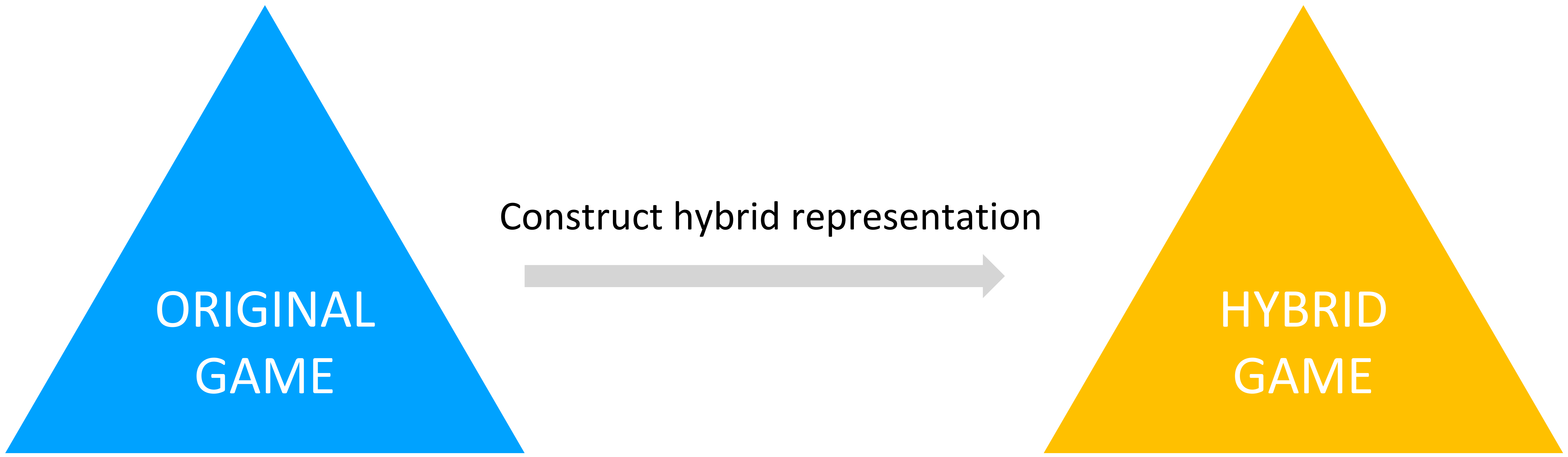
Algorithms for TME

Hybrid Column Generation [Celli and Gatti, 2017]
= Two LPs formulated on a progressively larger hybrid formulation of the game
= Integer LP oracle to find the next Joint Strategy to add



Algorithms for TME

Hybrid Column Generation [Celli and Gatti, 2017]
= *Two LPs formulated on a progressively larger hybrid formulation of the game*
= *Integer LP oracle to find the next Joint Strategy to add*

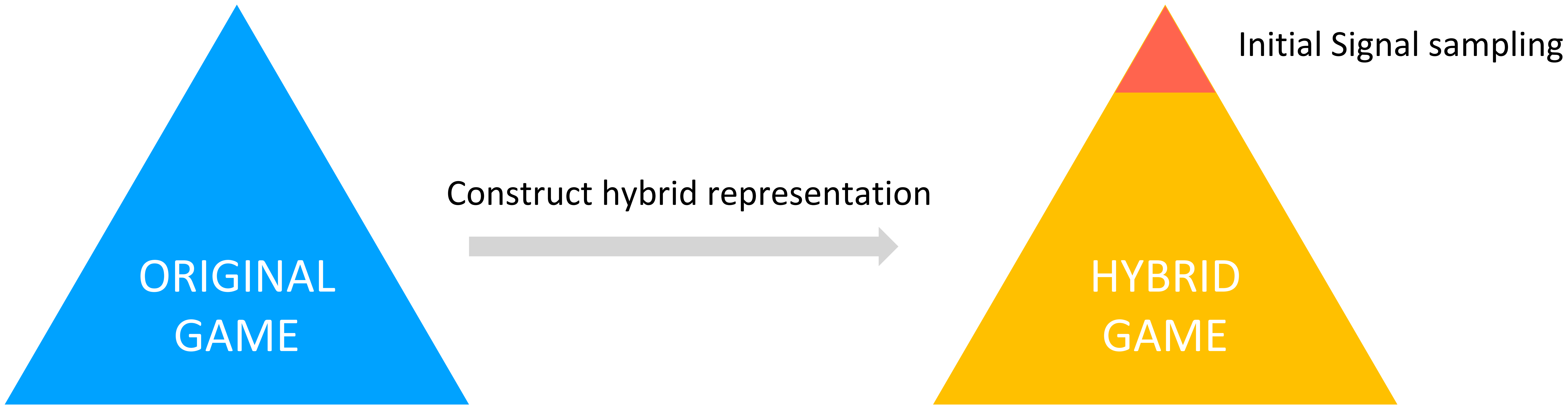


Algorithms for TME

Hybrid Column Generation [Celli and Gatti, 2017]

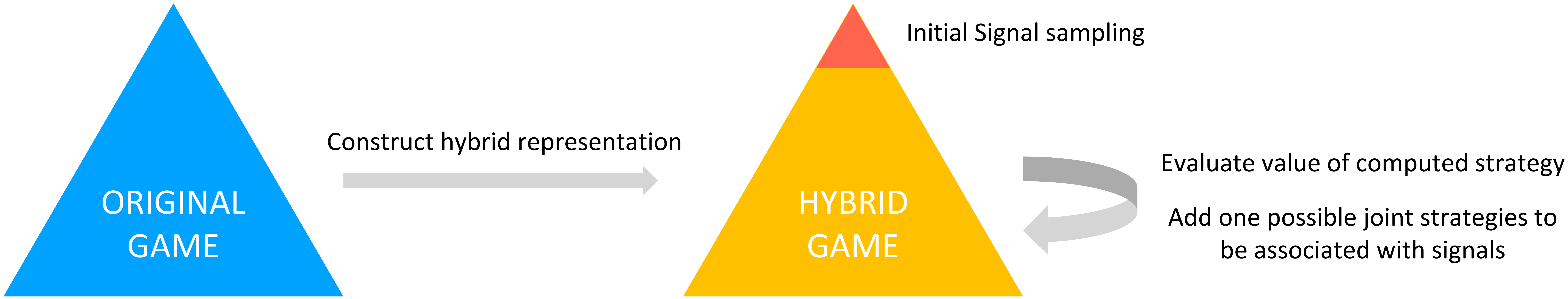
= Two LPs formulated on a progressively larger hybrid formulation of the game

= Integer LP oracle to find the next Joint Strategy to add



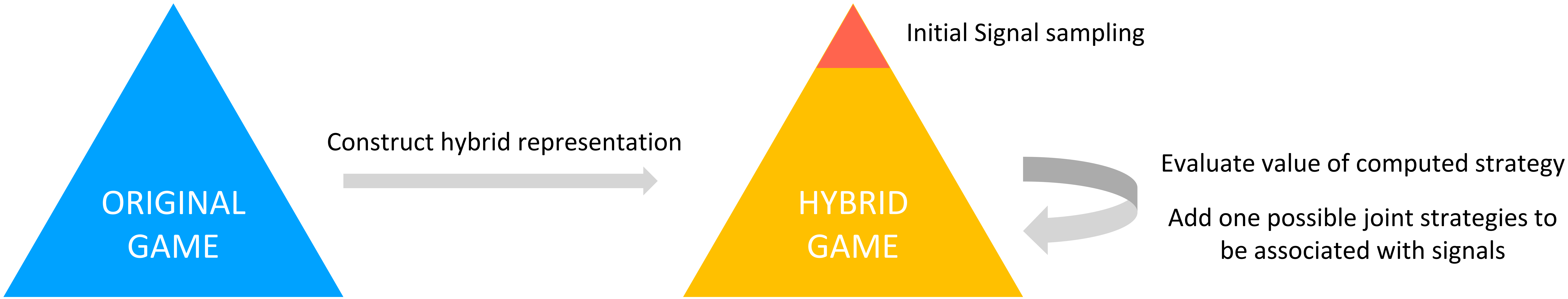
Algorithms for TME

Hybrid Column Generation [Celli and Gatti, 2017]
= *Two LPs formulated on a progressively larger hybrid formulation of the game*
= *Integer LP oracle to find the next Joint Strategy to add*



