# Partially Observable Stochastic Games with Neural Perception Mechanisms

Rui Yan[1] ✉, Gabriel Santos[1], Gethin Norman[1,2], David Parker[1], and Marta Kwiatkowska[1]

[1] University of Oxford, Oxford, OX1 2JD, UK
{rui.yan,gabriel.santos,david.parker,marta.kwiatkowska}@cs.ox.ac.uk
[2] University of Glasgow, Glasgow, G12 8QQ, UK
gethin.norman@glasgow.ac.uk

**Abstract.** Stochastic games are a well established model for multi-agent sequential decision making under uncertainty. In practical applications, though, agents often have only partial observability of their environment. Furthermore, agents increasingly perceive their environment using data-driven approaches such as neural networks trained on continuous data. We propose the model of neuro-symbolic partially-observable stochastic games (NS-POSGs), a variant of continuous-space concurrent stochastic games that explicitly incorporates neural perception mechanisms. We focus on a one-sided setting with a partially-informed agent using discrete, data-driven observations and another, fully-informed agent. We present a new method, called one-sided NS-HSVI, for approximate solution of one-sided NS-POSGs, which exploits the piecewise constant structure of the model. Using neural network pre-image analysis to construct finite polyhedral representations and particle-based representations for beliefs, we implement our approach and illustrate its practical applicability to the analysis of pedestrian-vehicle and pursuit-evasion scenarios.

## 1 Introduction

Strategic reasoning is essential to ensure stable multi-agent coordination in complex environments, e.g., autonomous driving or multi-robot planning. *Partially-observable stochastic games* (POSGs) are a natural model for settings involving multiple agents, uncertainty and partial information. They allow the synthesis of optimal (or near-optimal) strategies and equilibria that guarantee expected outcomes, even in adversarial scenarios. But POSGs also present significant challenges: key problems are undecidable, already for the single-agent case of partially observable Markov decision processes (POMDPs) [24], and practical algorithms for finding optimal values and strategies are lacking.

Computational tractability can be improved using *one-sided POSGs*, a subclass of two-agent, zero-sum POSGs where only one agent has partial information while the other agent is assumed to have full knowledge of the state [40,41]. This can be useful when making worst-case assumptions about one agent, such as

in an adversarial setting (e.g., an attacker-defender scenario) or a safety-critical domain (e.g., a pedestrian in an autonomous driving application).

From a computational perspective, one-sided POSGs avoid the need for nested beliefs [39], i.e., reasoning about beliefs not only over states but also over opponents' beliefs. This is because the fully-informed agent can reconstruct beliefs from observation histories. Recent advances [19] have led to the first practical variant of heuristic search value iteration (HSVI) [31] for computing approximately optimal values and strategies in (finite) one-sided POSGs.

However, in many realistic autonomous coordination scenarios, agents perceive *continuous* environments using *data-driven* observation functions, typically implemented as neural networks (NNs). Examples include autonomous vehicles using NNs to perform object recognition or to estimate pedestrian intention, and NN-enabled vision in an airborne pursuit-evasion scenario.

In this paper, we introduce *one-sided neuro-symbolic POSGs (NS-POSGs)*, a variant of continuous-space POSGs that explicitly incorporates neural perception mechanisms. We assume one partially-informed agent with a (finite-valued) observation function synthesised in a data-driven fashion, and a second agent with full observation of the (continuous) state. Continuous-space models with neural perception mechanisms have already been developed, but are limited to the simpler cases of POMDPs [36] and (fully-observable) stochastic games [33]. Our model provides the ability to reason about an agent with a realistic perception mechanism *and* operating in an adversarial or worst-case setting.

Solving continuous-space models, even approximately, is computationally challenging. One approach is to discretise and then use techniques for finite-state models (e.g., [19] in our case). But this can yield exponential growth of the state space, depending on the granularity and time-horizon used. Furthermore, decision boundaries for data-driven perception are typically irregular and can be misaligned with gridding schemes for discretisation, limiting precision.

An alternative is to exploit structure in the underlying model and work directly with the continuous-state model. For example, classic dynamic programming approaches to solving MDPs can be lifted to continuous-state variants [12]: a piecewise constant representation of the value function is computed, based on a partition of the state space created dynamically during solution. It is demonstrated that this approach can outperform discretisation and that it can also be generalised to solving POMDPs. We can adapt this approach to models with neural perception mechanisms [36], exploiting the fact that ReLU NN classifiers induce a finite decomposition of the continuous environment into polyhedra.

**Contributions.** The contributions of this paper are as follows. We first define the model of one-sided NS-POSGs and motivate it via an autonomous driving scenario based on a ReLU NN classifier for pedestrian intention learnt from public datasets [28]. We then prove that the (discounted reward) value function for NS-POSGs is continuous and convex, and is a fixed point of a minimax operator. Based on mild assumptions about the model, we give a piecewise linear and convex representation of the value function, which admits a finite polyhedral representation and which is closed with respect to the minimax operator.

In order to provide a feasible approach to approximating values of NS-POSGs, we present a variant of HSVI, which is a popular anytime algorithm for POMDPs that iteratively computes lower and upper bounds on values. We build on ideas from HSVI for finite one-sided POSGs [19] (but there are multiple challenges when moving to a continuous state space and NNs) and for POMDPs with neural perception mechanisms [36] (but, for us, the move to games brings a number of complications); see Section 6 for a detailed discussion.

We implement our one-sided NS-HSVI algorithm using the popular particle-based representation for beliefs and employing NN pre-image computation [25] to construct an initial finite polyhedral representation of perception functions. We apply this to the pedestrian-vehicle interaction scenario and a pursuit-evasion game inspired by mobile robotics applications, demonstrating the ability to synthesise agent strategies for models with complex perception functions, and to explore trade-offs when using perception mechanisms of varying precision.

**Related work.** Solving POSGs is largely intractable. Methods based on exact dynamic programming [17] and approximations [23,11] exist but have high computational cost. Further approaches exist for *zero-sum* POSGs, including conversion to extensive-form games [3], counterfactual regret minimisation [42,21,22] and methods based on reinforcement learning and search [5,26]. In [9], an HSVI-like finite-horizon solver that provably converges to an $\varepsilon$-optimal solution is proposed; [32] provides convexity and concavity results but no algorithmic solution.

Methods exist for *one-sided* POSGs: a space partition approach when actions are public [40], a point-based approximate algorithm when observations are continuous [41] and projection to POMDPs based on factored representations [7]. But these are all restricted to *finite-state* games. Closer to our work, but still for finite models, is [19], which proposes an HSVI method for POSGs.

For the *continuous-state* but *single-agent* (POMDP) setting, point-based value iteration [27,6,38] and discrete space approximation [4] can be used; the former also uses $\alpha$-functions but works with (approximate) Gaussian mixtures or beta-densities, whereas we exploit structure, similarly to [12]. As discussed above, in earlier work, we proposed models and techniques for extending several simpler probabilistic models with neural perception mechanisms [36,34,33]. Recent work [37] builds on the one-sided NS-POSG model proposed in this paper, but focuses instead on *online* methods for strategy synthesis.

## 2 Background

**POSGs.** The semantics of our models are continuous-state *partially observable concurrent stochastic games* (POSGs) [21,5,18]. Letting $\mathbb{P}(X)$ denote the space of probability measures on a Borel space $X$, POSGs are defined as follows.

A two-player POSG is a tuple $\mathsf{G} = (N, S, A, \delta, \mathcal{O}, Z)$, where: $N = \{1, 2\}$ is a set of two agents; $S$ a Borel measurable set of states; $A \triangleq A_1 \times A_2$ a finite set of joint actions where $A_i$ are actions of agent $i$; $\delta : (S \times A) \to \mathbb{P}(S)$ a probabilistic transition function; $\mathcal{O} \triangleq \mathcal{O}_1 \times \mathcal{O}_2$ a finite set of joint observations where $\mathcal{O}_i$ are observations of agent $i$; and $Z : (S \times A \times S) \to \mathcal{O}$ an observation function.

