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
[2] University of Glasgow, Glasgow, G12 8QQ, 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 [41,42]. 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 [40], 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) [33] 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 [37] and (fully-observable) stochastic games [35]. 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 [37], 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 [29]. 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 [37] (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 [43,21,22] and methods based on reinforcement learning and search [5,26]. In [9], an HSVI-like finite-horizon solver that provably converges to an ε-optimal solution is proposed; [34] provides convexity and concavity results but no algorithmic solution.

Methods exist for *one-sided* POSGs: a space partition approach when actions are public [41], a point-based approximate algorithm when observations are continuous [42] 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. As discussed above, our continuous-state model necessitates several new techniques.

For the *continuous-state* but *single-agent* (POMDP) setting, point-based value iteration [28,6,39] and discrete space approximation [4] can be used; the former also use α-functions to represent value functions but, unlike our approach, which exploits structure similarly to [12], they work with (approximate) Gaussian mixtures or beta-densities. As discussed above, in earlier work, we proposed HSVI for neuro-symbolic POMDPs [37], again exploiting the piecewise constant structure of the underlying continuous-state model. Methods for concurrent stochastic games enriched with neural perception mechanisms are proposed in [36,35], including a value iteration algorithm in [35], but partial observability is not considered, which is the main focus of this paper. Recent work [38] 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; $\delta\colon (S\times A) \to \mathbb{P}(S)$ 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 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 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 π, 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 π 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 $\mathit{FPaths}_{\mathsf{G}}$ and $\mathit{FPaths}_{\mathsf{G},i}$ denote the sets of finite histories of G and AOHs of agent i, respectively.

A (behaviour) *strategy* of agent i is a mapping $\sigma_i : \mathit{FPaths}_{\mathsf{G},i} \to \mathbb{P}(A_i)$. We denote by Σ_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 π 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 σ 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 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. We also present a motivating example of a pedestrian-vehicle interaction.

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* C *comprises agents* $\mathsf{Ag}_1 = (S_1,A_1,\mathit{obs}_1,\delta_1)$ *and* $\mathsf{Ag}_2{=}(A_2)$*, and environment* $E{=}(S_E,\delta_E)$*, where:*

- $S_1 = \mathit{Loc}_1\times\mathit{Per}_1$ *is a set of states for* 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* 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* Ag_1*'s perception function;*
- $\delta_1 : (S_1 \times A) \to \mathbb{P}(Loc_1)$ *is* 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.*

One-sided NS-POSGs are a subclass of two-agent, hybrid-state POSGs with discrete observations (S_1) and actions for Ag_1, and continuous observations ($S_1 \times S_E$) and discrete actions for Ag_2. Additionally, Ag_1 is informed of its own actions and Ag_2 of joint actions. Thus, Ag_1 is partially informed, without access to environment states and actions of Ag_2, and Ag_2 is fully informed. Since Ag_2 needs no percepts, its local state and transition function are omitted.

The game executes as follows. A global state of C comprises a state $s_1 = (loc_1, per_1)$ for 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 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 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 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 $obs_1(loc_1, s_E) = f_{loc_1}^{\max}(s_E)$, where $f_{loc_1}^{\max}(s_E)$ 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 [30,29]. 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 [29]. 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 [29].

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.

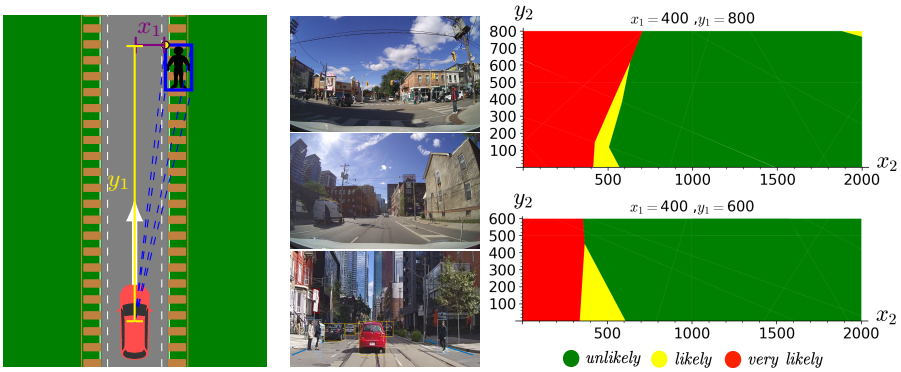


Fig. 1: Pedestrian-vehicle example. Left: Positions of two agents. Middle: Sample images from the PIE dataset [29]. Right: Slices of learnt perception function, where $(x_1, y_1), (x_2, y_2)$ are two successive (relative) positions of the pedestrian.

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 towards the roadside, 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 Appx. E.

One-sided NS-POSG semantics. A one-sided NS-POSG 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* 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 Ag_2 is fully informed, it can recover the beliefs of Ag_1, thus removing nested beliefs. Hence, the AOHs of 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 π of $[\![\mathsf{C}]\!]$, a stage strategy for Ag_1 is a distribution $u_1 \in \mathbb{P}(A_1)$ and a stage strategy for 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 Ag_1 is partially informed, it may need to infer the current state from its AOH. For an 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 Ag_1 are also the observations of Ag_1 and states of $[\![\mathsf{C}]\!]$ are percept compatible, a *belief* for Ag_1, which can also be reconstructed by 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 Ag_1.

Given a belief (s_1, b_1), if action a_1 is selected by Ag_1, 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 Ag_1 via Bayesian inference is denoted $(s_1', b_1^{s_1,a_1,u_2,s_1'})$; see Appx. A for details.

4 Values of One-Sided NS-POSGs

We establish the *value function* of a one-sided NS-POSG 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 β. The *value function* of 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 [14,2,31] 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 Appx. D.

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 $K(b_1, b_1') = \frac{1}{2}(U - L)\int_{s_E \in S_E^{s_1}} |b_1(s_E) - b_1'(s_E)| \mathrm{d}s_E$.*

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 over the beliefs S_B.

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 Γ 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 $s \in S$, we have the following:

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

where the sum over s_E' is due to the finite branching of $\delta(s,(a_1,a_2))$; see Defn. 2.