Algorithms for TME

Hybrid Column Generation [Celli and Gatti, 2017]
= *Two LPs formulated on a progressively larger hybrid formulation of the game*
= *Integer LP oracle to find the next Joint Strategy to add*



✓ *Approximation can be obtained by relaxing binary constraints of BR oracle*
✗ *Integer LP limits scalability*

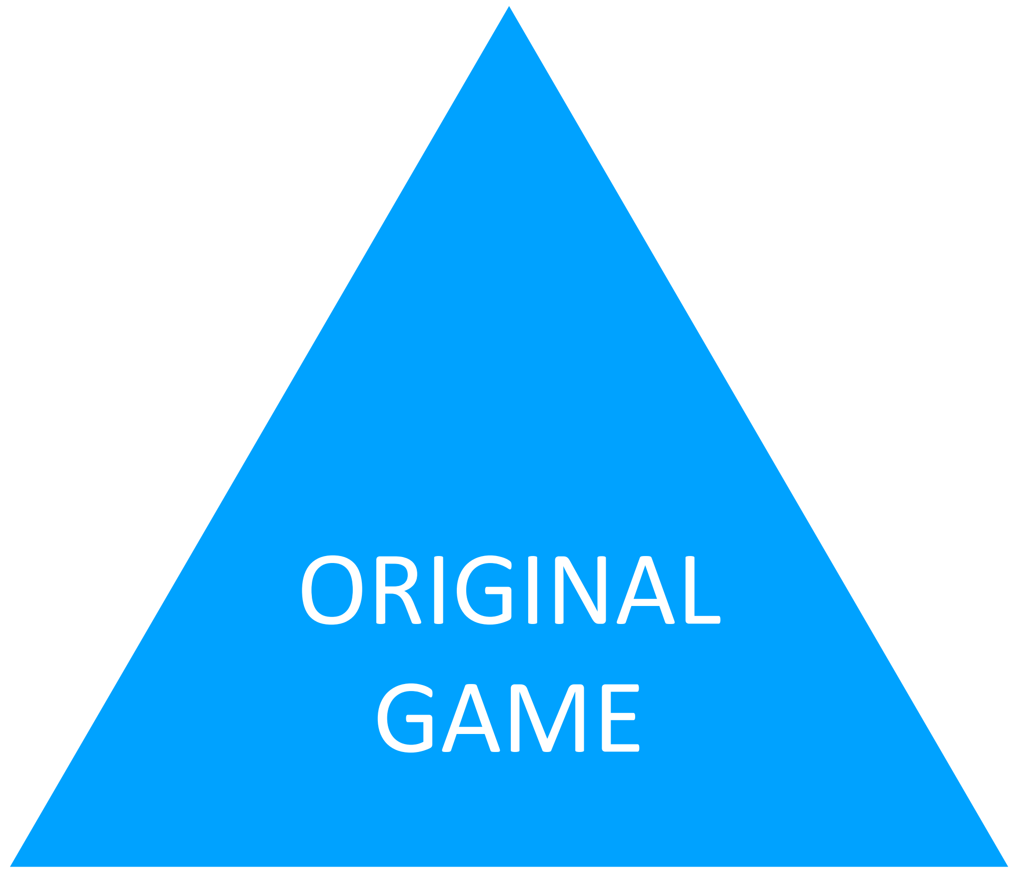
Algorithms for TME

Algorithms for TME

Fictitious Team Play [Farina and Celli, 2018]
= Iterative Best Response computation to average strategy of adversary
= Best Response as an MILP

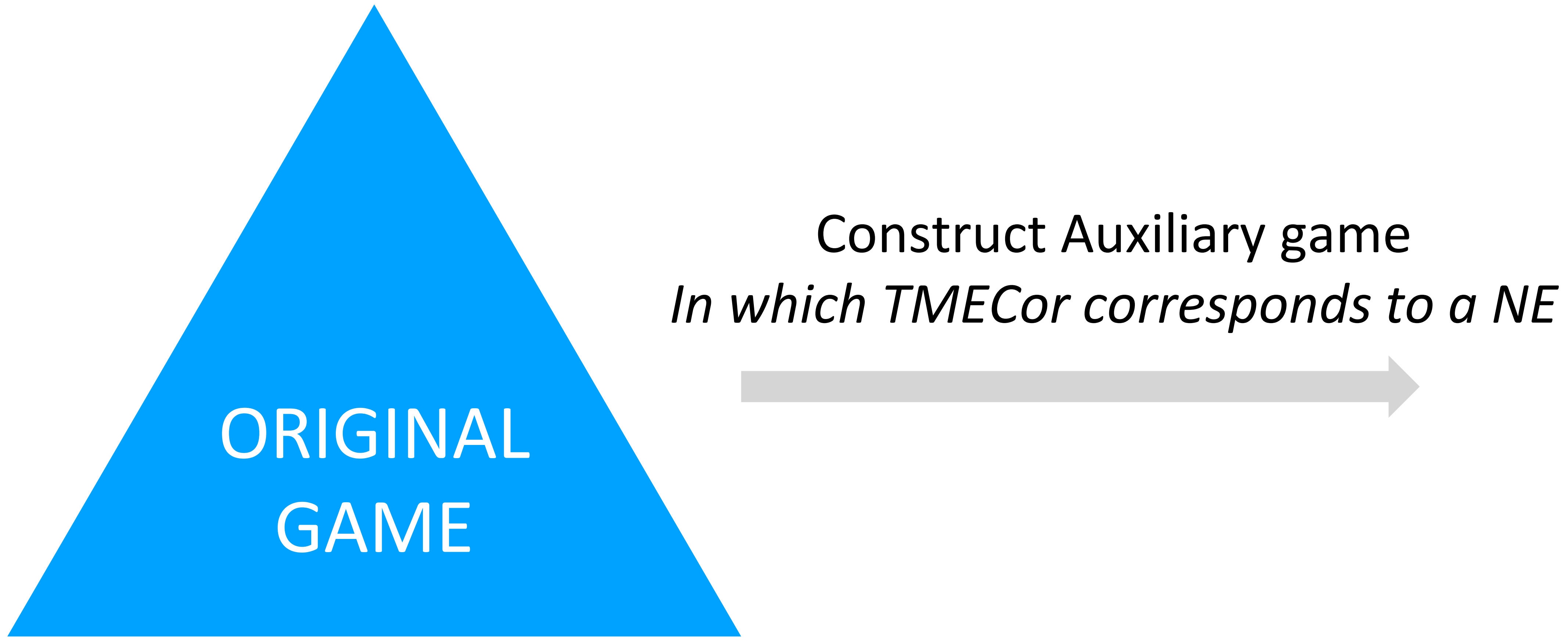
Algorithms for TME

Fictitious Team Play [Farina and Celli, 2018]
= Iterative Best Response computation to average strategy of adversary
= Best Response as an MILP



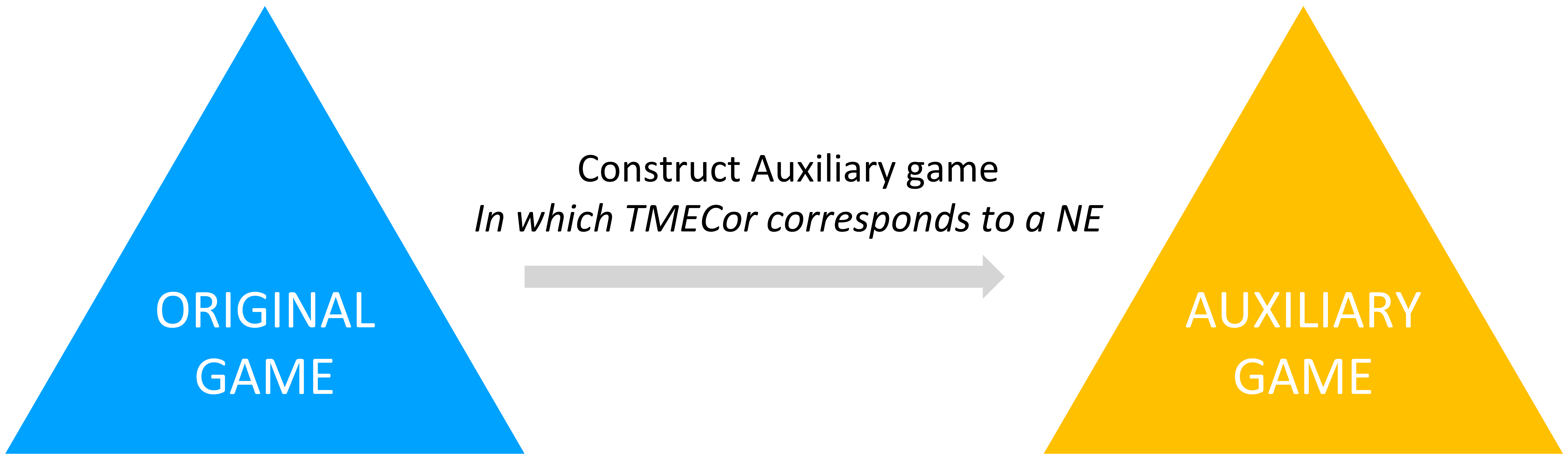
Algorithms for TME

Fictitious Team Play [Farina and Celli, 2018]
= Iterative Best Response computation to average strategy of adversary
= Best Response as an MILP