In a state $s$ of a POSG $\mathsf{G}$, each agent $i$ selects an action $a_i$ from $A_i$. The probability to move to a state $s'$ is $\delta(s,(a_1,a_2))(s')$, and the subsequent observation is $Z(s,(a_1,a_2),s')=(o_1,o_2)$, where agent $i$ can only observe $o_i$. A *history* of $\mathsf{G}$ is a sequence of states and joint actions $\pi=(s^0,a^0,s^1,\ldots,a^{t-1},s^t)$ such that $\delta(s^k,a^k)(s^{k+1})>0$ for each $k$. For a history $\pi$, we denote by $\pi(k)$ the $(k{+}1)$th state, and $\pi[k]$ the $(k{+}1)$th action. A (local) *action-observation history (AOH)* is the view of a history $\pi$ from agent $i$'s perspective: $\pi_i=(o_i^0,a_i^0,o_i^1,\ldots,a_i^{t-1},o_i^t)$. If an agent has full information about the state, then we assume the agent is also informed of the history of joint actions. Let $FPaths_{\mathsf{G}}$ and $FPaths_{\mathsf{G},i}$ denote the sets of finite histories of $\mathsf{G}$ and AOHs of agent $i$, respectively.

A (behaviour) *strategy* of agent $i$ is a mapping $\sigma_i : FPaths_{\mathsf{G},i} \to \mathbb{P}(A_i)$. We denote by $\Sigma_i$ the set of strategies of agent $i$. A *profile* $\sigma=(\sigma_1,\sigma_2)$ is a pair of strategies for each agent and we denote by $\Sigma=\Sigma_1\times\Sigma_2$ the set of profiles.

**Objectives.** Agents 1 and 2 maximise and minimise, respectively, the expected value of the *discounted reward* $Y(\pi)=\sum_{k=0}^{\infty}\beta^k r(\pi(k),\pi[k])$, where $\pi$ is an infinite history, $r:(S\times A)\to\mathbb{R}$ a reward structure and $\beta\in(0,1)$. The expected value of $Y$ starting from state distribution $b$ under profile $\sigma$ is denoted $\mathbb{E}_b^{\sigma}[Y]$.

**Values and minimax strategies.** If $V^\star(b)\triangleq\sup_{\sigma_1\in\Sigma_1}\inf_{\sigma_2\in\Sigma_2}\mathbb{E}_b^{\sigma_1,\sigma_2}[Y]=\inf_{\sigma_2\in\Sigma_2}\sup_{\sigma_1\in\Sigma_1}\mathbb{E}_b^{\sigma_1,\sigma_2}[Y]$ for all $b\in\mathbb{P}(S)$, then $V^\star$ is called the *value* of $\mathsf{G}$. A profile $\sigma^\star=(\sigma_1^\star,\sigma_2^\star)$ is a *minimax strategy profile* if, for any $b\in\mathbb{P}(S)$, $\mathbb{E}_b^{\sigma_1^\star,\sigma_2}[Y]\geq\mathbb{E}_b^{\sigma_1^\star,\sigma_2^\star}[Y]\geq\mathbb{E}_b^{\sigma_1,\sigma_2^\star}[Y]$ for all $\sigma_1\in\Sigma_1$ and $\sigma_2\in\Sigma_2$.

## 3 One-Sided Neuro-Symbolic POSGs

We now introduce our model, aimed at commonly deployed multi-agent scenarios with data-driven perception, necessitating the use of continuous environments.

**One-sided NS-POSGs.** A *one-sided neuro-symbolic POSG (NS-POSG)* comprises a *partially informed, neuro-symbolic* agent and a *fully informed* agent in a continuous-state environment. The first agent has a finite set of local states, and is endowed with a data-driven perception mechanism, through which (and only through which) it makes finite-valued observations of the environment's state, stored locally as *percepts*. The second agent can directly observe both the local state and percept of the first agent, and the state of the environment.

**Definition 1 (NS-POSG)** *A one-sided NS-POSG* $\mathsf{C}$ *comprises agents* $\mathsf{Ag}_1=(S_1,A_1,obs_1,\delta_1)$ *and* $\mathsf{Ag}_2{=}(A_2)$*, and environment* $E{=}(S_E,\delta_E)$*, where:*

- $S_1=Loc_1\times Per_1$ *is a set of states for* $\mathsf{Ag}_1$*, where* $Loc_1$ *and* $Per_1$ *are finite sets of local states and percepts, respectively;*
- $S_E\subseteq\mathbb{R}^e$ *is a closed set of continuous environment states;*
- $A_i$ *is a finite set of actions for* $\mathsf{Ag}_i$ *and* $A\triangleq A_1\times A_2$ *is a set of joint actions;*
- $obs_1:(Loc_1\times S_E)\to Per_1$ *is* $\mathsf{Ag}_1$*'s perception function;*
- $\delta_1:(S_1\times A)\to\mathbb{P}(Loc_1)$ *is* $\mathsf{Ag}_1$*'s local probabilistic transition function;*
- $\delta_E:(Loc_1\times S_E\times A)\to\mathbb{P}(S_E)$ *is a finitely-branching probabilistic transition function for the environment.*

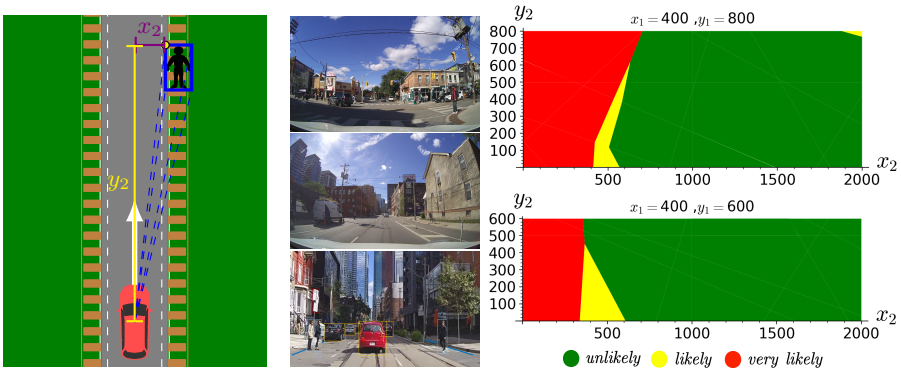



Fig. 1: Pedestrian-vehicle example. Left: Positions of two agents. Middle: Sample images from the PIE dataset [28]. Right: Slices of learnt perception function, where $(x_1, y_1), (x_2, y_2)$ are two successive (relative) positions of the pedestrian.

One-sided NS-POSGs are a subclass of two-agent, hybrid-state POSGs with discrete observations ($S_1$) and actions for $\mathsf{Ag}_1$, and continuous observations ($S_1 \times S_E$) and discrete actions for $\mathsf{Ag}_2$. Additionally, $\mathsf{Ag}_1$ is informed of its own actions and $\mathsf{Ag}_2$ of joint actions. Thus, $\mathsf{Ag}_1$ is partially informed, without access to environment states and actions of $\mathsf{Ag}_2$, and $\mathsf{Ag}_2$ is fully informed. Since $\mathsf{Ag}_2$ needs no percepts, its local state and transition function are omitted.

The game executes as follows. A global state of $\mathsf{C}$ comprises a state $s_1 = (loc_1, per_1)$ for $\mathsf{Ag}_1$ and an environment state $s_E$. In state $s = (s_1, s_E)$, the two agents concurrently choose one of their actions, resulting in a joint action $a = (a_1, a_2) \in A$. Next, the local state of $\mathsf{Ag}_1$ is updated to some $loc'_1 \in Loc_1$, according to $\delta_1(s_1, a)$. At the same time, the environment state is updated to some $s'_E \in S_E$ according to $\delta_E(loc_1, s_E, a)$. Finally, the first agent $\mathsf{Ag}_1$, based on $loc'_1$, generates a percept $per'_1 = obs_1(loc'_1, s'_E)$ by observing the environment state $s'_E$ and $\mathsf{C}$ reaches the global state $s' = ((loc'_1, per'_1), s'_E)$.

We focus on neural perception functions, i.e., for each local state $loc_1$, we associate an NN classifier $f_{loc_1} : S_E \to \mathbb{P}(Per_1)$ that returns a distribution over percepts for each environment state $s_E \in S_E$. Then $obs_1(loc_1, s_E) = f^{\max}_{loc_1}(s_E)$, where $f^{\max}_{loc_1}(s_E)$ is the percept with the largest probability in $f_{loc_1}(s_E)$ (a tie-breaking rule is applied if multiple percepts have the largest probability).

