

Strategy Synthesis for Zero-Sum Neuro-Symbolic Concurrent Stochastic Games

Rui Yan^a, Gabriel Santos^a, Gethin Norman^{a,b}, David Parker^a,
Marta Kwiatkowska^a

^a*Department of Computer Science, University of Oxford, Oxford, OX1 2JD, UK*

^b*School of Computing Science, University of Glasgow, Glasgow, G12 8QQ, UK*

Abstract

Neuro-symbolic approaches to artificial intelligence, which combine neural networks with classical symbolic techniques, are growing in prominence, necessitating formal approaches to reason about their correctness. We propose a novel modelling formalism called neuro-symbolic concurrent stochastic games (NS-CSGs), which comprise two probabilistic finite-state agents interacting in a shared continuous-state environment. Each agent observes the environment using a neural perception mechanism, which converts inputs such as images into symbolic percepts, and makes decisions symbolically. We focus on the class of NS-CSGs with Borel state spaces and prove the existence and measurability of the value function for zero-sum discounted cumulative rewards under piecewise-constant restrictions on the components of this class of models. To compute values and synthesise strategies, we present, for the first time, practical value iteration (VI) and policy iteration (PI) algorithms to solve this new subclass of continuous-state CSGs. These require a finite decomposition of the environment induced by the neural perception mechanisms of the agents and rely on finite abstract representations of value functions and strategies closed under VI or PI. First, we introduce a Borel measurable piecewise-constant (B-PWC) representation of value functions, extend minimax backups to this representation and propose a value iteration algorithm called B-PWC VI. Second, we introduce

Email addresses: `rui.yan@cs.ox.ac.uk` (Rui Yan), `gabriel.santos@cs.ox.ac.uk` (Gabriel Santos), `gethin.norman@glasgow.ac.uk` (Gethin Norman), `david.parker@cs.ox.ac.uk` (David Parker), `marta.kwiatkowska@cs.ox.ac.uk` (Marta Kwiatkowska)

two novel representations for the value functions and strategies, constant-piecewise-linear (CON-PWL) and constant-piecewise-constant (CON-PWC) respectively, and propose Minimax-action-free PI by extending a recent PI method based on alternating player choices for finite state spaces to Borel state spaces, which does not require normal-form games to be solved. We illustrate our approach with a dynamic vehicle parking example by generating approximately optimal strategies using a prototype implementation of the B-PWC VI algorithm.

Keywords: Stochastic games, neuro-symbolic systems, value iteration, policy iteration, Borel state spaces

1. Introduction

Game theory offers an attractive framework for analysing strategic interactions among agents, with application to, for instance, the game of Go [1], autonomous driving [2] and robotics [3]. An important class of dynamic games is *stochastic games* [4], which move between states according to transition probabilities controlled jointly by multiple agents (also called players). Extending both strategic-form games to dynamic environments and Markov decision processes (MDPs) to multiple players, stochastic games have long been used to model sequential decision-making problems with more than one agent, ranging from multi-agent reinforcement learning [5] to quantitative verification and synthesis for equilibria [6].

Increasingly, agents in these settings incorporate machine learning components. Recent years have witnessed encouraging advances in the use of neural networks (NNs) to approximate either value functions or strategies [7] for stochastic games that model large, complex environments. Such *end-to-end* NNs directly map environment states to Q-values or actions. This means that they have a relatively complex structure and a large number of weights and biases, since they interweave multiple tasks (e.g., object detection and recognition, decision making) within a single NN. An emerging trend in autonomous and robotic systems is *neuro-symbolic* approaches, where some components that are synthesized from data (e.g., perception modules) are implemented as NNs, while others (e.g., nonlinear controllers) are formulated using traditional symbolic methods. This can greatly simplify the design and training process, and yield smaller NNs.

Even with the above advances, there remains a lack of modelling and

verification frameworks which can reason formally about the correctness of neuro-symbolic systems. Progress has been made on techniques for both multi-agent verification [8, 9] and safe reinforcement learning [10] in this context, but without the ability to reason formally about stochasticity, which is crucial for modelling uncertainty. Elsewhere, concurrent stochastic games (CSGs) have been widely studied [11, 12, 13, 14, 15], and also integrated into formal modelling and verification frameworks [6], but primarily in the context of *finite* state spaces, which are insufficient for many real-life systems.

We propose a new modelling formalism called *neuro-symbolic concurrent stochastic games (NS-CSGs)*, overviewed in the conference paper [16], which comprise two finite-state probabilistic agents interacting in a shared continuous-state environment. Each agent observes the environment using a neural perception mechanism, which classifies inputs such as images and sensor values into symbolic *percepts* from a finite set and makes decisions using a conventional, symbolic mechanism. During execution, the agents alternate between invoking perception and symbolic decisions, and transition between states according to joint transition probabilities. The agent’s transition to its next local state is based on the current local states of the agents and the agent’s current percept, rather than directly accessing the continuous environment, and can thus model knowledge acquisition from the neural perception mechanism.

Under the assumption that agents have full state observability and working with Borel state spaces, we establish restrictions on the modelling formalism which ensure that NS-CSGs belong to a new subclass of uncountable state-space CSGs [17] that are *determined* for zero-sum discounted cumulative objectives, and therefore prove the existence and measurability of the value function for such objectives.

Our assumptions on the NS-CSG model enable symbolic reasoning with (exactly) learnt percepts (e.g., transition functions that preserve the decomposition of continuous-state spaces into finitely many regions), and induce finite representations of value functions that incorporate symbolic combinations of percepts and knowledge acquired by the agents through symbolic reasoning. We also show that our formalism can capture neural perception mechanisms instantiated as ReLU NN classifiers, which yield piecewise constant perception functions and impose a finite decomposition of the continuous environment into polytopes via preimage computation.

Next, we develop computationally tractable algorithms for optimal strategy synthesis for NS-CSGs, which exploit the Borel and piecewise constant

structure of the transition, perception and reward functions. We propose a new representation for the value function and show its closure under a min-max operator defined for it. Using this (finite) representation, we develop a value iteration (VI) algorithm for NS-CSGs that *approximates* the value of the game and prove the algorithm’s convergence.

Then, we present a policy iteration (PI) algorithm for NS-CSGs inspired by recent work for finite state spaces [18], which we generalise by using novel representations for the value functions and strategies, to ensure finite representability and measurability. This allows us to overcome the main issue that arises when solving Borel state space CSGs with PI, namely that the value function may change from a Borel measurable function to a non-Borel measurable function across iterations.

The PI algorithm adopts the alternating player choices proposed in [18] and removes the need to solve normal-form games and MDPs at each iteration. To the best of our knowledge, these are the first implementable algorithms for solving zero-sum CSGs over Borel state spaces with convergence guarantees. Finally, we illustrate our approach by modelling a dynamic vehicle parking as an NS-CSG and synthesizing (approximately optimal) strategies using a prototype implementation of our VI algorithm.

We note that we assume a *fully observable* game setting. While it is relatively straightforward to generalise the NS-CSG model to partial observability, since NS-CSGs already include perception functions that generate observations, there are no general algorithmic methods for value and strategy computation in the partially observable game setting; see [19] for a one-sided variant of neuro-symbolic partially observable stochastic games that shares its syntax with NS-CSGs. In the general two-sided case, we believe that an approach similar to [20, 21], which converts imperfect-information games to perfect-information, can potentially be used to enable the solution of partially observable NS-CSGs.

1.1. Executive summary

The modelling formalism of NS-CSGs introduced in this paper induces a new class of stochastic games with discrete observations and uncountable state spaces. The underlying continuous state space raises multiple technical issues that need to be resolved to ensure determinacy and enable practical algorithms for value computation and strategy synthesis, which are the main goal of this work. This is achieved through placing appropriate structural restrictions on NS-CSGs and relying on Borel measurability, which allows

us to obtain finite abstract representations of the uncountable state space during value computation.

The key modelling decisions that allowed us to obtain our results are summarised below. To help navigate through the technical sections, we have listed the main concepts and dependencies between them in Fig. 1. Table 1 summarises the role of the definitions, including where they are used.

In Section 3, the definition of NS-CSGs restricts the interface between an agent’s neural perception mechanism and its symbolic decision-making mechanism. In particular, when making decisions, the agent can only access the continuous environment through its perception mechanism, which is limited to a finite set of observations (percepts). This allows us to abstract the (uncountable) continuous state space by partitioning it into a finite set of regions (called a finite connected partition, or FCP), with the elements of each region mapped to the same percept. The initial partition is obtained via preimage computation applied to the perception mechanism and is then refined in subsequent iterations.

Section 4 requires Borel measurability over the continuous environment to constrain the abstractions of the environment to Borel FCPs (called BFCPs) and restrict the choices that the strategies can make to ensure measurability of the induced sets of paths. By requiring Borel measurability of the environment transition function and the fact that the structural BFCP property of the abstraction of the environment is preserved under this transition function (Assumption 1(i)), referred to as BFCP invertibility, we can derive a minimax operator for our setting. We can then apply, in Section 5, the result of [17] to prove the existence of the value for discounted expected reward as a fixed point of the minimax operator (Theorem 1) and convergence of value iteration (Proposition 2). However, this result does not ensure finite representability of the value functions.

In Section 6, structural assumptions are placed on the model to require Borel measurable piecewise constant (B-PWC) perception and reward functions (Assumption 1(ii) and (iii)). These assumptions allow us to derive BFCPs, in which the states in each region are equivalent with respect to either the perception, reward or transition function. Working with B-PWC functions as a finite representation of the value functions, in conjunction with proving that B-PWC functions are closed with respect to the minimax operator, we show that the value can be approximated by a sequence of B-PWC functions (Theorem 2), although the optimal value is not necessarily B-PWC representable.

Section 6 demonstrates that the above assumptions and requirements allow us to derive B-PWC VI (Algorithm 1), which uses the B-PWC representation of value functions. The algorithm refines the BFCP induced by the value function at each iteration by computing the preimage of the transition function (Algorithm 2). The refinement step is necessary because, for a given action, two states with the same percept may result in states that are not observationally equivalent.

In Section 7 we introduce two representations, CON-PWC (constant-piecewise-constant) for Borel measurable strategies and CON-PWL (constant-piecewise-linear) for Borel measurable functions, and, under the assumptions and requirements above, prove that these representations are closed under Max-Min and Min-Max operators (Theorem 3). This allows us to generalise the recent Minimax-action-free PI algorithm [18] for finite CSGs to NS-CSGs (Algorithm 3) by ensuring finite representability and Borel measurability at each iteration. At each iteration, the algorithm refines the BFCPs used for representing strategies and value functions through a preimage computation (Algorithm 4) and then computes values for regions in the resulting BFCPs by Max-Min and Min-Max operators (Algorithm 5).

Finally, we show that our framework is amenable to neural perception mechanisms instantiated as ReLU NNs trained from data. This yields polytope BFCPs (Section 6), which suffice as a symbolic representation of perception BFCPs, thus enabling value computation and policy iteration with the (exactly) learnt percepts. We also provide a prototype implementation of Algorithm 1 and evaluate it on a dynamic parking example with the simpler perception mechanism given by a regression function, demonstrating the feasibility of strategy synthesis for NS-CSGs in practice.

1.2. Related work

Stochastic games were introduced by Shapley [4], who assumed a finite state space. Since then, many researchers have considered CSGs with uncountable state spaces, e.g., [17, 22, 23]. Maitra and Parthasarathy [22] were the first to study discounted zero-sum CSGs in this setting, assuming that the state space is a compact metric space. Following this, more general results for discounted zero-sum CSGs with Borel state spaces have been derived, e.g., [17, 24, 23, 25]. These aim at providing sufficient conditions for the *existence* of either values or optimal strategies for players.

Another important and practical problem for zero-sum CSGs with uncountable state spaces is the *computation* of values and optimal strategies.

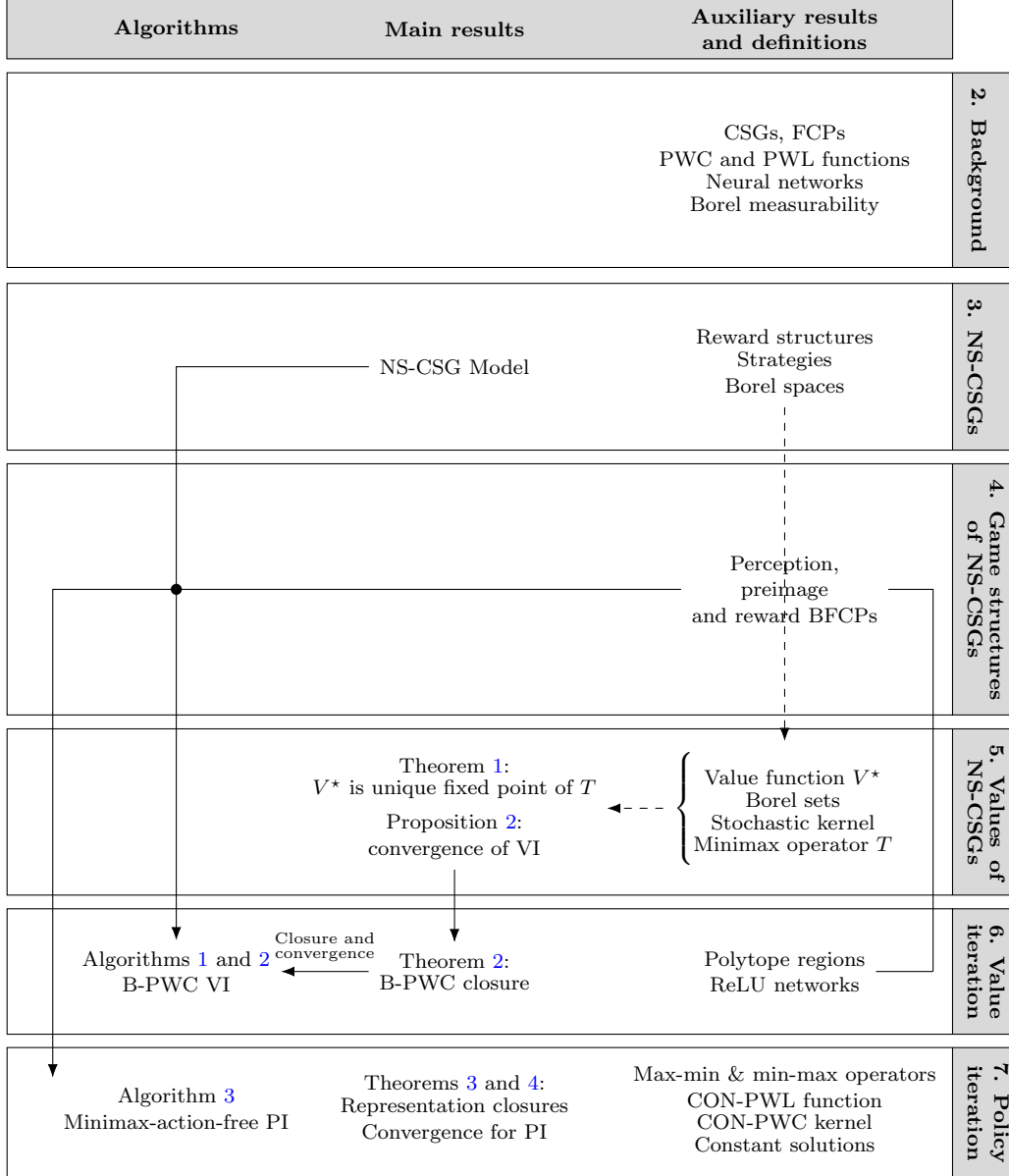


Figure 1: Outline of the main concepts and contributions of our paper, as well as dependencies between them.

Since the seminal policy iteration (PI) methods were introduced by Hoffman and Karp [26] and Pollatschek and Avi-Itzhak [27], a wide range of fixed-point algorithms have been developed for zero-sum CSGs with finite state spaces

Section 2: Background	
1: FCP and Borel FCP	Decomposes uncountable state spaces into finite sets of regions for a finite abstraction of the environment. Used in Definitions 2, 3, 13-17.
2: PWC Borel measurable	Provides measurable finite representations. Used in Definitions 13-17; Lemmas 2, 3, 6-9, 11 Theorems 2-4.
3: PWL Borel measurable	Provides measurable finite representations. Used in Definitions 13, 15; Lemmas 6-9; Theorems 3, 4.
4: BFCP invertible	Preserves the BFCP representation. Used in Assumption 1
5: CSGs	Recalls the definition of discrete CSGs.
Section 3: NS-CSGss	
6: NS-CSGs	Introduces our new neuro-symbolic game model.
7: Semantics of an NS-CSG	A formal semantic definition for the model.
8: Strategy	Defines the stationary strategy for each agent.
Section 5: Values of zero-sum NS-CSGs	
9: Value function	Formal definition of the game's value. Used in Theorems 1, 2, 4; Proposition 2.
10: Minimax operator	Characterises and compute the value function. Used in Theorems 1, 2; Proposition 2.
Section 7: Policy iteration	
11: Operator for the Max-min value	Induces max-min policy evaluation of maximizer. Used in Lemma 6; Theorem 3; Corollary 2.
12: Operator for the Min-max value	Induces min-max policy evaluation of minimizer. Used in Lemma 8; Theorem 3; Corollary 2.
13: CON-PWL Borel measurable function	Finite representations for minimizer's functions. Used in Definitions 15; Lemmas 6-9; Theorem 3; Corollary 2.
14: CON-PWC stochastic kernel	Finite representation for minimizer's strategies. Used in Lemmas 8, 9; Theorem 3; Corollary 2.
15: CON-1 solution	Induces policy improvement of maximizer. Used in Lemma 7; Theorem 3; Corollary 2.
16: CON-2 solution	Induces policy improvement of minimizer. Used in Lemma 9; Theorem 3; Corollary 2.
17: CON-3 solution	Extracts PWC strategies from value functions Used in Lemma 11.

Table 1: Listing of definitions (by section) and their roles.

[11, 12, 13, 14]. Recent work by Bertsekas [18] proposed a distributed optimistic abstract PI algorithm, which inherits the attractive structure of the Pollatschek and Avi-Itzhak algorithm while resolving its convergence difficulties. Value iteration (VI) and PI algorithms have been improved for simple stochastic games [28, 29]. However, all of the above approaches assume fi-

nite state spaces and, to the best of our knowledge, there are no existing VI or PI algorithms for CSGs with uncountable, or more specifically Borel, state spaces. VI and PI algorithms for stochastic control (i.e., the one player case) with Borel state spaces can be found in [30, 31]. Other problems for zero-sum CSGs with uncountable state spaces have been studied and include information structure [32], specialized strategy spaces [33], continuous time setup [34] and payoff criteria [25].

A variety of other objectives, for instance, mean-payoff [35, 36], ratio [36] and reachability [37, 38] objectives, have also been studied for CSGs [11, 12, 13, 14]. But these are primarily in the context of finite/countable state spaces which, as argued above, are insufficient for our setting, where uncountable real vector spaces are usually supplied as inputs to NNs. Building on an earlier version of this work [39], there has been recent progress on solving NS-CSGs [40], but focusing on finite-horizon objectives and using equilibria-based (nonzero-sum) properties.

Compared to the above works, NS-CSGs are shown in Section 3 to form a new subclass of uncountable state-space CSGs, whose determinacy (see Section 5) relies on [17] under some restrictions. The closure of B-PWC functions under the minimax operator in Section 6 is a new result; combining the closure with determinacy, the new B-PWC VI algorithm directly relies on Banach’s fixed point theorem. Our Minimax-action-free PI algorithm takes ideas from recent work [18], which proposed a new PI method to solve zero-sum stochastic games with finite state spaces, but has to resolve a number of issues (see Section 7) due to the uncountability of the underlying state space and the need to ensure Borel measurability at each iteration, including finite representability of measurable functions and strategies, strategy generation in the policy improvement and the division of the uncountable state space into a finite set of regions at each iteration.

Finally, we note that this paper assumes a *fully observable* game setting; a natural extension would be partially observable stochastic games (POSGs), for which there are no general VI and PI computation algorithms. A variant of POSGs, called factored-observation stochastic games (FOSGs), was recently proposed [21] that distinguishes between private and public observations in a similar fashion to our model, but for finite-state models without NNs. Partial observability in FOSGs is dealt with via a mechanism that converts imperfect-information games into continuous-state (public belief state) perfect-information games [20, 21], such that many techniques for perfect-information games can also be applied. Our fully observable model can ar-

guably serve as a vehicle to later solve the more complex case with imperfect information. In recent work [19], a point-based value iteration algorithm was presented for one-sided neuro-symbolic POSGs, thus dealing with partial observability under some mild assumptions.

2. Background

In this section we summarise the background notation, definitions and concepts used in this paper. Our results crucially rely on Borel measurability over the continuous environment and piecewise constant functions as a representation of value functions. The latter is inspired by the observation that a neural perception mechanism such as an NN classifier is a piecewise constant function of the environment and results in a finite decomposition of the environment into regions.

2.1. Borel measurable spaces and functions

Given a non-empty set X , we denote its Borel σ -algebra by $\mathcal{B}(X)$, and the sets in $\mathcal{B}(X)$ are called *Borel sets* of X . The pair $(X, \mathcal{B}(X))$ is a (standard) *Borel space* if there exists a metric on X that makes it a complete separable metric space (unless required for clarity, $\mathcal{B}(X)$ will be omitted). For convenience we will work with real vector spaces; however, this is not essential and any complete separable metric spaces could be used. For Borel spaces X and Y , a function $f : X \rightarrow Y$ is *Borel measurable* if $f^{-1}(B) \in \mathcal{B}(X)$ for all $B \in \mathcal{B}(Y)$ and *bimeasurable* if it is Borel measurable and $f(B) \in \mathcal{B}(Y)$ for all $B \in \mathcal{B}(X)$.

We denote by $\mathbb{F}(X)$ the space of all bounded, Borel measurable real-valued functions on a Borel space X , with respect to the unweighted sup-norm $\|J\| = \sup_{x \in X} |J(x)|$ for $J \in \mathbb{F}(X)$. For functions $J, K \in \mathbb{F}(X)$, we use $\max[J, K]$ and $\min[J, K]$ to denote the respective pointwise maximum and minimum functions of J and K , i.e., we have $\text{opt}[J, K](x) := \text{opt}\{J(x), K(x)\}$ for $\text{opt} \in \{\min, \max\}$ and $x \in X$.