Definition 4 (Maxsup) *If $\Gamma \subseteq \mathbb{F}(S)$ and $V(s_1,b_1) = \sup_{\alpha \in \Gamma} \langle \alpha, (s_1,b_1) \rangle$ 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 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 Γ. The minimisation by 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 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). *If $\Gamma \subseteq \mathbb{F}(S)$ and $V(s_1,b_1) = \sup_{\alpha \in \Gamma} \langle \alpha, (s_1,b_1) \rangle$ for $(s_1,b_1) \in S_B$, then the minimax operator T and maxsup operator T_Γ 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) α-functions, called P-PWLC (*piecewise linear and convex under PWC*), originally introduced in [37]. This representation extends the α-functions of [28,6,39] 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 [28] 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 Φ, 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 Φ 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 Ag_1's perception function obs_1, it is PWC (as for the one-agent case [37]) 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* Φ_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 Φ_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 Φ_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 [37]). Informally, we require that, for any decomposition Φ' of the state-space into regions (i.e., an FCP), there is a second decomposition Φ, the *pre-image FCP*, such that states in regions of Φ have the same rewards and transition probabilities into regions of Φ'. The transitions of the (continuous) environment must also be decomposable into regions.

Assumption 1 (Transitions and rewards) *Given any FCP Φ' of S, there exists an FCP Φ of S, called the* pre-image FCP *of Φ', 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, δ_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 [37,28], that $V^\star$ is the limit of a sequence of α-functions, called *piecewise linear and convex under PWC α-functions*, first introduced in [37] for NS-POMDPs.

Definition 6 (P-PWLC function) *A function $V : S_B \to \mathbb{R}$ is* piecewise linear and convex under PWC α-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 Γ are called PWC α-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 $\langle \alpha, (s_1, b_1) \rangle = \int_{s_E \in S_E} \alpha(s_1, s_E) b_1(s_E) \mathrm{d}s_E$, so 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 (Lemma 5 in Appx. D), we first 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 α-function involved is PWC and thus allows for a finite representation. However, as the number of α-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) [33], 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. Rather than using PWLC functions for lower bounds as in [19], we switch to P-PWLC functions, resulting in different key ingredients to prove convergence. Value computations are also much more complex because the NN-based perception function induces a connected partition of regions (called 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 [37]. 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 α-functions requires closure properties with respect to the maxsup operator. Motivated by [37], 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 α-functions (see Definition 6), for which the

closure is guaranteed by Theorem 3. The lower bound V_{lb}^{Γ} has a finite representation as each α-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 [37], 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 Υ 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 for NS-POMDPs [37], K_{ub} here needs to satisfy an additional triangle property in (4) to ensure the continuity of V_{ub}^{Υ}, 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}^{Υ} is initialised as in [19].

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

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

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 Φ_P, Algorithm 1 computes a new α-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^ϕ is the unique agent state appearing in ϕ) 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 ϕ 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 ϕ 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 [37]. The details of the ISPP backup for one-sided NS-POSGs are in Appx. B. The lemma also shows that the lower bound function increases and is valid after each update.

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

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: $\Upsilon \leftarrow \Upsilon \cup \{((s_1, b_1), y^\star)\}$

Upper bound updates. For the upper bound V_{ub}^{Υ}, 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 Υ 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}^{Γ} 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^{Γ}_{lb} and V^{Υ}_{ub} via sets Γ and Υ
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* [28,10], which can approximate arbitrary beliefs and handle non-Gaussian systems. However, compared to region-based representations [37], it is more vulnerable to disturbances and can require many particles for good approximation.

Particle-based beliefs. A *particle-based belief* $(s_1, b_1) \in S_B$ is represented by a weighted particle set $\{(s_E^i, \kappa_i)\}_{i=1}^{N_b}$ with a normalised weight κ_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^{Γ}_{lb} and V^{Υ}_{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) (Appx. A).

Lower bound. Since V^{Γ}_{lb} is P-PWLC with PWC α-functions Γ, 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^{Υ}_{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 Appx. C).

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, α-functions and reward structures.

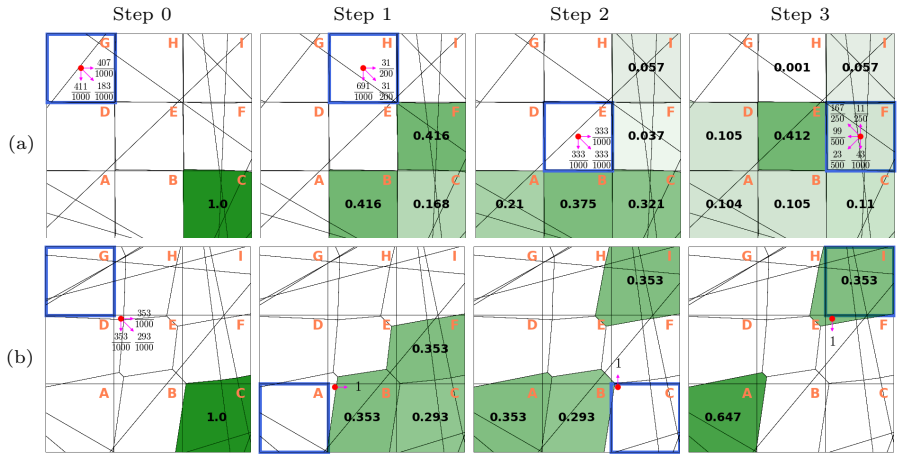


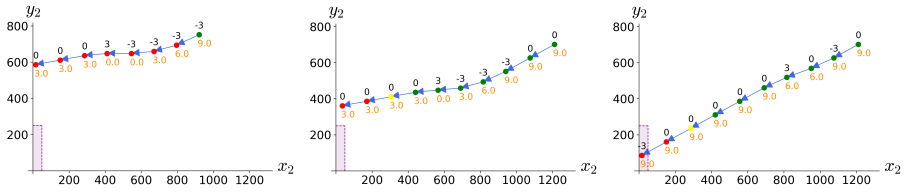
Fig. 2: Simulations of strategies for the pursuer, showing actual location (red), perceived location (blue), belief of evader location (green) and strategy (pink) for two different NN perception functions: (a) more precise; (b) coarser.

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 Appx. E for more details of the models, including the training of the ReLU NN classifiers, and emprirical 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 3×3 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 illustrate that our approach can be used to synthesise and explore non-trivial strategies for agents using NN-based perception in a partially observable setting. We can further study the impact of a poorly trained perception function. Fig. 2(b), for the coarser NN, shows that the pursuer repeatedly misdetects its location because the shapes of grid cells are poorly approximated, and subsequently takes incorrect actions. This is exploited by the evader, leading to considerably worse performance for the pursuer.



(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).