**Motivating example: Pedestrian-vehicle interaction.** A key challenge for autonomous driving in urban environments is predicting pedestrians' intentions or actions. One solution is NN classifiers, e.g., trained on video datasets [29,28]. To illustrate our NS-POSG model, we consider decision making for an autonomous vehicle using an NN-based intention estimation model for a pedestrian at a crossing [28]. We use their simpler "vanilla" model, which takes two successive (relative) locations of the pedestrian (the top-left coordinates $(x_1, y_1)$ and $(x_2, y_2)$ of two fixed size bounding boxes around the pedestrian) and classifies its intention as: *unlikely*, *likely* or *very likely* to cross. We train a feed-forward NN classifier with ReLU activation functions over the PIE dataset [28].

We build this perception mechanism into an NS-POSG model of a vehicle yielding at a pedestrian crossing, based on [13], illustrated in Fig. 1. A pedestrian further ahead at the side of the road may decide to cross and the vehicle must decide how to adapt its speed. The first, partially-informed agent represents the vehicle. It observes the environment (comprising the successive pedestrian locations) using the NN-based perception mechanism to predict the pedestrian's intention. This is stored as a percept and its speed as its local state. The vehicle chooses between selected (positive or negative) acceleration actions. The second agent, the pedestrian, is fully informed, providing a worst-case analysis of the vehicle decisions, and can decide to cross or return to the roadside. The goal of the vehicle is to minimise the likelihood of a collision with the pedestrian, which is achieved by associating a negative reward with this event.

Fig. 1 also shows selected slices of the state space decomposition obtained by computing the pre-image [25] of the learnt NN classifier, for each of the three predicted intentions. The decision boundaries are non-trivial, justifying our goal of performing a formal analysis, but some intuitive characteristics can be seen. When $x_2 \geq x_1$, meaning that the pedestrian is stationary or moving away from the road, it will generally be classified as *unlikely* to cross. We also see the prediction model is *cautious* when trying to make an estimation if its first observation is made from greater distance. More details are in [35].

**One-sided NS-POSG semantics.** A one-sided NS-POSG $\mathsf{C}$ induces a POSG $[\![\mathsf{C}]\!]$, where we restrict to states that are *percept compatible*, i.e., where $per_1 = obs_1(loc_1, s_E)$ for $s = ((loc_1, per_1), s_E)$. The semantics of a one-sided NS-POSG is closed with respect to percept compatible states.

**Definition 2 (Semantics)** *Given a one-sided NS-POSG* $\mathsf{C}$*, as in Definition 1, its semantics is the POSG* $[\![\mathsf{C}]\!] = (N, S, A, \delta, \mathcal{O}, Z)$ *where:*

- $N = \{1, 2\}$ *is a set of two agents and* $A = A_1 \times A_2$*;*
- $S \subseteq S_1 \times S_E$ *is the set of percept compatible states;*
- *for* $s = (s_1, s_E), s' = (s'_1, s'_E) \in S$ *and* $a \in A$ *where* $s_1 = (loc_1, per_1)$ *and* $s'_1 = (loc'_1, per'_1)$*, we have* $\delta(s, a)(s') = \delta_1(s_1, a)(loc'_1)\delta_E(loc_1, s_E, a)(s'_E)$*;*
- $\mathcal{O} = \mathcal{O}_1 \times \mathcal{O}_2$*, where* $\mathcal{O}_1 = S_1$ *and* $\mathcal{O}_2 = S$*;*
- $Z(s, a, s') = (s'_1, s')$ *for* $s \in S$*,* $a \in A$ *and* $s' = (s'_1, s'_E) \in S$*.*

**Strategies.** As $[\![\mathsf{C}]\!]$ is a POSG, we consider (behaviour) *strategies* for the two agents. Since $\mathsf{Ag}_2$ is fully informed, it can recover the beliefs of $\mathsf{Ag}_1$, thus removing nested beliefs. Hence, the AOHs of $\mathsf{Ag}_2$ are equal to the histories of $[\![\mathsf{C}]\!]$, i.e., $FPaths_{[\![\mathsf{C}]\!],2} = FPaths_{[\![\mathsf{C}]\!]}$. We also consider the *stage strategies* at a history of $[\![\mathsf{C}]\!]$, which will later be required for solving the induced zero-sum normal-form games in the minimax operator. For a history $\pi$ of $[\![\mathsf{C}]\!]$, a stage strategy for $\mathsf{Ag}_1$ is a distribution $u_1 \in \mathbb{P}(A_1)$ and a stage strategy for $\mathsf{Ag}_2$ is a function $u_2 : S \to \mathbb{P}(A_2)$, i.e., $u_2 \in \mathbb{P}(A_2 \mid S)$.

**Beliefs.** Since $\mathsf{Ag}_1$ is partially informed, it may need to infer the current state from its AOH. For an $\mathsf{Ag}_1$ state $s_1 = (loc_1, per_1)$, we let $S_E^{s_1}$ be the set of environment states compatible with $s_1$, i.e., $S_E^{s_1} = \{s_E \in S_E \mid obs_1(loc_1, s_E) =$

$per_1\}$. Since the states of $\mathsf{Ag}_1$ are also the observations of $\mathsf{Ag}_1$ and states of $[\![\mathsf{C}]\!]$ are percept compatible, a *belief* for $\mathsf{Ag}_1$, which can also be reconstructed by $\mathsf{Ag}_2$, can be represented as a pair $b = (s_1, b_1)$, where $s_1 \in S_1$, $b_1 \in \mathbb{P}(S_E)$ and $b_1(s_E) = 0$ for all $s_E \in S_E \setminus S_E^{s_1}$. We denote by $S_B$ the set of beliefs of $\mathsf{Ag}_1$.

Given a belief $(s_1, b_1)$, if action $a_1$ is selected by $\mathsf{Ag}_1$, $\mathsf{Ag}_2$ is *assumed* to take stage strategy $u_2 \in \mathbb{P}(A_2 \mid S)$ and $s_1'$ is observed, then the updated belief of $\mathsf{Ag}_1$ via Bayesian inference is denoted $(s_1', b_1^{s_1,a_1,u_2,s_1'})$; see [35] for details.

## 4 Values of One-Sided NS-POSGs

We establish the *value function* of a one-sided NS-POSG $\mathsf{C}$ with semantics $[\![\mathsf{C}]\!]$, which gives the minimax expected reward from an initial belief, and show its convexity and continuity. Next, to compute it, we introduce minimax and maxsup operators specialised for one-sided NS-POSGs, and prove their equivalence. Finally, we provide a fixed-point characterisation of the value function.

**Value function.** We assume a fixed reward structure $r$ and discount factor $\beta$. The *value function* of $\mathsf{C}$ represents the minimax expected reward in each possible initial belief of the game, given by $V^\star : S_B \to \mathbb{R}$, where $V^\star(s_1, b_1) = \mathbb{E}^{\sigma^\star}_{(s_1,b_1)}[Y]$ for all $(s_1, b_1) \in S_B$ and $\sigma^\star$ is a minimax strategy profile of $[\![\mathsf{C}]\!]$.

The value function for zero-sum POSGs may not exist when the state space is uncountable [14,2,30] as in our case. In this paper, we only consider one-sided NS-POSGs that are determined, i.e., for which the value function exists.

**Convexity and continuity.** Since $r$ is bounded, the value function $V^\star$ has lower and upper bounds $L = \min_{s\in S, a\in A} r(s,a)/(1-\beta)$ and $U = \max_{s\in S, a\in A} r(s,a)/(1-\beta)$. The proof of the following and all other results can be found in [35].

**Theorem 1 (Convexity and continuity).** *For $s_1 \in S_1$, $V^\star(s_1, \cdot) : \mathbb{P}(S_E) \to \mathbb{R}$ is convex and continuous, and for $b_1, b_1' \in \mathbb{P}(S_E)$ : $|V^\star(s_1, b_1) - V^\star(s_1, b_1')| \leq K(b_1, b_1')$ where $K(b_1, b_1') = \frac{1}{2}(U - L)\int_{s_E \in S_E^{s_1}} \left|b_1(s_E) - b_1'(s_E)\right| \mathrm{d}s_E$.*

**Minimax and maxsup operators.** We give a fixed-point characterisation of the value function $V^\star$, first introducing a minimax operator and then simplifying to an equivalent maxsup variant. The latter will be used in Section 5 to prove closure of our representation for value functions and in Section 6 to formulate HSVI. For $f : S \to \mathbb{R}$ and belief $(s_1, b_1)$, let $\langle f, (s_1, b_1)\rangle = \int_{s_E \in S_E} f(s_1, s_E) b_1(s_E) \mathrm{d}s_E$ and $\mathbb{F}(S_B)$ denote the space of functions mapping the beliefs $S_B$ to reals $\mathbb{R}$.