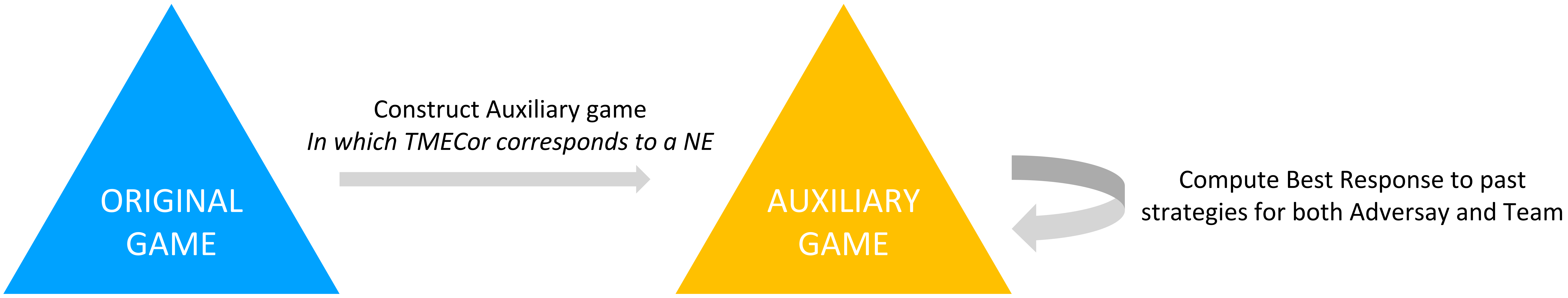
Algorithms for TME

Fictitious Team Play [Farina and Celli, 2018]
= *Iterative Best Response computation to average strategy of adversary*
= *Best Response as an MILP*



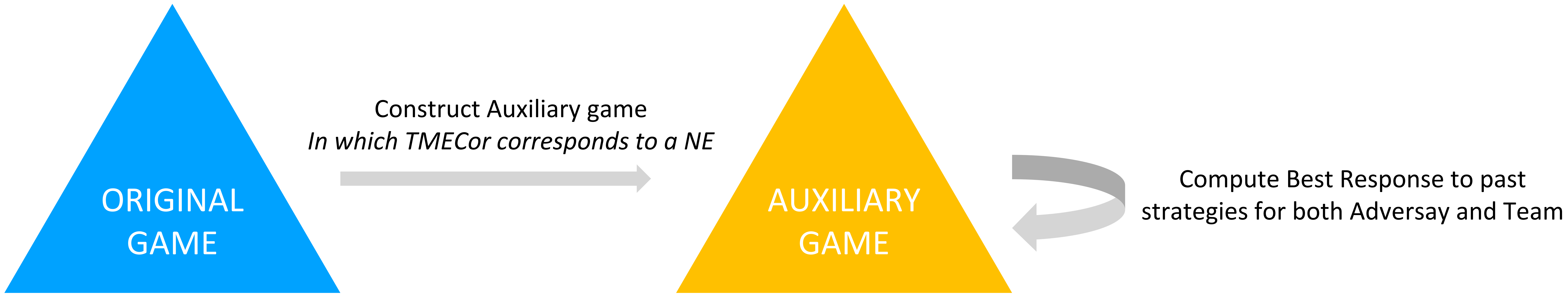
Algorithms for TME

Fictitious Team Play [Farina and Celli, 2018]
= *Iterative Best Response computation to average strategy of adversary*
= *Best Response as an MILP*



Algorithms for TME

Fictitious Team Play [Farina and Celli, 2018]
= Iterative Best Response computation to average strategy of adversary
= Best Response as an MILP



✓ *Faster than HCG*
✗ *Slower empirical convergence rate of FP*
✗ *MILP limits scalability*

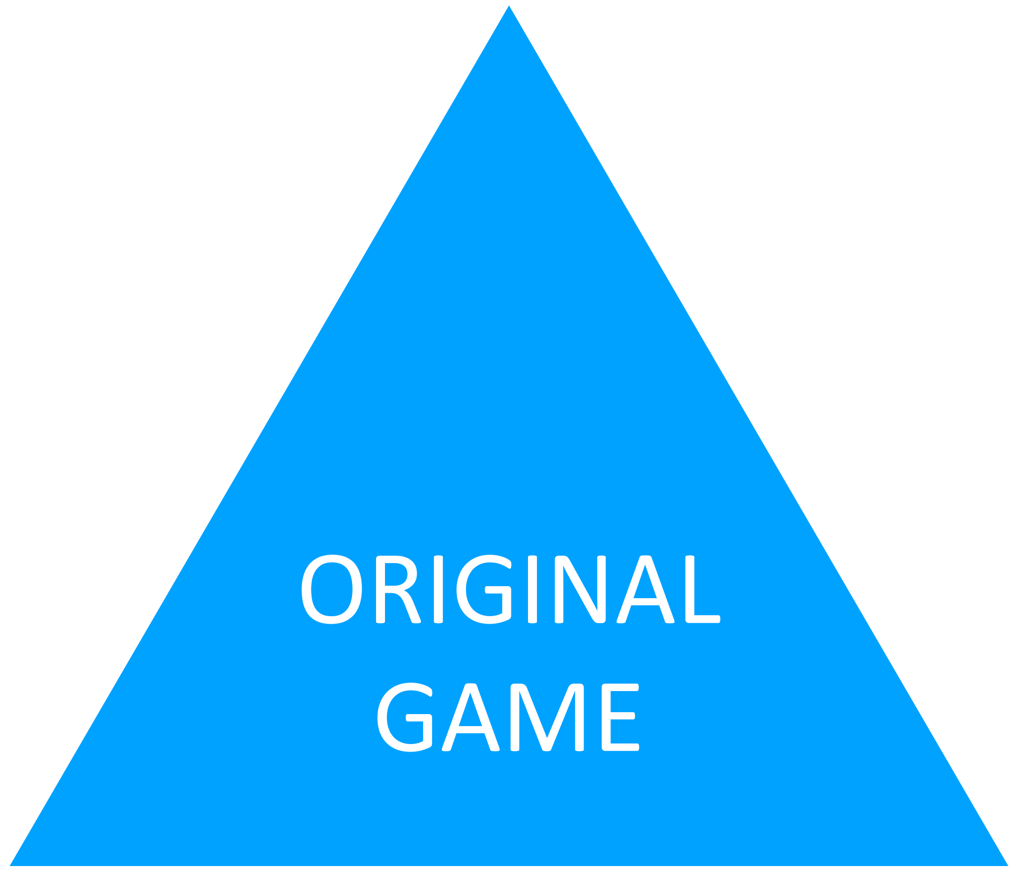
Algorithms for TME

Algorithms for TME

Soft Team Actor Critic [Celli et al, 2019]
= Iterative gradient descent over the space of possible parameters
= Actor-Critic RL Framework

Algorithms for TME

Soft Team Actor Critic [Celli et al, 2019]
= Iterative gradient descent over the space of possible parameters
= Actor-Critic RL Framework