Pedestrian-vehicle interaction. Fig. 3 shows several simulations from strategies synthesised for the pedestrian-vehicle example described in Section 3 (Fig. 1), 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 use our approach to investigate the feasibility of generating strategies for agents deploying realistic NN-based perception functions. In this case, 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 a much later detection; (c) shows we were then unable to synthesise strategy that is always safe.

8 Conclusions

We 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 polyhedra operations. Nevertheless, our method provide 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 study restricted two-sided NS-POSGs, e.g., 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. Sci. Comput. Program. **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 Trans. 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. vol. 147, 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. v. Neumann, J.: Zur theorie der gesellschaftsspiele. Mathematische annalen **100**(1), 295–320 (1928)
28. Porta, J.M., Vlassis, N., Spaan, M.T., Poupart, P.: Point-based value iteration for continuous POMDPs. JMLR **7**, 2329–2367 (2006)
29. 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)
30. 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)
31. Saha, S.: Zero-sum stochastic games with partial information and average payoff. Journal of Optimization Theory and Applications **160**(1), 344–354 (2014)
32. Sion, M.: On general minimax theorems. Pacific J. Math. **8**(1), 171–176 (1958)
33. Smith, T., Simmons, R.: Heuristic search value iteration for POMDPs. In: Proc. UAI'04. p. 520–527. AUAI (2004)
34. 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)
35. Yan, R., Santos, G., Norman, G., Parker, D., Kwiatkowska, M.: Strategy synthesis for zero-sum neuro-symbolic concurrent stochastic games. arXiv.2202.06255, under revision for *Information & Computation* (2022)
36. 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)
37. Yan, R., Santos, G., Norman, G., Parker, D., Kwiatkowska, M.: Point-based value iteration for neuro-symbolic POMDPs. arXiv.2306.17639, under revision for *Artificial Intelligence Journal (AIJ)* (2023)
38. 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), to appear, available as arXiv.2404.10679
39. Zamani, Z., Sanner, S., Poupart, P., Kersting, K.: Symbolic dynamic programming for continuous state and observation POMDPs. Adv. Neural Inf. Process. Syst. **25** (2012)

40. Zettlemoyer, L., Milch, B., Kaelbling, L.: Multi-agent filtering with infinitely nested beliefs. Advances in neural information processing systems **21** (2008)
41. Zheng, W., Jung, T., Lin, H.: The Stackelberg equilibrium for one-sided zero-sum partially observable stochastic games. Automatica **140**, 110231 (2022)
42. 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)
43. Zinkevich, M., Johanson, M., Bowling, M., Piccione, C.: Regret minimization in games with incomplete information. Advances in neural information processing systems **20** (2007)

A Probability Measure Computations

The main paper omits details of how to compute several required quantities in terms of probability measures via closed forms. We provide these details below.

Belief updates. Section 3 (p. 6) discusses belief updates for agent Ag_1 of a one-sided NS-POSG. Given a belief (s_1, b_1), if action a_1 is selected by Ag_1, Ag_2 is *assumed* to take the stage strategy $u_2 \in \mathbb{P}(A_2 \mid S)$ and s_1' is observed, then the updated belief of Ag_1 via Bayesian inference is $(s_1', b_1^{s_1,a_1,u_2,s_1'})$ where for $s_E' \in S_E$:

$$b_1^{s_1,a_1,u_2,s_1'}(s_E') = \frac{P((s_1', s_E') \mid (s_1, b_1), a_1, u_2)}{P(s_1' \mid (s_1, b_1), a_1, u_2)} \text{ if } s_E' \in S_E^{s_1'} \text{ and 0 otherwise.} \quad (7)$$

On the other hand, if it is *assumed* that a joint action a is taken, then the updated belief of Ag_1 is $(s_1', b_1^{s_1,a,s_1'})$, where for $s_E' \in S_E$:

$$b_1^{s_1,a,s_1'}(s_E') = \frac{P((s_1', s_E') \mid (s_1, b_1), a)}{P(s_1' \mid (s_1, b_1), a)} \text{ if } s_E' \in S_E^{s_1'} \text{ and 0 otherwise.} \quad (8)$$

We now show how to compute the probability values given in the belief updates (7) and (8). Recalling that $s_1 = (loc_1, per_1)$, for (7), using the syntax in Definition 1, $P(s_1' \mid (s_1, b_1), a_1, u_2)$ equals

$$\int_{s_E \in S_E} b_1(s_E) \sum_{a_2 \in A_2} u_2(a_2 \mid s_1, s_E) \int_{s_E' \in S_E} \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \mathrm{d}s_E \quad (9)$$

and if $s_E' \in S_E^{s_1'}$, then $P((s_1', s_E') \mid (s_1, b_1), a_1, u_2)$ equals

$$\int_{s_E \in S_E} b_1(s_E) \sum_{a_2 \in A_2} u_2(a_2 \mid s_1, s_E) \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \mathrm{d}s_E \,.$$

For (8), we have that $P(s_1' \mid (s_1, b_1), a)$ equals

$$\int_{s_E \in S_E} b_1(s_E) \int_{s_E' \in S_E} \delta((s_1, s_E), a)(s_1', s_E') \mathrm{d}s_E$$

and if $s_E' \in S_E^{s_1'}$, then $P((s_1', s_E') \mid (s_1, b_1), a)$ equals

$$\int_{s_E \in S_E} b_1(s_E) \delta((s_1, s_E), a)(s_1', s_E') \mathrm{d}s_E \,.$$

Particle-based beliefs. Section 6.3 discusses computation of particle-based beliefs. For a particle-based belief (s_1, b_1) with weighted particle set $\{(s_E^i, \kappa_i)\}_{i=1}^{N_b}$, it follows from (7) that for belief $b_1^{s_1,a_1,u_2,s_1'}$ we have, for any $s_E' \in S_E$, that $b_1^{s_1,a_1,u_2,s_1'}(s_E')$ equals

$$\frac{\sum_{i=1}^{N_b} \kappa_i \left(\sum_{a_2 \in A_2} u_2(a_2 \mid s_1, s_E^i) \delta((s_1, s_E^i), (a_1, a_2))(s_1', s_E') \right)}{\sum_{i=1}^{N_b} \kappa_i \left(\sum_{a_2 \in A_2} u_2(a_2 \mid s_1, s_E^i) \left(\sum_{s_E'' \in S_E} \delta((s_1, s_E^i), (a_1, a_2))(s_1', s_E'') \right) \right)} \quad (10)$$

if $s_E' \in S_E^{s_1'}$ and equals 0 otherwise. Similarly, we can compute $\langle \alpha, (s_1, b_1) \rangle$, $\langle r, (s_1, b_1) \rangle$ and $P(a_1, s_1' \mid (s_1, b_1), u_1, u_2)$ as simple summations.