**Definition 3 (Minimax)** *The minimax operator $T : \mathbb{F}(S_B) \to \mathbb{F}(S_B)$ is given by:*

$$[TV](s_1, b_1) = \max_{u_1 \in \mathbb{P}(A_1)} \min_{u_2 \in \mathbb{P}(A_2|S)} \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] + \beta \sum_{(a_1,s_1')\in A_1\times S_1} P(a_1, s_1' \mid (s_1, b_1), u_1, u_2) V(s_1', b_1^{s_1,a_1,u_2,s_1'}) \quad (1)$$

*for $V \in \mathbb{F}(S_B)$ and $(s_1, b_1) \in S_B$, where $\mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] = \int_{s_E \in S_E} b_1(s_E) \sum_{(a_1,a_2)\in A} u_1(a_1) u_2(a_2 \mid s_1, s_E) r((s_1, s_E), (a_1, a_2)) \mathrm{d}s_E$.*

Motivated by [19], which proposed an equivalent operator for the discrete case, we instead prove that the minimax operator has an equivalent simplified form over convex continuous functions of $\mathbb{F}(S_B)$.

For $\Gamma \subseteq \mathbb{F}(S)$, we let $\Gamma^{A_1 \times S_1}$ denote the set of vectors of elements of the convex hull of $\Gamma$ indexed by $A_1 \times S_1$. Furthermore, for $u_1 \in \mathbb{P}(A_1)$, $\overline{\alpha} = (\alpha^{a_1,s_1'})_{(a_1,s_1') \in A_1 \times S_1} \in \Gamma^{A_1 \times S_1}$ and $a_2 \in A_2$, we define $f_{u_1,\overline{\alpha},a_2} : S \to \mathbb{R}$ to be the function such that, for $s \in S$:

$$f_{u_1,\overline{\alpha},a_2}(s) = \sum_{a_1 \in A_1} u_1(a_1) r(s,(a_1,a_2) + \beta \sum_{(a_1,s_1') \in A_1 \times S_1} u_1(a_1) \sum_{s_E' \in S_E} \delta(s,(a_1,a_2))(s_1',s_E') \alpha^{a_1,s_1'}(s_1',s_E') \quad (2)$$

where the sum over $s_E'$ is due to the finite branching of $\delta(s,(a_1,a_2))$.

**Definition 4 (Maxsup)** *For $\varnothing \neq \Gamma \subseteq \mathbb{F}(S)$, if $V(s_1,b_1) = \sup_{\alpha \in \Gamma} \langle \alpha, (s_1,b_1) \rangle$ for $(s_1,b_1) \in S_B$, then the maxsup operator $T_\Gamma : \mathbb{F}(S_B) \to \mathbb{F}(S_B)$ is defined as $[T_\Gamma V](s_1,b_1) = \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \langle f_{u_1,\overline{\alpha}}, (s_1,b_1) \rangle$ for $(s_1,b_1) \in S_B$ where $f_{u_1,\overline{\alpha}}(s) = \min_{a_2 \in A_2} f_{u_1,\overline{\alpha},a_2}(s)$ for $s \in S$.*

In the maxsup operator, $u_1$ and $\overline{\alpha}$ are aligned with $\mathsf{Ag}_1$'s goal of maximising the objective, where $u_1$ is over action distributions and $\overline{\alpha}$ is over convex combinations of elements of $\Gamma$. The minimisation by $\mathsf{Ag}_2$ is simplified to an optimisation over its finite action set in the function $f_{u_1,\overline{\alpha}}$. Note that each state may require a different minimiser $a_2$, as $\mathsf{Ag}_2$ knows the current state before taking an action.

The maxsup operator avoids the minimisation over Markov kernels with continuous states in the original minimax operator. Given $u_1$ and $\overline{\alpha}$, the minimisation can induce a pure best-response stage strategy $u_2 \in \mathbb{P}(A_2 \mid S)$ such that, for any $s \in S$, $u_2(a_2' \mid s) = 1$ for some $a_2' \in \arg\min_{a_2 \in A_2} f_{u_1,\overline{\alpha},a_2}(s)$. Using Theorem 1, the operator equivalence and fixed-point result are as follows.

**Theorem 2 (Operator equivalence and fixed point).** *For $\varnothing \neq \Gamma \subseteq \mathbb{F}(S)$, if $V(s_1,b_1) = \sup_{\alpha \in \Gamma} \langle \alpha, (s_1,b_1) \rangle$ for $(s_1,b_1) \in S_B$, then the minimax operator $T$ and maxsup operator $T_\Gamma$ are equivalent and their unique fixed point is $V^\star$.*

## 5 P-PWLC Value Iteration

We next discuss a representation for value functions using *piecewise constant* (PWC) $\alpha$-functions, called P-PWLC (*piecewise linear and convex under PWC*), originally introduced in [36]. This representation extends the $\alpha$-functions of [27,6,38] for continuous-state POMDPs, but a key difference is that we work with polyhedral representations (induced precisely from NNs) rather than approximations based on Gaussian mixtures [27] or beta densities [15].

We show that, given PWC representations for an NS-POSG's perception, reward and transition functions, and under mild assumptions on model structure, P-PWLC value functions are closed with respect to the minimax operator. This yields a (non-scalable) *value iteration* algorithm and, subsequently, the basis for a more practical point-based HSVI algorithm in Section 6.

**PWC representations.** A *finite connected partition* (FCP) of $S$, denoted $\Phi$, is a finite collection of disjoint connected *regions* (subsets) of $S$ that cover it.

**Definition 5 (PWC function)** *A function $f : S \to \mathbb{R}$ is piecewise constant (PWC) if there exists an FCP $\Phi$ of $S$ such that $f : \phi \to \mathbb{R}$ is constant for $\phi \in \Phi$. Let $\mathbb{F}_C(S)$ be the set of PWC functions in $\mathbb{F}(S)$.*

Since we focus on NNs for $\mathsf{Ag}_1$'s perception function $obs_1$, it is PWC (as for the one-agent case [36]) and the state space $S$ of a one-sided NS-POSG can be decomposed into a finite set of *regions*, each with the same observation. Formally, there exists a *perception FCP* $\Phi_P$, the smallest FCP of $S$ such that all states in any $\phi \in \Phi_P$ are observationally equivalent, i.e., if $(s_1, s_E), (s'_1, s'_E) \in \phi$, then $s_1 = s'_1$. We can use $\Phi_P$ to find the set $S_E^{s_1}$ for any agent state $s_1 \in S_1$. Given an NN representation of $obs_1$, the corresponding FCP $\Phi_P$ can be extracted (or approximated) offline by analysing its pre-image [25].

We also need to make some assumptions about the transitions and rewards of one-sided NS-POSGs (in a similar style to [36]). Informally, we require that, for any decomposition $\Phi'$ of the state-space into regions (i.e., an FCP), there is a second decomposition $\Phi$, the *pre-image FCP*, such that states in regions of $\Phi$ have the same rewards and transition probabilities into regions of $\Phi'$. The transitions of the (continuous) environment must also be decomposable into regions.

**Assumption 1 (Transitions and rewards)** *Given any FCP $\Phi'$ of $S$, there exists an FCP $\Phi$ of $S$, called the* pre-image FCP *of $\Phi'$, where for $\phi \in \Phi$, $a \in A$ and $\phi' \in \Phi'$ there exists $\delta_\Phi : (\Phi \times A) \to \mathbb{P}(\Phi')$ and $r_\Phi : (\Phi \times A) \to \mathbb{R}$ such that $\delta(s,a)(s') = \delta_\Phi(\phi,a)(\phi')$ and $r(s,a) = r_\Phi(\phi,a)$ for $s \in \phi$ and $s' \in \phi'$. In addition, $\delta_E$ can be expressed in the form $\sum_{i=1}^{n} \mu_i \delta_E^i$, where $n \in \mathbb{N}$, $\mu_i \in [0,1]$, $\sum_{i=1}^{n} \mu_i = 1$ and $\delta_E^i : (Loc_1 \times S_E \times A) \to S_E$ are piecewise continuous functions.*

The need for this assumption also becomes clear in our later algorithms, which compute a representation for an NS-POSG's value function over a (polyhedral) partition of the state space. This partition is created dynamically over the iterations of the solution, using a pre-image based splitting operation.