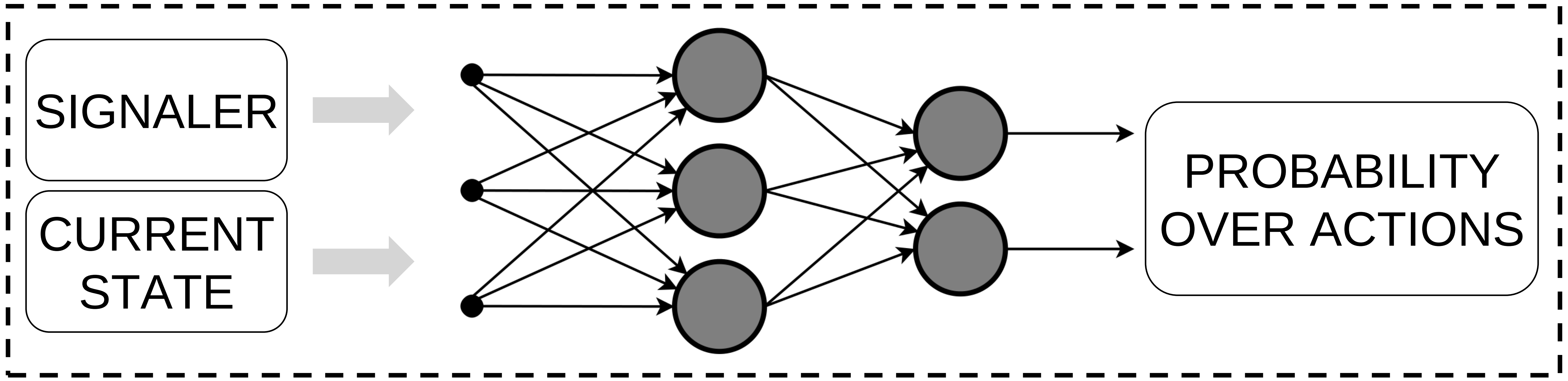


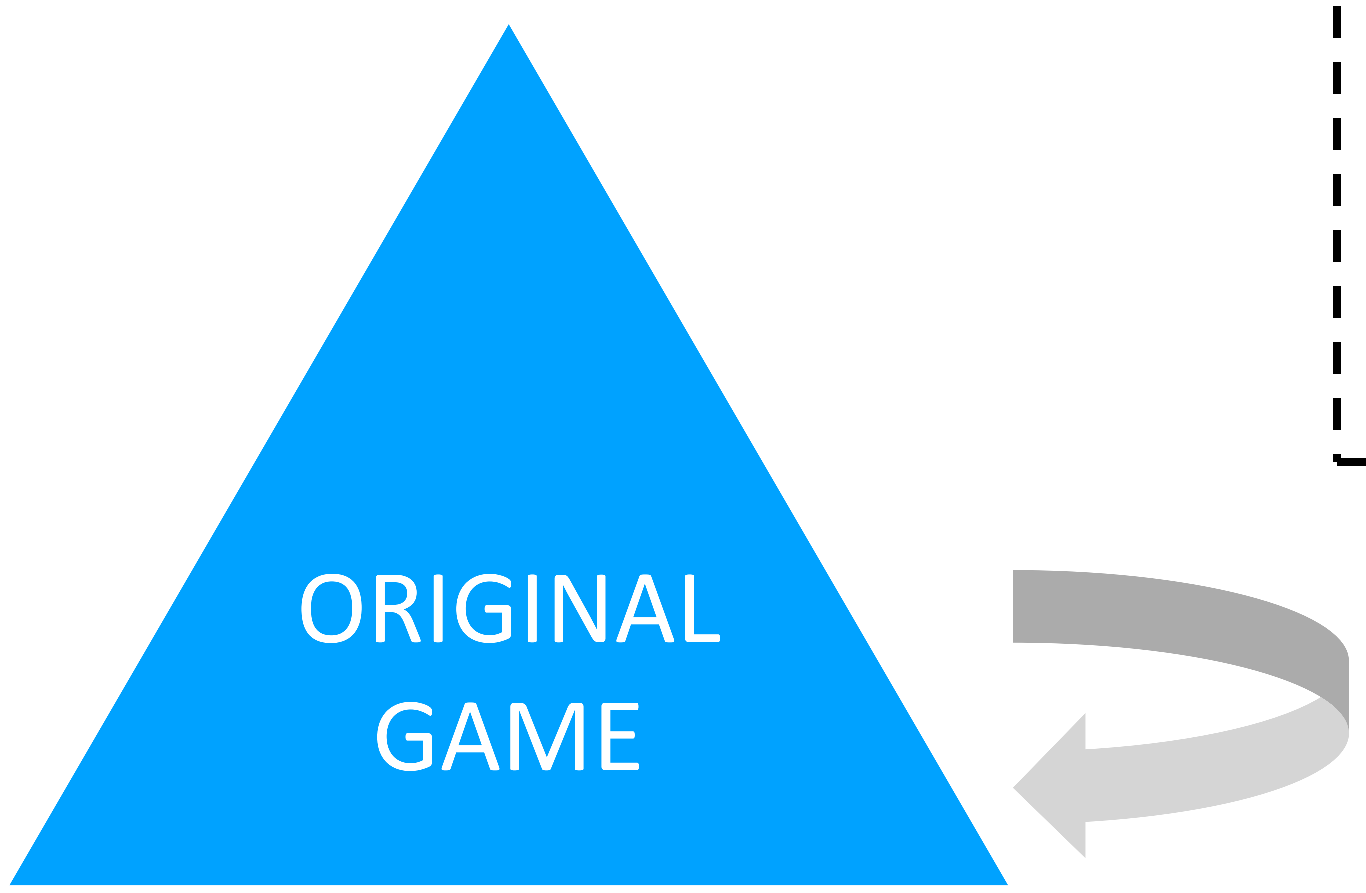
Algorithms for TME

Soft Team Actor Critic [Celli et al, 2019]

= Iterative gradient descent over the space of possible parameters

= Actor-Critic RL Framework

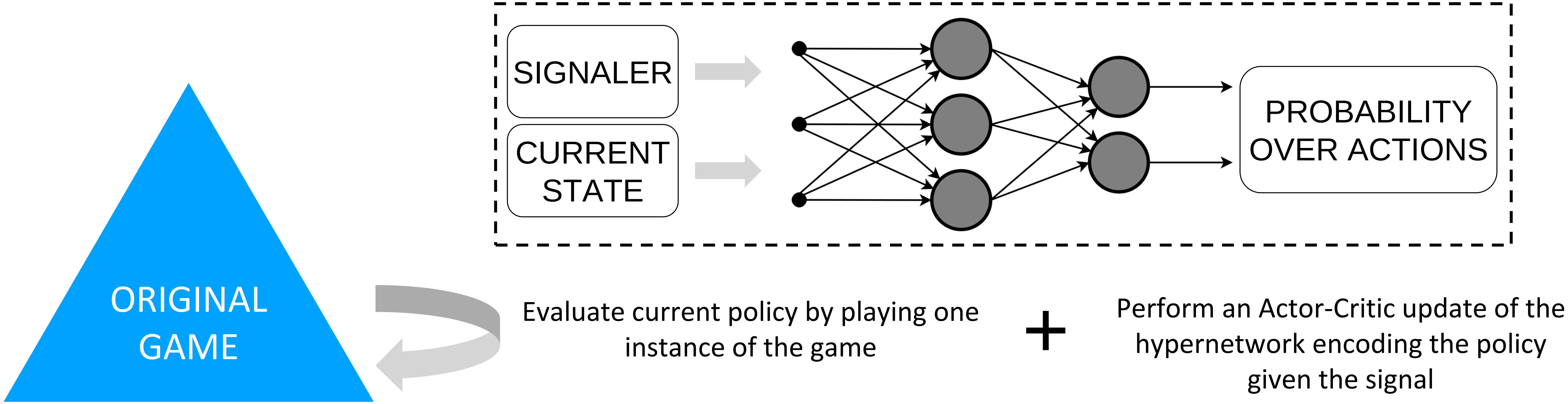




Evaluate current policy by playing one instance of the game + Perform an Actor-Critic update of the hypernetwork encoding the policy given the signal

Algorithms for TME

Soft Team Actor Critic [Celli et al, 2019]
= Iterative gradient descent over the space of possible parameters
= Actor-Critic RL Framework



✓ *No requirements of model available ⇒ no manipulation of original game needed*
✗ *Fixed number of uniform signals ⇒ no guarantees of convergence*
✗ *No Robustness of result due to noise from fixed uniform signals and gradient descent*

Algorithms for TME

Algorithms for TME

Signal Mediated Strategies [Cacciamani et al, 2020]