ALGORITHM 3 Image-Split-Preimage-Product (ISPP) backup over a region

Input: region ϕ, action $\overline{p}_1^\star$, PWC functions $\overline{\alpha}^\star$

1: $\bar{A}_1 \leftarrow \{a_1 \in A_1 \mid \overline{p}_1^\star(a_1) > 0\}$
2: $Loc'_a \leftarrow \{loc'_1 \in Loc_1 \mid \delta_1(s_1^\phi, a)(loc'_1) > 0\}$ for $a \in \bar{A}_1 \times A_2$, $\Phi_{\text{product}} \leftarrow \phi$
3: **for** $a = (a_1, a_2) \in \bar{A}_1 \times A_2, loc'_1 \in Loc'_a, i = 1, \ldots, N_e$ **do**
4: $\quad \phi'_E \leftarrow \{\delta_E^i(s_E, a) \mid (s_1^\phi, s_E) \in \phi\}$ ▷ Image
5: $\quad \Phi_{\text{image}} \leftarrow$ divide ϕ'_E into regions over S by $obs_1(loc'_1, \cdot)$
6: $\quad \Phi_{\text{split}} \leftarrow \varnothing$ ▷ Split
7: $\quad$ **for** $\phi_{\text{image}} \in \Phi_{\text{image}}$ **do**
8: $\quad\quad \Phi_\alpha \leftarrow$ a constant-FCP of S for the PWC function $\alpha^{\star a_1, s_1^{\phi_{\text{image}}}}$
9: $\quad\quad \Phi_{\text{split}} \leftarrow \Phi_{\text{split}} \cup \{\phi_{\text{image}} \cap \phi' \mid \phi' \in \Phi_\alpha\}$
10: $\quad \Phi_{\text{pre}} \leftarrow \varnothing$ ▷ Preimage
11: $\quad$ **for** $\phi_{\text{image}} \in \Phi_{\text{split}}$ **do**
12: $\quad\quad \Phi_{\text{pre}} \leftarrow \Phi_{\text{pre}} \cup \{(s_1^\phi, s_E) \in \phi \mid \delta_E^i(s_E, a) \in \phi_{\text{image}}\}$
13: $\quad \Phi_{\text{product}} \leftarrow \{\phi_1 \cap \phi_2 \mid \phi_1 \in \Phi_{\text{pre}} \wedge \phi_2 \in \Phi_{\text{product}}\}$ ▷ Product
14: $\Phi_{\text{product}} \leftarrow \{\phi_1 \cap \phi_2 \mid \phi_1 \in \Phi_{\text{product}} \wedge \phi_2 \in \sum_{a \in \bar{A}_1 \times A_2} \Phi_R^a\}$
15: **for** $\phi_{\text{product}} \in \Phi_{\text{product}}$ **do** ▷ Value backup
16: $\quad$ Take one state $(\hat{s}_1, \hat{s}_E) \in \phi_{\text{product}}$
17: $\quad \alpha^\star(\phi_{\text{product}}) \leftarrow f_{\overline{p}_1^\star, \overline{\alpha}^\star}(\hat{s}_1, \hat{s}_E)$
18: **return:** $(\Phi_{\text{product}}, \alpha^\star)$

B Image-Split-Preimage-Product (ISPP) Backup

We provide here the Image-Split-Preimage-Product (ISPP) backup for one-sided NS-POSGs, adapted from the single-agent variant in [37], as used for a region-by-region backup in line 4 of Algorithm 1 (Section 6.1).

For FCPs Φ_1 and Φ_2 of S, we denote by $\Phi_1 + \Phi_2$ the smallest FCP of S 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. We call the FCP Φ in Definition 5 the *constant-FCP* of S for a PWC function $f \in \mathbb{F}_C(S)$. Recall from Assumption 1 that δ_E can be represented as $\sum_{i=1}^{N_e} \mu_i \delta_E^i$.

Algorithm 3 shows the ISPP backup method. This method, inspired by Lemma 2, is to divide a region ϕ into subregions where for each subregion $\alpha^\star$ is constant. Given any reachable local state loc'_1 under a and continuous transition function δ_E^i, the *image* of ϕ under a and δ_E^i to loc'_1 is divided into *image* regions Φ_{image} such that the states in each region have a unique agent state. Each image region ϕ_{image} is then split into subregions by a constant-FCP of the PWC function $\alpha^{a_1, s_1^{\phi_{\text{image}}}}$ by pairwise intersections where $a = (a_1, a_2)$, and thus Φ_{image} is *split* into a set of refined image regions Φ_{split}. An FCP over ϕ, denoted by Φ_{pre}, is constructed by computing the *pre-image* of each $\phi_{\text{image}} \in \Phi_{\text{split}}$ to ϕ. Finally, the *product* of these FCPs Φ_{pre} for all reachable local states and environment functions and reward FCPs $\{\Phi_R^a \mid a \in \bar{A}_1 \times A_2\}$, denoted Φ_{product}, is computed. The following lemma demonstrates that $\alpha^\star$ is constant in each region of Φ_{product}, and therefore that line 4 of Algorithm 1 can be computed by finite backups.

Lemma 4 (ISPP backup) *The FCP Φ_{product} returned by Algorithm 3 is a constant-FCP of ϕ for $\alpha^\star$ and the region-by-region backup for α^* satisfies the line 4 of Algorithm 1.*

Proof. For the PWC α-functions in the input of Algorithm 3, if $\Phi_{a_1,s_1'}$ is an FCP of S for $\alpha^{a_1,s_1'}$, then let $\Phi = \sum_{a_1\in\bar{A}_1,s_1'\in S_1}\Phi_{a_1,s_1'}$, i.e., Φ is the smallest refinement of these FCPs.