We now show, using results for continuous-state POMDPs [36,27], that $V^\star$ is the limit of a sequence of $\alpha$-functions, called *piecewise linear and convex under PWC $\alpha$-functions*, first introduced in [36] for neuro-symbolic POMDPs.

**Definition 6 (P-PWLC function)** *A function $V : S_B \to \mathbb{R}$ is* piecewise linear and convex under PWC $\alpha$-functions (P-PWLC) *if there exists a finite set $\Gamma \subseteq \mathbb{F}_C(S)$ such that $V(s_1, b_1) = \max_{\alpha \in \Gamma} \langle \alpha, (s_1, b_1) \rangle$ for $(s_1, b_1) \in S_B$, where the functions in $\Gamma$ are called PWC $\alpha$-functions.*

If $V \in \mathbb{F}(S_B)$ is P-PWLC, then it can be represented by a set of PWC functions over $S$, i.e., as a finite set of FCP regions and a value vector. Recall that $\langle \alpha, (s_1, b_1) \rangle = \int_{s_E \in S_E} \alpha(s_1, s_E) b_1(s_E) \mathrm{d}s_E$, and therefore computing the value for a belief involves integration. For one-sided NS-POSGs, we demonstrate, under Assumption 1, closure of the P-PWLC representation for value functions under the minimax operator and the convergence of value iteration.

**LP, closure property and convergence.** By showing that $f_{u_1,\overline{\alpha},a_2}$ in (2) is PWC in $S$ (see [35]), we use Theorem 2 to demonstrate that, if $V$ is P-PWLC, the minimax operation can be computed by solving an LP.

**Lemma 1 (LP for minimax and P-PWLC)** *If $V \in \mathbb{F}(S_B)$ is P-PWLC, then $[TV](s_1, b_1)$ is given by an LP for $(s_1, b_1) \in S_B$.*

Using Lemma 1, we show that the P-PWLC representation is closed under the minimax operator. This closure property enables iterative computation of a sequence of such functions to approximate $V^\star$ to within a convergence guarantee.

**Theorem 3 (P-PWLC closure and convergence).** *If $V \in \mathbb{F}(S_B)$ is P-PWLC, then so is $[TV]$. If $V^0 \in \mathbb{F}(S_B)$ is P-PWLC, then the sequence $(V^t)_{t=0}^{\infty}$, such that $V^{t+1} = [TV^t]$, is P-PWLC and converges to $V^\star$.*

An implementation of value iteration for one-sided NS-POSGs is therefore feasible, since each $\alpha$-function involved is PWC and thus allows for a finite representation. However, as the number of $\alpha$-functions grows exponentially in the number of iterations, it is not scalable in practice.

## 6 Heuristic Search Value Iteration for NS-POSGs

To provide a more practical approach to solving one-sided NS-POSGs, we now present a variant of HSVI (heuristic search value iteration) [31], an anytime algorithm that approximates the value function $V^\star$ via lower and upper bound functions, updated through heuristically generated beliefs.

Our approach broadly follows the structure of HSVI for *finite* POSGs [19], but every step presents challenges when extending to continuous states and NN-based observations. In particular, we must work with integrals over beliefs and deal with uncountability, using P-PWLC (rather than PWLC) functions for lower bounds, and therefore different ingredients to prove convergence. Value computations are also much more complex because NN perception function induce FCPs, which are used to compute images, pre-images and intersections.

We also build on ideas from HVSI for (single-agent) neuro-symbolic POMDPs in [36]. The presence of two opposing agents brings three main challenges. First, value backups at belief points require solving normal-form games instead of maximising over one agent's actions. Second, since the first agent is not informed of the joint action, in the value backups and belief updates of the maxsup operator uncountably many stage strategies of the second agent have to be considered, whereas, in the single-agent variant, the agent can decide the transition probabilistically on its own. Third, the forward exploration heuristic is more complex as it depends on the stage strategies of the agents in two-stage games.

### 6.1 Lower and Upper Bound Representations

We first discuss representing and updating the lower and upper bound functions.

**Lower bound function.** Selecting an appropriate representation for $\alpha$-functions requires closure properties with respect to the maxsup operator. Motivated

by [36], we represent the lower bound $V_{lb}^{\Gamma} \in \mathbb{F}(S_B)$ as the P-PWLC function for a finite set $\Gamma \subseteq \mathbb{F}_C(S)$ of PWC $\alpha$-functions (see Definition 6), for which the closure is guaranteed by Theorem 3. The lower bound $V_{lb}^{\Gamma}$ has a finite representation as each $\alpha$-function is PWC, and is initialised as in [19].

**Upper bound function.** The upper bound $V_{ub}^{\Upsilon} \in \mathbb{F}(S_B)$ is represented by a finite set of belief-value points $\Upsilon = \{((s_1^i, b_1^i), y_i) \in S_B \times \mathbb{R} \mid i \in I\}$, where $y_i$ is an upper bound of $V^{\star}(s_1^i, b_1^i)$. Similarly to [36], for any $(s_1, b_1) \in S_B$, the upper bound $V_{ub}^{\Upsilon}(s_1, b_1)$ is the lower envelope of the lower convex hull of the points in $\Upsilon$ satisfying the following LP problem: minimise

$$\textstyle\sum_{i \in I_{s_1}} \lambda_i y_i + K_{ub}(b_1, \sum_{i \in I_{s_1}} \lambda_i b_1^i) \text{ subject to } \lambda_i \geq 0 \text{ and } \sum_{i \in I_{s_1}} \lambda_i = 1 \qquad (3)$$

for $i \in I_{s_1}$ where $I_{s_1} = \{i \in I \mid s_1^i = s_1\}$ and $K_{ub} : \mathbb{P}(S_E) \times \mathbb{P}(S_E) \to \mathbb{R}$ measures the difference between two beliefs such that, if $K$ is the function from Theorem 1, then for any $b_1, b_1', b_1'' \in \mathbb{P}(S_E)$: $K_{ub}(b_1, b_1) = 0$,

$$K_{ub}(b_1, b_1') \geq K(b_1, b_1') \quad \text{and} \quad |K_{ub}(b_1, b_1') - K_{ub}(b_1, b_1'')| \leq K_{ub}(b_1', b_1'')\,. \qquad (4)$$

Note that (3) is close to the upper bound in regular HSVI for finite-state spaces, except for the function $K_{ub}$ that measures the difference between two beliefs (two continuous-state functions). With respect to the upper bound used in [36], $K_{ub}$ here needs to satisfy an additional triangle property in (4) to ensure the continuity of $V_{ub}^{\Upsilon}$, for the convergence of the point-based algorithm below. The properties of $K_{ub}$ imply that (3) is an upper bound after a value backup, as stated in Lemma 3 below. The upper bound $V_{ub}^{\Upsilon}$ is initialised as in [19].

**Lower bound updates.** For the lower bound $V_{lb}^{\Gamma}$, in each iteration we add a new PWC $\alpha$-function $\alpha^{\star}$ to $\Gamma$ at a belief $(s_1, b_1) \in S_B$ such that:

$$\langle \alpha^{\star}, (s_1, b_1) \rangle = [TV_{lb}^{\Gamma}](s_1, b_1) = \langle f_{\overline{p}_1^{\star}, \overline{\alpha}^{\star}}, (s_1, b_1) \rangle \qquad (5)$$

where the second equality follows from Lemma 1 and $(\overline{p}_1^{\star}, \overline{\alpha}^{\star})$ is computed via the optimal solution to the LP in Lemma 1 at $(s_1, b_1)$.

Using $\overline{p}_1^{\star}$, $\overline{\alpha}^{\star}$ and the perception FCP $\Phi_P$, Algorithm 1 computes a new $\alpha$-function $\alpha^{\star}$ at belief $(s_1, b_1)$. To guarantee (5) and improve efficiency, we only compute the backup values for regions $\phi \in \Phi_P$ over which $(s_1, b_1)$ has positive probabilities, i.e., $s_1^{\phi} = s_1$ (where $s_1^{\phi}$ is the unique agent state appearing in $\phi$) and $\int_{(s_1, s_E) \in \phi} b_1(s_E) \mathrm{d}s_E > 0$, and assign the trivial lower bound $L$ otherwise.