We now introduce notation and definitions for concepts that are fundamental to the abstraction on which our algorithms are performed. The abstraction is based on a decomposition of the uncountable state space into finitely many abstract regions. In the definitions below, let $X \subseteq \mathbb{R}^{n_1}$ and $Y \subseteq \mathbb{R}^{n_2}$ for $n_1, n_2 \in \mathbb{N}$.

Definition 1 (FCP and Borel FCP). *A finite connected partition (FCP) of X , denoted Φ , is a finite collection of disjoint connected subsets (regions)*

that cover X . Furthermore, Φ is a Borel FCP (BFCP) if each region $\phi \in \Phi$ is a Borel set of X .

Definition 2 (PWC Borel measurable). A function $f : X \rightarrow Y$ is piecewise constant Borel measurable (B-PWC) if there exists a BFCP Φ of X such that $f : \phi \rightarrow Y$ is constant for all $\phi \in \Phi$ and Φ is called a constant-BFCP of X for f .

Definition 3 (PWL Borel measurable). A function $f : X \rightarrow Y$ is piecewise linear Borel measurable (B-PWL) if there exists a BFCP Φ of X such that $f : \phi \rightarrow Y$ is linear and bounded for all $\phi \in \Phi$.

Definition 4 (BFCP invertible). A function $f : X \rightarrow Y$ is BFCP invertible if, for any BFCP Φ_Y of Y , there exists a BFCP Φ_X of X , called a preimage BFCP of Φ_Y for f , such that for any $\phi_X \in \Phi_X$ we have $\{f(x) \mid x \in \phi_X\} \subseteq \phi_Y$ for some $\phi_Y \in \Phi_Y$.

For BFCPs Φ_1 and Φ_2 of X , we denote by $\Phi_1 + \Phi_2$ the smallest BFCP of X such that $\Phi_1 + \Phi_2$ is a refinement of both Φ_1 and Φ_2 , which can be obtained by taking all the intersections between regions of Φ_1 and Φ_2 .

2.2. Probability measures

Let X be a Borel space. A function $f : \mathcal{B}(X) \rightarrow [0, 1]$ is a probability measure on X if $f(X) = 1$ and $\sum_{i \in I} f(B_i) = f(\cup_{i \in I} B_i)$ for any countable disjoint family of Borel sets $(B_i)_{i \in I}$. We denote the space of all probability measures on a Borel space X by $\mathbb{P}(X)$. For Borel spaces X and Y , a Borel measurable function $\sigma : Y \rightarrow \mathbb{P}(X)$ is called a *stochastic kernel* on X given Y (also known as a transition probability function from Y to X), and we denote by $\mathbb{P}(X \mid Y)$ the set of all stochastic kernels on X given Y . If $\sigma \in \mathbb{P}(X \mid Y)$, $y \in Y$ and $B \in \mathcal{B}(X)$, then we write $\sigma(B \mid y)$ for $\sigma(y)(B)$. It follows that $\sigma \in \mathbb{P}(X \mid Y)$ if and only if $\sigma(\cdot \mid y) \in \mathbb{P}(X)$ for all $y \in Y$ and $\sigma(B \mid \cdot)$ is Borel measurable for all $B \in \mathcal{B}(X)$.

2.3. Neural networks

A *neural network* (NN) is a real vector-valued function $f : \mathbb{R}^m \rightarrow \mathbb{R}^c$, where $m, c \in \mathbb{N}$, composed of a sequence of *layers* $h_1, \dots, h_k$, where $h_i : \mathbb{R}^{m_i} \rightarrow \mathbb{R}^{c_i}$ for $1 \leq i \leq k$, $m_1 = m$, $c_i = m_{i+1}$ for $1 \leq i \leq k-1$ and $c_k = c$. Each layer h_i is a data-processing module explicitly formulated as

$h_i(x_i) = \text{act}_i(W_i x_i + b_i)$, where x_i is the input to the i th layer given by the output $h_{i-1}(x_{i-1})$ of the $(i-1)$ th layer, act_i is an activation function, and $W_i x_i + b_i$ is a weighted sum of x_i for a weight matrix W_i and a bias vector b_i . An NN f is continuous for all popular activation functions, e.g., Rectified Linear Unit (ReLU), Sigmoid and Softmax [41].

An NN f is said to be a *classifier* for a set of classes C of size c if, for any input $x \in \mathbb{R}^m$, the output $f(x) \in \mathbb{R}^c$ is a probability vector where the i th element of $f(x)$ represents the confidence probability of the i th class of C , i.e., a classifier is a function $f : \mathbb{R}^m \rightarrow \mathbb{P}(C)$. Let $f^{\max} : \mathbb{R}^m \rightarrow C$ denote a function that returns the class with the largest confidence probability in $f(x)$, and call $f^{\max}(x)$ the *class* of x . To allow for situations where the class with the highest probability returned by f is not unique, and hence $f^{\max}(x)$ would be undefined, we assume the classifier includes a *tie-breaking rule* defined by a function $\kappa : 2^C \rightarrow C$ which, given a set of classes, i.e., those with the highest probability, returns the selected class.

Given an NN classifier f with the tie-breaking rule κ , the *preimage* of f divides $\mathbb{R}^m$ into a BFCP Φ of $\mathbb{R}^m$, i.e., for any $\phi \in \Phi$, there exists a class y such that $f^{\max}(x) = y$ for all $x \in \phi$. The preimage of an NN classifier with PWL activation functions can be computed via polyhedral decomposition as described in [42].

2.4. Concurrent stochastic games

Finally, in this section, we recall the model of two-player *concurrent stochastic games*.

Definition 5. A (two-player) concurrent stochastic game (CSG) is a tuple $G = (N, S, A, \Delta, \delta)$ where:

- $N = \{1, 2\}$ is a set of two players;
- S is a finite set of states;
- $A = (A_1 \cup \{\perp\}) \times (A_2 \cup \{\perp\})$ where A_i is a finite set of actions available to player $i \in N$ and $\perp$ is an idle action disjoint from the set $A_1 \cup A_2$;
- $\Delta : S \rightarrow 2^{(A_1 \cup A_2)}$ is an action available function;
- $\delta : (S \times A) \rightarrow \mathbb{P}(S)$ is a probabilistic transition function.

In a state s of a CSG $\mathbf{G}$, each player $i \in N$ selects an action from its available actions, i.e., from the set $\Delta(s) \cap A_i$, if this set is non-empty, and selects the idle action $\perp$ otherwise. We denote the action choices for each player i in state s by $A_i(s)$, i.e., $A_i(s)$ equals $\Delta(s) \cap A_i$ if $\Delta(s) \cap A_i \neq \emptyset$ and equals $\{\perp\}$ otherwise, and by $A(s)$ the possible joint actions in a state, i.e., $A(s) = A_1(s) \times A_2(s)$. Supposing each player i chooses action a_i , then with probability $\delta(s, (a_1, a_2))(s')$ there is a transition to state $s' \in S$. A *path* π of $\mathbf{G}$ is a sequence $\pi = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \dots$ such that $s_k \in S$, $\alpha_k \in A(s_k)$ and $\delta(s_k, \alpha_k)(s_{k+1}) > 0$ for all $k \geq 0$. We let $FPaths_{\mathbf{G}}$ and $IPaths_{\mathbf{G}}$ denote the sets of finite and infinite paths of $\mathbf{G}$, respectively. For a path π , we denote by $\pi(k)$ the $(k+1)$ th state, and $\pi[k]$ the action for the transition from $\pi(k)$ to $\pi(k+1)$.

A *strategy* for a player of a CSG $\mathbf{G}$ resolves its action choices in each state. These choices can depend on the history of the CSG's execution and can be randomised. Formally, a strategy for player i is a function $\sigma_i : FPaths_{\mathbf{G}} \rightarrow \mathbb{P}(A_i \cup \{\perp\})$ mapping finite paths to distributions over available actions, such that, if $\sigma_i(\pi)(a_i) > 0$, then $a_i \in A_i(last(\pi))$ where $last(\pi)$ is the final state of π . A strategy is said to be *stationary* if it makes the same choices for paths that end in the same state. Furthermore, a *strategy profile* of $\mathbf{G}$ is a pair $\sigma = (\sigma_1, \sigma_2)$ of strategies for each player. Given a strategy profile σ and state s , letting $IPaths_s^\sigma$ denote the set of infinite paths from s under the choices of σ , we can define a probability measure $Prob_s^\sigma \in \mathbb{P}(IPaths_s^\sigma)$ [43].

3. Zero-sum neuro-symbolic concurrent stochastic games

This section introduces our model of *neuro-symbolic concurrent stochastic games* (NS-CSGs). We restrict our attention to two-agent (which we also refer to as two-player) games, as we are concerned with zero-sum games, in which there are two agents with directly opposing objectives. However, the approach extends to multi-agent games, by allowing the agents to form two coalitions with directly opposing objectives. Our results depend on the uniqueness of value functions, and therefore do not extend to more than two coalitions.

A (two-agent) NS-CSG comprises two interacting *neuro-symbolic agents* acting in a *shared, continuous-state environment*. Each agent $\mathbf{Ag}_i$ has finitely many local states and actions; it observes the environment through a (trained) *neural perception* mechanism (the perception function obs_i), which depends on the local states of both agents, and relies on *symbolic* decision-making

mechanisms (the transition function δ_i). During execution, the agents alternate between invoking perception and symbolic decisions, where the interface between them is suitably constrained to enable symbolic reasoning with the (exactly) learnt concepts (regions of the continuous inputs space), which we call *percepts* to distinguish them from local states. When invoking perception, continuous inputs are converted into symbolic percepts, and the agent's transition to the next local state is based on the current local state and percept, rather than the environment state, and can thus model knowledge acquisition from the neural perception mechanism.

Definition 6. *A (two-agent) neuro-symbolic concurrent stochastic game (NS-CSG) $\mathbf{C}$ comprises agents $(\mathbf{Ag}_i)_{i \in N}$ for $N = \{1, 2\}$ and environment E where: $\mathbf{Ag}_i = (S_i, A_i, \Delta_i, obs_i, \delta_i)$, $E = (S_E, \delta_E)$ and we have:*

- $S_i = Loc_i \times Per_i$ is a set of states for $\mathbf{Ag}_i$, and Loc_i and Per_i are finite sets of local states and percepts, respectively;
- $S_E \subseteq \mathbb{R}^e$ for $e \in \mathbb{N}$ is a closed uncountable set of environment states;
- A_i is a nonempty finite set of actions for $\mathbf{Ag}_i$, and $A := (A_1 \cup \{\perp\}) \times (A_2 \cup \{\perp\})$ is the set of joint actions, where $\perp$ is an idle action disjoint from $A_1 \cup A_2$;
- $\Delta_i : S_i \rightarrow 2^{A_i}$ is an available action function for $\mathbf{Ag}_i$, defining the actions the agent can take in each of its states;
- $obs_i : (Loc_1 \times Loc_2 \times S_E) \rightarrow Per_i$ is a perception function for $\mathbf{Ag}_i$, mapping the local states of the agents and environment state to a percept of the agent;
- $\delta_i : (S_i \times A) \rightarrow \mathbb{P}(Loc_i)$ is a probabilistic transition function for $\mathbf{Ag}_i$ determining the distribution over the agent's local states given its current state and joint action;
- $\delta_E : (S_E \times A) \rightarrow S_E$ is a deterministic transition function for the environment determining its next state given its current state and joint action.

Remark 1. *We restrict the range of observation functions induced by perception mechanisms to finite sets, which admits a wide class of functions,*

including ReLU neural network classifiers. For an example instantiation of an observation function obs_i , for each pair of local states $(loc_1, loc_2) \in Loc_1 \times Loc_2$ we can associate an NN classifier $f_{loc_1, loc_2} : S_E \rightarrow \mathbb{P}(Per_i)$, see Section 2.3, such that $obs_i(loc_1, loc_2, s_E) = f_{loc_1, loc_2}^{\max}(s_E)$ for all $s_E \in S_E$. These NNs need not be distinct for different pairs of local agent states, but we have allowed this modelling choice to reflect the design of existing NN-enabled systems that we target, e.g., in the VCAS collision avoidance system [44] there are nine NN advisories and which of these advisories is selected is based on the local state of the system. Assuming the range of the perception functions is finite, our framework is also capable of modelling perception mechanisms that input (continuous) numerical sensor information (e.g., via output discretisation) and other machine learning models (which may require additional assumptions to ensure finite representability of percepts).

In an NS-CSG $\mathcal{C}$ the agents and environment execute concurrently and agents move between their local states probabilistically. For simplicity, we consider deterministic environments, but all the results extend directly to probabilistic environments with finite branching.

A (global) state of an NS-CSG comprises a state $s_i = (loc_i, per_i)$ for each agent $\mathbf{Ag}_i$ (a local-state-percept pair) and an environment state s_E . A state $s = ((loc_1, per_1), (loc_2, per_2), s_E)$ is *percept compatible* if $per_i = obs_i(loc_1, loc_2, s_E)$ for $1 \leq i \leq 2$. Percept compatibility indicates that each agent always accesses its percept via the perception function, and observes the environment state only through the perception function. In state $s = (s_1, s_2, s_E)$, each $\mathbf{Ag}_i$ simultaneously chooses one of the actions available in its state s_i (if no action is available, i.e., $\Delta_i(s_i) = \emptyset$, then $\mathbf{Ag}_i$ chooses the idle action $\perp$), resulting in a joint action $\alpha = (a_1, a_2) \in A$. Next, each $\mathbf{Ag}_i$ updates its local state to some $loc'_i \in Loc_i$, according to the distribution $\delta_i(s_i, \alpha)$. At the same time, the environment updates its state to some $s'_E \in S_E$ according to the transition $\delta_E(s_E, \alpha)$. Finally, each $\mathbf{Ag}_i$, based on its new local state, observes the new local state of the other agent and the new environment state to generate a new percept $per'_i = obs_i(loc'_1, loc'_2, s'_E)$. Thus, the game reaches the state $s' = (s'_1, s'_2, s'_E)$, where $s'_i = (loc'_i, per'_i)$ for $1 \leq i \leq 2$.

Example 1. As an illustration, we present an NS-CSG model of a dynamic vehicle parking problem (a static version is presented in [45]). Fig. 2 (left) shows two agents, $\mathbf{Ag}_1$ (the red vehicle) and $\mathbf{Ag}_2$ (the blue vehicle), in a

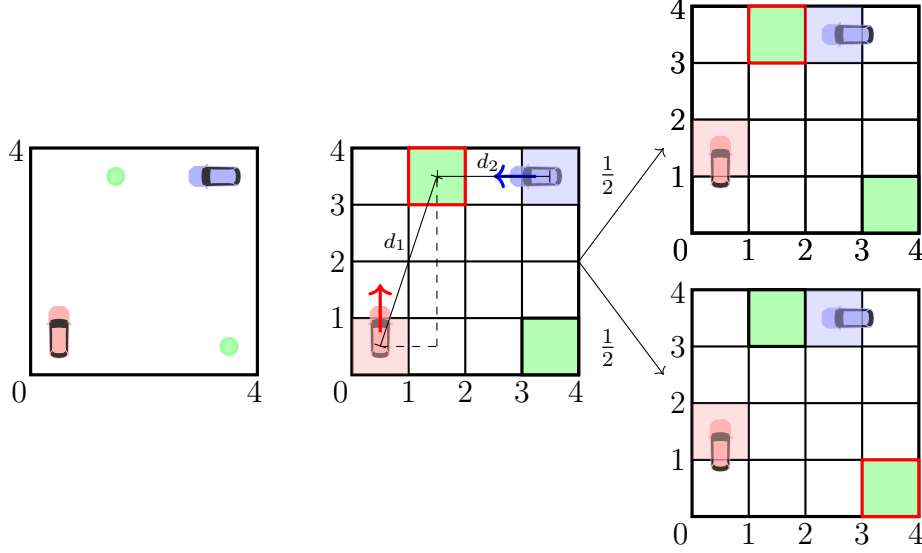


Figure 2: Dynamic vehicle parking: continuous environment $[0, 4]^2$ (left); discrete percepts corresponding the 4×4 abstract grid cells (middle) and probabilistic transitions following joint action (*up, left*) (right). Red vehicle's parking preference is also indicated in red.

(continuous) environment $\mathcal{R} = \{(x, y) \in \mathbb{R}^2 \mid 0 \leq x, y \leq 4\}$ and two preferred parking spots $ps_1, ps_2 \in \mathcal{R}$ (the green circles), which are known to the agents. This example employs a classifier $f_{\mathcal{R}} : \mathcal{R} \rightarrow \mathbb{P}(\text{Grid})$, where $\text{Grid} = \{(i, j) \mid i, j \in \{1, 2, 3, 4\}\}$, which takes the coordinates of a vehicle (or parking spot) as input and outputs a probability distribution over 16 abstract grid cells, see Fig. 2 (centre).

The actions of the agents are to move either up, down, left or right, or park. The vehicles of the agents start from different positions in $\mathcal{R}$ and have the same speed. The red agent initially chooses one parking spot and changes its parking spot with probability 0.5 when the blue agent is observed to be closer to its chosen parking spot and both agents move towards this spot, see Fig. 2 (centre and right). Formally, the agents and the environment are defined as follows.

- $Loc_1 = \{ps_1, ps_2\}$ and $Loc_2 = \{\perp\}$, i.e., the local state of $\mathbf{Ag}_1$ is its current chosen parking spot and the local state of $\mathbf{Ag}_2$ is a dummy state. For $1 \leq i \leq 2$, the set of percepts of $\mathbf{Ag}_i$ is given by $Per_i = \text{Grid} \times \text{Grid}$, representing the abstract grid cells that each agent perceives as the positions of the two vehicles.

- $S_E = \mathcal{R} \times \mathcal{R}$, i.e., the environment is in state $s_E = (w_1, w_2)$ if w_i is the continuous coordinate of $\mathbf{Ag}_i$'s vehicle for $1 \leq i \leq 2$.
- $A_i = \{up, down, left, right, park\}$ for $1 \leq i \leq 2$.
- For $1 \leq i \leq 2$, local state loc_i and percept $per_i = (cell_1, cell_2)$, where the perceived grid cell of agent $\mathbf{Ag}_j$ is $cell_j$, we let $\Delta_i(loc_i, per_i)$ equal A_i if $cell_i \in \{f_{\mathcal{R}}^{\max}(ps_1), f_{\mathcal{R}}^{\max}(ps_2)\}$ and equal $A_i \setminus \{park\}$ otherwise, i.e., an agent's available actions are to move up, down, left and right, and additionally park when the agent is perceived to have reached a parking spot.
- For $1 \leq i \leq 2$, local states loc_1 and loc_2 and environment state (w_1, w_2) , we let $obs_i(loc_1, loc_2, (w_1, w_2)) = (f_{\mathcal{R}}^{\max}(w_1), f_{\mathcal{R}}^{\max}(w_2))$, i.e., independently of the local states of the agents, the perception function returns the perceived grid cell of each agent under the classifier $f_{\mathcal{R}}$.
- For any $\mathbf{Ag}_1$ state $s_1 = (loc_1, (cell_1, cell_2))$ and joint action α , to define δ_1 we have the following two cases to consider:
 - if $\|f_{\mathcal{R}}^{\max}(loc_1) - cell_1\|_2 > \|f_{\mathcal{R}}^{\max}(loc_1) - cell_2\|_2$, where $\|\cdot\|_2$ is the Euclidean norm, i.e. $\mathbf{Ag}_1$ observes $\mathbf{Ag}_2$ is closer to its currently chosen parking spot, and the joint action α indicates both agents are approaching loc_1 , then $\delta_1(s_1, \alpha)(ps_j) = 0.5$ for $1 \leq j \leq 2$, i.e., $\mathbf{Ag}_1$ changes its chosen parking spot with probability 0.5;
 - otherwise $\delta_1(s_1, \alpha)(loc_1) = 1$, i.e., $\mathbf{Ag}_1$ sticks with its chosen parking spot.

Considering δ_2 , since $Loc_2 = \{\perp\}$, we have $\delta_2(s_2, \alpha)(\perp) = 1$ for any $\mathbf{Ag}_2$ state $s_2 = (\perp, (cell_1, cell_2)) \in S_2$ and joint action α .

- For any environment state (w_1, w_2) and joint action $\alpha = (a_1, a_2)$, we let $\delta_E((w_1, w_2), \alpha) = (w'_1, w'_2)$ where, for $1 \leq i \leq 2$, we have $w'_i = w_i + d_{a_i} \Delta t$ if $(w_i + d_{a_i} \Delta t) \in \mathcal{R}$ and $w'_i = w_i$ otherwise, and d_{a_i} is the direction of movement of the action a_i , e.g., $d_{up} = (0, 1)$, and $\Delta t = 0.5$ is the time step. ■

3.1. Semantics of an NS-CSG

The semantics of an NS-CSG $\mathbf{C}$ is a CSG $\llbracket \mathbf{C} \rrbracket$ over the product of the states of the agents and the environment formally defined as follows.

Definition 7 (Semantics of an NS-CSG). *Given an NS-CSG $\mathbf{C}$ consisting of two agents and an environment, its semantics is the CSG $\llbracket \mathbf{C} \rrbracket = (N, S, A, \Delta, \delta)$ where:*

- $S \subseteq S_1 \times S_2 \times S_E$ is the set of percept compatible states;
- $A = (A_1 \cup \{\perp\}) \times (A_2 \cup \{\perp\})$;
- $\Delta(s_1, s_2, s_E) = \Delta_1(s_1) \cup \Delta_2(s_2)$;
- $\delta : (S \times ((A_1 \cup \{\perp\}) \times (A_2 \cup \{\perp\}))) \rightarrow \mathbb{P}(S)$ is the probabilistic transition function, where for states $s = (s_1, s_2, s_E)$, $s' = (s'_1, s'_2, s'_E) \in S$ and joint action $\alpha = (a_1, a_2) \in A$, if $a_i \in \Delta_i(s_i)$ when $\Delta_i(s_i) \neq \emptyset$ and $a_i = \perp$ otherwise for $1 \leq i \leq 2$, then $\delta(s, \alpha)$ is defined and, if $s'_i = (loc'_i, per'_i)$, $per'_i = obs_i(loc'_1, loc'_2, s'_E)$ for $1 \leq i \leq 2$ and $s'_E = \delta_E(s_E, \alpha)$, then

$$\delta(s, \alpha)(s') = \delta_1(s_1, \alpha)(loc'_1) \delta_2(s_2, \alpha)(loc'_2)$$

and otherwise $\delta(s, \alpha)(s') = 0$.