According to Assumption 1, there exists a preimage-FCP of Φ for each joint action a. Through the image, split, pre-image and product operations of Algorithm 3, all the states in any region $\phi' \in \Phi_{\text{product}}$ reach the same regions of Φ. Since each α-function $\alpha^{a_1,s_1'}$ is constant over each region in Φ, all states in ϕ' have the same backup value from $\alpha^{a_1,s_1'}$ for $a_1 \in \bar{A}_1$ and $s_1' \in S_1$. This implies that Φ_{product} is the product of the preimage-FCPs of Φ for all $a \in \bar{A}_1 \times A_2$. Since the value backup in line 4 of Algorithm 1 is used for each region in Φ_{product} and the image is from the region ϕ, then Φ_{product} is a constant-FCP of ϕ for $\alpha^\star$, and thus the value backup in line 4 of Algorithm 1 for $\alpha^\star$ is achieved by considering the regions of Φ_{product}. □

C Linear Programs

We provide some linear programs (LPs) and their dual versions, omitted for space reasons in the main paper, in particular for the stage games $[TV_{lb}^{\Gamma}](s_1,b_1)$ and $[TV_{ub}^{\Upsilon}](s_1,b_1)$. Consider a particle-based belief (s_1,b_1) represented by $\{(s_E^i,\kappa_i)\}_{i=1}^{N_b}$.

Stage game over the lower bound. Using Lemma 1 and its extended version Lemma 6, the LP (33) for the stage game $[TV_{lb}^{\Gamma}](s_1,b_1)$ is simplified to the LP over the variables:

- $(v_{s_E^i})_{i=1}^{N_b}$;
- $(\lambda_\alpha^{a_1,s_1'})_{(a_1,s_1')\in A_1\times S_1,\alpha\in\Gamma}$;
- $(p^{a_1})_{a_1\in A_1}$;

and is given by maximise $\sum_{i=1}^{N_b}\kappa_i v_{s_E^i}$ subject to:

$$\begin{aligned} v_{s_E^i} &\le \textstyle\sum_{a_1\in A_1} p^{a_1} r((s_1,s_E^i),(a_1,a_2)) + \beta\sum_{(a_1,s_1')\in A_1\times S_1,s_E'\in S_E} \\ &\qquad \delta((s_1,s_E^i),(a_1,a_2))(s_1',s_E')\textstyle\sum_{\alpha\in\Gamma}\lambda_\alpha^{a_1,s_1'}\alpha(s_1',s_E') \\ \lambda_\alpha^{a_1,s_1'} &\ge 0 \\ p^{a_1} &= \textstyle\sum_{\alpha\in\Gamma}\lambda_\alpha^{a_1,s_1'} \\ \textstyle\sum_{a_1\in A_1} p^{a_1} &= 1 \end{aligned} \tag{11}$$

for all $1 \le i \le N_b$, $a_2 \in A_2$, $(a_1,s_1') \in A_1 \times S_1$ and $\alpha \in \Gamma$.

The dual of LP problem (11) is over the variables:

- v;

- $(v_{a_1,s_1'})_{(a_1,s_1')\in A_1\times S_1}$;
- $(p_{a_2}^{s_1,s_E^i})_{a_2\in A_2,1\le i\le N_b}$;

and is given by minimise v subject to:

$$\begin{aligned} v &\ge \textstyle\sum_{i=1}^{N_b}\sum_{a_2\in A_2} p_{a_2}^{s_1,s_E^i} r((s_1,s_E^i),(a_1,a_2)) + \beta\sum_{s_1'\in S_1} v_{a_1,s_1'} \\ v_{a_1,s_1'} &\ge \textstyle\sum_{i=1}^{N_b}\sum_{a_2\in A_2} p_{a_2}^{s_1,s_E^i}\delta((s_1,s_E^i),(a_1,a_2))(s_1',s_E')\alpha(s_1',s_E') \\ \textstyle\sum_{a_2\in A_2} p_{a_2}^{s_1,s_E^i} &= \kappa_i \end{aligned} \tag{12}$$

for all $a_1\in A_1$, $(a_1,s_1')\in A_1\times S_1$, $\alpha\in\Gamma$ and $1\le i\le N_b$.

By solving (11) and (12), we obtain the minimax strategy profile in the stage game $[TV_{lb}^{\Gamma}](s_1,b_1)$: $u_1^{lb}(a_1)=p^{\star a_1}$ for $a_1\in A_1$ and $u_2^{lb}(a_2\mid s_1,s_E^i)=p_{a_2}^{\star s_1,s_E^i}/\kappa_i$ for $1\le i\le N_b$ and $a_2\in A_2$.

Stage game over the upper bound. The LP for the stage game $[TV_{ub}^{\Upsilon}](s_1,b_1)$ is over the variables:

- v;
- $(c_{s_E'}^{a_1,s_1'})_{(a_1,s_1')\in A_1\times S_1\wedge s_E'\in S_E^{a_1,s_1'}}$;
- $(\lambda_k^{a_1,s_1'})_{(a_1,s_1')\in A_1\times S_1,k\in I_{s_1'}}$;
- $(p_{a_2}^{s_1,s_E^i})_{1\le i\le N_b,a_2\in A_2}$

and is given by minimise v subject to:

$$\begin{aligned} v &\ge \textstyle\sum_{i=1}^{N_b}\sum_{a_2\in A_2}\kappa_i p_{a_2}^{s_1,s_E^i} r((s_1,s_E^i),(a_1,a_2)) \\ &\quad + \beta\textstyle\sum_{s_1'\in S_1}\sum_{k\in I_{s_1'}}\lambda_k^{a_1,s_1'}y_k + \frac{1}{2}\beta(U-L)\sum_{s_1'\in S_1}\sum_{s_E'\in S_E^{a_1,s_1'}} c_{s_E'}^{a_1,s_1'} \\ c_{s_E'}^{a_1,s_1'} &\ge \Big|\textstyle\sum_{i=1}^{N_b}\sum_{a_2\in A_2}\kappa_i p_{a_2}^{s_1,s_E^i}\delta((s_1,s_E^i),(a_1,a_2))(s_1',s_E') \\ &\quad - \textstyle\sum_{k\in I_{s_1'}}\lambda_k^{a_1,s_1'}P(s_E';b_1^k)\Big| \\ \textstyle\sum_{k\in I_{s_1'}}\lambda_k^{a_1,s_1'} &= \textstyle\sum_{i=1}^{N_b}\sum_{a_2\in A_2,s_E'\in S_E}\kappa_i p_{a_2}^{s_1,s_E^i}\delta((s_1,s_E^i),(a_1,a_2))(s_1',s_E') \\ \lambda_k^{a_1,s_1'} &\ge 0 \\ p_{a_2}^{s_1,s_E^i} &\ge 0 \\ \textstyle\sum_{a_2\in A_2} p_{a_2}^{s_1,s_E^i} &= 1 \end{aligned} \tag{13}$$

for all $a_1\in A_1$, $(a_1,s_1')\in A_1\times S_1$ and $s_E'\in S_E^{a_1,s_1'}$, $k\in I_{s_1'}$, $a_2\in A_2$ and $1\le i\le N_b$ where $S_E^{a_1,s_1'}=\{s_E'\in S_E\mid \sum_{a_2\in A_2} b_1^{s_1,a_1,a_2,s_1'}(s_E')+\sum_{k\in I_{s_1'}} b_1^k(s_E')>0\}$.