= Centralized Training merging team players and creating a joint strategy

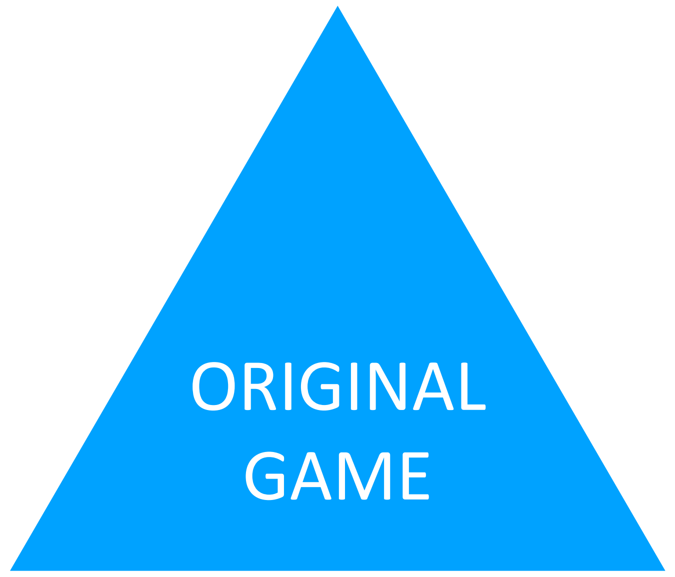
= Learn marginalized policies for decentralized execution conditioned by a signal

Algorithms for TME

Signal Mediated Strategies [Cacciamani et al, 2020]

= *Centralized Training merging team players and creating a joint strategy*

= *Learn marginalized policies for decentralized execution conditioned by a signal*

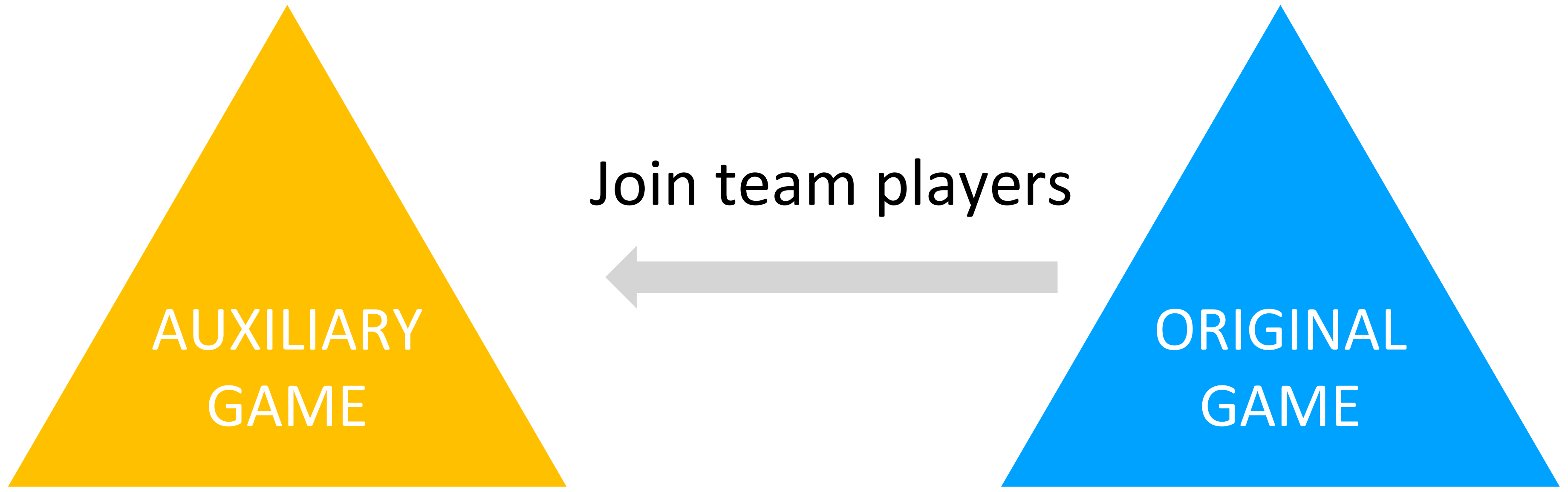


Algorithms for TME

Signal Mediated Strategies [Cacciamani et al, 2020]

= Centralized Training merging team players and creating a joint strategy

= Learn marginalized policies for decentralized execution conditioned by a signal

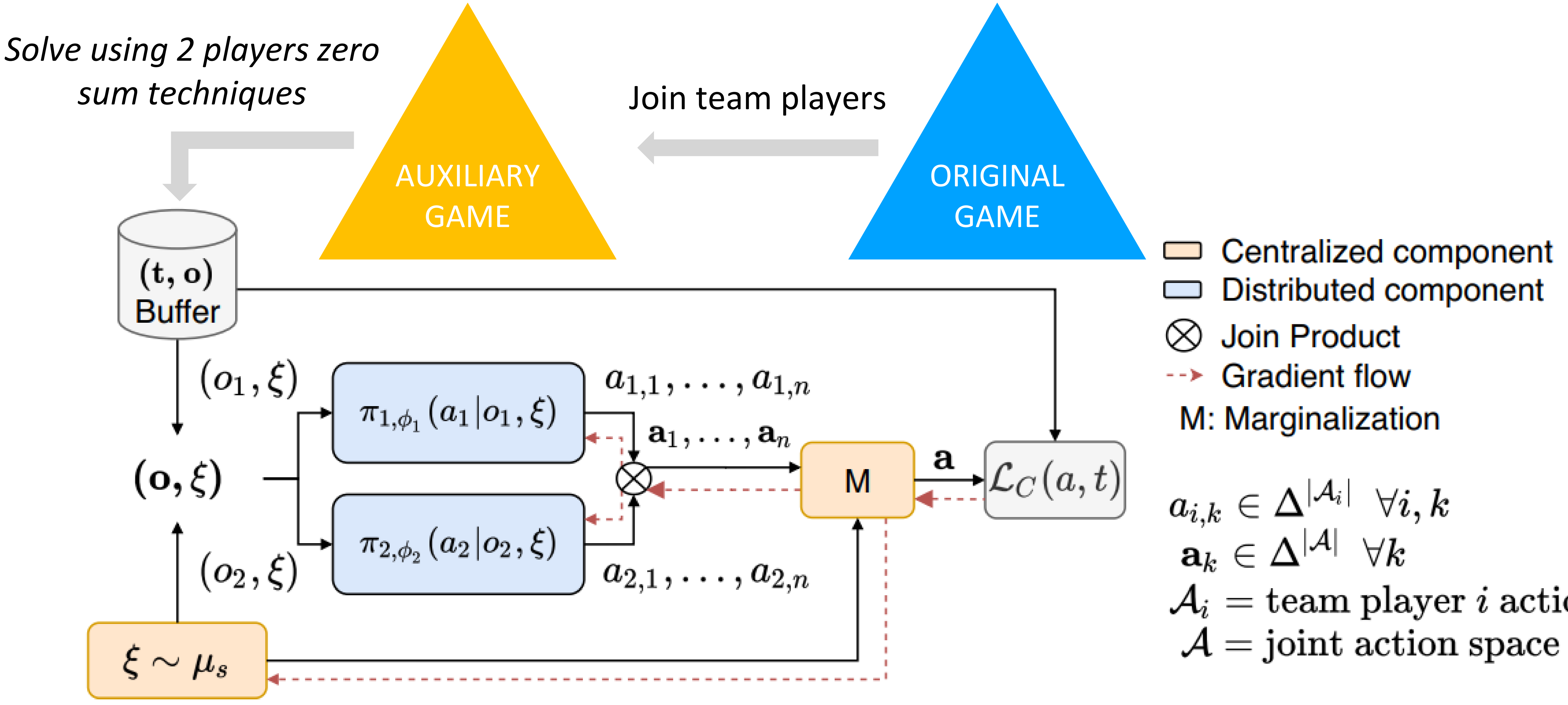


Algorithms for TME

Signal Mediated Strategies [Cacciamani et al, 2020]

= Centralized Training merging team players and creating a joint strategy

= Learn marginalized policies for decentralized execution conditioned by a signal