Notice that the CSG $\llbracket \mathbf{C} \rrbracket$ is over percept compatible states and that, by definition of obs_i for each agent $\mathbf{Ag}_i$, the underlying transition relation δ is closed with respect to percept compatible states. Since δ_E is deterministic and Loc_i is a finite set, the set of successors of s under α , denoted $\Theta_s^\alpha = \{s' \mid \delta(s, \alpha)(s') > 0\}$, is finite for all $s \in S$ and $\alpha \in A(s)$. While the semantics of an NS-CSG is an instance of the general class of uncountable state space CSGs, its particular structure induced by perception functions (see Definition 6) will be important in order to establish measurability and finite representability to allow us to derive our algorithms.

3.2. Zero-sum NS-CSGs

For an NS-CSG $\mathbf{C}$, the objectives we consider are *discounted accumulated rewards*, and we assume the first agent tries to maximise the expected value of this objective and the second tries to minimise it. More precisely, for a reward structure $r = (r_A, r_S)$, where $r_A : (S \times A) \rightarrow \mathbb{R}$ is an action reward function and $r_S : S \rightarrow \mathbb{R}$ is a state reward function, and discount factor $\beta \in (0, 1)$, the accumulated discounted reward for a path π of $\llbracket \mathbf{C} \rrbracket$ over the infinite-horizon is defined by:

$$Y(\pi) = \sum_{k=0}^{\infty} \beta^k (r_A(\pi(k), \pi[k]) + r_S(\pi(k))) . \quad (1)$$

Example 2. Returning to the dynamic vehicle parking model of Example 1, we suppose the objective for $\mathbf{Ag}_1$ is to try and park at its currently preferred parking spot without crashing into $\mathbf{Ag}_2$ and, since we consider zero-sum NS-CSGs whose objectives must be directly opposing, the objective of $\mathbf{Ag}_2$ is to try to crash into $\mathbf{Ag}_1$ and prevent it from parking. We can represent this scenario using a discounted reward structure, where all action rewards are zero and for the state rewards we set: there is a negative reward if it is perceived that $\mathbf{Ag}_1$ has yet to reach its current parking spot and the agents have crashed; a positive reward if it is observed that $\mathbf{Ag}_1$ has reached its parking spot, which is higher if the agents are not perceived to have crashed; and 0 otherwise.

Formally, for $s = (s_1, s_2, (w_1, w_2))$ where $s_1 = (loc_1, (cell_1, cell_2))$, we define the state reward function as follows:

$$r_S(s) = \begin{cases} -1000 & \text{if } cell_1 \neq f_{\mathcal{R}}^{\max}(loc_1) \text{ and } cell_1 = cell_2 \\ 500 & \text{if } cell_1 = f_{\mathcal{R}}^{\max}(loc_1) \text{ and } cell_1 = cell_2 \\ 1000 & \text{if } cell_1 = f_{\mathcal{R}}^{\max}(loc_1) \text{ and } cell_1 \neq cell_2 \\ 0 & \text{otherwise.} \end{cases}$$

For the discount factor, we let $\beta = 0.6$. ■

3.3. Strategies of NS-CSGs

Since the state space S is uncountable due to the continuous environment state space, we follow the approach of [17] and require Borel measurable conditions on the choices that the strategies can make to ensure the measurability of the induced sets of paths.

The semantics of any NS-CSG will turn out to be an instance of the class of CSGs from [17], for which *stationary strategies* achieve optimal values [17, Theorem 2(ii), Theorem 3], and therefore, to simplify the presentation, we restrict our attention to stationary strategies and refer to them simply as strategies. Before we give their formal definition, since we work with real vector spaces we require the following lemma.

Lemma 1 (Borel spaces). *The sets S , S_i , S_E and A_i for $1 \leq i \leq 2$ are Borel spaces.*

Proof. By Theorem 27 [46, Chapter 9.6] and Theorem 12 [46, Chapter 9.4], S_1 and S_2 are finite and S_E are complete separable metric spaces, and hence are Borel spaces. Furthermore, we have that $S_1 \times S_2 \times S_E$ is the Cartesian product of Borel spaces, and therefore, using Theorem 1.10 [47, Chapter 1], is also a Borel space. Since we assume obs_i is Borel measurable for $1 \leq i \leq 2$ (see Assumption 1 below), for $(loc_i, per_i) \in S_i$ and $1 \leq i \leq 2$, the set:

$$\{((loc_1, per_1), (loc_2, per_2), s_E) \in S \mid obs_i(loc_1, loc_2, s_E) = per_i \text{ for } 1 \leq i \leq 2\}$$

is a Borel subset of $S_1 \times S_2 \times S_E$. Hence, since S_1 and S_2 are finite, it follows that S is a Borel space. Finally, for $1 \leq i \leq 2$, since A_i is finite it is a Borel space. $\square$

Definition 8 (Strategy). A (stationary) strategy for $\mathbf{Ag}_i$ of an NS-CSG $\mathcal{C}$ is a stochastic kernel $\sigma_i : S \rightarrow \mathbb{P}(A_i)$, i.e., $\sigma_i \in \mathbb{P}(A_i \mid S)$, such that $\sigma_i(A_i(s) \mid s) = 1$ for all $s \in S$. A (strategy) profile $\sigma = (\sigma_1, \sigma_2)$ is a pair of strategies for each agent. We denote by Σ_i the set of all strategies of $\mathbf{Ag}_i$ and by $\Sigma = \Sigma_1 \times \Sigma_2$ the set of profiles.

For $s \in S$ and $1 \leq i \leq 2$, we let $\mathbb{P}(A_i(s)) = \{u_i \in \mathbb{P}(A_i) \mid u_i(A_i(s)) = 1\}$.

3.4. Assumptions on NS-CSGs

Finally, in this section we summarise and motivate the assumptions over NS-CSGs that are required for the results presented in the remainder of the paper. First, NS-CSGs are designed to model neuro-symbolic agents, whose operation depends on particular perception functions, which may result in imperfect information. However, we assume *full observability*, i.e., where agents' decisions can depend on the full state space. It is straightforward to extend the semantics above to *partially observable* CSGs (POSGs) [48, 49] where, for any state, each agent's observation function returns the agent's observable component of the state, by restricting to observationally-equivalent strategies, but this comes at a significant increase in complexity. Instead, we focus on full observability, which can serve as a vehicle to solve the more complex imperfect information game via an appropriate adaptation of the belief-space construction.

Regarding the structure of NS-CSGs, we make the following assumptions to ensure determinacy and that our finite abstract representations of value functions and strategies are closed under both value and policy iteration. Recall that the BFCP, bimeasurable function, B-PWC function and BFCP intertible function are defined in Section 2.

Assumption 1. For any NS-CSG $\mathbf{C}$ and reward structure $r = (r_A, r_S)$:

- (i) $\delta_E(\cdot, \alpha) : S_E \rightarrow S_E$ is bimeasurable and BFCP invertible for $\alpha \in A$;
- (ii) $obs_i(loc_1, loc_2, \cdot) : S_E \rightarrow Per_i$ is B-PWC for $loc_i \in Loc_i$ and $1 \leq i \leq 2$;
- (iii) $r_A(\cdot, \alpha)$, $r_S : S \rightarrow \mathbb{R}$ are B-PWC for $\alpha \in A$.

The above assumptions for NS-CSGs differ from existing stochastic games with Borel state spaces [17, 24, 25] in that the states have both discrete and continuous elements, while the perception and reward functions are required to be B-PWC. The B-PWC requirements in Assumption 1(ii) and (iii) and BFCP invertibility in Assumption 1(i) are needed to achieve B-PWC closure, and hence ensure finitely many abstract state regions at each transition (and are used in Lemmas 2, 3, 4 and Theorem 2 below). The B-PWC closure in Theorem 2 allows us to derive a VI algorithm in Section 6 that employs B-PWC functions, whose convergence follows from the classical Banach’s fixed point theorem. Bimeasurability in Assumption 1(i) enables the application of the results in [17] and ensures the existence of the value of an NS-CSG with respect to a reward structure in Section 5 (and is used in Lemma 4).

In the case that, for each pair of local states of the agents, the perception function obs_i of $\mathbf{Ag}_i$ is implemented via an NN classifier $f : S_E \rightarrow \mathbb{P}(Per_i)$ (see Remark 1), we have that, since f is continuous, it is also Borel measurable.

Example 3. Returning to Example 1, we now give two possible implementations for the classifier $f_{\mathcal{R}} : \mathcal{R} \rightarrow \mathbb{P}(Grid)$ used in the example, where $Grid = \{(i, j) \mid i, j \in \{1, 2, 3, 4\}\}$. It takes the coordinates of a vehicle (or parking spot) as input and outputs a probability distribution over 16 abstract grid cells. This leads to two different implementations for perception functions for the agents, which we recall are defined by $obs_i(loc_1, loc_2, (w_1, w_2)) = (f_{\mathcal{R}}^{\max}(w_1), f_{\mathcal{R}}^{\max}(w_2))$ for $loc_1 \in Loc_1$, $loc_2 \in Loc_2$, $(w_1, w_2) \in \mathcal{R} \times \mathcal{R}$ and $1 \leq i \leq 2$.

The first implementation of the classifier $f_{\mathcal{R}}$ is via the linear regression model for multi-class classification, i.e., $f_{\mathcal{R}}^{\max}(x, y) = (\lceil x \rceil, \lceil y \rceil)$ for all $(x, y) \in \mathcal{R}$, where $\lceil \cdot \rceil$ is the ceiling function, see Fig. 2 (centre).

The second implementation of the classifier $f_{\mathcal{R}}$ is a feed-forward NN classifier, which has one hidden ReLU layer with 10 neurons, and is trained from labelled data. We break ties using a total order over the abstract grid cells, which is Borel measurable. ■

We remark that, while Assumption 1(ii) allows a wide range of perception functions, our main focus is on NNs. We discuss the case when perception functions are instantiated using ReLU neural networks in Section 6.

4. Game structures for NS-CSGs

In this section, we present three finite abstract representations for the continuous state space of an NS-CSG to enable value and policy iterations. These take the form of finite decompositions of the environment, represented as BFCPs, with respect to the perception, reward and transition functions of the NS-CSG. Recall, from Section 2, that a BFCP of a set is a finite family of disjoint Borel sets (regions) that cover the set. Using Assumption 1, we construct these BFCPs over the state space such that the states in each region are *equivalent* with respect to either the perception, reward or transition function, e.g., for any region of the perception BFCP all states in the region yield *the same* percept. These BFCPs allow us to abstract an uncountable state space into a finite set of regions when performing our VI and PI algorithms. In particular, Sections 6 and 7 demonstrate how these different BFCPs can be used, together with intersection, image and preimage operations, to iteratively refine the abstract, symbolic representations of the environment while maintaining the necessary conditions for correctness and convergence of value functions.

For the remainder of this section we fix an NS-CSG $\mathcal{C}$ and reward structure r .

Lemma 2 (Perception BFCP). *There exists a smallest BFCP of S , called the perception BFCP, denoted Φ_P , such that, for any $\phi \in \Phi_P$, all states in ϕ have the same agents' states, i.e., if $(s_1, s_2, s_E), (s'_1, s'_2, s'_E) \in \phi$, then $s_i = s'_i$ for $1 \leq i \leq 2$.*

Proof. For $1 \leq i \leq 2$, since obs_i is PWC and S_i is finite, using Definition 6 we have that, for any $s_i = (loc_i, per_i) \in S_i$, the set $S_E^{s_1, s_2} = \{s_E \in S_E \mid obs_i(loc_1, loc_2, s_E) = per_i \wedge 1 \leq i \leq 2\}$ can be expressed as a number of disjoint regions of S_E and we let $\Phi_E^{s_1, s_2}$ be such a representation that minimises the number of the regions. It then follows that $\Phi_P := \{\{(s_1, s_2, s_E) \mid s_E \in \phi_E\} \mid \phi_E \in \Phi_E^{s_1, s_2} \wedge s_1 \in S_1 \wedge s_2 \in S_2\}$ is a smallest FCP of S such that all states in any region have the same agents' states.

Next we prove that Φ_P is a BFCP of S . We consider a region $\phi \in \Phi_P$. Thus all states in ϕ have the same agents' states, say $s_1 = (loc_1, per_1)$ and

$s_2 = (loc_2, per_2)$. According to Assumption 1, $obs_i(loc_1, loc_2, \cdot) : S_E \rightarrow Per_i$ for $1 \leq i \leq 2$ is B-PWC. The preimage of (per_1, per_2) under obs_1 and obs_2 over S given $s_1 = (loc_1, per_1)$ and $s_2 = (loc_2, per_2)$, denoted $obs^{-1}(per_1, per_2 \mid s_1, s_2)$, equals:

$$\{(s_1, s_2, s_E) \in S \mid obs_1(loc_1, loc_2, s_E) = per_1 \wedge obs_2(loc_1, loc_2, s_E) = per_2\}$$

and therefore is a Borel set of S . Since Φ_P is the smallest such partition of S , the regions in Φ_P , which lead to the percept (per_1, per_2) given s_1 and s_2 , have no common boundary. Thus, $obs^{-1}(per_1, per_2 \mid s_1, s_2)$ is a finite union of disjoint regions in Φ_P , which include the agents' states s_1 and s_2 . Thus, each such region is a Borel set of S , meaning that $\phi \in \mathcal{B}(S)$. Thus, Φ_P is a BFCP of S . $\square$

Lemma 3 (Reward BFCP). *For each $\alpha \in A$, there exists a smallest BFCP of S , called the reward BFCP of S under α and denoted Φ_R^α , such that for any $\phi \in \Phi_R^\alpha$ all states in ϕ have the same state reward and action reward when α is chosen, i.e., if $s, s' \in \phi$, then $r_A(s, \alpha) = r_A(s', \alpha)$ and $r_S(s) = r_S(s')$.*

Proof. For any $\alpha \in A$, since $r_A(\cdot, \alpha) + r_S(\cdot) : S \rightarrow \mathbb{R}$ is B-PWC by Assumption 1, we can show that Φ_R^α is a BFCP of S by a similar argument to that in the proof of Lemma 2. $\square$

Using Assumption 1, we show that, given any joint action α , the perception BFCP Φ_P can be refined into a new BFCP, such that the states in each region of this BFCP all reach, under the transition function of $\llbracket C \rrbracket$, the same regions of the image of Φ_P under the transition function. This result, referred to as *reachability consistency*, will be used for the existence of the value of $\llbracket C \rrbracket$ and in our algorithms.

Lemma 4 (Preimage BFCP). *For each $\alpha \in A$, there exists a refinement BFCP of Φ_P , denoted Φ_P^α , such that, for each $\phi \in \Phi_P^\alpha$ and $\phi' \in \Phi_P$, if $\delta(s, \alpha)$ is defined for $s \in \phi$, then there exists $p_\alpha(\phi, \phi') \in [0, 1]$ such that:*

1. *either $\delta(s, \alpha)(s') = p_\alpha(\phi, \phi') = 0$ for all $s \in \phi$ and $s' \in \phi'$;*
2. *or (i) if $s, \tilde{s} \in \phi$, then there exist unique states $s', \tilde{s}' \in S$ such that $\{s'\} = \Theta_s^\alpha \cap \phi'$, $\{\tilde{s}'\} = \Theta_{\tilde{s}}^\alpha \cap \phi'$ and $\delta(s, \alpha)(s') = \delta(\tilde{s}, \alpha)(\tilde{s}') = p_\alpha(\phi, \phi') > 0$, and (ii) there exists a bimeasurable, BFCP invertible function $q_\alpha : \phi \rightarrow \phi'$ such that $\{q_\alpha(s)\} = \Theta_s^\alpha \cap \phi'$ and $\delta(s, \alpha)(q_\alpha(s)) = p_\alpha(\phi, \phi')$ for all $s \in \phi$.*

Proof. We compute the refinement Φ_P^α of Φ_P by dividing each ϕ of Φ_P such that the required (reachability consistency) property holds. Now, for any $\alpha \in A$ and $\phi \in \Phi_P$, by Lemma 2, all states in ϕ have the same agents' states, say s_1 and s_2 . To aid the proof, for each $\phi' \in \Phi_P$, we will construct a BFCP of ϕ based on ϕ' , denoted $\Phi'(\phi, \phi')$, such that the reachability consistency to the region ϕ' holds in each region of $\Phi'(\phi, \phi')$. If $\delta(s, \alpha)$ is not defined for $s \in \phi$, we do not divide ϕ and let $\Phi'(\phi, \phi') = \{\phi\}$ for all $\phi' \in \Phi_P$ and the reachability consistency to ϕ' is preserved.

It remains to consider the case when $\delta(s, \alpha)$ is defined. Considering any $\phi' \in \Phi_P$, by Lemma 2 there exists agent states $s'_1 = (loc'_1, per'_1)$ and $s'_2 = (loc'_2, per'_2)$ such that if $(s''_1, s''_2, s''_E) \in \phi'$ then $s''_1 = s'_1$ and $s''_2 = s'_2$. We have the following two cases.

- If $\{(s'_1, s'_2, \delta_E(s_E, \alpha)) \in S \mid (s_1, s_2, s_E) \in \phi\} \cap \phi' = \emptyset$, $\delta_1(s_1, \alpha)(loc'_1) = 0$ or $\delta_2(s_2, \alpha)(loc'_2) = 0$, then we do not divide ϕ and let $\Phi'(\phi, \phi') = \{\phi\}$ and we have $\delta(s, \alpha)(s') = p_\alpha(\phi, \phi') = 0$ for all $s \in \phi$ and $s' \in \phi'$.
- If $(\cup_{s \in \phi} \Theta_s^\alpha) \cap \phi'$ is non-empty, then since $\delta_E(\cdot, \alpha) : S_E \rightarrow S_E$ is BFCP invertible using Assumption 1 and ϕ' is a Borel measurable region, there exists a BFCP $\Phi'(\phi, \phi')$ of ϕ such that for each $\phi_1 \in \Phi'(\phi, \phi')$:
 - either $\delta(s, \alpha)(s') = p_\alpha(\phi_1, \phi') = 0$ for all $s \in \phi_1$ and $s' \in \phi'$;
 - or for $s, \tilde{s} \in \phi_1$ there exist unique states $s', \tilde{s}' \in S$ such that $s' = \Theta_s^\alpha \cap \phi'$, $\tilde{s}' = \Theta_{\tilde{s}}^\alpha \cap \phi'$ and $\delta(s, \alpha)(s') = \delta(\tilde{s}, \alpha)(\tilde{s}') = p_\alpha(\phi_1, \phi') > 0$.

It remains to show that the bimeasurable, BFCP invertible function q_α of 2.(ii) exists, which follows from the fact that $\delta_E(\cdot, \alpha) : S_E \rightarrow S_E$ is bimeasurable and BFCP invertible.

Finally, we divide ϕ into a BFCP $\sum_{\phi' \in \Phi_P} \Phi'(\phi, \phi')$, and therefore each region of this BFCP has the required reachability consistency. $\square$

Example 4. Returning to Example 1, we now give the perception BFCPs for the two implementations of the classifier $f_{\mathcal{R}}$ proposed in Example 3. In each case the perception BFCP is of the form $\Phi_P = Loc_1 \times Loc_2 \times \Phi_E$, where Φ_E is a BFCP for the environment state space and the perception BFCP is also the reward BFCP Φ_R^α for $\alpha \in A$. In this example, all pairs of local states correspond to the same classifier for both agents.

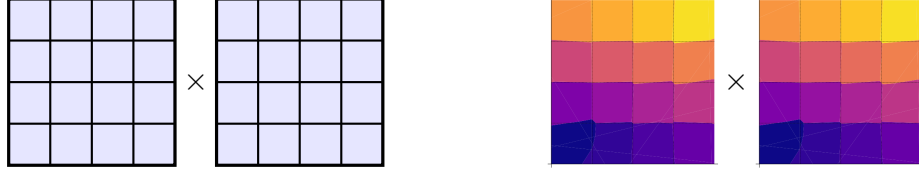


Figure 3: Perception BFCP of the environment states Φ_E . Left: linear regression model. Right: feed-forward NN model, where each subfigure, computed as an exact preimage of the feed-forward NN classifier in Example 3 using [42], depicts the boundaries of the 16 abstract grid cells learnt by the classifier. Each abstract grid cell in the right figure is a union of polytopes and is indicated by a different colour.

For the first implementation of $f_{\mathcal{R}}$, which employs a linear regression model, the BFCP Φ_E for the environment state space is given by:

$$\left\{ \{(x, y) \in \mathcal{R} \mid (i < x \leq i + 1) \wedge (j < y \leq j + 1)\} \mid i, j \in \{0, 1, 2, 3\} \right\}^2$$

as shown in Fig. 3 (left). For the second implementation, the BFCP Φ_E can be found by computing the (exact) preimage of the feed-forward NN classifier of Example 3 using [42], and is shown in Fig. 3 (right). Since an environment state represents the position of each agent, in each case the perception BFCP is two copies, one for each agent, of how the corresponding NN classifier divides the bounded area into 16 abstract grid cells. Each abstract grid cell in Fig. 3 (right) is a union of polytopes and is indicated by a different colour. ■

5. Values of zero-sum NS-CSGs

We now proceed by establishing the value of an NS-CSG $\mathbb{C}$ with respect to a discounted accumulated reward objective Y , i.e., for a reward structure r and discount factor β . We prove the existence of this value, which is a fixed point of a minimax operator. Using Banach’s fixed-point theorem, a sequence of bounded, Borel measurable functions converging to this value is constructed.