The dual of LP problem (13) is the following LP problem over the variables:

- $(v_{s_E^i})_{1\leq i\leq N_b}$;
- $(v_{a_1,s_1'})_{(a_1,s_1')\in A_1\times S_1}$;
- $(p^{a_1})_{a_1\in A_1}$;
- $(d_{a_1,s_1',s_E'})_{(a_1,s_1')\in A_1\times S_1\wedge s_E'\in S_E^{a_1,s_1'}}$;
- $(e_{a_1,s_1',s_E'})_{(a_1,s_1')\in A_1\times S_1\wedge s_E'\in S_E^{a_1,s_1'}}$;

and is given by maximise $\sum_{i=1}^{N_b}\kappa_i v_{s_E^i}$ subject to:

$$\begin{aligned}
v_{s_E^i} &\leq \textstyle\sum_{a_1\in A_1} p^{a_1} r((s_1,s_E^i),(a_1,a_2)) + \beta\sum_{a_1\in A_1, s_1'\in S_1, s_E'\in S_E^{a_1,s_1'}} \\
&\qquad \delta((s_1,s_E^i),(a_1,a_2))(s_1',s_E')(v_{a_1,s_1'} + d_{a_1,s_1',s_E'} - e_{a_1,s_1',s_E'}) \\
v_{a_1,s_1'} &\leq y_k p^{a_1} - \textstyle\sum_{s_E'\in S_E^{a_1,s_1'}} (d_{a_1,s_1',s_E'} - e_{a_1,s_1',s_E'}) P(s_E'; b_1^k) \\
d_{a_1,s_1',s_E'} - e_{a_1,s_1',s_E'} &\leq \tfrac{1}{2}(U-L) \\
d_{a_1,s_1',s_E'} &\geq 0 \\
e_{a_1,s_1',s_E'} &\geq 0 \\
p^{a_1} &\geq 0 \\
\textstyle\sum_{a_1\in A_1} p^{a_1} &= 1
\end{aligned} \tag{14}$$

for all $a_2\in A_2$ and $1\leq i\leq N_b$, $(a_1,s_1')\in A_1\times S_1$, $k\in I_{s_1'}$ and $s_E'\in S_E^{a_1,s_1'}$ where $S_E^{a_1,s_1'} = \{s_E'\in S_E \mid \exists 1\leq i\leq N_b.\, \exists a_2\in A_2.\, \delta((s_1,s_E^i),(a_1,a_2))(s_1',s_E')>0\}$.

By solving (13) and (14), we obtain the minimax strategy profile in stage game $[TV_{ub}^{\Upsilon}](s_1,b_1)$: $u_1^{ub}(a_1) = p^{\star a_1}$ for $a_1\in A_1$ and $u_2^{ub}(a_2 \mid s_1, s_E^i) = p_{a_2}^{\star s_1,s_E^i}$ for $1\leq i\leq N_b$ and $a_2\in A_2$.

D Proofs of Main Results

We provide here the proofs of the results from the main paper.

*Proof (**Proof of Theorem 1**).* Given $s_1\in S_1$, we first prove that $V^\star(s_1,\cdot)$ is convex and continuous. For any $b_1\in\mathbb{P}(S_E)$, since $V^\star(s_1,b_1)$ is the lower value of Y, then $V^\star(s_1,b_1) = \sup_{\sigma_1\in\Sigma_1}\inf_{\sigma_2\in\Sigma_2}\mathbb{E}^{\sigma_1,\sigma_2}_{(s_1,b_1)}[Y]$. We define a payoff function $V_{\sigma_1}:\mathbb{P}(S_E)\to\mathbb{R}$ to be the objective of the sup optimisation in the lower value such that for $b_1\in\mathbb{P}(S_E)$ we have $V_{\sigma_1}(s_1,b_1) = \inf_{\sigma_2\in\Sigma_2}\mathbb{E}^{\sigma_1,\sigma_2}_{(s_1,b_1)}[Y]$. Note that the value $V_{\sigma_1}(s_1,b_1)$ is the expected reward of σ_1 against the best-response strategy σ_2, from the initial belief (s_1,b_1). Since Ag_2 can observe the true initial state (s_1,s_E) where s_E is sampled from b_1, and thus can play a state-wise best-response to each initial state (s_1,s_E), the value $V_{\sigma_1}(s_1,b_1)$ can be rewritten as:

$$V_{\sigma_1}(s_1,b_1) = \textstyle\int_{s_E\in S_E} b_1(s_E)\big(\inf_{\sigma_2\in\Sigma_2}\mathbb{E}^{\sigma_1,\sigma_2}_{(s_1,s_E)}[Y]\big)\mathrm{d}s_E\,. \tag{15}$$

Thus, $V_{\sigma_1}(s_1, \cdot)$ is a linear function in the belief $b_1 \in \mathbb{P}(S_E)$. Since $V^\star(s_1, b_1) = \sup_{\sigma_1 \in \Sigma_1} V_{\sigma_1}(s_1, b_1)$ and any point-wise supremum of linear functions is convex and continuous (it follows from the convexity and continuity in the discrete case, see [19, Proposition 5.9]), we can conclude that $V^\star(s_1, \cdot)$ is convex and continuous.