$a_{i,k} \in \Delta^{|\mathcal{A}_i|} \quad \forall i, k$

$\mathbf{a}_k \in \Delta^{|\mathcal{A}|} \quad \forall k$

$\mathcal{A}_i =$ team player i action space

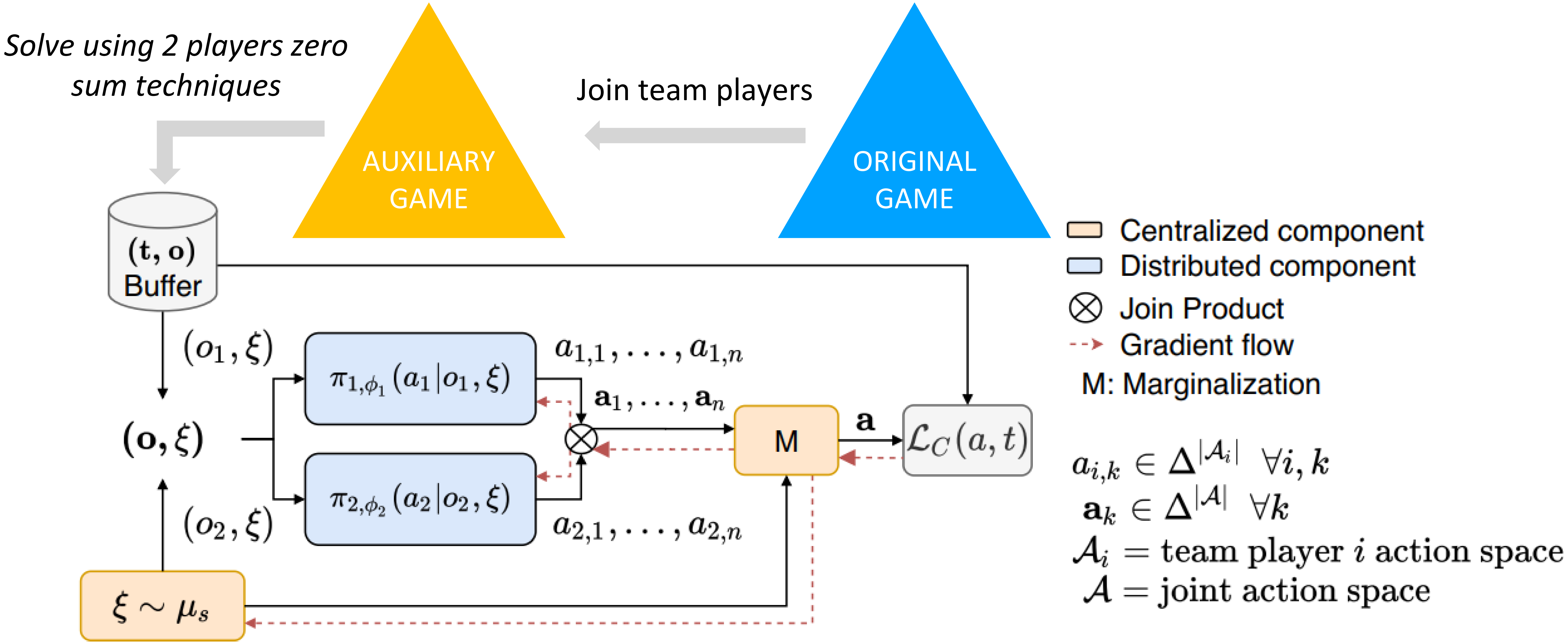
$\mathcal{A} =$ joint action space

Algorithms for TME

Signal Mediated Strategies [Cacciamani et al, 2020]

= Centralized Training merging team players and creating a joint strategy

= Learn marginalized policies for decentralized execution conditioned by a signal



✓ *Model free but convergence to a TMECor guaranteed*

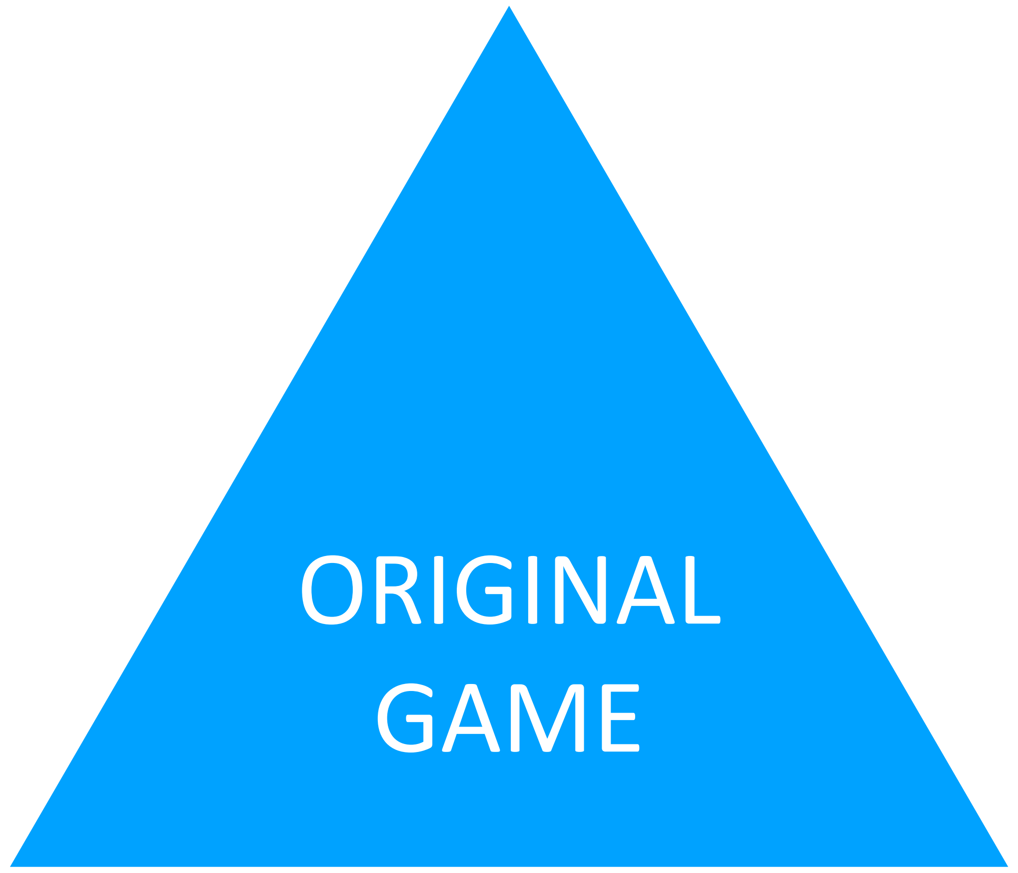
✗ *Strong Assumptions on game structure*

Overview

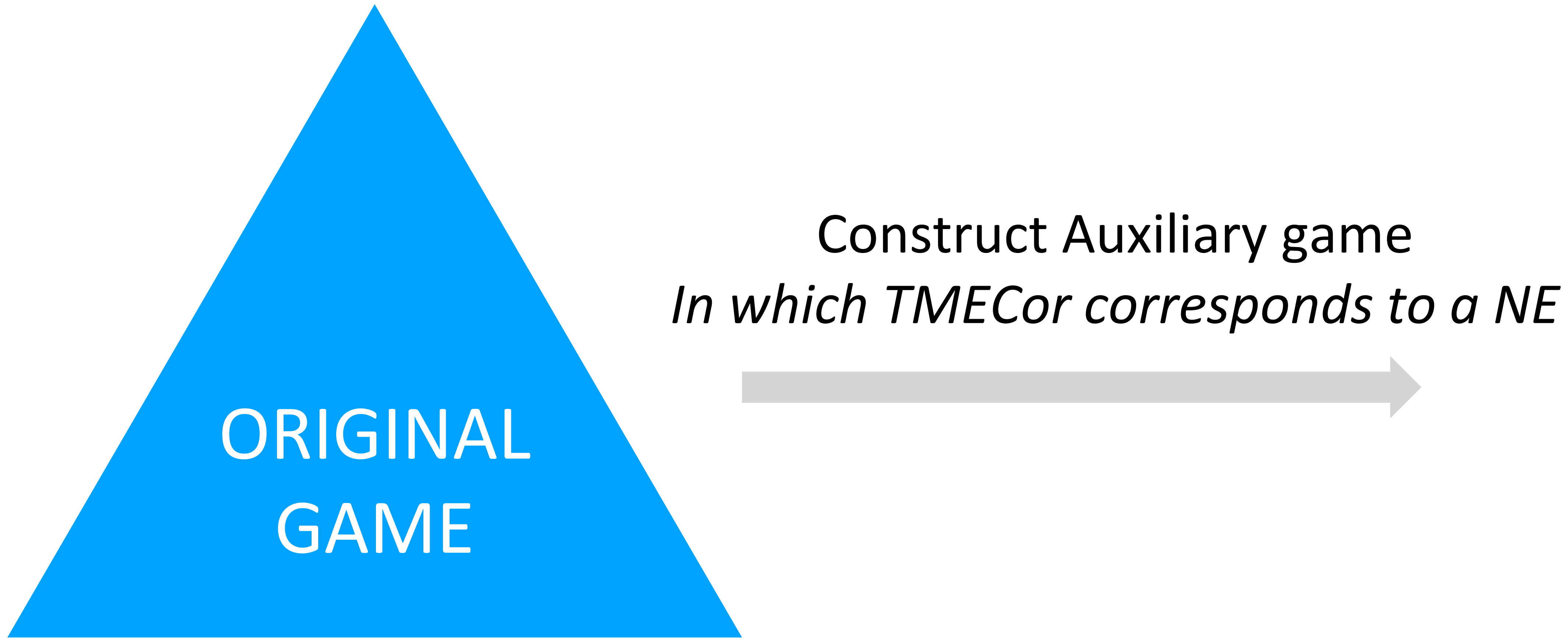
1. Introduction to Algorithmic Game Theory
2. Main Questions
3. Preliminaries
4. State of the art
5. Project proposal