For each region $\phi$ either $\alpha^{\star}(\hat{s}_1, \hat{s}_E) = f_{\overline{p}_1^{\star}, \overline{\alpha}^{\star}}(\hat{s}_1, \hat{s}_E)$ or $\alpha^{\star}(\hat{s}_1, \hat{s}_E) = L$ for all $(\hat{s}_1, \hat{s}_E) \in \phi$. Computing the backup values in line 4 of Algorithm 1 state by state is computationally intractable, as $\phi$ contains an infinite number of states. However, the following lemma shows that $\alpha^{\star}$ is PWC, allowing a tractable region-by-region backup, called Image-Split-Preimage-Product (ISPP) backup, which is adapted from the single-agent variant in [36]. The details of the ISPP backup for one-sided NS-POSGs are in [35]. The lemma also shows that the lower bound function increases and is valid after each update.

ALGORITHM 1 Point-based *Update*$(s_1, b_1)$ of $(V_{lb}^{\Gamma}, V_{ub}^{\Upsilon})$

1: $(\overline{p}_1^\star, \overline{\alpha}^\star) \leftarrow [TV_{lb}^{\Gamma}](s_1, b_1)$ via an LP in Lemma 1
2: **for** $\phi \in \Phi_P$ **do**
3: **if** $s_1^\phi = s_1$ and $\int_{(s_1, s_E)\in\phi} b_1(s_E)\mathrm{d}s_E > 0$ **then**
4: $\alpha^\star(\hat{s}_1, \hat{s}_E) \leftarrow f_{\overline{p}_1^\star, \overline{\alpha}^\star}(\hat{s}_1, \hat{s}_E)$ for $(\hat{s}_1, \hat{s}_E) \in \phi$ ▷ ISPP backup
5: **else** $\alpha^\star(\hat{s}_1, \hat{s}_E) \leftarrow L$ for $(\hat{s}_1, \hat{s}_E) \in \phi$
6: $\Gamma \leftarrow \Gamma \cup \{\alpha^\star\}$
7: $y^\star \leftarrow [TV_{ub}^{\Upsilon}](s_1, b_1)$ via (1) and (3)
8: $\Upsilon \leftarrow \Upsilon \cup \{((s_1, b_1), y^\star)\}$

**Lemma 2 (Lower bound)** *The function $\alpha^\star$ generated by Algorithm 1 is a PWC $\alpha$-function satisfying (5), and if $\Gamma' = \Gamma \cup \{\alpha^\star\}$, then $V_{lb}^{\Gamma} \leq V_{lb}^{\Gamma'} \leq V^\star$.*

**Upper bound updates.** For the upper bound $V_{ub}^{\Upsilon}$, due to representation (3), at a belief $(s_1, b_1) \in S_B$ in each iteration, we add a new belief-value point $((s_1, b_1), y^\star)$ to $\Upsilon$ such that $y^\star = [TV_{ub}^{\Upsilon}](s_1, b_1)$. Computing $[TV_{ub}^{\Upsilon}](s_1, b_1)$ via (1) and (3) requires the concrete formula for $K_{ub}$ and the belief representations. Thus, we will show how to compute $[TV_{ub}^{\Upsilon}](s_1, b_1)$ when introducing belief representations below. The following lemma shows that $y^\star \geq V^\star(s_1, b_1)$ required by (3), and the upper bound function is decreasing and is valid after each update.

**Lemma 3 (Upper bound)** *Given a belief $(s_1, b_1) \in S_B$, if $y^\star = [TV_{ub}^{\Upsilon}](s_1, b_1)$, then $y^\star$ is an upper bound of $V^\star$ at $(s_1, b_1)$, i.e., $y^\star \geq V^\star(s_1, b_1)$, and if $\Upsilon' = \Upsilon \cup \{((s_1, b_1), y^\star)\}$, then $V_{ub}^{\Upsilon} \geq V_{ub}^{\Upsilon'} \geq V^\star$.*

### 6.2 One-Sided NS-HSVI

Algorithm 2 presents our NS-HSVI algorithm for one-sided NS-POSGs.

**Forward exploration heuristic.** The algorithm uses a heuristic approach to select which belief will be considered next. Similarly to finite-state one-sided POSGs [19], we focus on a belief that has the highest *weighted excess gap*. The excess gap at a belief $(s_1, b_1)$ with depth $t$ from the initial belief is defined by $excess_t(s_1, b_1) = V_{ub}^{\Upsilon}(s_1, b_1) - V_{lb}^{\Gamma}(s_1, b_1) - \rho(t)$, where $\rho(0) = \varepsilon$ and $\rho(t{+}1) = (\rho(t) - 2(U - L)\bar{\varepsilon})/\beta$, and $\bar{\varepsilon} \in (0, (1-\beta)\varepsilon/(2U - 2L))$. Using this excess gap, the next action-observation pair $(\hat{a}_1, \hat{s}_1)$ for exploration is selected from:

$$\mathrm{argmax}_{(a_1, s_1')\in A_1\times S_1} P(a_1, s_1' \mid (s_1, b_1), u_1^{ub}, u_2^{lb}) excess_{t+1}(s_1', b_1^{s_1, a_1, u_2^{lb}, s_1'}). \quad (6)$$

To compute the next belief via lines 8 and 9 of Algorithm 2, the minimax strategy profiles in stage games $[TV_{lb}^{\Gamma}](s_1, b_1)$ and $[TV_{ub}^{\Upsilon}](s_1, b_1)$, i.e., $(u_1^{ub}, u_2^{lb})$, are required. Since $V_{lb}^{\Gamma}$ is P-PWLC, using Lemma 1, the strategy $u_2^{lb}$ is obtained by solving an LP. However, the computation of the strategy $u_1^{ub}$ depends on the representation of $(s_1, b_1)$ and the measure function $K_{ub}$, and thus will be discussed later. One-sided NS-HSVI has the following convergence guarantees.

**Theorem 4 (One-sided NS-HSVI).** *For any $(s_1^{init}, b_1^{init}) \in S_B$ and $\varepsilon > 0$, Algorithm 2 will terminate and upon termination: $V_{ub}^{\Upsilon}(s_1^{init}, b_1^{init}) - V_{lb}^{\Gamma}(s_1^{init}, b_1^{init}) \leq \varepsilon$ and $V_{lb}^{\Gamma}(s_1^{init}, b_1^{init}) \leq V^\star(s_1^{init}, b_1^{init}) \leq V_{ub}^{\Upsilon}(s_1^{init}, b_1^{init})$.*

ALGORITHM 2 One-sided NS-HSVI for one-sided NS-POSGs

**while** $V^{\Upsilon}_{ub}(s_1^{init}, b_1^{init}) - V^{\Gamma}_{lb}(s_1^{init}, b_1^{init}) > \varepsilon$ **do** *Explore*$((s_1^{init}, b_1^{init}), 0)$
**return** $V^{\Gamma}_{lb}$ and $V^{\Upsilon}_{ub}$ via sets $\Gamma$ and $\Upsilon$
**function** *Explore*$((s_1, b_1), t)$
    $(u_1^{lb}, u_2^{lb}) \leftarrow$ minimax strategy profile in $[TV^{\Gamma}_{lb}](s_1, b_1)$
    $(u_1^{ub}, u_2^{ub}) \leftarrow$ minimax strategy profile in $[TV^{\Upsilon}_{ub}](s_1, b_1)$
    *Update*$(s_1, b_1)$ ▷ Algorithm 1
    $(\hat{a}_1, \hat{s}_1) \leftarrow$ select according to forward exploration heuristic
    **if** $P(\hat{a}_1, \hat{s}_1 \mid (s_1, b_1), u_1^{ub}, u_2^{lb}) excess_{t+1}(\hat{s}_1, b_1^{s_1, \hat{a}_1, u_2^{lb}, \hat{s}_1}) > 0$ **then**
        *Explore*$((\hat{s}_1, b_1^{s_1, \hat{a}_1, u_2^{lb}, \hat{s}_1}), t+1)$
        *Update*$(s_1, b_1)$ ▷ Algorithm 1

### 6.3 Belief Representation and Computations

Implementing one-sided NS-HSVI depends on belief representations, as closed forms are needed. We use the popular *particle-based representation* [27,10], which can approximate arbitrary beliefs and handle non-Gaussian systems. However, compared to region-based representations [36], it is more vulnerable to disturbances and can require many particles for a good approximation.