Regarding the inequality in Theorem 1, for any $b_1, b_1' \in \mathbb{P}(S_E)$, we have:

$$\int_{s_E \in S_E^{s_1}} b_1(s_E) \mathrm{d}s_E = \int_{s_E \in S_E^{s_1}} b_1'(s_E) \mathrm{d}s_E = 1\,. \tag{16}$$

Now, letting $S_E^{>} = \{s_E \in S_E^{s_1} \mid b_1(s_E) - b_1'(s_E) > 0\}$ and $S_E^{\leq} = \{s_E \in S_E^{s_1} \mid b_1(s_E) - b_1'(s_E) \leq 0\}$, rearranging (16) and using the fact that $S_E^{>} \cup S_E^{\leq} = S_E^{s_1}$ it follows that:

$$\int_{s_E \in S_E^{\leq}} (b_1(s_E) - b_1'(s_E)) \mathrm{d}s_E = -\int_{s_E \in S_E^{>}} (b_1(s_E) - b_1'(s_E)) \mathrm{d}s_E$$

from which we have:

$$\begin{aligned}
&\int_{s_E \in S_E^{s_1}} |b_1(s_E) - b_1'(s_E)| \mathrm{d}s_E = \int_{s_E \in S_E^{>} \cup S_E^{\leq}} |b_1(s_E) - b_1'(s_E)| \mathrm{d}s_E \\
&= \int_{s_E \in S_E^{>}} (b_1(s_E) - b_1'(s_E)) \mathrm{d}s_E - \int_{s_E \in S_E^{\leq}} (b_1(s_E) - b_1'(s_E)) \mathrm{d}s_E \\
&= 2 \int_{s_E \in S_E^{>}} (b_1(s_E) - b_1'(s_E)) \mathrm{d}s_E
\end{aligned} \tag{17}$$

and thus, using (17) and [37, Theorem 2], the inequality in Theorem 1 holds. □

Theorem 6 (Operator equivalence and fixed point, extended version of Theorem 2). *If $\Gamma \subseteq \mathbb{F}(S)$ and $V(s_1, b_1) = \sup_{\alpha \in \Gamma} \langle \alpha, (s_1, b_1) \rangle$ for $(s_1, b_1) \in S_B$, then the minimax operator T and maxsup operator T_Γ are equivalent, i.e., for $(s_1, b_1) \in S_B$ we have:*

$$\begin{aligned}
&[TV](s_1, b_1) = \max_{u_1 \in \mathbb{P}(A_1)} \min_{u_2 \in \mathbb{P}(A_2 | S)} \mathbb{E}_{(s_1, b_1), u_1, u_2}[r(s, a)] \\
&\quad + \beta \sum_{a_1 \in A_1} \sum_{s_1' \in S_1} P((a_1, s_1') \mid (s_1, b_1), u_1, u_2) V(s_1', b_1^{s_1, a_1, u_2, s_1'}) && (18) \\
&= \min_{u_2 \in \mathbb{P}(A_2 | S)} \max_{u_1 \in \mathbb{P}(A_1)} \mathbb{E}_{(s_1, b_1), u_1, u_2}[r(s, a)] \\
&\quad + \beta \sum_{a_1 \in A_1} \sum_{s_1' \in S_1} P((a_1, s_1') \mid (s_1, b_1), u_1, u_2) V(s_1', b_1^{s_1, a_1, u_2, s_1'}) && (19) \\
&= \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \langle f_{u_1, \overline{\alpha}}, (s_1, b_1) \rangle && (20) \\
&= [T_\Gamma V](s_1, b_1)\,.
\end{aligned}$$

Moreover, the unique fixed point of T and T_Γ is $V^\star$.

Proof. Considering any $V \in \mathbb{F}(S_B)$ and $\Gamma \subseteq \mathbb{F}(S)$ such that:

$$V(s_1, b_1) = \sup_{\alpha \in \Gamma} \langle \alpha, (s_1, b_1) \rangle \quad \text{for all } (s_1, b_1) \in S_B. \tag{21}$$

Operator equivalence. We first show that the operators T and T_Γ are equivalent. We first define a payoff function $J : \mathbb{P}(A_1) \times \mathbb{P}(A_2 \mid S) \to \mathbb{R}$ to be the

objective of the maximin and minimax optimisation in (18) and (19) such that for $u_1 \in \mathbb{P}(A_1)$ and $u_2 \in \mathbb{P}(A_2 \mid S)$:

$$J(u_1,u_2) \;=\; \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)]+ \\ \beta\textstyle\sum_{a_1\in A_1}\sum_{s_1'\in S_1} P(a_1,s_1' \mid (s_1,b_1),u_1,u_2)V(s_1',b_1^{s_1,a_1,u_2,s_1'})\,. \tag{22}$$

Now for any belief $(s_1,b_1) \in S_B$ such that $s_1 = (loc_1, per_1)$, action $a_1 \in A_1$, agent state $s_1' \in S_1$ and stage strategy $u_2 \in \mathbb{P}(A_2 \mid S)$, letting $P_1 \triangleq P(s_1' \mid (s_1,b_1),a_1,u_2)$ by (21) we have:

$$\begin{aligned}
&V(s_1',b_1^{s_1,a_1,u_2,s_1'}) \;=\; \sup_{\alpha\in\Gamma}\langle\alpha,(s_1',b_1^{s_1,a_1,u_2,s_1'})\rangle \\
&= \sup_{\alpha\in\Gamma}\textstyle\int_{s_E'\in S_E}\alpha(s_1',s_E')b_1^{s_1,a_1,u_2,s_1'}(s_E')\mathrm{d}s_E' && \text{rearranging} \\
&= \sup_{\alpha\in\Gamma}\textstyle\int_{s_E'\in S_E}\alpha(s_1',s_E')\dfrac{P((s_1',s_E') \mid (s_1,b_1),a_1,u_2)}{P(s_1' \mid (s_1,b_1),a_1,u_2)}\mathrm{d}s_E' && \text{by (7)} \\
&= \frac{1}{P_1}\sup_{\alpha\in\Gamma}\textstyle\int_{s_E'\in S_E}\alpha(s_1',s_E')P((s_1',s_E') \mid (s_1,b_1),a_1,u_2)\mathrm{d}s_E' && \text{rearranging} \\
&= \frac{1}{P_1}\sup_{\alpha\in\Gamma}\Big(\textstyle\int_{s_E'\in S_E}\alpha(s_1',s_E')\int_{s_E'\in S_E^{s_1'}\wedge s_E\in S_E} b_1(s_E)\sum_{a_2\in A_2}u_2(a_2 \mid s_1,s_E) \\
&\qquad\cdot\delta((s_1,s_E),(a_1,a_2))(s_1',s_E')\mathrm{d}s_E\Big)\mathrm{d}s_E' && \text{by (9)} \\
&= \frac{1}{P_1}\sup_{\alpha\in\Gamma}\Big(\textstyle\int_{s_E\in S_E}\Big(\int_{s_E'\in S_E^{s_1'}}\alpha(s_1',s_E')\sum_{a_2\in A_2}u_2(a_2 \mid s_1,s_E) \\
&\qquad\cdot\delta((s_1,s_E),(a_1,a_2))(s_1',s_E')\mathrm{d}s_E'\Big)b_1(s_E)\mathrm{d}s_E && \text{rearranging.}
\end{aligned} \tag{23}$$