Our Proposed Approach

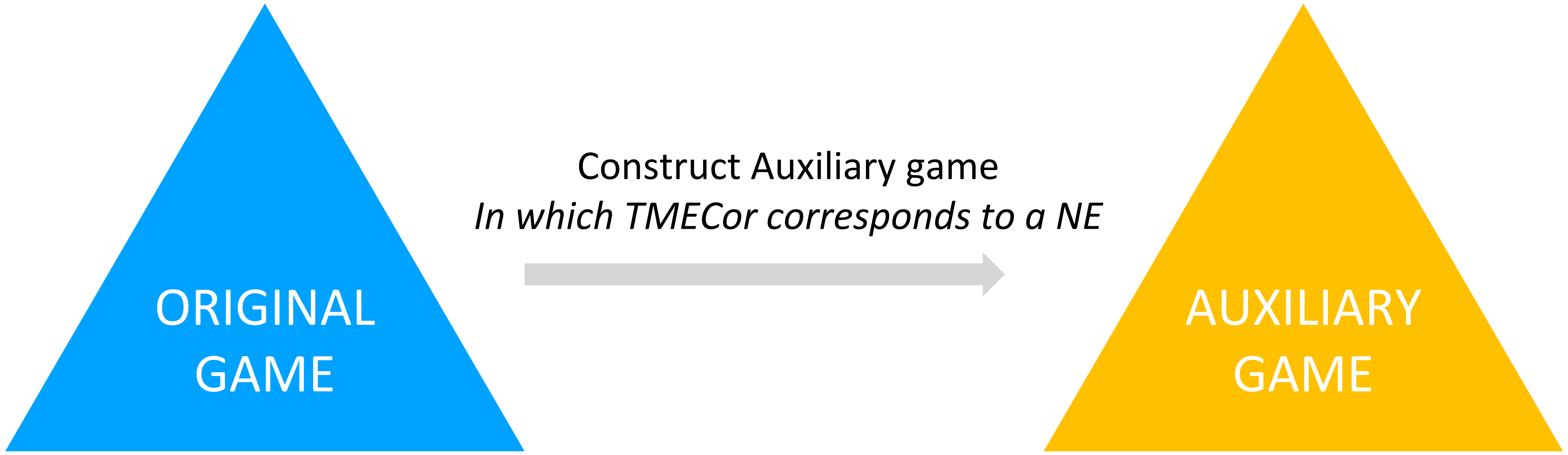
Our Proposed Approach



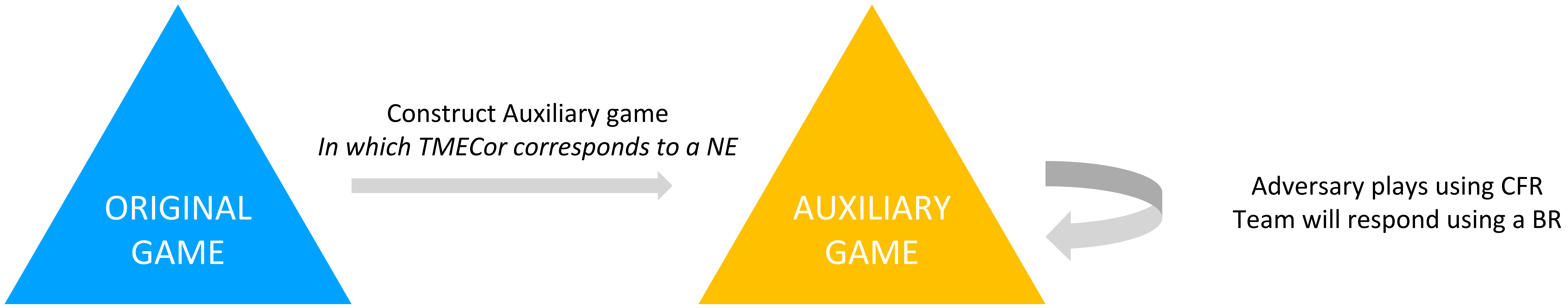
Our Proposed Approach



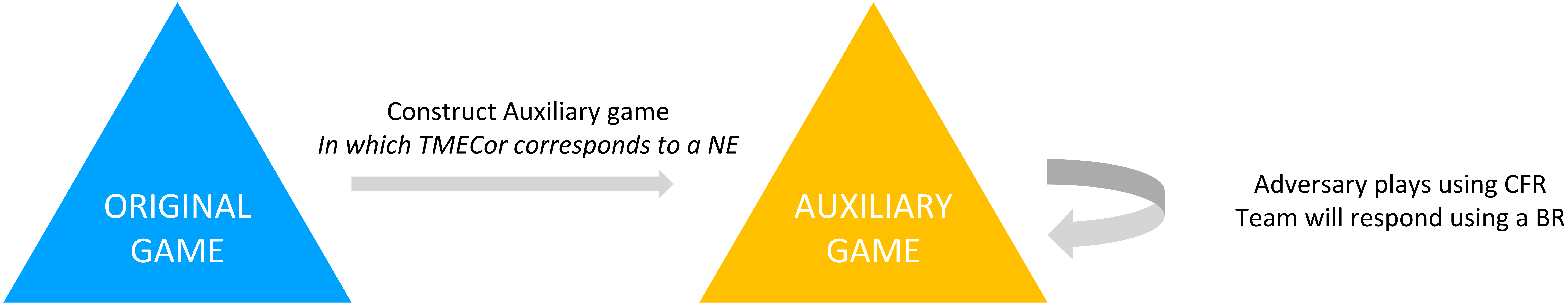
Our Proposed Approach



Our Proposed Approach



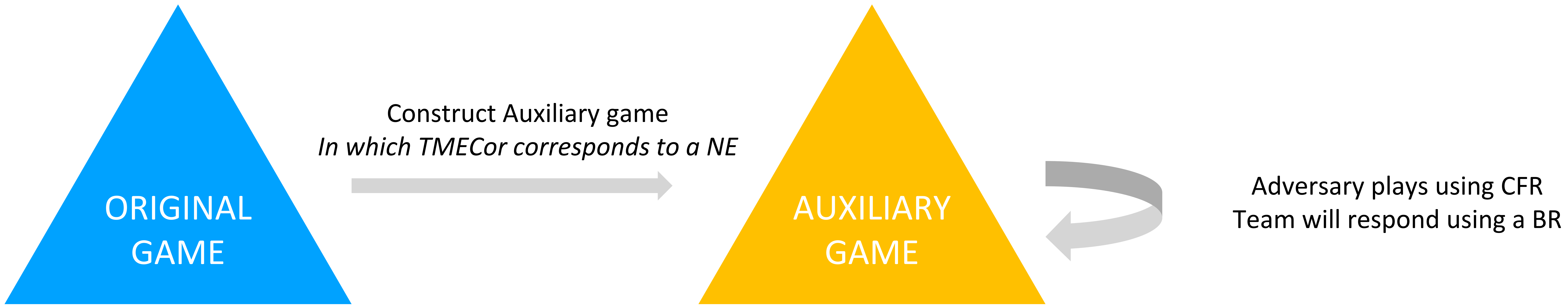
Our Proposed Approach



Build on Top of Auxiliary Game Framework used in Fictitious Team Play
BUT

- Employ ***CFR-BR*** to have a faster convergence rate in place of FP
- Use an approximated RL approach with fewer guarantees to solve BR problem