Given a state s and (strategy) profile $\sigma = (\sigma_1, \sigma_2)$ of $\llbracket \mathbb{C} \rrbracket$, we denote by $\mathbb{E}_s^\sigma[Y]$ the expected value of the objective Y when starting from state s , given by (1). The functions $\underline{V}, \overline{V} : S \rightarrow \mathbb{R}$, where $s \in S$:

$$\underline{V}(s) := \sup_{\sigma_1 \in \Sigma_1} \inf_{\sigma_2 \in \Sigma_2} \mathbb{E}_s^{\sigma_1, \sigma_2}[Y]$$

$$\bar{V}(s) := \inf_{\sigma_2 \in \Sigma_2} \sup_{\sigma_1 \in \Sigma_1} \mathbb{E}_s^{\sigma_1, \sigma_2}[Y]$$

are called the *lower value* and *upper value* of Y , respectively.

Definition 9 (Value function). *If $\underline{V}(s) = \bar{V}(s)$ for all $s \in S$, then $\llbracket \mathbf{C} \rrbracket$ is determined with respect to the objective Y and the common function is called the value of $\llbracket \mathbf{C} \rrbracket$, denoted by V^* , with respect to Y .*

We next introduce the spaces of feasible state-action pairs and state-action-distribution tuples, and present properties of these spaces. More precisely, for $1 \leq i \leq 2$, we let:

$$\begin{aligned} \Xi_i &:= \{(s, a_i) \in S \times A_i \mid a_i \in A_i(s)\} \\ \Lambda_i &:= \{(s, u_i) \in S \times \mathbb{P}(A_i) \mid u_i \in \mathbb{P}(A_i(s))\} \\ \Xi_{12} &:= \{(s, (a_1, a_2)) \in S \times (A_1 \times A_2) \mid a_1 \in A_1(s) \wedge a_2 \in A_2(s)\} \\ \Lambda_{12} &:= \{(s, (u_1, u_2)) \in S \times (\mathbb{P}(A_1) \times \mathbb{P}(A_2)) \mid u_1 \in \mathbb{P}(A_1(s)) \wedge u_2 \in \mathbb{P}(A_2(s))\}. \end{aligned}$$

Lemma 5 (Borel sets). *For $1 \leq i \leq 2$, the sets Ξ_i and Λ_i are Borel sets of $S \times A_i$ and $S \times \mathbb{P}(A_i)$, respectively. Furthermore, the sets Ξ_{12} and Λ_{12} are Borel sets of $S \times (A_1 \times A_2)$ and $S \times (\mathbb{P}(A_1) \times \mathbb{P}(A_2))$, respectively.*

Proof. We first consider Ξ_i and Λ_i for $i = 1$ (the case for $i = 2$ follows similarly). Since A_1 is finite, the sets Ξ_1 and Λ_1 can be rearranged as:

$$\begin{aligned} \Xi_1 &= \bigcup_{\hat{A}_1 \subseteq A_1} (\{s_1 \mid \Delta_1(s_1) = \hat{A}_1\} \times S_2 \times S_E \times \hat{A}_1) \cap (S \times A_1) \\ \Lambda_1 &= \bigcup_{\hat{A}_1 \subseteq A_1} (\{s_1 \mid \Delta_1(s_1) = \hat{A}_1\} \times S_2 \times S_E \times \mathbb{P}(\hat{A}_1)) \cap (S \times \mathbb{P}(A_1)). \end{aligned}$$

Since $\hat{A}_1$ is a subset of the finite set A_1 , the sets $\hat{A}_1$ and $\mathbb{P}(\hat{A}_1)$ are Borel sets of A_1 and $\mathbb{P}(A_1)$, respectively. Since S_1 is a finite set, for any $\hat{A}_1 \subseteq A_1$, the set $\{s_1 \mid \Delta_1(s_1) = \hat{A}_1\}$ is a Borel set of S_1 . Since S_2 and S_E are both Borel sets by Lemma 1, the result follows by Theorem 1.10 [47, Chapter 1]. Using similar reasoning, it follows that Ξ_{12} and Λ_{12} are also Borel sets of the respective spaces. $\square$

Proposition 1 (Stochastic kernel transition function). *The probabilistic transition function δ of $\llbracket \mathbf{C} \rrbracket$ is a stochastic kernel.*

Proof. From Definition 7, it follows that, for any $(s, \alpha) \in \Xi_{12}$, we have $\delta(s, \alpha)(\cdot) \in \mathbb{P}(S)$. We show that, if $B \in \mathcal{B}(S)$, then $\delta(\cdot, \cdot)(B) : (S \times A) \rightarrow \mathbb{R}$ is Borel measurable on Ξ_{12} . More precisely, we prove that, for any $c \in \mathbb{R}$, the preimage of the Borel set $[c, \infty)$ of $\mathbb{R}$ under $\delta(\cdot, \cdot)(B)$ which is given by:

$$\delta^{-1}([c, \infty))(B) = \{(s, \alpha) \in \Xi_{12} \mid \delta(s, \alpha)(B) \geq c\}$$

is an element of $\mathcal{B}(\Xi_{12})$. If $c > 1$, then $\delta^{-1}([c, \infty))(B) = \emptyset \in \mathcal{B}(\Xi_{12})$, and if $c \leq 0$, then $\delta^{-1}([c, \infty))(B) = \Xi_{12} \in \mathcal{B}(\Xi_{12})$.

Therefore, it remains to consider the case when $0 < c \leq 1$. Consider any $\alpha \in A$ and let Φ_P^α be the refinement of Φ_P of Lemma 4. For each $\phi \in \Phi_P^\alpha$ and $\phi' \in \Phi_P$ such that $p_\alpha(\phi, \phi') > 0$, let $q_\alpha : \phi \rightarrow \phi'$ be the associated bimeasurable, BFCP invertible function from Lemma 4. The image of ϕ under q_α into ϕ' is given by:

$$\hat{q}_\alpha(\phi, \phi') = \{s' \in \phi' \mid s' = q_\alpha(s) \wedge s \in \phi\}.$$

By Lemmas 2 and 4, both ϕ and ϕ' are Borel sets and q_α is bimeasurable, and therefore $\hat{q}_\alpha(\phi, \phi')$ is a Borel set. Next, since q_α is Borel measurable, the preimage of the Borel set $\hat{q}_\alpha(\phi, \phi') \cap B$ under q_α over the region ϕ , which is given by:

$$\hat{q}_\alpha^{-1}(\phi, \hat{q}_\alpha(\phi, \phi') \cap B) = \{s \in \phi \mid q_\alpha(s) \in \hat{q}_\alpha(\phi, \phi') \cap B\}$$

is a Borel set. By combining this result with Lemma 4, each state in $\hat{q}_\alpha^{-1}(\phi, \hat{q}_\alpha(\phi, \phi') \cap B)$ under α transitions to B with probability $p_\alpha(\phi, \phi')$. We denote the set of all transition probabilities from ϕ under α by $P_\alpha(\phi) = \{p_\alpha(\phi, \phi') > 0 \mid \phi' \in \Phi_P\}$. Then, the collection of the subsets of $P_\alpha(\phi)$ for which the sum of their elements is greater or equal to c is given by:

$$P_\alpha^{\geq c}(\phi) := \{P' \subseteq P_\alpha(\phi) \mid \sum_{p' \in P'} p' \geq c\}$$

and is finite. Now for each set $P' \in P_\alpha^{\geq c}(\phi)$, the states in the set:

$$O_\alpha(\phi, P') = \bigcap_{p_\alpha(\phi, \phi') \in P'} \hat{q}_\alpha^{-1}(\phi, \hat{q}_\alpha(\phi, \phi') \cap B)$$

transition to B under α with probability greater or equal to c and $O_\alpha(\phi, P')$ is a Borel set as P' is a finite set. Thus, the states in ϕ reaching B under α with probability greater or equal to c are given by:

$$O_\alpha(\phi) = \bigcup_{P' \in P_\alpha^{\geq c}(\phi)} O_\alpha(\phi, P')$$

which is a Borel set since $P_\alpha^{\geq c}(\phi)$ is a finite set. Finally, we have:

$$\delta^{-1}([c, \infty))(B) = \bigcup_{\alpha \in A} \bigcup_{\phi \in \Phi_P^\alpha} \{(s, \alpha) \in \Xi_{12} \mid s \in O_\alpha(\phi)\}$$

and therefore, combining Lemmas 4 and 5, it follows that $\delta^{-1}([c, \infty))(B) \in \mathcal{B}(\Xi_{12})$ as required. $\square$

Before presenting properties of the value function, we introduce the following operator based on the classical Bellman equation. Recall that r_A and r_S are the action and state reward functions defined in Section 3.2, respectively, and δ is the transition function of the NS-CSG (Definition 7). Further, we require that the set of successors of s under α , denoted $\Theta_s^\alpha = \{s' \mid \delta(s, \alpha)(s') > 0\}$, is finite for all $s \in S$ and $\alpha \in A(s)$ (see Section 3.1).

Definition 10 (Minimax operator). *Given a bounded, Borel measurable real-valued function $V \in \mathbb{F}(S)$, the minimax operator $T : \mathbb{F}(S) \rightarrow \mathbb{F}(S)$ is defined, for any $s \in S$, by:*

$$[TV](s) := \max_{u_1 \in \mathbb{P}(A_1(s))} \min_{u_2 \in \mathbb{P}(A_2(s))} \sum_{a_1 \in A_1(s)} \sum_{a_2 \in A_2(s)} Q(s, (a_1, a_2), V) u_1(a_1) u_2(a_2)$$

where for any $\alpha \in A(s)$:

$$Q(s, \alpha, V) := r_A(s, \alpha) + r_S(s) + \beta \sum_{s' \in \Theta_s^\alpha} \delta(s, \alpha)(s') V(s').$$

We use the standard notation $[TV]$ for $T(V)$ in the remainder of the paper to simplify the presentation.

Theorem 1 (Value function). *If $\mathbb{C}$ is an NS-CSG and Y is a discounted zero-sum objective, then*

- (i) $\llbracket \mathbb{C} \rrbracket$ is determined with respect to Y , i.e., $V^\star$ exists;
- (ii) $V^\star$ is the unique fixed point of the operator T ;
- (iii) $V^\star$ is a bounded, Borel measurable function.

Proof. The proof follows through showing that $\llbracket \mathbb{C} \rrbracket$ is an instance of a zero-sum stochastic game that satisfies the conditions of the Borel model presented in [17].

From Lemma 1, we have that A_1 , A_2 and S are complete and separable metric spaces. By Lemma 5, the spaces Ξ_i and Λ_i are Borel sets of $S \times A_i$ and $S \times \mathbb{P}(A_i)$ for $1 \leq i \leq 2$, respectively. By Proposition 1, δ is a Borel stochastic

kernel. Furthermore, from Assumption 1 we have that $r_A + r_S : (S \times A) \rightarrow \mathbb{R}$ is bounded, and therefore it follows that $\llbracket C \rrbracket$ with respect to the zero-sum objective Y is an instance of a zero-sum stochastic game with Borel model and discounted payoffs introduced in [17]. Hence, (i) follows from [17, Theorems 2 and 3], and (ii) from the discounted case of [17, Theorem 1]. Finally, for (iii), since $\beta \in (0, 1)$, we have that V^* is bounded, and therefore V^* is Borel measurable using [17, Lemma 3]. $\square$

The following guarantees that value iteration (VI) converges to the value function.

Proposition 2 (Convergence). *For any $V^0 \in \mathbb{F}(S)$, the sequence $(V^t)_{t \in \mathbb{N}}$, where $V^{t+1} = [TV^t]$, converges to V^* . Moreover, each V^t is bounded, Borel measurable.*

Proof. Since $r_A + r_S : (S \times A) \rightarrow \mathbb{R}$ is bounded, using [17, Lemma 2] we have that, if V^t is bounded, Borel measurable, then $[TV^t]$ is also bounded. The result then follows from the fact that $V^*(s) = \lim_{t \rightarrow \infty} V^t(s)$ for all $s \in S$ if $V^{t+1} = [TV^t]$ for all $t \in \mathbb{N}$ [17]. $\square$

6. Value iteration

Despite the convergence result of Proposition 2, in practice there may not exist finite representations of general bounded Borel measurable functions $(V^t)_{t \in \mathbb{N}}$ due to the uncountable state space. We now show how VI can be used to *approximate* the values of $\llbracket C \rrbracket$, based on a sequence of Borel piecewise constant (B-PWC) functions (Section 2).

6.1. B-PWC closure and convergence

For NS-CSGs, we demonstrate that, under Assumption 1, a B-PWC representation of value functions is closed under the minimax operator and ensures convergence of value iteration.

Theorem 2 (B-PWC closure and convergence). *If $V \in \mathbb{F}(S)$ and B-PWC, then so is $Q(\cdot, \alpha, V)$ and $[TV]$ for $\alpha \in A$. If $V^0 \in \mathbb{F}(S)$ and B-PWC, the sequence $(V^t)_{t \in \mathbb{N}}$ such that $V^{t+1} = [TV^t]$ converges to V^* , and each V^t is B-PWC.*

Proof. Considering any B-PWC function $V \in \mathbb{F}(S)$ and joint action $\alpha \in A$, since $r_A(\cdot, \alpha) + r_S(\cdot)$ is B-PWC by Assumption 1, the fact that $Q(\cdot, \alpha, V)$ is B-PWC follows if, by Definition 10, we can show that the function $\overline{Q}(\cdot, \alpha, V)$ where:

$$\overline{Q}(\cdot, \alpha, V) := \sum_{s' \in \Theta^\alpha} \delta(\cdot, \alpha)(s') V(s')$$

is B-PWC. Boundedness follows because V is bounded. The indicator function of a subset $S' \subseteq S$ is the function $\chi_{S'} : S \rightarrow \mathbb{R}$ such that $\chi_{S'}(s) = 1$ if $s \in S'$ and 0 otherwise. Now $\chi_{S'}$ is Borel measurable if and only if S' is a Borel set of S [46]. For clarity, we use $q_\alpha(s; \phi, \phi')$ to refer to q_α from Lemma 4 for $\alpha \in A$, $s \in \phi$, $\phi \in \Phi_P^\alpha$ and $\phi' \in \Phi_P$ (where again Φ_P^α is from Lemma 4). For any $s \in S$ such that $\delta(s, \alpha)$ is defined, we have:

$$\begin{aligned} \overline{Q}(s, \alpha, V) &= \sum_{\phi \in \Phi_P^\alpha} \chi_\phi(s) \sum_{s' \in \Theta^\alpha} \delta(s, \alpha)(s') V(s') \\ &= \sum_{\phi \in \Phi_P^\alpha} \chi_\phi(s) \sum_{\phi' \in \Phi_P} p_\alpha(\phi, \phi') V(q_\alpha(s; \phi, \phi')) \quad \text{by Lemma 4} \\ &= \sum_{\phi \in \Phi_P^\alpha} \sum_{\phi' \in \Phi_P} p_\alpha(\phi, \phi') \chi_\phi(s) V(q_\alpha(s; \phi, \phi')) \quad \text{rearranging.} \end{aligned}$$

Since ϕ is a Borel set of S , we have that χ_ϕ is Borel measurable. Next, we show that $V(q_\alpha(\cdot; \phi, \phi'))$ is Borel measurable on ϕ . Let Φ_V be a constant-BFCP of S for V . Given $c \in \mathbb{R}$, we denote by $\Phi_V^{\geq c}$ the set of regions in Φ_V on which $V \geq c$ holds. The preimage of $[c, \infty)$ under $V(q_\alpha(\cdot; \phi, \phi'))$ defined on ϕ is given by:

$$\begin{aligned} V^{-1}(q_\alpha([c, \infty); \phi, \phi')) &= \{s \in \phi \mid V(q_\alpha(s; \phi, \phi')) \geq c\} \\ &= \bigcup_{\phi_V \in \Phi_V^{\geq c}} \{s \in \phi \mid q_\alpha(s; \phi, \phi') \in \phi_V\}. \end{aligned}$$

Since $q_\alpha(s; \phi, \phi')$ is Borel measurable in $s \in \phi$ (see Lemma 4) and ϕ_V is a Borel set of S , then $\{s \in \phi \mid q_\alpha(s; \phi, \phi') \in \phi_V\}$ is a Borel set of ϕ . Since $V^{-1}(q_\alpha([c, \infty); \phi, \phi'))$ is also a Borel set of ϕ by noting that $\Phi_V^{\geq c}$ is finite, it follows that $V(q_\alpha(\cdot; \phi, \phi'))$ is Borel measurable on ϕ . Therefore $\overline{Q}(\cdot, \alpha, V)$ is Borel measurable.

Next, since $q_\alpha(\cdot; \phi, \phi')$ is BFCP invertible on ϕ by Lemma 4, there exists a BFCP Φ_q of ϕ such that all states in each region of Φ_q are mapped into the same region of Φ_V under $q_\alpha(\cdot; \phi, \phi')$. Following this, $V(q_\alpha(\cdot; \phi, \phi'))$ is constant on each region of Φ_q . Therefore, using the fact that χ_ϕ is PWC, it follows that $\overline{Q}(\cdot, \alpha, V)$ is PWC, which completes the proof that $\overline{Q}(\cdot, \alpha, V)$ is B-PWC.

From Proposition 2 we have that $[TV]$ is bounded, Borel measurable. Since $Q(\cdot, \alpha, V)$ is PWC for any joint action $\alpha \in A$, $A(s)$ is PWC and

ALGORITHM 1 B-PWC VI

```

1: Input: NS-CSG  $\mathcal{C}$ , perception FCP  $\Phi_P$ , reward FCPs  $(\Phi_R^\alpha)_{\alpha \in A}$ , error  $\varepsilon$ 
2: Output: Approximate value function  $V$ 
3: Initialize  $(\Phi_{V^0}, V^0)$ 
4:  $\bar{\varepsilon} \leftarrow 2\varepsilon$ ,  $t \leftarrow 0$ 
5: while  $\bar{\varepsilon} > \varepsilon$  do
6:    $\Phi_{V^{t+1}} \leftarrow \text{Preimage\_BFCP}(\Phi_{V^t}, \Phi_P, (\Phi_R^\alpha)_{\alpha \in A})$  (Algorithm 2)
7:   for  $\phi \in \Phi_{V^{t+1}}$  do
8:     Take one state  $s \in \phi$ ,  $V_\phi^{t+1} \leftarrow [TV^t](s)$ 
9:    $\bar{\varepsilon} \leftarrow \text{Dist}(V^{t+1}, V^t)$ 
10:   $t \leftarrow t + 1$ 
11: return  $V \leftarrow V^t$ 

```

A is finite, it follows that $[TV]$ is PWC using the fact that the value of a zero-sum normal-form game induced at every $s \in S$ is unique. Thus, $[TV]$ is B-PWC. The remainder of the proof follows directly from Banach's fixed point theorem and the fact we have proved that, if $V \in \mathbb{F}(S)$ and B-PWC, so is $[TV]$. $\square$

6.2. B-PWC VI algorithm

We use the closure property of B-PWC value functions under the minimax operator from Theorem 2 to iteratively construct a sequence $(V^t)_{t \in \mathbb{N}}$ of such functions to approximate V^* to within a convergence guarantee. Algorithm 1 presents our B-PWC VI scheme, where the BFCP of the B-PWC value function at each iteration is refined (line 6) and subsequently the B-PWC value function is updated via minimax computations (line 8) for a state sampled from each of its regions.

Initialization. The function V^0 is initialised as a 0-valued B-PWC function defined over the BFCP $\Phi_{V^0} = \Phi_P + \sum_{\alpha \in A} \Phi_R^\alpha$ of S , i.e., $V_\phi^0 = 0$ for $\phi \in \Phi_{V^0}$.

The algorithm. The steps of our B-PWC VI algorithm are illustrated in Fig. 4. These steps use $\text{Preimage_BFCP}(\Phi_{V^t}, \Phi_P, (\Phi_R^\alpha)_{\alpha \in A})$, see Algorithm 2, to compute a refinement of $\Phi_P + \sum_{\alpha \in A} \Phi_R^\alpha$ that is a preimage BFCP of Φ_{V^t} for δ . Then, in order to compute the value V_ϕ^{t+1} over each region $\phi \in \Phi$, we take one state $s \in \phi$ and then find the value of a zero-sum normal form game [50] at s induced by Definition 10.

As a convergence criterion for B-PWC VI in Algorithm 1, we detect when

ALGORITHM 2 BFCP iteration for B-PWC VI

```

1: procedure Preimage_BFCP( $\Phi, \Phi_P, (\Phi_R^\alpha)_{\alpha \in A}$ )
2:    $\Phi_{\text{pre}} \leftarrow \emptyset$ 
3:   for  $\phi \in \Phi_P + \sum_{\alpha \in A} \Phi_R^\alpha$  do
4:      $\Phi_{\text{pre}}^\phi \leftarrow \emptyset$ 
5:     for  $\alpha \in A, \phi' \in \{\phi' \in \Phi \mid (\cup_{s \in \phi} \Theta_s^\alpha) \cap \phi' \neq \emptyset\}$  do
6:        $\Phi_{\text{pre}}^\phi \leftarrow \Phi_{\text{pre}}^\phi \cup \{s \in \phi \mid \Theta_s^\alpha \cap \phi' \neq \emptyset\}$ 
7:    $\Phi_{\text{pre}} \leftarrow \Phi_{\text{pre}} \cup \{\phi_1 \in \text{Intersect}(\phi, \Phi_{\text{pre}}^\phi)\}$ 
8:   return  $\Phi_{\text{pre}}$ 
  
```

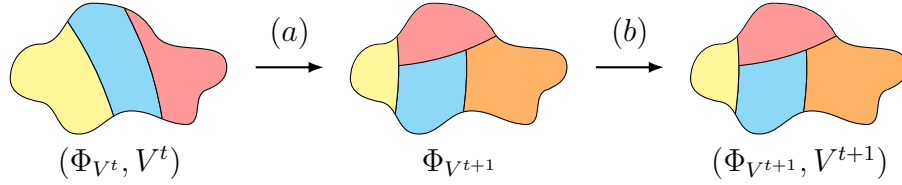


Figure 4: B-PWC VI in Algorithm 1. (a) Find new BFCP $\Phi_{V^{t+1}}$: refine $\Phi_P + \sum_{\alpha \in A} \Phi_R^\alpha$ to be a pre-image BFCP of Φ_{V^t} for δ ; (b) compute a value for each $\phi \in \Phi_{V^{t+1}}$: take one state $s \in \phi$ and compute V^{t+1} by assigning to each region ϕ the value $[TV^t](s)$.

the difference between successive value approximations falls below a threshold ε (as usual for VI, this does *not* guarantee an ε -optimal solution). The function $\text{Dist}(V^{t+1}, V^t)$ computes the difference between V^{t+1} and V^t , which may have different regions due to the possible inconsistency between $\Phi_{V^{t+1}}$ and Φ_{V^t} . An intuitive method is to evaluate V^{t+1} and V^t at a finite set of points, and then compute the maximum difference. In the usual manner for VI, an approximately optimal strategy can be extracted from the final step of the computation.