**Particle-based beliefs.** A *particle-based belief* $(s_1, b_1) \in S_B$ is represented by a weighted particle set $\{(s_E^i, \kappa_i)\}_{i=1}^{n_s}$ with a normalised weight $\kappa_i$ for each particle $s_E^i \in S_E$, where $b_1(s_E) = \sum_{i=1}^{n_b} \kappa_i D(s_E - s_E^i)$ for $s_E \in S_E$ and $D(s_E - s_E^i)$ is a Dirac delta function centred at 0.

To implement one-sided NS-HSVI using particle-based beliefs, we prove that $V^{\Gamma}_{lb}$ and $V^{\Upsilon}_{ub}$ are eligible representations, as the belief update $b_1^{s_1, a_1, u_2, s_1'}$, expected values $\langle \alpha, (s_1, b_1) \rangle$, $\langle r, (s_1, b_1) \rangle$ and probability $P(a_1, s_1' \mid (s_1, b_1), u_1, u_2)$ are computed as simple summations for a particle-based belief $(s_1, b_1)$ ([35]).

**Lower bound.** Since $V^{\Gamma}_{lb}$ is P-PWLC with PWC $\alpha$-functions $\Gamma$, for a particle-based belief $(s_1, b_1)$ represented by $\{(s_E^i, \kappa_i)\}_{i=1}^{n_b}$, using Definition 6, $V^{\Gamma}_{lb}(s_1, b_1) = \max_{\alpha \in \Gamma} \sum_{i=1}^{n_b} \kappa_i \alpha(s_1, s_E^i)$. The stage game $[TV^{\Gamma}_{lb}](s_1, b_1)$ and minimax strategy profile $(u_1^{lb}, u_2^{lb})$ follow from solving the LP in Lemma 1.

**Upper bound.** To compute $V^{\Upsilon}_{ub}$ in (3), we need a function $K_{ub}$ to measure belief differences that satisfies (4). We take $K_{ub} = K$, which does so by definition. Given $\Upsilon = \{((s_1^i, b_1^i), y_i) \mid i \in I\}$, the upper bound and stage game can be computed by solving an LP, respectively, as demonstrated by the following theorem, and then the minimax strategy profile $(u_1^{ub}, u_2^{ub})$ is synthesised (see [35]).

**Theorem 5 (LPs for upper bound).** *For a particle-based belief $(s_1, b_1) \in S_B$, $V^{\Upsilon}_{ub}(s_1, b_1)$ and $[TV^{\Upsilon}_{ub}](s_1, b_1)$ are the optimal value of an LP, respectively.*

## 7 Experimental Evaluation

We have built a prototype implementation in Python, using Gurobi [16] to solve the LPs needed for computing lower and upper bound values, and the minimax values and strategies of one-shot games. We use the Parma Polyhedra Library [1] to operate over polyhedral pre-images of NNs, $\alpha$-functions and reward structures.

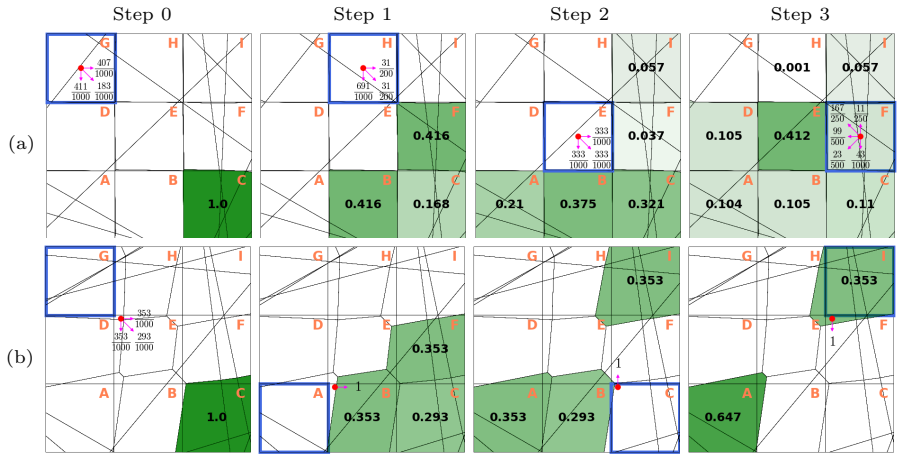



Fig. 2: Simulations of strategies for the pursuer, showing actual location (red), perceived location (blue), belief of evader location (green) and strategy (pink) for two different NN perception functions: (a) more precise; (b) coarser.

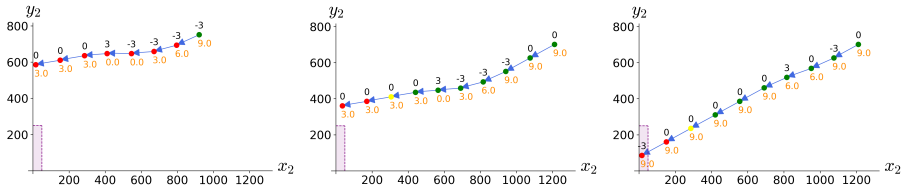
Our evaluation uses two one-sided NS-POSG examples: a *pursuit-evasion* game and the *pedestrian-vehicle* scenario from Section 3. Below, we discuss the applicability and usefulness of our techniques on these examples. Due to limited space, we refer to [35] for more details of the models, including the training of the ReLU NN classifiers, and empirical results on performance.

**Pursuit-evasion.** A pursuit-evasion game models a *pursuer* trying to catch an *evader* aiming to avoid capture. We build a continuous-space variant of the model from [19] inspired by mobile robotics applications [8,20]. The environment includes the exact position of both agents. The (partially informed) pursuer uses an NN classifier to perceive its own location, which maps to one of $3\times3$ grid cells. To showcase the ability of our methodology to assess the performance of realistic NN perception functions, we train two NNs, the second with a coarser accuracy.

Fig. 2 shows simulations of strategies synthesised for the pursuer, using the two different NNs. Its actual location is a red dot, and the pink arrows denote the strategy. Blue squares show the cell that is output by the pursuer's perception function, and black lines mark the underlying polyhedral decomposition. The pursuer's belief over the evader's location is shown by the green shading and annotated probabilities; it initially (correctly) believes that the evader is in cell $C$ and the belief evolves based on the optimal counter-strategy of the evader.

The plots show we can synthesise non-trivial strategies for agents using NN-based perception in a partially observable setting. We can also study the impact of a poorly trained perception function. Fig. 2(b), for the coarser NN, shows the pursuer repeatedly mis-detecting its location because the grid cells shapes are poorly approximated, and subsequently taking incorrect actions. This is exploited by the evader, leading to considerably worse performance for the pursuer.

**Pedestrian-vehicle interaction.** Fig. 3 shows several simulations from strategies synthesised for the pedestrian-vehicle example described in Section 3 (Fig. 1),




(a) (b) (c)

Fig. 3: Simulations of strategies for the vehicle, plotted as the pedestrian's current position $(x_2, y_2)$ relative to it. Also shown: perceived pedestrian intention (green/yellow/red = *unlikely*/*likely*/*very likely* to cross), current speed (orange), acceleration (black) and crash region (shaded purple region).

plotting the position $(x_2, y_2)$ of the pedestrian, relative to the vehicle. We fix the pedestrian's strategy, to simulate a crossing scenario: it moves from right to left, i.e., decreasing $x_2$. The (partially informed) vehicle's perception function predicts the intention of the pedestrian (green/yellow/red = *unlikely*/*likely*/*very likely* to cross), shown as coloured dots. Above and below each circle, we indicate the acceleration actions taken (black) and current speeds (orange), respectively, which determine the distance $y_2$ to the pedestrian crossing.

Again, we investigate the feasibility of generating strategies for agents with realistic NN-based perception. Here, the goal is to avoid a crash scenario, denoted by the shaded region at the bottom left of the plots. We find that, in many cases, safe strategies can be synthesised. Fig. 3(a) shows an example; notice that the pedestrian intention is detected early. This is not true in (b) and (c), which show two simulations from a strategy and starting point where the perception function results in much later detection; (c) shows we were then unable to synthesise a strategy for the vehicle that is always safe.