Our Proposed Approach



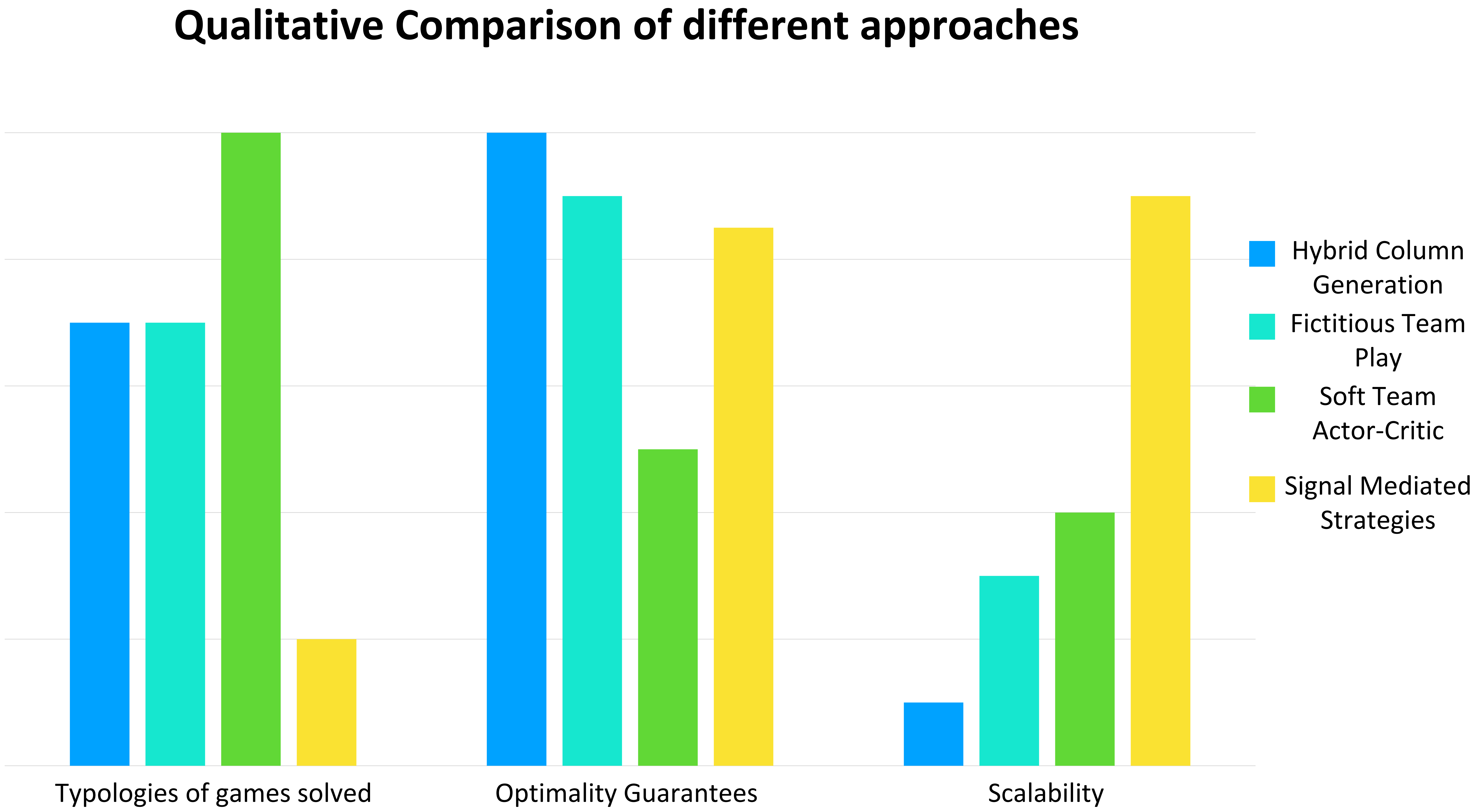
Build on Top of Auxiliary Game Framework used in Fictitious Team Play

BUT

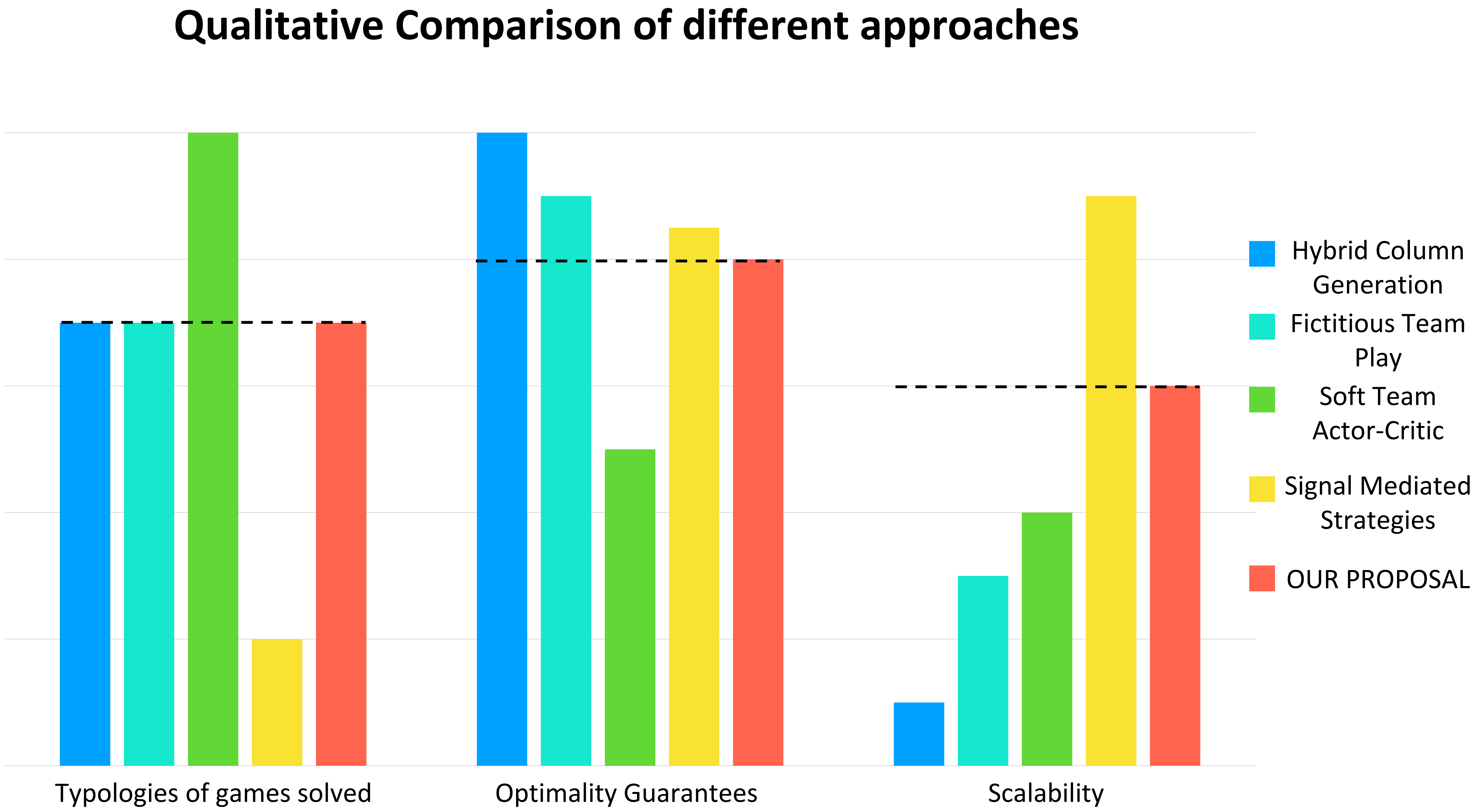
- Employ ***CFR-BR*** to have a faster convergence rate in place of FP
- Use an approximated RL approach with fewer guarantees to solve BR problem

$\Rightarrow$ Probabilistic Guarantee of convergence

Our Proposed Approach



Our Proposed Approach



Our Proposed Approach

Validation Procedures:

Our Proposed Approach

Validation Procedures:

- Comparison of approximate Team-BR procedures on Random Games:

  → *Mixed ILP formulation of HCG and FTP*
  → *Approximate BR with Iterated LP*
  → *Approximate BR using RL*

Our Proposed Approach

Validation Procedures:

- Comparison of approximate Team-BR procedures on Random Games:

  → *Mixed ILP formulation of HCG and FTP*
  → *Approximate BR with Iterated LP*
  → *Approximate BR using RL*

- Comparison with Fictitious Team Play and Hybrid Column Generation algorithms:

  → *Kuhn Poker 2vs1* → *for small environment and preliminary results [16 infosets per player]*

  → *Leduc Poker 2vs1* → *for more extensive environment, and testing scalability capabilities by changing number of cards [456 infosets per player]*

Thanks for the attention!

Any Question?