Algorithm 2 requires region-wise computations involving the image and preimage of a region, region intersection and the sum of BFCPs. In particular, $\text{Intersect}(\phi, \Phi_{\text{pre}}^\phi)$ is the refinement of ϕ obtained by computing all pairwise intersections of ϕ with regions in Φ_{pre}^ϕ and, by construction, is a preimage BFCP of Φ for δ over ϕ . The following corollary then follows from Lemma 4 and Theorem 2.

Corollary 1 (BFCP iteration for B-PWC VI). *In Algorithm 2, Φ_{pre} is a refinement of $\Phi_P + \sum_{\alpha \in A} \Phi_R^\alpha$ and is a preimage BFCP of Φ for δ .*

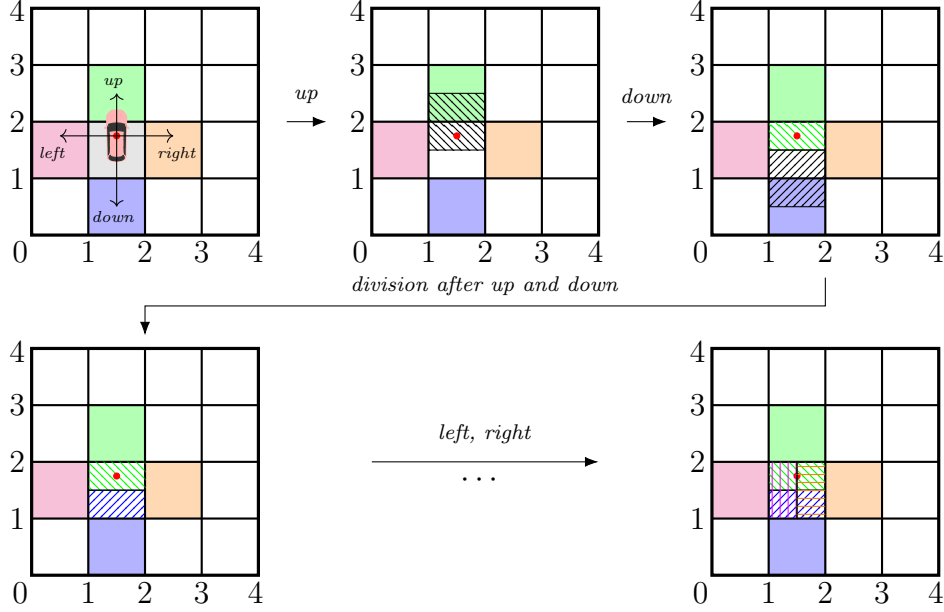


Figure 5: Illustration of the region refinement process.

Example 5. Returning to Example 1, in Fig. 5 we illustrate the refinement process for a region in the perception BFCP when performing Algorithm 2. We focus on the perception function that is implemented via the linear regression model given in Example 3. The preimage BFCP is constructed as described in Lemma 4 and Corollary 1 for the case $\Phi = \Phi_P$. For simplicity, here we only consider actions and coordinates of $\mathbf{Ag}_1$.

Fig. 5 demonstrates how the grey region in the initial BFCP in the top left grid is subdivided when considering each action of $\mathbf{Ag}_1$. In general, when subdividing a region, we also have to take into account the reward BFCP as indicated in Algorithms 1 and 2. The shaded squares (diagonal lines) in the top centre and top right grids are the images of the initial grey region under the actions *up* and *down*, respectively. Based on their intersections and the corresponding preimages, the grey region in the top left grid is subdivided into green and purple subregions in the bottom left grid such that all points in each subregion lead to the same new percepts under both actions, *up* and *down*. The bottom right grid then shows the subdivision of the grey region in the top left grid after considering all actions.

Polytope regions. Our B-PWC VI algorithm assumes that each region in

a BFCP is finitely representable. We now briefly discuss the use of BFCPs defined by *polytopes*, which suffice for a symbolic representation of perception BCFPs of ReLU NNs (discussed below). The focus is the region-based computations required by Algorithm 2. A polytope $\phi \subseteq \mathbb{R}^m$ is an intersection of ℓ halfspaces $\{x \in \mathbb{R}^m \mid g_k(x) \geq 0 \text{ for } 1 \leq k \leq \ell\}$, where $g_k(x) = W_k^\top x + b_k$ is a linear function, i.e., $W_k \in \mathbb{R}^m$ and $b_k \in \mathbb{R}$, for $1 \leq k \leq \ell$. If ϕ_1 and ϕ_2 are polytopes, represented by $\{(W_k, b_k)\}_{k=1}^{\ell'}$ and $\{(W_k, b_k)\}_{k=\ell'+1}^\ell$, respectively, then the intersection $\phi_1 \cap \phi_2$, is the intersection of ℓ halfspaces and can be represented as $\{(W_k, b_k)\}_{k=1}^\ell$. Therefore, the sum $\Phi_1 + \Phi_2$ of two BFCPs Φ_1 and Φ_2 can be computed by considering the intersection $\phi_1 \cap \phi_2$ of all pairwise combinations of regions $\phi_1 \in \Phi_1$ and $\phi_2 \in \Phi_2$.

The *image* of a polytope $\phi = \{x \in \mathbb{R}^m \mid g_k(x) \geq 0 \text{ for } 1 \leq k \leq \ell\}$ under a linear function $f : \mathbb{R}^m \rightarrow \mathbb{R}^m$, where $f(x) = Dx + b$, $D \in \mathbb{R}^{m \times m}$ is non-singular and $b \in \mathbb{R}^m$, is the polytope $f(\phi) = \{x \in \mathbb{R}^m \mid W_k^\top D^{-1}x + b_k - W_k^\top D^{-1}b \geq 0 \text{ for } 1 \leq k \leq \ell\}$ with the representation $\{(D^{-\top}W_k, b_k - W_k^\top D^{-1}b)\}_{k=1}^\ell$. The *preimage* of ϕ under f is the polytope $f^{-1}(\phi) = \{x \in \mathbb{R}^m \mid W_k^\top Dx + b_k + W_k^\top b \geq 0 \text{ for } 1 \leq k \leq \ell\}$ with the representation $\{(D^\top W_k, b_k + W_k^\top b)\}_{k=1}^\ell$. Checking the feasibility of a set constrained by a set of linear inequalities can be solved by a linear program solver [51].

ReLU networks. If each perception function obs_i is implemented via a ReLU NN classifier, where the activation function is B-PWL, then the preimages of the ReLU NN for each percept [42] have linear boundaries, and therefore all regions in the corresponding perception BFCP Φ_P can be (symbolically) represented by polytopes (see Example 4). If there exist polytope constant-BFCPs for B-PWC $r_A(\cdot, \alpha)$ and r_S for all $\alpha \in A$, then all regions in Φ_R^α for $\alpha \in A$ are polytopes. If $\delta_E(\cdot, \alpha)$ is piecewise linear and invertible and ϕ' is a polytope (line 5 in Algorithm 2), then $\{s \in \phi \mid \Theta_s^\alpha \cap \phi' \neq \emptyset\}$ is a polytope. Therefore, each region in Φ_{pre} is a polytope after every iteration and the operations over polytopes, including intersections, image and preimage computations, directly follow from the computation above.

Example 6. We now return to the NS-CSG model, presented in Example 1, of a dynamic vehicle parking problem with the perception functions implemented via the linear regression model given in Example 3. To demonstrate the practicality of our approach we synthesise strategies using a prototype Python implementation of the B-PWC VI algorithm.

The implementation uses a polyhedral representation of regions and the values of the zero-sum normal-form games involved in the minimax operator

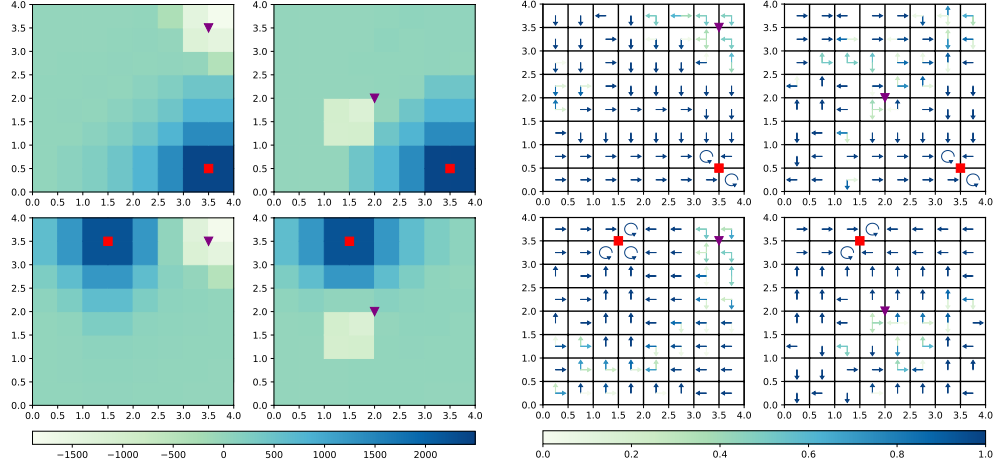


Figure 6: Strategy synthesis for Example 6. Value function (left) and optimal strategy (right) over different coordinates of $\mathbf{Ag}_1$ for a fixed local state of $\mathbf{Ag}_1$ (red square) and a fixed coordinate of $\mathbf{Ag}_2$ (purple triangle).

at step 8 of Algorithm 1 are found by solving the corresponding linear program [50] using the SciPy library [51]. We have partitioned the state space of the game into two sets corresponding to the two possible local states of $\mathbf{Ag}_1$. The B-PWC VI algorithm converges after 46 iterations when $\varepsilon = 10^{-6}$ and takes 3,825s to complete. For each set in the partition of the state space, the BFCP of this set converges to the product of two 8×8 grids. For the currently preferred parking spot of $\mathbf{Ag}_1$ (red square) and coordinate of $\mathbf{Ag}_2$ (purple triangle), the value function with respect to the coordinate of $\mathbf{Ag}_1$ is presented in Fig. 6 (left) and shows that, the closer $\mathbf{Ag}_1$ is to its preferred parking spot, the higher the (approximate) optimal value. The lightest-colour class is caused by an immediate crash, and its position follows from the observation function.

An (approximately) optimal strategy for $\mathbf{Ag}_1$ is presented in Fig. 6 (right), where the colour intensity of an arrow is proportional to the probability of moving in that direction and the rotating arrow represents the parking action. We see that there are several choices that may not at first appear intuitive. As an example, $\mathbf{Ag}_1$ moves left when in cell $[1.0, 1.5] \times [3.5, 4.0]$ (top left), which is away from its current chosen parking spot. However, since $\mathbf{Ag}_2$ is closer to its current parking spot, keeping away from this parking spot will avoid a crash with $\mathbf{Ag}_2$. In addition, under $\mathbf{Ag}_1$'s transition function, the chosen parking spot will change with probability 0.5 at each step, so with

high probability $\mathbf{Ag}_1$ will be able to park before $\mathbf{Ag}_2$ can reach it and a crash can occur. $\blacksquare$

7. Policy iteration

It is known that, for MDPs, PI algorithms generally converge faster than VI algorithms, since policy improvement can jump over policies directly [52]. Motivated by this fact, in this section we show how PI can be used to approximate the values and optimal strategies of an NS-CSG $\mathbf{C}$ with respect to a discounted accumulated reward objective Y . Our algorithm takes ideas from recent work [18], which proposed a new PI method to solve zero-sum stochastic games with finite state spaces, and is the first PI algorithm for CSGs with Borel state spaces and with a convergence guarantee. Our PI algorithm ensures that the strategies and value functions generated during each iteration never leave a finitely representable class of functions. In addition, when computing values of CSGs, efficiencies are gained over alternative algorithms as there is no need to solve normal-form games, which is required by our B-PWC VI and Pollatschek-Avi-Itzhak’s PI algorithm [27], nor to solve MDPs, which adds complexity to Hoffman-Karp’s PI algorithm [26]. This results in cheaper computations and faster convergence over these alternatives, as for PI over VI for MDPs.

7.1. Operators, functions and solutions

Before presenting the algorithm, the following operators, functions and solutions are proposed. Let $\gamma \in \mathbb{R}$ be a constant such that $\gamma > 1$ and $\gamma\beta < 1$, which will be used to distribute the discount factor β between policy evaluation and policy improvement of the two agents.

Operators for Max-Min and Min-Max. Before introducing operators for Max-Min and Min-Max, we require the notion of a stationary Stackelberg (follower) strategy for $\mathbf{Ag}_2$, which is a stochastic kernel $\bar{\sigma}_2 : \Lambda_1 \rightarrow \mathbb{P}(A_2)$, i.e., $\bar{\sigma}_2 \in \mathbb{P}(A_2 \mid \Lambda_1)$ such that $\bar{\sigma}_2(A_2(s) \mid (s, u_1)) = 1$ for $(s, u_1) \in \Lambda_1$. This strategy is introduced only for the PI algorithm and implies that $\mathbf{Ag}_2$ makes decisions conditioned on the current state s and the current choice of $\mathbf{Ag}_1$, i.e. action distribution u_1 , and thus allows us to split the maximum and minimum operations of the two agents. We denote by $\bar{\Sigma}_2$ the set of all stationary Stackelberg strategies for $\mathbf{Ag}_2$.

Definition 11 (Operator for the Max-Min value). For strategy $\sigma_1 \in \Sigma_1$ of $\mathbf{Ag}_1$ and function $V_2 \in \mathbb{F}(\Lambda_1)$, we define the operator $H_{\sigma_1, V_2}^1 : \mathbb{F}(\Lambda_1) \rightarrow \mathbb{F}(S)$ such that for $J_2 \in \mathbb{F}(\Lambda_1)$ and $s \in S$:

$$\begin{aligned} [H_{\sigma_1, V_2}^1 J_2](s) &= \gamma^{-1} \min\{J_2(s, \sigma_1(s)), V_2(s, \sigma_1(s))\} \\ &= \gamma^{-1} \min\{J_2(s, u_1), V_2(s, u_1)\} \end{aligned}$$

where $\sigma_1(s) = u_1 \in \mathbb{P}(A_1(s))$.

Definition 12 (Operator for the Min-Max value). For Stackelberg (follower) strategy $\bar{\sigma}_2 \in \bar{\Sigma}_2$ of $\mathbf{Ag}_2$ and function $V_1 \in \mathbb{F}(S)$, we define the operator $H_{\bar{\sigma}_2, V_1}^2 : \mathbb{F}(S) \rightarrow \mathbb{F}(\Lambda_1)$ such that for $J_1 \in \mathbb{F}(S)$ and $(s, u_1) \in \Lambda_1$:

$$\begin{aligned} [H_{\bar{\sigma}_2, V_1}^2 J_1](s, u_1) &= \sum_{(a_1, a_2) \in A(s)} Q(s, (a_1, a_2), \gamma \max[J_1, V_1]) u_1(a_1) \bar{\sigma}_2(a_2 | (s, u_1)) \\ &= \sum_{(a_1, a_2) \in A(s)} Q(s, (a_1, a_2), \gamma \max[J_1, V_1]) u_1(a_1) u_2(a_2) \end{aligned}$$

where $\bar{\sigma}_2(\cdot | (s, u_1)) = u_2 \in \mathbb{P}(A_2(s))$.

Unlike the classical PI algorithms by Hoffman and Karp [26] and Pollatschek and Avi-Itzhak [27], following [18], our PI algorithm separates the policy evaluation and policy improvement of the maximiser ($\mathbf{Ag}_1$) and the minimiser ($\mathbf{Ag}_2$) through the use of the operators of Definition 11 and Definition 12, respectively. To track the value functions after performing policy evaluation of $\mathbf{Ag}_1$ and $\mathbf{Ag}_2$, our PI algorithm introduces value functions J_1 and J_2 . In addition, the value functions V_1 and V_2 are introduced to avoid the oscillatory behavior of the Pollatschek and Avi-Itzhak PI algorithm [27], thus ensuring convergence, and are updated only during policy improvement. The role of γ is to split the discount factor β such that all the operators corresponding to policy evaluation and policy improvement of the two agents are contraction mappings, which then ensures convergence.

Two function representations. We next define two classes of functions, which play a key role in characterizing the functions and strategies generated during each iteration of our PI algorithm.

Definition 13 (CON-PWL Borel measurable function). A function $f \in \mathbb{F}(\Lambda_1)$ is a constant-piecewise-linear (CON-PWL) Borel measurable function if there exists a BFCP Φ of S such that, for each $\phi \in \Phi$, $A_1(s) = A_1(s')$ for $s, s' \in \phi$, and Φ generates $\Theta = \{\theta(\phi) \mid \phi \in \Phi\}$ where $\theta(\phi) = \{(s, u_1) \in \Lambda_1 \mid s \in \phi\}$, a BFCP of Λ_1 , such that for $\theta(\phi) \in \Theta$:

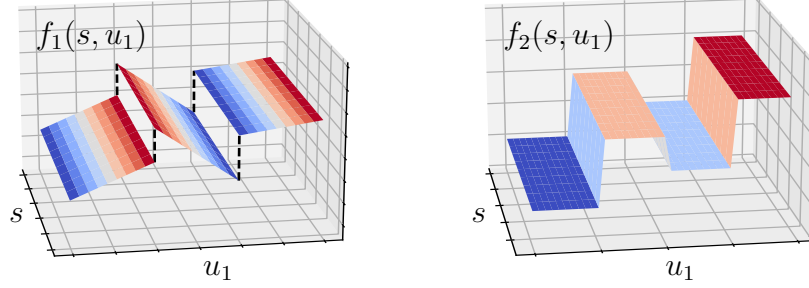


Figure 7: Two functions over one region of a BFCP of Λ_1 . Constant-piecewise-linear (CON-PWL) Borel measurable function (left): given u_1 , $f_1(s, u_1)$ is constant in s , and given s , $f_1(s, u_1)$ is B-PWL in u_1 . Constant-piecewise-constant (CON-PWC) stochastic kernel (right): given u_1 , $f_2(s, u_1)$ is constant in s , and given s , $f_2(s, u_1)$ is B-PWC in u_1 .

- (i) $f(\cdot, u_1) : \phi \rightarrow \mathbb{R}$ is constant for $u_1 \in \mathbb{P}(A_1(s))$ where $s \in \phi$;
- (ii) $f(s, \cdot) : \mathbb{P}(A_1(s)) \rightarrow \mathbb{R}$ is B-PWL for $s \in \phi$.

Definition 14 (CON-PWC stochastic kernel). A function $f \in \bar{\Sigma}_2$ is a constant-piecewise-constant (CON-PWC) stochastic kernel if there exists a BFCP Φ of S such that, for each $\phi \in \Phi$, $A(s) = A(s')$ for $s, s' \in \phi$, and Φ generates $\Theta = \{\theta(\phi) \mid \phi \in \Phi\}$ where $\theta(\phi) = \{(s, u_1) \in \Lambda_1 \mid s \in \phi\}$, a BFCP of Λ_1 , such that for $\theta(\phi) \in \Theta$:

- (i) $f(\cdot, u_1) : \phi \rightarrow \mathbb{P}(A_2(s))$ is constant for $u_1 \in \mathbb{P}(A_1(s))$ where $s \in \phi$;
- (ii) $f(s, \cdot) : \mathbb{P}(A_1(s)) \rightarrow \mathbb{P}(A_2(s))$ is B-PWC for $s \in \phi$.

Fig. 7 presents an example of a CON-PWL Borel measurable function and CON-PWC stochastic kernel over a region. We now show that these two functions can be represented by finite sets of vectors. Each CON-PWL Borel measurable function f can be represented by a finite set of vectors $\{(D_{\phi, \phi'}, b_{\phi, \phi'}) \in \mathbb{R}^{|A_1|} \times \mathbb{R} \mid \phi \in \Phi \wedge \phi' \in \Phi'(\phi)\}$ such that $f(s, u_1) = D_{\phi, \phi'}^\top u_1 + b_{\phi, \phi'}$ for $s \in \phi$ and $u_1 \in \phi'$, where Φ is a BFCP of S for f using Definition 13 and $\Phi'(\phi)$ is a BFCP of $\{u_1 \in \mathbb{P}(A_1) \mid (s, u_1) \in \theta(\phi)\}$, and $\theta(\phi) \in \Theta$ again using Definition 13 is such that, over each region $\phi' \in \Phi'(\phi)$, $f(s, u_1)$ is linear in u_1 given $s \in \phi$. Similarly using Definition 14, each CON-PWC stochastic kernel f can be represented by a finite set of vectors $\{D_{\phi, \phi'} \in \mathbb{P}(A_2) \mid \phi \in \Phi \wedge \phi' \in \Phi'(\phi)\}$ such that $f(s, u_1) = D_{\phi, \phi'}$ for $s \in \phi$ and $u_1 \in \phi'$, where Φ is a BFCP of S for f using Definition 14, $\Phi'(\phi)$ is a BFCP of $\{u_1 \in \mathbb{P}(A_1) \mid (s, u_1) \in \theta(\phi)\}$, $\theta(\phi) \in \Theta$ using Definition 14 is such that, over each region $\phi' \in \Phi'(\phi)$, $f(s, u_1)$ is constant in u_1 given $s \in \phi$.

Maximum or minimum solutions. We introduce a criterion for selecting the maximum or minimum solution over a region, by which the strategies from policy improvement are finitely representable.

Definition 15 (CON-1 solution). Let $f \in \mathbb{F}(\Lambda_1)$ be a CON-PWL Borel measurable function. Using Definition 13 there exists a BFCP Φ of S for f . Now, for each $\phi \in \Phi$, if there exists $u_1^\phi \in \mathbb{P}(A_1(s))$ such that:

$$f(s, u_1^\phi) = \max_{u_1 \in \mathbb{P}(A_1(s))} f(s, u_1)$$

for $s \in \phi$, and σ_1 is a strategy of $\mathbf{Ag}_1$ such that $\sigma_1(s) = u_1^\phi$ for $s \in \phi$, then σ_1 is a constant-1 (CON-1) solution of f over ϕ .