## 8 Conclusions

We have proposed one-sided neuro-symbolic POSGs, designed to reason formally about partially observable agents equipped with neural perception mechanisms. We characterised the value function for discounted infinite-horizon rewards, and designed, implemented and evaluated a HSVI algorithm for approximate solution. Computational complexity is high due to expensive polyhedral operations. Nevertheless, our method provides an important baseline that can reason about true decision boundaries for game models with NN-based perception, against which efficiency improvements can later be benchmarked. We plan to investigate ways to improve performance, e.g., merging of adjacent polyhedra or Monte-Carlo planning methods, and to study restricted cases of two-sided NS-POSGs, e.g., those with public observations [18].

**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. Bagnara, R., Hill, P.M., Zaffanella, E.: The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. Science of Computer Programming **72**(1), 3–21 (2008), bugseng.com/ppl
2. Bhabak, A., Saha, S.: Partially observable discrete-time discounted Markov games with general utility. arXiv:2211.07888 (2022)
3. Bosansky, B., Kiekintveld, C., Lisy, V., Pechoucek, M.: An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information. Journal of Artificial Intelligence Research **51**, 829–866 (2014)
4. Brechtel, S., Gindele, T., Dillmann, R.: Solving continuous POMDPs: Value iteration with incremental learning of an efficient space representation. In: Proc. ICML'13. pp. 370–378. PMLR (2013)
5. Brown, N., Bakhtin, A., Lerer, A., Gong, Q.: Combining deep reinforcement learning and search for imperfect-information games. In: Proc. NeurIPS'20. pp. 17057–17069. Curran Associates, Inc. (2020)
6. Burks, L., Loefgren, I., Ahmed, N.R.: Optimal continuous state POMDP planning with semantic observations: A variational approach. IEEE Transactions on Robotics **35**(6), 1488–1507 (2019)
7. Carr, S., Jansen, N., Bharadwaj, S., Spaan, M.T., Topcu, U.: Safe policies for factored partially observable stochastic games. In: Robotics: Science and System XVII (2021)
8. Chung, T.H., Hollinger, G.A., Isler, V.: Search and pursuit-evasion in mobile robotics. Autonomous Robots **31**(4), 299–316 (2011)
9. Delage, A., Buffet, O., Dibangoye, J.S., Saffidine, A.: HSVI can solve zero-sum partially observable stochastic games. Dynamic Games and Applications pp. 1–55 (2023)
10. Doucet, A., De Freitas, N., Gordon, N.J. (eds.): Sequential Monte Carlo methods in practice, vol. 1(2). Springer (2001)
11. Emery-Montemerlo, R., Gordon, G., Schneider, J., Thrun, S.: Approximate solutions for partially observable stochastic games with common payoffs. In: Proc. AAMAS'04. pp. 136–143. IEEE (2004)
12. Feng, Z., Dearden, R., Meuleau, N., Washington, R.: Dynamic programming for structured continuous Markov decision problems. In: Proc. UAI'04. p. 154–161 (2004)
13. Fu, T., Miranda-Moreno, L., Saunier, N.: A novel framework to evaluate pedestrian safety at non-signalized locations. Accident Analysis & Prevention **111**, 23–33 (2018)
14. Ghosh, M.K., McDonald, D., Sinha, S.: Zero-sum stochastic games with partial information. Journal of Optimization Theory and Applications **121**, 99–118 (2004)
15. Guestrin, C., Hauskrecht, M., Kveton, B.: Solving factored MDPs with continuous and discrete variables. In: Proc. UAI'04. p. 235–242 (2004)
16. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2021), gurobi.com
17. Hansen, E.A., Bernstein, D.S., Zilberstein, S.: Dynamic programming for partially observable stochastic games. In: Proc. AAAI'04. vol. 4, pp. 709–715 (2004)
18. Horák, K., Bošanskỳ, B.: Solving partially observable stochastic games with public observations. In: Proc. AAAI'19. vol. 33, pp. 2029–2036 (2019)

19. Horák, K., Bošanskỳ, B., Kovařík, V., Kiekintveld, C.: Solving zero-sum one-sided partially observable stochastic games. Artificial Intelligence **316**, 103838 (2023)
20. Isler, V., Nikhil, K.: The role of information in the cop-robber game. Theoretical Computer Science **399**(3), 179–190 (2008)
21. Kovařík, V., Schmid, M., Burch, N., Bowling, M., Lisỳ, V.: Rethinking formal models of partially observable multiagent decision making. Artificial Intelligence **303**, 103645 (2022)
22. Kovařík, V., Seitz, D., Lisỳ, V., Rudolf, J., Sun, S., Ha, K.: Value functions for depth-limited solving in zero-sum imperfect-information games. Artificial Intelligence **314**, 103805 (2023)
23. Kumar, A., Zilberstein, S.: Dynamic programming approximations for partially observable stochastic games. In: Proc. FLAIRS'09. pp. 547–552 (2009)
24. Madani, O., Hanks, S., Condon, A.: On the undecidability of probabilistic planning and related stochastic optimization problems. Artificial Intelligence **147**(1-2), 5–34 (2003)
25. Matoba, K., Fleuret, F.: Computing preimages of deep neural networks with applications to safety (2020), openreview.netforum?id=FN7_BUOG78e
26. Moravčík, M., Schmid, M., Burch, N., Lisỳ, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M., Bowling, M.: Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. Science **356**(6337), 508–513 (2017)
27. Porta, J.M., Vlassis, N., Spaan, M.T., Poupart, P.: Point-based value iteration for continuous POMDPs. Journal of Machine Learning Research **7**, 2329–2367 (2006)
28. Rasouli, A., Kotseruba, I., Kunic, T., Tsotsos, J.K.: PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In: Proc. ICCV'19. pp. 6262–6271 (2019)
29. Rasouli, A., Kotseruba, I., Tsotsos, J.K.: Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In: Proc. ICCV'17. pp. 206–213 (2017)
30. Saha, S.: Zero-sum stochastic games with partial information and average payoff. Journal of Optimization Theory and Applications **160**(1), 344–354 (2014)
31. Smith, T., Simmons, R.: Heuristic search value iteration for POMDPs. In: Proc. UAI'04. p. 520–527. AUAI (2004)
32. Wiggers, A.J., Oliehoek, F.A., Roijers, D.M.: Structure in the value function of two-player zero-sum games of incomplete information. Frontiers in Artificial Intelligence and Applications **285**, 1628 – 1629 (2016)
33. Yan, R., Santos, G., Norman, G., Parker, D., Kwiatkowska, M.: Strategy synthesis for zero-sum neuro-symbolic concurrent stochastic games. arXiv.2202.06255 (2022)
34. Yan, R., Santos, G., Duan, X., Parker, D., Kwiatkowska, M.: Finite-horizon equilibria for neuro-symbolic concurrent stochastic games. In: Proc. UAI'22. pp. 2170–2180. AUAI Press (2022)
35. Yan, R., Santos, G., Norman, G., Parker, D., Kwiatkowska, M.: Partially observable stochastic games with neural perception mechanisms. arXiv:2310.11566 (2023)
36. Yan, R., Santos, G., Norman, G., Parker, D., Kwiatkowska, M.: Point-based value iteration for POMDPs with neural perception mechanisms. arXiv.2306.17639 (2023)
37. Yan, R., Santos, G., Norman, G., Parker, D., Kwiatkowska, M.: HSVI-based online minimax strategies for partially observable stochastic games with neural perception mechanisms. In: Proc. L4DC'24 (2024)
38. Zamani, Z., Sanner, S., Poupart, P., Kersting, K.: Symbolic dynamic programming for continuous state and observation POMDPs. Advances in Neural Information Processing Systems **25** (2012)

39. Zettlemoyer, L., Milch, B., Kaelbling, L.: Multi-agent filtering with infinitely nested beliefs. Advances in Neural Information Processing Systems **21** (2008)
40. Zheng, W., Jung, T., Lin, H.: The Stackelberg equilibrium for one-sided zero-sum partially observable stochastic games. Automatica **140**, 110231 (2022)
41. Zheng, W., Jung, T., Lin, H.: Continuous-observation one-sided two-player zero-sum partially observable stochastic game with public actions. IEEE Transactions on Automatic Control pp. 1–15 (2023)
42. Zinkevich, M., Johanson, M., Bowling, M., Piccione, C.: Regret minimization in games with incomplete information. Advances in Neural Information Processing Systems **20** (2007)