Definition 16 (CON-2 solution). Let $f \in \mathbb{F}(\Lambda_{12})$ be a Borel measurable function. If there exists a BFCP Θ of Λ_1 where, for each $\theta \in \Theta$, $A_2(s)$ is constant for $(s, u_1) \in \theta$ and there exists $u_2^\theta \in \mathbb{P}(A_2(s))$ such that:

$$f(s, u_1, u_2^\theta) = \min_{u_2 \in \mathbb{P}(A_2(s))} f(s, u_1, u_2)$$

for $(s, u_1) \in \theta$, and $\bar{\sigma}_2$ is a Stackelberg strategy for $\mathbf{Ag}_2$ such that $\bar{\sigma}_2(s, u_1) = u_2^\theta$ for $(s, u_1) \in \theta$, then $\bar{\sigma}_2$ is a constant-2 (CON-2) solution of f over θ .

7.2. Minimax-action-free PI

We now use the operators for the Max-Min and Min-Max values (Definitions 11 and 12) to derive a PI algorithm called *Minimax-action-free PI* (Algorithm 3) for strategy synthesis for NS-CSGs with Borel state spaces. Our algorithm closely follows the PI method of [18] for finite state spaces, but has to resolve a number of issues due to the uncountability of the underlying state space and the need to ensure Borel measurability at each iteration. To overcome these issues we (i) introduce CON-PWL Borel measurable functions (Definition 13) and CON-PWC Borel measurable strategies (Definition 14) to ensure measurability and finite representability; (ii) work with CON-1 and CON-2 solutions (Definition 15 and (Definition 16) for policy improvement to ensure that the strategies generated are finitely representable and consistent; and (iii) propose a BFCP iteration algorithm (Algorithm 4) and a BFCP-based computation algorithm (Algorithm 5) to compute a new BFCP of the state space and the values or strategies over this BFCP. We also provide a simpler proof than that presented in [18], which does not require the introduction of any new concepts except those used in the algorithm.

ALGORITHM 3 Iteration t of Minimax-action-free PI

- 1: **Input:** NS-CSG $\mathbf{C}$, PWC $\sigma_1^t \in \Sigma_1$, CON-PWC $\bar{\sigma}_2^t \in \bar{\Sigma}_2$, PWC $J_1^t, V_1^t \in \mathbb{F}(S)$, CON-PWL $J_2^t, V_2^t \in \mathbb{F}(\Lambda_1)$
 - 2: **Perform one of the following four iterations.**
 - 3: Policy evaluation of $\mathbf{Ag}_1$:
 - 4: $J_1^{t+1} \leftarrow [H_{\sigma_1^t, V_2^t}^1 J_2^t]$ via *PE1*, $\sigma_1^{t+1} \leftarrow \sigma_1^t$,
 - 5: $V_1^{t+1} \leftarrow V_1^t$, $\bar{\sigma}_2^{t+1} \leftarrow \bar{\sigma}_2^t$, $J_2^{t+1} \leftarrow J_2^t$, $V_2^{t+1} \leftarrow V_2^t$
 - 6: Policy improvement of $\mathbf{Ag}_1$ by CON-1 solution:
 - 7: $\sigma_1^{t+1}(s) \in \operatorname{argmax}_{u_1 \in \mathbb{P}(A_1(s))} [H_{u_1, V_2^t}^1 J_2^t](s)$,
 - 8: $V_1^{t+1} \leftarrow [H_{\sigma_1^{t+1}, V_2^t}^1 J_2^t]$ via *PI1*,
 - 9: $J_1^{t+1} \leftarrow J_1^t$, $\bar{\sigma}_2^{t+1} \leftarrow \bar{\sigma}_2^t$, $J_2^{t+1} \leftarrow J_2^t$, $V_2^{t+1} \leftarrow V_2^t$
 - 10: Policy evaluation of $\mathbf{Ag}_2$:
 - 11: $J_2^{t+1} \leftarrow [H_{\bar{\sigma}_2^t, V_1^t}^2 J_1^t]$ via *PE2*, $\sigma_1^{t+1} \leftarrow \sigma_1^t$,
 - 12: $J_1^{t+1} \leftarrow J_1^t$, $V_1^{t+1} \leftarrow V_1^t$, $\bar{\sigma}_2^{t+1} \leftarrow \bar{\sigma}_2^t$, $V_2^{t+1} \leftarrow V_2^t$
 - 13: Policy improvement of $\mathbf{Ag}_2$ by CON-2 solution:
 - 14: $\bar{\sigma}_2^{t+1}(s, u_1) \in \operatorname{argmin}_{u_2 \in \mathbb{P}(A_2(s))} [H_{u_2, V_1^t}^2 J_1^t](s, u_1)$,
 - 15: $V_2^{t+1} \leftarrow [H_{\bar{\sigma}_2^{t+1}, V_1^t}^2 J_1^t]$ via *PI2*,
 - 16: $\sigma_1^{t+1} \leftarrow \sigma_1^t$, $J_1^{t+1} \leftarrow J_1^t$, $V_1^{t+1} \leftarrow V_1^t$, $J_2^{t+1} \leftarrow J_2^t$
 - 17: $t \leftarrow t + 1$
-

Initialization. The Minimax-action-free PI algorithm is initialized with strategies σ_1^0 and $\bar{\sigma}_2^0$ for each player, which are uniform distributions over available actions/state-action pairs, i.e., $\sigma_1^0(s) = \frac{1}{|A_1(s)|}$ for all $s \in S$ and $\bar{\sigma}_2^0(s, u_1) = \frac{1}{|A_2(s)|}$ for all $(s, u_1) \in \Lambda_1$, and four 0-valued functions, $J_1^0, V_1^0, J_2^0, V_2^0$, i.e., $J_1^0(s) = V_1^0(s) = 0$ for all $s \in S$ and $J_2^0(s, u_1) = V_2^0(s, u_1) = 0$ for all $(s, u_1) \in \Lambda_1$, and Algorithm 4 gives one BFCP for each strategy and function,

The algorithm. An iteration of the Minimax-action-free PI is given in Algorithm 3. As shown later, the order and frequency by which the possible four iterations of Algorithm 3 are run do not affect the convergence, as long as each is performed infinitely often. This permits an asynchronous implementation of the Minimax-action-free PI algorithm, as discussed in [18] and for its single-agent counterparts in [53].

For each of the four iterations, Algorithm 4 provides a way to compute new BFCPs and the results below demonstrate that, over each region of

ALGORITHM 4 BFCP iteration t for Minimax-action-free PI

```

1: Input: Perception FCP  $\Phi_P$ , reward FCPs  $(\Phi_R^\alpha)_{\alpha \in A}$ 
2: Output: BFCPs  $\langle \Phi_{J_1^t}, \Phi_{V_1^t}, \Phi_{\sigma_1^t}, \Theta_{J_2^t}, \Theta_{V_2^t}, \Theta_{\bar{\sigma}_2^t} \rangle_{t \in \mathbb{N}}$ 
   for  $\langle J_1^t, V_1^t, \sigma_1^t, J_2^t, V_2^t, \bar{\sigma}_2^t \rangle_{t \in \mathbb{N}}$ 
3:  $\Phi_{J_1^0}, \Phi_{V_1^0}, \Phi_{\sigma_1^0} \leftarrow \{S\}, \Theta_{J_2^0}, \Theta_{V_2^0}, \Theta_{\bar{\sigma}_2^0} \leftarrow \{\Lambda_1\}$ 
4: while Algorithm 3 performs iteration  $t$  do
5:   if policy evaluation of  $\mathbf{Ag}_1$  is chosen then
6:      $Preprocess\_maximiser()$ ,
7:      $\Phi_{J_1^{t+1}} \leftarrow \Phi_{\sigma_1^t} + \Phi_{J_2^t} + \Phi_{V_2^t}, \Phi_{V_1^{t+1}} \leftarrow \Phi_{V_1^t}, \Phi_{\sigma_1^{t+1}} \leftarrow \Phi_{\sigma_1^t}$ 
8:   if policy improvement of  $\mathbf{Ag}_1$  is chosen then
9:      $Preprocess\_maximiser()$ ,
10:     $\Phi_{\sigma_1^{t+1}} \leftarrow \Phi_{J_2^t} + \Phi_{V_2^t}, \Phi_{V_1^{t+1}} \leftarrow \Phi_{\sigma_1^{t+1}}, \Phi_{J_1^{t+1}} \leftarrow \Phi_{J_1^t}$ 
11:   if policy evaluation of  $\mathbf{Ag}_2$  is chosen then
12:      $Preprocess\_minimiser()$ ,
13:      $\Phi_{\bar{\sigma}_2^t} \leftarrow \{\{s \mid (s, u_1) \in \theta\} \mid \theta \in \Theta_{\bar{\sigma}_2^t}\},$ 
14:      $\Theta_{J_2^{t+1}} \leftarrow \{\{(s, u_1) \in \Lambda_1 \mid s \in \phi\} \mid \phi \in \Phi_{\hat{Q}^{t+1}} + \Phi_{\bar{\sigma}_2^t}\},$ 
15:      $\Theta_{V_2^{t+1}} \leftarrow \Theta_{V_2^t}, \Theta_{\bar{\sigma}_2^{t+1}} \leftarrow \Theta_{\bar{\sigma}_2^t}$ 
16:   if policy improvement of  $\mathbf{Ag}_2$  is chosen then
17:      $Preprocess\_minimiser()$ ,
18:      $\Theta_{\bar{\sigma}_2^{t+1}} \leftarrow \{\{(s, u_1) \in \Lambda_1 \mid s \in \phi\} \mid \phi \in \Phi_{\hat{Q}^{t+1}}\},$ 
19:      $\Theta_{V_2^{t+1}} \leftarrow \Theta_{\bar{\sigma}_2^{t+1}}, \Theta_{J_2^{t+1}} \leftarrow \Theta_{J_2^t}$ 
20: return  $\langle \Phi_{J_1^t}, \Phi_{V_1^t}, \Phi_{\sigma_1^t}, \Theta_{J_2^t}, \Theta_{V_2^t}, \Theta_{\bar{\sigma}_2^t} \rangle_{t \in \mathbb{N}}$ 
21:
22: procedure  $Preprocess\_maximiser()$ 
23:    $\Theta_{J_2^{t+1}} \leftarrow \Theta_{J_2^t}, \Theta_{V_2^{t+1}} \leftarrow \Theta_{V_2^t}, \Phi_{J_2^t} \leftarrow \{\{s \mid (s, u_1) \in \theta\} \mid \theta \in \Theta_{J_2^t}\},$ 
24:    $\Phi_{V_2^t} \leftarrow \{\{s \mid (s, u_1) \in \theta\} \mid \theta \in \Theta_{V_2^t}\}, \Theta_{\bar{\sigma}_2^{t+1}} \leftarrow \Theta_{\bar{\sigma}_2^t}$ 
25: procedure  $Preprocess\_minimiser()$ 
26:    $\Phi_{J_1^{t+1}} \leftarrow \Phi_{J_1^t}, \Phi_{V_1^{t+1}} \leftarrow \Phi_{V_1^t}, \Phi_{\sigma_1^{t+1}} \leftarrow \Phi_{\sigma_1^t},$ 
27:    $\Phi_{\hat{Q}^{t+1}} \leftarrow Preimage\_BFCP(\Phi_{J_1^t} + \Phi_{V_1^t}, \Phi_P, (\Phi_R^\alpha)_{\alpha \in A})$ 

```

ALGORITHM 5 BFCP based computation for Minimax-action-free PI

```

1: Input:  $J_1^t, V_1^t, \sigma_1^t, J_2^t, V_2^t, \bar{\sigma}_2^t, \Phi_{J_1^{t+1}}, \Phi_{\sigma_1^{t+1}}, \Theta_{J_2^{t+1}}, \Theta_{\bar{\sigma}_2^{t+1}}$ 
2: procedure PE1
3:   for  $\phi \in \Phi_{J_1^{t+1}}$  do
4:     Take one state  $s \in \phi$ , and then  $J_{1,\phi}^{t+1} \leftarrow [H_{\sigma_1^t, V_2^t}^1 J_2^t](s)$ 
5:   return  $J_1^{t+1} \leftarrow (J_{1,\phi}^{t+1})_{\phi \in \Phi_{J_1^{t+1}}}$ 
6: procedure PI1
7:   for  $\phi \in \Phi_{\sigma_1^{t+1}}$  do
8:     Take  $s \in \phi$ , and then  $u_1 \in \operatorname{argmax}_{u_1 \in \mathbb{P}(A_1(s))} [H_{u_1, V_2^t}^1 J_2^t](s)$ 
9:      $\sigma_{1,\phi}^{t+1} \leftarrow u_1, V_{1,\phi}^{t+1} \leftarrow \max_{u_1 \in \mathbb{P}(A_1(s))} [H_{u_1, V_2^t}^1 J_2^t](s)$ 
10:  return  $\sigma_1^{t+1} \leftarrow (\sigma_{1,\phi}^{t+1})_{\phi \in \Phi_{\sigma_1^{t+1}}}, V_1^{t+1} \leftarrow (V_{1,\phi}^{t+1})_{\phi \in \Phi_{\sigma_1^{t+1}}}$ 
11: procedure PE2
12:   for  $\theta \in \Theta_{J_2^{t+1}}$  do
13:      $\phi \leftarrow \{s \mid (s, u_1) \in \theta\}$ 
14:     Take  $s \in \phi$ , and then compute a BFCP  $\Phi_u$  of  $\mathbb{P}(A_1(s))$  such that
        over  $\phi_u \in \Phi_u$ ,  $[H_{\bar{\sigma}_2^t, V_1^t}^2 J_1^t](s, u_1)$  is linear in  $u_1$ 
15:      $J_{2,\theta}^{t+1} \leftarrow [H_{\bar{\sigma}_2^t, V_1^t}^2 J_1^t](s, u_1)$  is linear in  $u_1$ 
16:   return  $J_2^{t+1} \leftarrow (J_{2,\theta}^{t+1})_{\theta \in \Theta_{J_2^{t+1}}}$ 
17: procedure PI2
18:   for  $\theta \in \Theta_{\bar{\sigma}_2^{t+1}}$  do
19:      $\phi \leftarrow \{s \mid (s, u_1) \in \theta\}$ 
20:     Take  $s' \in \phi$ , and then compute a BFCP  $\Phi_u$  of  $\mathbb{P}(A_1(s'))$  such that
        over  $\phi_u \in \Phi_u$ ,  $\min_{u_2 \in \mathbb{P}(A_2(s'))} [H_{u_2, V_1^t}^2 J_1^t](s', u_1)$  is constant for  $u_1 \in \phi_u$ 
21:     Take  $u'_1 \in \phi_u$  and  $u'_2 \in \operatorname{argmin}_{u_2 \in \mathbb{P}(A_2(s'))} [H_{u_2, V_1^t}^2 J_1^t](s', u'_1)$  for  $\phi_u \in \Phi_u$ 
22:      $\bar{\sigma}_{2,\theta}^{t+1} \leftarrow u'_2, V_{2,\theta}^{t+1} \leftarrow [H_{u'_2, V_1^t}^2 J_1^t](s', u_1)$  is linear in  $u_1$ 
23:   return  $\bar{\sigma}_2^{t+1} \leftarrow (\bar{\sigma}_{2,\theta}^{t+1})_{\theta \in \Theta_{\bar{\sigma}_2^{t+1}}}, V_2^{t+1} \leftarrow (V_{2,\theta}^{t+1})_{\theta \in \Theta_{\bar{\sigma}_2^{t+1}}}$ 

```

these BFCPs, the corresponding computed strategies and value functions are either constant, PWC or PWL. Therefore, we can follow similar steps to our VI algorithm (see Algorithm 1) to compute the value functions of these new strategies and value functions (see Algorithm 5). The idea is to first compute the BFCPs $\Phi_{J_1^{t+1}}$, $\Phi_{V_1^{t+1}}$, $\Phi_{\sigma_1^{t+1}}$, $\Theta_{J_2^{t+1}}$, $\Theta_{V_2^{t+1}}$ and $\Theta_{\bar{\sigma}_2^{t+1}}$ via Algorithm 4 and then use them to compute strategies and value functions using Algorithm 5. For instance, if policy improvement of $\mathbf{Ag}_2$ is chosen at iteration $t \in \mathbb{N}$ then we proceed as follows. First, new BFCPs are computed via Algorithm 4. Second, procedure *PI2* of Algorithm 5 is performed. In this step we take each region $\theta \in \Theta_{\bar{\sigma}_2^{t+1}}$, let $\phi = \{s \mid (s, u_1) \in \theta\}$, then take one state $s' \in \phi$, and compute a BFCP Φ_u of $\mathbb{P}(A_1(s'))$ such that $\min_{u_2 \in \mathbb{P}(A_2(s'))} [H_{u_2, V_1^t}^2 J_1^t](s', u_1)$ is constant over $\phi_u \in \Phi_u$ and for $u_1 \in \phi_u$. Third, take one $u'_1 \in \phi_u$ and find $u'_2 \in \mathbb{P}(A_2(s'))$ that minimises $[H_{u_2, V_1^t}^2 J_1^t](s', u'_1)$. Fourth, we let $\bar{\sigma}_2^{t+1}(s, u_1) = u'_2$ for $s \in \phi$ and $u_1 \in \phi_u$, which is a CON-2 solution of $[H_{u_2, V_1^t}^2 J_1^t](s, u_1)$ over $\{(s, u_1) \mid s \in \phi \wedge u_1 \in \phi_u\}$ by Lemma 9 and $V_2^{t+1}(s, u_1)$ is CON-linear in $s \in \phi$ and $u_1 \in \phi_u$. Finally, we copy the other strategies and value functions for the next iteration.

Representation closures. The following lemmas show the strategies and value functions generated during each iteration of the Minimax-action-free PI algorithm are closed under B-PWC, CON-PWL and CON-PWC functions, and are thus finitely representable.

Lemma 6 (Evaluation closure for $\mathbf{Ag}_1$). *If $\sigma_1^t \in \Sigma_1$ is a PWC stochastic kernel, $J_2^t, V_2^t \in \mathbb{F}(\Lambda_1)$ are CON-PWL Borel measurable and policy evaluation of $\mathbf{Ag}_1$ is performed (procedure *PE1*), then $J_1^{t+1} = [H_{\sigma_1^t, V_2^t}^1 J_2^t]$ is B-PWC.*

Proof. Suppose $\sigma_1^t \in \Sigma_1$ is a PWC stochastic kernel and $J_2^t, V_2^t \in \mathbb{F}(\Lambda_1)$ are CON-PWL Borel measurable. Since σ_1^t is a PWC stochastic kernel, there exists a constant-BFCP $\Phi_{\sigma_1^t}$ of S for σ_1^t . Since J_2^t is a CON-PWL Borel measurable function, there exists a BFCP $\Phi_{J_2^t}$ of S satisfying the properties of Definition 13 for J_2^t . Therefore $J_2^t(s, \sigma_1^t(s))$ is constant on each region of the BFCP $\Phi_{\sigma_1^t} + \Phi_{J_2^t}$. We can similarly show that $V_2^t(s, \sigma_1^t(s))$ is constant on each region of the BFCP $\Phi_{\sigma_1^t} + \Phi_{V_2^t}$, where $\Phi_{V_2^t}$ is a BFCP of S from Definition 13 for V_2^t . Consider the policy evaluation of $\mathbf{Ag}_1$ (procedure *PE1*). Using Definition 11 we have that $J_1^{t+1} = [H_{\sigma_1^t, V_2^t}^1 J_2^t]$ is constant on each region of the BFCP $\Phi_{\sigma_1^t} + \Phi_{J_2^t} + \Phi_{V_2^t}$, which also implies that J_1^{t+1} is Borel measurable. Since J_2^t and V_2^t are bounded, then J_1^{t+1} is also bounded as required. $\square$

Lemma 7 (Improvement closure for Ag_1). *If $J_2^t, V_2^t \in \mathbb{F}(\Lambda_1)$ are CON-PWL Borel measurable and policy improvement of Ag_1 is performed (procedure PI1), then $\sigma_1^{t+1}(s) \in \arg\max_{u_1 \in \mathbb{P}(A_1(s))} [H_{u_1, V_2^t}^1 J_2^t](s)$ is a PWC stochastic kernel, and $V_1^{t+1} = [H_{\sigma_1^{t+1}, V_2^t}^1 J_2^t]$ is B-PWC.*

Proof. Suppose $J_2^t, V_2^t \in \mathbb{F}(\Lambda_1)$ are CON-PWL Borel measurable functions. Using [46, Chapter 18.1] and Definition 13 it follows that the function $K^t := \min[J_2^t, V_2^t]$ is Borel measurable. Note that, over each region of $\Phi_{J_2^t} + \Phi_{V_2^t}$, $K^t(s, u_1)$ is constant in s given u_1 , and PWL in u_1 given s (where $\Phi_{J_2^t}$ and $\Phi_{V_2^t}$ are from Lemma 6), and therefore K^t is CON-PWL.

Let $\Phi_{K^t} = \Phi_{J_2^t} + \Phi_{V_2^t}$ and Θ_{K^t} be a BFCP of Λ_1 satisfying the properties of Definition 13 for K^t . Every state in each region of the BFCP Φ_{K^t} has the same set of available actions for Ag_1 and same strategy u_1 that maximises $K^t(s, u_1)$ on a region of Θ_{K^t} . Therefore, using the CON-1 solution in Definition 15, the strategy of Ag_1 :

$$\sigma_1^{t+1}(s) \in \arg\max_{u_1 \in \mathbb{P}(A_1(s))} [H_{u_1, V_2^t}^1 J_2^t](s)$$

is constant on each region of Φ_{K^t} , which also implies that σ_1^{t+1} is Borel measurable. Since σ_1^{t+1} is a PWC stochastic kernel, then Lemma 6 implies that V_1^{t+1} is B-PWC as required. $\square$

Lemma 8 (Evaluation closure for Ag_2). *If $J_1^t, V_1^t \in \mathbb{F}(S)$ are B-PWC and $\bar{\sigma}_2^t \in \bar{\Sigma}_2$ is a CON-PWC stochastic kernel and policy evaluation of Ag_2 is performed (procedure PE2), then $J_2^{t+1} = [H_{\bar{\sigma}_2^t, V_1^t}^2 J_1^t]$ is CON-PWL Borel measurable.*

Proof. Suppose J_1^t and V_1^t are B-PWC and $\bar{\sigma}_2^t \in \bar{\Sigma}_2$ is a CON-PWC stochastic kernel. Using [46, Chapter 18.1] it follows that $\gamma \max[J_1^t, V_1^t]$ is B-PWC. In view of the B-PWC function $Q(\cdot, \alpha, V)$ in Theorem 2, for each $\alpha \in A$ the function:

$$\hat{Q}_\alpha^t(s) := Q(s, \alpha, \gamma \max[J_1^t, V_1^t])$$

is B-PWC. Let $\Phi_{\hat{Q}^t}$ be a BFCP of S such that $\hat{Q}_\alpha^t$ is constant on each region of $\Phi_{\hat{Q}^t}$ for $\alpha \in A$. It follows that $A(s)$ is constant on each region of $\Phi_{\hat{Q}^t}$.

Next, let $\Phi_{\bar{\sigma}_2^t}$ be a BFCP of S satisfying the properties of Definition 14 for the CON-PWC stochastic kernel $\bar{\sigma}_2^t$. For the BFCP $\Phi_{\hat{Q}^t} + \Phi_{\bar{\sigma}_2^t}$ of S , we generate a BFCP Θ_1^t of Λ_1 such that each region $\theta_1^t(\phi) \in \Theta_1^t$, induced by a region $\phi \in \Phi_{\hat{Q}^t} + \Phi_{\bar{\sigma}_2^t}$, is given by $\theta_1^t(\phi) = \{(s, u_1) \in \Lambda_1 \mid s \in \phi\}$.

Finally, consider the policy evaluation of $\mathbf{Ag}_2$. According to Definition 12, for $(s, u_1) \in \theta_1^t(\phi)$, $J_2^{t+1}(s, u_1) = [H_{\bar{\sigma}_2^t, V_1^t}^2 J_1^t](s, u_1)$ is constant in s for a fixed u_1 , and PWC in u_1 for a fixed $s \in S$. Thus, J_2^{t+1} is CON-PWL. Since $\hat{Q}_\alpha^t$ and $\bar{\sigma}_2^t$ are bounded, Borel measurable, then so is J_2^{t+1} by Definition 12 as required. $\square$

Lemma 9 (Improvement closure for $\mathbf{Ag}_2$). *If $J_1^t, V_1^t \in \mathbb{F}(S)$ are B-PWC and policy improvement of $\mathbf{Ag}_2$ is performed (procedure PI2), then $\bar{\sigma}_2^{t+1}(s, u_1) \in \operatorname{argmin}_{u_2 \in \mathbb{P}(A_2(s))} [H_{u_2, V_1^t}^2 J_1^t](s, u_1)$ is a CON-PWC stochastic kernel, and $V_2^{t+1} = [H_{\bar{\sigma}_2^{t+1}, V_1^t}^2 J_1^t]$ is CON-PWL Borel measurable.*

Proof. Suppose $J_1^t, V_1^t \in \mathbb{F}(S)$ are B-PWC. For the BFCP $\Phi_{\hat{Q}^t}$ of S , we generate a BFCP Θ_2^t of Λ_1 such that each region $\theta_2^t(\phi)$ in Θ_2^t induced by a region $\phi \in \Phi_{\hat{Q}^t}$ is given by $\theta_2^t(\phi) = \{(s, u_1) \in \Lambda_1 \mid s \in \phi\}$, where $\Phi_{\hat{Q}^t}$ is from the proof of Lemma 8. Consider the policy improvement of $\mathbf{Ag}_2$ (procedure PI2). According to Definition 12, by using the CON-2 solution in Definition 16, for $(s, u_1) \in \theta_2^t(\phi)$, the Stackelberg strategy of $\mathbf{Ag}_2$:

$$\bar{\sigma}_2^{t+1}(s, u_1) \in \operatorname{argmin}_{u_2 \in \mathbb{P}(A_2(s))} [H_{u_2, V_1^t}^2 J_1^t](s, u_1)$$

is constant in s for a fixed u_1 , and PWC in u_1 for a fixed s . Thus, $\bar{\sigma}_2^{t+1}$ is CON-PWC. Since $\bar{\sigma}_2^{t+1}$ is a CON-PWC stochastic kernel, then Lemma 8 implies that V_2^{t+1} is CON-PWL Borel measurable as required. $\square$

By fusing Lemmas 6, 7, 8 and 9 we can prove that the strategies and value functions generated during each iteration of Algorithm 3 never leave a finitely representable class of functions, and Algorithm 4 constructs new BFCPs such that the strategies and value functions after one iteration of the Minimax-action-free PI algorithm remain constant, PWC, or PWL on each region of the constructed BFCPs.

Theorem 3 (Representation closure). *In any iteration of the Minimax-action-free PI algorithm (see Algorithm 3), if*

- (i) $J_1^t, V_1^t \in \mathbb{F}(S)$ are B-PWC and $\sigma_1^t \in \Sigma_1$ is a PWC stochastic kernel;
- (ii) $J_2^t, V_2^t \in \mathbb{F}(\Lambda_1)$ are CON-PWL Borel measurable and $\bar{\sigma}_2^t \in \bar{\Sigma}_2$ is a CON-PWC stochastic kernel;

then so are $J_1^{t+1}, V_1^{t+1}, \sigma_1^{t+1}, J_2^{t+1}, V_2^{t+1}$ and $\bar{\sigma}_2^{t+1}$, respectively, regardless of which one of the four iterations is performed.

Proof. The conclusion follows from one of Lemmas 6, 7, 8 and 9, depending on which one of the four iterations is executed. $\square$

Corollary 2 (BFCP iteration for Minimax-action-free PI). *After performing Algorithm 4:*

- (i) $\Phi_{J_1^{t+1}}, \Phi_{V_1^{t+1}}$ and $\Phi_{\sigma_1^{t+1}}$ are constant-BFCPs of S for $J_1^{t+1} = [H_{\sigma_1^t, V_2^t}^1 J_2^t]$, $V_1^{t+1} = [H_{\sigma_1^{t+1}, V_2^t}^1 J_2^t]$ and $\sigma_1^{t+1}(s) \in \operatorname{argmax}_{u_1 \in \mathbb{P}(A_1(s))} [H_{u_1, V_2^t}^1 J_2^t](s)$;
- (ii) $\Theta_{J_2^{t+1}}$ and $\Theta_{V_2^{t+1}}$ are BFCPs of Λ_1 for $J_2^{t+1} = [H_{\bar{\sigma}_2^t, V_1^t}^2 J_1^t]$ and $V_2^{t+1} = [H_{\bar{\sigma}_2^{t+1}, V_1^t}^2 J_1^t]$ meeting the conditions of Definition 13, and $\Theta_{\bar{\sigma}_2^{t+1}}$ is a BFCP of Λ_1 for $\bar{\sigma}_2^{t+1}(s, u_1) \in \operatorname{argmin}_{u_2 \in \mathbb{P}(A_2(s))} [H_{u_2, V_1^t}^2 J_1^t](s, u_1)$ meeting the conditions of Definition 14.

7.3. Convergence analysis and strategy computation

We next prove the convergence of the Minimax-action-free PI algorithm by showing that there exists an operator from the product space of the function spaces over which J_1, V_1, J_2 and V_2 are defined to itself, which is a contraction mapping with a unique fixed point, one of whose components is the value function multiplied by a known constant. The proof closely follows the steps for finite state spaces given in [18], but is more complex due to the underlying uncountable state space and the need to deal with the requirement of Borel measurability and finite representation of strategies and value functions.

Convergence analysis. Given PWC $\sigma_1 \in \Sigma_1$ and CON-PWC $\bar{\sigma}_2 \in \bar{\Sigma}_2$, we define the operator $G_{\sigma_1, \bar{\sigma}_2} : (\mathbb{F}(S) \times \mathbb{F}(S) \times \mathbb{F}(\Lambda_1) \times \mathbb{F}(\Lambda_1)) \rightarrow (\mathbb{F}(S) \times \mathbb{F}(S) \times \mathbb{F}(\Lambda_1) \times \mathbb{F}(\Lambda_1))$ such that:

$$G_{\sigma_1, \bar{\sigma}_2}(J_1, V_1, J_2, V_2) := (M_{\sigma_1}^1(J_2, V_2), K^1(J_2, V_2), M_{\bar{\sigma}_2}^2(J_1, V_1), K^2(J_1, V_1)) \quad (2)$$

where we assume $J_1, V_1 \in \mathbb{F}(S)$ are B-PWC, $J_2, V_2 \in \mathbb{F}(\Lambda_1)$ are CON-PWL, and the four operators $M_{\sigma_1}^1, K^1, M_{\bar{\sigma}_2}^2$ and K^2 represent the four iterations of the Minimax-action-free PI algorithm from lines 3 to 16, and are defined as follows.

- $M_{\sigma_1}^1 : \mathbb{F}(\Lambda_1) \times \mathbb{F}(\Lambda_1) \rightarrow \mathbb{F}(S)$ corresponds to the policy evaluation of Ag_1 (procedure *PE1*) where for any $s \in S$:

$$M_{\sigma_1}^1(J_2, V_2)(s) := [H_{\sigma_1, V_2}^1 J_2](s) \quad (3)$$

and is B-PWC using Lemma 6.

- $K^1 : \mathbb{F}(\Lambda_1) \times \mathbb{F}(\Lambda_1) \rightarrow \mathbb{F}(S)$ corresponds to the policy improvement of $\mathbf{Ag}_1$ (procedure *PI1*) where for any $s \in S$:

$$K^1(J_2, V_2)(s) := \max_{u_1 \in \mathbb{P}(A_1(s))} [H_{u_1, V_2}^1 J_2](s) \quad (4)$$

and is B-PWC using Lemma 7.

- $M_{\sigma_2}^2 : \mathbb{F}(S) \times \mathbb{F}(S) \rightarrow \mathbb{F}(\Lambda_1)$ corresponds to the policy evaluation of $\mathbf{Ag}_2$ (procedure *PE2*) where for any $(s, u_1) \in \Lambda_1$:

$$M_{\sigma_2}^2(J_1, V_1)(s, u_1) := [H_{\sigma_2, V_1}^2 J_1](s, u_1) \quad (5)$$

and is CON-PWL Borel measurable using Lemma 8.

- $K^2 : \mathbb{F}(S) \times \mathbb{F}(S) \rightarrow \mathbb{F}(\Lambda_1)$ corresponds to the policy improvement of $\mathbf{Ag}_2$ (procedure *PI2*) where any $(s, u_1) \in \Lambda_1$:

$$K^2(J_1, V_1)(s, u_1) := \min_{u_2 \in \mathbb{P}(A_2(s))} [H_{u_2, V_1}^2 J_1](s, u_1) \quad (6)$$

and is CON-PWL Borel measurable using Lemma 9.

For the spaces $\mathbb{F}(S) \times \mathbb{F}(S)$ and $\mathbb{F}(\Lambda_1) \times \mathbb{F}(\Lambda_1)$, we consider the norm $\|(J, V)\| = \max\{\|J\|, \|V\|\}$, and for the space $\mathbb{F}(S) \times \mathbb{F}(S) \times \mathbb{F}(\Lambda_1) \times \mathbb{F}(\Lambda_1)$ the norm $\|(J_1, V_1, J_2, V_2)\| = \max\{\|J_1\|, \|V_1\|, \|J_2\|, \|V_2\|\}$. We next require the following properties of these norms, which follow from [18].

Lemma 10. *For any $J_1, V_1, J'_1, V'_1 \in \mathbb{F}(S)$ and $J_2, V_2, J'_2, V'_2 \in \mathbb{F}(\Lambda_1)$:*

$$\begin{aligned} \|\max[J_1, V_1] - \max[J'_1, V'_1]\| &\leq \max\{\|J_1 - J'_1\|, \|V_1 - V'_1\|\} \\ \|\min[J_2, V_2] - \min[J'_2, V'_2]\| &\leq \max\{\|J_2 - J'_2\|, \|V_2 - V'_2\|\}. \end{aligned}$$

Proof. Consider any $J_1, V_1, J'_1, V'_1 \in \mathbb{F}(S)$. The norm for the space $\mathbb{F}(S)$ implies that for any $s \in S$:

$$J_1(s) \leq J'_1(s) + \max\{\|J_1 - J'_1\|, \|V_1 - V'_1\|\} \quad (7)$$

$$V_1(s) \leq V'_1(s) + \max\{\|J_1 - J'_1\|, \|V_1 - V'_1\|\} \quad (8)$$

from which we have:

$$\max\{J_1(s), V_1(s)\} \leq \max\{J'_1(s), V'_1(s)\} + \max\{\|J_1 - J'_1\|, \|V_1 - V'_1\|\}. \quad (9)$$

Exchanging (J_1, V_1) with (J'_1, V'_1) in (7) and (8) derives an inequality similar to (9), and combining it with (9) leads to the inequality:

$$|\max\{J_1(s), V_1(s)\} - \max\{J'_1(s), V'_1(s)\}| \leq \max\{\|J_1 - J'_1\|, \|V_1 - V'_1\|\} \quad (10)$$

for any $s \in S$. Since J_1, V_1, J'_1 and V'_1 are bounded, Borel measurable, so is $\max[J_1, V_1] - \max[J'_1, V'_1]$ by [46, Chapter 18.1], i.e., $\max[J_1, V_1] - \max[J'_1, V'_1] \in \mathbb{F}(S)$. Thus, since (10) holds for any $s \in S$:

$$\|\max[J_1, V_1] - \max[J'_1, V'_1]\| \leq \max\{\|J_1 - J'_1\|, \|V_1 - V'_1\|\}.$$

The second inequality of the lemma can be proved following the same steps for $J_2, V_2, J'_2, V'_2 \in \mathbb{F}(\Lambda_1)$. $\square$

Using the above operators and results, we are now in a position to prove the convergence of the Minimax-action-free PI algorithm.

Theorem 4 (Convergence guarantee). *If each of the four iterations of the Minimax-action-free PI algorithm (Algorithm 3) from lines 3 to 16 is performed infinitely often, then the sequence $(\gamma V_1^t)_{t \in \mathbb{N}}$ generated by the algorithm converges to V^* .*

Proof. We prove each component $G_{\sigma_1, \bar{\sigma}_2}$ satisfies a contraction property. Suppose that $J_1, V_1, J'_1, V'_1 \in \mathbb{F}(S)$ are B-PWC and $J_2, V_2, J'_2, V'_2 \in \mathbb{F}(\Lambda_1)$ are CON-PWL Borel measurable.

- For $M_{\sigma_1}^1$, since $M_{\sigma_1}^1(J_2, V_2) - M_{\sigma_1}^1(J'_2, V'_2) \in \mathbb{F}(S)$ by [46, Chapter 18.1]. By Definition 11, the sup-norm for $\mathbb{F}(S)$ and rearranging we have:

$$\begin{aligned} & \|M_{\sigma_1}^1(J_2, V_2) - M_{\sigma_1}^1(J'_2, V'_2)\| \\ &= \gamma^{-1} \sup_{s \in S} |\min\{J_2(s, \sigma_1(s)), V_2(s, \sigma_1(s))\} \\ &\quad - \min\{J'_2(s, \sigma_1(s)), V'_2(s, \sigma_1(s))\}| \\ &\leq \gamma^{-1} \sup_{(s, u_1) \in \Lambda_1} |\min\{J_2(s, u_1), V_2(s, u_1)\} - \min\{J'_2(s, u_1), V'_2(s, u_1)\}| \\ &\quad \text{since } \{(s, \sigma_1(s)) \mid s \in S\} \subseteq \Lambda_1 \\ &= \gamma^{-1} \|\min[J_2, V_2] - \min[J'_2, V'_2]\| \\ &\quad \text{since } \min[J_2, V_2] - \min[J'_2, V'_2] \in \mathbb{F}(\Lambda_1) \text{ using [46, Chapter 18.1]} \\ &\leq \gamma^{-1} \max\{\|J_2 - J'_2\|, \|V_2 - V'_2\|\} \quad \text{by Lemma 10} \\ &\leq \gamma^{-1} \max\{\|J_1 - J'_1\|, \|V_1 - V'_1\|, \|J_2 - J'_2\|, \|V_2 - V'_2\|\}. \quad (11) \end{aligned}$$

- For K^1 , since $K^1(J_2, V_2) - K^1(J'_2, V'_2) \in \mathbb{F}(S)$ by Definition 11 and the sup-norm for $\mathbb{F}(S)$:

$$\begin{aligned}
& \|K^1(J_2, V_2) - K^1(J'_2, V'_2)\| \\
&= \sup_{s \in S} \left| \max_{u_1 \in \mathbb{P}(A_1(s))} \gamma^{-1} \min\{J_2(s, u_1), V_2(s, u_1)\} \right. \\
&\quad \left. - \max_{u_1 \in \mathbb{P}(A_1(s))} \gamma^{-1} \min\{J'_2(s, u_1), V'_2(s, u_1)\} \right| \\
&\leq \gamma^{-1} \sup_{(s, u_1) \in \Lambda_1} \left| \min\{J_2(s, u_1), V_2(s, u_1)\} - \min\{J'_2(s, u_1), V'_2(s, u_1)\} \right| \\
&\quad \text{rearranging and since } \{(s, u_1) \mid u_1 \in \mathbb{P}(A_1(s))\} \subseteq \Lambda_1 \\
&\leq \gamma^{-1} \max\{\|J_1 - J'_1\|, \|V_1 - V'_1\|, \|J_2 - J'_2\|, \|V_2 - V'_2\|\} \quad (12)
\end{aligned}$$

where the final inequality follows from similar arguments used in (11).

- For $M_{\bar{\sigma}_2}^2$, since $M_{\bar{\sigma}_2}^2(J_1, V_1) - M_{\bar{\sigma}_2}^2(J'_1, V'_1) \in \mathbb{F}(\Lambda_1)$ by Definition 12 and the sup-norm for $\mathbb{F}(\Lambda_1)$ we have:

$$\begin{aligned}
& \|M_{\bar{\sigma}_2}^2(J_1, V_1) - M_{\bar{\sigma}_2}^2(J'_1, V'_1)\| \\
&= \sup_{(s, u_1) \in \Lambda_1} \left| \sum_{(a_1, a_2) \in A(s)} (Q(s, (a_1, a_2), \gamma \max[J_1, V_1]) \right. \\
&\quad \left. - Q(s, (a_1, a_2), \gamma \max[J'_1, V'_1])) u_1(a_1) \bar{\sigma}_2(a_2 \mid (s, u_1)) \right| \\
&= \sup_{(s, u_1) \in \Lambda_1} \left| \sum_{(a_1, a_2) \in A(s)} \gamma \beta \sum_{s' \in \Theta(s, (a_1, a_2))} \delta(s, (a_1, a_2))(s') \right. \\
&\quad \left. (\max\{J_1(s'), V_1(s')\} - \max\{J'_1(s'), V'_1(s')\}) u_1(a_1) \bar{\sigma}_2(a_2 \mid (s, u_1)) \right| \\
&\quad \text{rearranging, by Definition 10 and the sup-norm for } \mathbb{F}(\Lambda_1) \\
&\leq \gamma \beta \sup_{(s, u_1) \in \Lambda_1} \sum_{(a_1, a_2) \in A(s)} \sum_{s' \in \Theta(s, (a_1, a_2))} \delta(s, (a_1, a_2))(s') \\
&\quad \left| \max\{J_1(s'), V_1(s')\} - \max\{J'_1(s'), V'_1(s')\} \right| u_1(a_1) \bar{\sigma}_2(a_2 \mid (s, u_1)) \\
&\quad \text{rearranging and since } \delta, u_1 \text{ and } \bar{\sigma}_2 \text{ are non-negative} \\
&\leq \gamma \beta \sup_{(s, u_1) \in \Lambda_1} \sum_{(a_1, a_2) \in A(s)} \sum_{s' \in \Theta(s, (a_1, a_2))} \delta(s, (a_1, a_2))(s') \\
&\quad \sup_{s'' \in S} \left| \max\{J_1(s''), V_1(s'')\} - \max\{J'_1(s''), V'_1(s'')\} \right| u_1(a_1) \bar{\sigma}_2(a_2 \mid (s, u_1)) \\
&\quad \text{since } f(s') \leq \sup_{s'' \in S} f(s'') \text{ for any } f \in \mathbb{F}(S)
\end{aligned}$$

$$\begin{aligned}
&= \gamma\beta \sup_{s'' \in S} |\max\{J_1(s''), V_1(s'')\} - \max\{J'_1(s''), V'_1(s'')\}| \\
&\quad \sup_{(s, u_1) \in \Lambda_1} \sum_{(a_1, a_2) \in A(s)} \sum_{s' \in \Theta(s, (a_1, a_2))} \delta(s, (a_1, a_2))(s') u_1(a_1) \bar{\sigma}_2(a_2 \mid (s, u_1)) \\
&\hspace{25em} \text{rearranging} \\
&= \gamma\beta \sup_{s'' \in S} |\max\{J_1(s''), V_1(s'')\} - \max\{J'_1(s''), V'_1(s'')\}| \\
&\quad \text{since } \delta \in \mathbb{P}(S \times A), u_1 \in \mathbb{P}(A_1) \text{ and } \bar{\sigma}_2 \in \mathbb{P}(A_2 \mid \Lambda_1) \\
&= \gamma\beta \|\max[J_1, V_1] - \max[J'_1, V'_1]\| \\
&\quad \text{since } \max[J_1, V_1] - \max[J'_1, V'_1] \in \mathbb{F}(S) \\
&\leq \gamma\beta \max\{\|J_1 - J'_1\|, \|V_1 - V'_1\|\} \hspace{10em} \text{by Lemma 10} \\
&\leq \gamma\beta \max\{\|J_1 - J'_1\|, \|V_1 - V'_1\|, \|J_2 - J'_2\|, \|V_2 - V'_2\|\}. \tag{13}
\end{aligned}$$

- For K^2 , since $K^2(J_1, V_1) - K^2(J'_1, V'_1) \in \mathbb{F}(\Lambda_1)$, by the sup-norm for $\mathbb{F}(\Lambda_1)$:

$$\begin{aligned}
&\|K^2(J_1, V_1) - K^2(J'_1, V'_1)\| \\
&= \sup_{(s, u_1) \in \Lambda_1} \left| \min_{u_2 \in \mathbb{P}(A_2(s))} \sum_{(a_1, a_2) \in A(s)} \gamma\beta \sum_{s' \in \Theta(s, (a_1, a_2))} \delta(s, (a_1, a_2))(s') \right. \\
&\quad \left. (\max\{J_1(s'), V_1(s')\} - \max\{J'_1(s'), V'_1(s')\}) u_1(a_1) u_2(a_2) \right| \\
&\leq \gamma\beta \sup_{(s, u_1) \in \Lambda_1} \min_{u_2 \in \mathbb{P}(A_2(s))} \sum_{(a_1, a_2) \in A(s)} \sum_{s' \in \Theta(s, (a_1, a_2))} \delta(s, (a_1, a_2))(s') \\
&\quad |\max\{J_1(s'), V_1(s')\} - \max\{J'_1(s'), V'_1(s')\}| u_1(a_1) u_2(a_2) \hspace{1em} \text{rearranging} \\
&\leq \gamma\beta \max\{\|J_1 - J'_1\|, \|V_1 - V'_1\|, \|J_2 - J'_2\|, \|V_2 - V'_2\|\} \tag{14}
\end{aligned}$$

where the final inequality follows from similar arguments used in (13).

Next we prove that $G_{\sigma_1, \bar{\sigma}_2}$ is a contraction mapping using the above inequalities. More precisely, by definition, see (2), we have:

$$\begin{aligned}
&\|G_{\sigma_1, \bar{\sigma}_2}(J_1, V_1, J_2, V_2) - G_{\sigma_1, \bar{\sigma}_2}(J'_1, V'_1, J'_2, V'_2)\| \\
&= \|(M_{\sigma_1}^1(J_2, V_2) - M_{\sigma_1}^1(J'_2, V'_2), K^1(J_2, V_2) - K^1(J'_2, V'_2), \\
&\quad M_{\bar{\sigma}_2}^2(J_1, V_1) - M_{\bar{\sigma}_2}^2(J'_1, V'_1), K^2(J_1, V_1) - K^2(J'_1, V'_1))\| \\
&= \max\{\|M_{\sigma_1}^1(J_2, V_2) - M_{\sigma_1}^1(J'_2, V'_2)\|, \|K^1(J_2, V_2) - K^1(J'_2, V'_2)\|, \\
&\quad \|M_{\bar{\sigma}_2}^2(J_1, V_1) - M_{\bar{\sigma}_2}^2(J'_1, V'_1)\|, \|K^2(J_1, V_1) - K^2(J'_1, V'_1)\|\},
\end{aligned}$$

$$\begin{aligned} & \|M_{\bar{\sigma}_2}^2(J_1, V_1) - M_{\bar{\sigma}_2}^2(J'_1, V'_1)\|, \|K^2(J_1, V_1) - K^2(J'_1, V'_1)\| \} \quad \text{rearranging} \\ & \leq \max\{\gamma^{-1}, \gamma\beta\} \max\{\|J_1 - J'_1\|, \|V_1 - V'_1\|, \|J_2 - J'_2\|, \|V_2 - V'_2\|\} \end{aligned}$$

where the final inequality follows from (11), (13), (12) and (14).

Therefore, since $\max\{\gamma^{-1}, \gamma\beta\} < 1$ and assuming σ_1 is PWC and $\bar{\sigma}_2$ is CON-PWC, we have that $G_{\sigma_1, \bar{\sigma}_2}$ is a contraction mapping for $(\sigma_1, \bar{\sigma}_2) \in \Sigma_1 \times \bar{\Sigma}_2$. Now since $\mathbb{F}(S) \times \mathbb{F}(S) \times \mathbb{F}(\Lambda_1) \times \mathbb{F}(\Lambda_1)$ is a complete metric space with respect to the sup norm, we conclude that $G_{\sigma_1, \bar{\sigma}_2}$ has a unique fixed point $(J_1^*, V_1^*, J_2^*, V_2^*)$. In view of (3)–(6), this fixed point satisfies for each $(s, u_1) \in \Lambda_1$:

$$J_1^*(s) = \gamma^{-1} \min\{J_2^*(s, \sigma_1(s)), V_2^*(s, \sigma_1(s))\} \quad (15)$$

$$V_1^*(s) = \gamma^{-1} \max_{u_1 \in \mathbb{P}(A_1(s))} \min\{J_2^*(s, u_1), V_2^*(s, u_1)\} \quad (16)$$

$$J_2^*(s, u_1) = \sum_{(a_1, a_2) \in A(s)} Q(s, (a_1, a_2), \gamma \max[J_1^*, V_1^*]) u_1(a_1) \bar{\sigma}_2(a_2 \mid (s, u_1)) \quad (17)$$

$$V_2^*(s, u_1) = \min_{u_2 \in \mathbb{P}(A_2(s))} \sum_{(a_1, a_2) \in A(s)} Q(s, (a_1, a_2), \gamma \max[J_1^*, V_1^*]) u_1(a_1) u_2(a_2). \quad (18)$$

By combining (15)–(18), we have for each $(s, u_1) \in \Lambda_1$:

$$J_1^*(s) \leq V_1^*(s) \text{ and } J_2^*(s, u_1) \geq V_2^*(s, u_1)$$

from which (16) and (18) can be simplified to:

$$\begin{aligned} V_1^*(s) &= \max_{u_1 \in \mathbb{P}(A_1(s))} \gamma^{-1} V_2^*(s, u_1) \\ V_2^*(s, u_1) &= \min_{u_2 \in \mathbb{P}(A_2(s))} \sum_{(a_1, a_2) \in A(s)} Q(s, (a_1, a_2), \gamma V_1^*) u_1(a_1) u_2(a_2) \end{aligned}$$

implying that $\gamma V_1^*(s)$ equals:

$$\max_{u_1 \in \mathbb{P}(A_1(s))} \min_{u_2 \in \mathbb{P}(A_2(s))} \sum_{(a_1, a_2) \in A(s)} Q(s, (a_1, a_2), \gamma V_1^*) u_1(a_1) u_2(a_2) = [T(\gamma V_1^*)].$$

Thus, we have $\gamma V_1^* = V^*$, which completes the proof. $\square$

Strategy computation. Next, introducing a criterion for selecting the minimax solution over a region, we compute the strategies for the agents based on the function returned by the Minimax-action-free PI algorithm.

Definition 17 (CON-3 solution). *Let $f \in \mathbb{F}(\Lambda_{12})$. If there exists a BFCP Φ of S where, for each $\phi \in \Phi$: $A(s) = A(s')$ for $s, s' \in \phi$ there exists a pair*

of probability measures $u_1^\phi \in \mathbb{P}(A_1(s))$ and $u_2^\phi \in \mathbb{P}(A_2(s))$ for $s \in \phi$ such that $f(s, u_1^\phi, u_2^\phi) = \max_{u_1 \in \mathbb{P}(A_1(s))} \min_{u_2 \in \mathbb{P}(A_2(s))} f(s, u_1, u_2)$ for $s \in \phi$, and $\sigma_1 \in \Sigma_1$, $\sigma_2 \in \Sigma_2$ are such that $\sigma_1(s) = u_1^\phi$ and $\sigma_2(s) = u_2^\phi$ for $s \in \phi$, then (σ_1, σ_2) is a constant-3 (CON-3) solution of f over ϕ .

Lemma 11 (PWC strategies). *If $V = \gamma V_1^t$, where V_1^t is from iteration $t \in \mathbb{N}$ of the Minimax-action-free PI algorithm, and $(\sigma_1, \sigma_2) \in \Sigma$ achieves the maximum and the minimum in Definition 10 for V and all $s \in S$ via a CON-3 solution, then σ_1 and σ_2 are PWC stochastic kernels.*

Proof. By Theorems 3 and 4, V is B-PWC. For any $\alpha \in A$, the function $Q(\cdot, \alpha, V) : S \rightarrow \mathbb{R}$ is B-PWC by Theorem 2. Let Φ_Q be a BFCP of S such that $Q(\cdot, \alpha, V)$ is constant on each region of Φ_Q for $\alpha \in A$, and Φ_A be a BFCP of S such that $A(s)$ is constant on each region of Φ_A . Then, for $u_1 \in \mathbb{P}(A_1(s))$ and $u_2 \in \mathbb{P}(A_2(s))$, the function $Q'(\cdot, u_1, u_2) : S \rightarrow \mathbb{R}$, where:

$$Q'(s, u_1, u_2) = \sum_{(a_1, a_2) \in A(s)} Q(s, (a_1, a_2), V) u_1(a_1) u_2(a_2)$$

for $s \in S$, is constant in each region of $\Phi_Q + \Phi_A$. Therefore, there exists a CON-3 solution (σ_1, σ_2) of $Q'(s, u_1, u_2)$ and, since $\Phi_Q + \Phi_A$ is a BFCP, the result follows. $\square$

8. Conclusions

We have proposed a novel modelling formalism called neuro-symbolic concurrent stochastic games (NS-CSGs) for representing probabilistic finite-state agents with neural perception mechanisms interacting in a shared, continuous-state environment. NS-CSGs have the advantage of allowing for the perception of a complex environment to be synthesised from data and implemented via NNs, while the safety-critical decision-making module is symbolic, explainable and knowledge-based. We constrain the interface of the neural perception so that agents can reason symbolically using percepts, representing (exactly) learnt subregions of the continuous environment.

For zero-sum discounted cumulative reward problems, we proved the existence and measurability of the value function of NS-CSGs under Borel measurability and piecewise constant restrictions. We then presented the first computationally practical B-PWC VI and Minimax-action-free PI algorithms with finite representations for computing the values and optimal

strategies of NS-CSGs, assuming a fully observable setting, by proposing B-PWC, CON-PWL and CON-PWC functions.

The B-PWC VI algorithm is, at the region level, the same as VI for finite state spaces, but involves, at each iteration, a division of the uncountable state space into a possibly refined finite set of regions (i.e., a BFCP). This is because taking the same action in two distinct states of some region in the current BFCP can yield states that need not be observationally equivalent. The Minimax-action-free PI algorithm requires multiple divisions of the uncountable state space into BFCPs at each iteration. We resolve a number of technical issues to apply [18], which ensures convergence, and, by not requiring the solution of normal-form games or MDPs at each iteration, reduces computational complexity. However, implementation of the Minimax-action-free PI algorithm is more challenging, requiring a distributed, asynchronous framework. We illustrated our approach by modelling a dynamic vehicle parking problem as an NS-CSG and synthesising approximately optimal values and strategies using B-PWC VI.

Future work will involve improving efficiency of the algorithms, generalising to more complex classes of neural perception mechanisms by exploiting preimage approximations [54], and moving to equilibria-based (nonzero-sum) properties, where initial progress has been made in a simpler setting of finite unfolding of our NS-CSG model [40]. We are also planning to extend the approach to two-sided partially observable neuro-symbolic POSGs, based on a recently proposed one-sided variant, which exploits a finite representation that generalises α -vectors to approximate value computation [19], and for which *online* strategy synthesis methods have been developed [55].

Acknowledgements. This project was funded by the ERC under the European Union’s Horizon 2020 research and innovation programme ([FUN2MODEL](#), grant agreement No. 834115).

References

- [1] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of Go with deep neural networks and tree search, *Nature* 529 (7587) (2016) 484–489.
- [2] S. Shalev-Shwartz, S. Shammah, A. Shashua, Safe, multi-agent, reinforcement learning for autonomous driving, [arXiv:1610.03295](#) (2016).

- [3] J. Gupta, M. Egorov, M. Kochenderfer, Cooperative multi-agent control using deep reinforcement learning, in: Proc. 16th Int. Conf. Autonomous Agents and Multiagent Systems (AAMAS'17), Springer, 2017, pp. 66–83.
- [4] L. S. Shapley, Stochastic games, PNAS 39 (10) (1953) 1095–1100.
- [5] R. Yan, X. Duan, Z. Shi, Y. Zhong, J. Marden, F. Bullo, Policy evaluation and seeking for multi-agent reinforcement learning via best response, IEEE Trans. Automat. Contr. 67 (4) (2022) 1898–1913.
- [6] M. Kwiatkowska, G. Norman, D. Parker, G. Santos, Automatic verification of concurrent stochastic systems, Form. Methods Syst. Des. (2021) 1–63.
- [7] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, I. Mordatch, Multi-agent actor-critic for mixed cooperative-competitive environments, in: Proc. 31st Int. Conf. Neural Information Processing Systems (NIPS'17), Curran Associates Inc., 2017, pp. 6382–6393.
- [8] M. E. Akintunde, E. Botoeva, P. Kouvaros, A. Lomuscio, Verifying strategic abilities of neural-symbolic multi-agent systems, in: Proc. 17th Int. Conf. Principles of Knowledge Representation and Reasoning (KR'20), IJCAI Organization, 2020, pp. 22–32.
- [9] L. D. Raedt, S. Dumancic, R. Manhaeve, G. Marra, From statistical relational to neural-symbolic artificial intelligence, in: Proc. 29th Int. Conf. Artificial Intelligence (IJCAI'20), IJCAI Organization, 2020, pp. 4943–4950.
- [10] G. Anderson, A. Verma, I. Dillig, S. Chaudhuri, Neurosymbolic reinforcement learning with formally verified exploration, in: Proc. 34th Int. Conf. Advances in Neural Information Processing Systems (NeurIPS'20), Curran Associates, Inc., 2020, pp. 6172–6183.
- [11] J. Van Der Wal, Discounted Markov games: Generalized policy iteration method, J. Optim. Theory Appl. 25 (1) (1978) 125–138.
- [12] B. Tolwinski, Newton-type methods for stochastic games, in: Differential games and applications, Springer, 1989, pp. 128–144.

- [13] J. Filar, K. Vrieze, Competitive Markov decision processes, Springer, 1997.
- [14] J. Perolat, B. Scherrer, B. Piot, O. Pietquin, Approximate dynamic programming for two-player zero-sum Markov games, in: Proc. 32nd Int. Conf. Machine Learning (ICML’15), Vol. 37, PMLR, 2015, pp. 1321–1329.
- [15] D. Bertsekas, Abstract dynamic programming, Athena Scientific, 2018.
- [16] M. Kwiatkowska, G. Norman, D. Parker, G. Santos, R. Yan, Probabilistic Model Checking for Strategic Equilibria-Based Decision Making: Advances and Challenges, in: 47th Int. Symp. Mathematical Foundations of Computer Science (MFCS’22), Vol. 241 of LIPIcs, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, pp. 4:1–4:22.
- [17] P. Kumar, T.-H. Shiau, Existence of value and randomized strategies in zero-sum discrete-time stochastic dynamic games, SIAM. J. Control. Optim. 19 (5) (1981) 617–634.
- [18] D. Bertsekas, Distributed asynchronous policy iteration for sequential zero-sum games and minimax control, [arXiv:2107.10406](https://arxiv.org/abs/2107.10406) (2021).
- [19] R. Yan, G. Santos, G. Norman, D. Parker, M. Kwiatkowska, Partially observable stochastic games with neural perception mechanisms, in: Proc. 26th Int. Symp. Formal Methods (FM’24), LNCS, Springer, 2024.
- [20] N. Brown, A. Bakhtin, A. Lerer, Q. Gong, Combining deep reinforcement learning and search for imperfect-information games, in: Proc. 34th Int. Conf. Advances in Neural Information Processing Systems (NeurIPS’20), Curran Associates, Inc., 2020, pp. 17057–17069.
- [21] V. Kovařík, M. Schmid, N. Burch, M. Bowling, V. Lisý, Rethinking formal models of partially observable multiagent decision making, Artif. Intell. 303 (2022) 103645.
- [22] A. Maitra, T. Parthasarathy, On stochastic games, J. Optim. Theory Appl. 5 (4) (1970) 289–300.
- [23] A. Nowak, Optimal strategies in a class of zero-sum ergodic stochastic games, Math. Methods. Oper. Res. 50 (3) (1999) 399–419.

- [24] A. Nowak, Universally measurable strategies in zero-sum stochastic games, *Ann. Probab.* 13 (1) (1985) 269–287.
- [25] O. Hernández-Lerma, J. Lasserre, Zero-sum stochastic games in borel spaces: average payoff criteria, *SIAM. J. Control. Optim.* 39 (5) (2000) 1520–1539.
- [26] A. Hoffman, R. Karp, On non-terminating stochastic games, *Manage Sci.* 12 (5) (1966) 359–370.
- [27] M. A. Pollatschek, B. Avi-Itzhak, Algorithms for stochastic games with geometrical interpretation, *Manage. Sci.* 15 (7) (1969) 399–415.
- [28] J. Křetínskỳ, E. Ramneantu, A. Slivinskiy, M. Weininger, Comparison of algorithms for simple stochastic games, *Inf. Comput.* 289 (2022) 104885.
- [29] J. Eisentraut, E. Kelmendi, J. Křetínskỳ, M. Weininger, Value iteration for simple stochastic games: Stopping criterion and learning algorithm, *Inf. Comput.* 285 (2022) 104886.
- [30] H. Yu, D. Bertsekas, A mixed value and policy iteration method for stochastic control with universally measurable policies, *Math. Oper. Res.* 40 (4) (2015) 926–968.
- [31] H. Yu, On convergence of value iteration for a class of total cost Markov decision processes, *SIAM. J. Control. Optim.* 53 (4) (2015) 1982–2016.
- [32] I. Hogeboom-Burr, S. Yuksel, Comparison of information structures for zero-sum games and a partial converse to Blackwell ordering in standard borel spaces, *SIAM. J. Control. Optim.* 59 (3) (2021) 1781–1803.
- [33] A. Basu, L. Stettner, Zero-sum Markov games with impulse controls, *SIAM. J. Control. Optim.* 58 (1) (2020) 580–604.
- [34] A. Cosso, Stochastic differential games involving impulse controls and double-obstacle quasi-variational inequalities, *SIAM. J. Control. Optim.* 51 (3) (2013) 2102–2131.
- [35] K. Chatterjee, R. Ibsen-Jensen, Qualitative analysis of concurrent mean-payoff games, *Inf. Comput.* 242 (2015) 2–24.

- [36] N. Basset, M. Kwiatkowska, C. Wiltsche, Compositional strategy synthesis for stochastic games with multiple objectives, *Inf. Comput.* 261 (2018) 536–587.
- [37] T. Brázdil, V. Forejt, J. Krčál, J. Křetínský, A. Kučera, Continuous-time stochastic games with time-bounded reachability, *Inf. Comput.* 224 (2013) 46–70.
- [38] J. Fearnley, M. N. Rabe, S. Schewe, L. Zhang, Efficient approximation of optimal control for continuous-time Markov games, *Inf. Comput.* 247 (2016) 106–129.
- [39] R. Yan, G. Santos, G. Norman, D. Parker, M. Kwiatkowska, Strategy synthesis for zero-sum neuro-symbolic concurrent stochastic games, [arXiv:2202.06255](https://arxiv.org/abs/2202.06255) (2022).
- [40] R. Yan, G. Santos, X. Duan, D. Parker, M. Kwiatkowska, Finite-horizon equilibria for neuro-symbolic concurrent stochastic games, in: *Proc. 38th Conf. Uncertainty in Artificial Intelligence (UAI’22)*, AUA Press, 2022, pp. 2170–2180.
- [41] S. Sharma, S. Sharma, A. Athaiya, Activation functions in neural networks, *IJEAST* 6 (12) (2020) 310–316.
- [42] K. Matoba, F. Fleuret, Computing preimages of deep neural networks with applications to safety, [openreview.netforum?id=FN7_BUOG78e](https://openreview.net/forum?id=FN7_BUOG78e) (2020).
- [43] J. Kemeny, J. Snell, A. Knapp, *Denumerable Markov Chains*, Springer, 1976.
- [44] K. D. Julian, M. J. Kochenderfer, A reachability method for verifying dynamical systems with deep neural network controllers, [arXiv.1903.00520](https://arxiv.org/abs/1903.00520) (2019).
- [45] D. Ayala, O. Wolfson, B. Xu, B. Dasgupta, J. Lin, Parking slot assignment games, in: *Proc. 19th ACM SIGSPATIAL Int. Conf. Advances in Geographic Information Systems (GIS’11)*, ACM, 2011, p. 299–308.
- [46] H. L. Royden, P. Fitzpatrick, *Real analysis* (fourth edition), Macmillan New York, 2010.

- [47] K. Parthasarathy, Probability measures on metric spaces, AMS., 1967.
- [48] J. Reif, Universal games of incomplete information, in: Proc. 11th ACM Symp. Theory of Computing (STOC'79), ACM, 1979, pp. 288–308.
- [49] J. Reif, The complexity of two-player games of incomplete information, J. Comput. Syst. Sci. 29 (1984) 274–301.
- [50] J. von Neumann, O. Morgenstern, H. Kuhn, A. Rubinstein, Theory of Games and Economic Behavior, Princeton University Press, 1944.
- [51] P. Virtanen, R. Gommers, T. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. van der Walt, M. Brett, J. Wilson, K. Millman, N. Mayorov, A. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, Í. Polat, Y. Feng, E. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. Quintero, C. Harris, A. Archibald, A. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272.
- [52] D. Bertsekas, Abstract dynamic programming, Athena Scientific, 2022.
- [53] D. Bertsekas, H. Yu, Q-learning and enhanced policy iteration in discounted dynamic programming, Math. Oper. Res. 37 (1) (2012) 66–94.
- [54] X. Zhang, B. Wang, M. Kwiatkowska, Provable preimage under-approximation for neural networks, in: Proc. 30th Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'24), Vol. 14572 of LNCS, Springer, 2024, pp. 3–23.
- [55] R. Yan, G. Santos, G. Norman, D. Parker, M. Kwiatkowska, HSVI-based online minimax strategies for partially observable stochastic games with neural perception mechanisms, in: Proc. 6th Learning for Dynamics & Control Conference L4DC'24, Vol. 424 of Proceedings of Machine Learning Research, PMLR, 2024, pp. 80–91.