Next, for any $\alpha \in \mathbb{F}(S)$, $s_1' \in S_1$, $a_1 \in A_1$ and $u_2 \in \mathbb{P}(A_2 \mid S)$ we let $\alpha^{a_1,u_2,s_1'} : S \to \mathbb{R}$ be the function where for any $s = ((loc_1, per_1), s_E) \in S$:

$$\begin{aligned}
\alpha^{a_1,u_2,s_1'}(s) &= \textstyle\int_{s_E'\in S_E^{s_1'}}\alpha(s_1',s_E')\sum_{a_2}u_2(a_2 \mid s)\delta(s,(a_1,a_2))(s_1',s_E')\mathrm{d}s_E' \\
&= \textstyle\sum_{a_2}u_2(a_2 \mid s)\sum_{s_E'\in S_E}\delta(s,(a_1,a_2))(s_1',s_E')\alpha(s_1',s_E')
\end{aligned} \tag{24}$$

and the summation in s_E' is due to the finite branching of δ. Combining (23) and (24) we have:

$$\begin{aligned}
V(s_1',b_1^{s_1,a_1,u_2,s_1'}) &= \frac{1}{P_1}\sup_{\alpha\in\Gamma}\textstyle\int_{s_E\in S_E}\alpha^{a_1,u_2,s_1'}(s_1,s_E)b_1(s_E)\mathrm{d}s_E \\
&= \frac{1}{P(s_1' \mid (s_1,b_1),a_1,u_2)}\sup_{\alpha\in\Gamma}\langle\alpha^{a_1,u_2,s_1'},(s_1,b_1)\rangle
\end{aligned} \tag{25}$$

by definition of P_1. Substituting (25) into (22), the payoff function $J(u_1,u_2)$ equals:

$$\begin{aligned}
&\mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] + \beta\textstyle\sum_{a_1,s_1'}u_1(a_1)P(s_1' \mid (s_1,b_1),a_1,u_2)V(s_1',b_1^{s_1,a_1,u_2,s_1'}) \\
&= \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] + \beta\textstyle\sum_{a_1,s_1'}u_1(a_1)\sup_{\alpha\in\Gamma}\langle\alpha^{a_1,u_2,s_1'},(s_1,b_1)\rangle\,.
\end{aligned} \tag{26}$$

We next show that the von Neumann's Minimax Theorem [27] applies to the game $[\![\mathsf{C}]\!]$ with the payoff function J and strategy spaces $\mathbb{P}(A_1)$ and $\mathbb{P}(A_2 \mid S)$. This theorem requires that $\mathbb{P}(A_1)$ and $\mathbb{P}(A_2 \mid S)$ are compact convex sets (which is straightforward to show) and that J is a continuous function that is concave-convex, i.e.,

- $J(\cdot, u_2)$ is concave for fixed $u_2 \in \mathbb{P}(A_2 \mid S)$;
- $J(u_1, \cdot)$ is convex for fixed $u_1 \in \mathbb{P}(A_1)$.

By Definition 3 the expectation $\mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)]$ can be rewritten as:

$$\sum_{a_1} u_1(a_1) \int_{s_E \in S_E} b_1(s_E) \sum_{a_2} u_2(a_2 \mid s_1, s_E) r((s_1, s_E), (a_1, a_2)) \mathrm{d}s_E$$

and thus, $\mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)]$ is bilinear in u_1 and u_2, and thus concave in $\mathbb{P}(A_1)$ and convex in $\mathbb{P}(A_2 \mid S)$.

We next show that $u_1(a_1) \sup_{\alpha \in \Gamma} \langle \alpha^{a_1,u_2,s_1'}, (s_1, b_1) \rangle$ is continuous and concave in $u_1 \in \mathbb{P}(A_1)$ and convex in $u_2 \in \mathbb{P}(A_2 \mid S)$. The continuity and concavity in $u_1 \in \mathbb{P}(A_1)$ follows directly as it is linear in $u_1 \in \mathbb{P}(A_1)$. For $u_2 \in \mathbb{P}(A_2 \mid S)$, we consider the function $f(u_2) = \langle \alpha^{a_1,u_2,s_1'}, (s_1, b_1) \rangle$. By (24) we have that $f(u_2)$ equals:

$$\int_{s_E \in S_E} \sum_{a_2} u_2(a_2 \mid s_1, s_E) \sum_{s_E' \in S_E} \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \alpha(s_1', s_E') b_1(s_E) \mathrm{d}s_E$$

and therefore $f(u_2)$ is linear in u_2. Since the point-wise maximum over linear functions is continuous and convex, it follows that $\sup_{\alpha \in \Gamma} f(u_2)$ is continuous and convex in $u_2 \in \mathbb{P}(A_2 \mid S)$, and hence $u_1(a_1) \sup_{\alpha \in \Gamma} \langle \alpha^{a_1,u_2,s_1'}, (s_1, b_1) \rangle$ is continuous and convex in $u_2 \in \mathbb{P}(A_2 \mid S)$. According to von Neumann's Minimax theorem:

$$\max_{u_1 \in \mathbb{P}(A_1)} \min_{u_2 \in \mathbb{P}(A_2|S)} J(u_1, u_2) = \min_{u_2 \in \mathbb{P}(A_2|S)} \max_{u_1 \in \mathbb{P}(A_1)} J(u_1, u_2)$$

and hence the equality between (18) and (19) holds.

Next we prove the equality of (18) and (20). Letting $\mathrm{Conv}(\Gamma)$ be the convex hull of Γ, recall that $\Gamma^{A_1 \times S_1}$ is the set of vectors of functions in $\mathrm{Conv}(\Gamma)$ indexed by the elements of $A_1 \times S_1$. The function $J(u_1, u_2)$ in (26) can be rewritten as follows:

$$\sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \Big(\mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] + \beta \sum_{a_1 \in A_1, s_1' \in S_1} u_1(a_1) \langle \alpha^{a_1,u_2,s_1'}, (s_1, b_1) \rangle \Big) \tag{27}$$

where $\bar{\alpha} = (\alpha^{a_1,s_1'})_{a_1 \in A_1, s_1' \in S_1}$, and given u_1 and u_2, the supremum over Γ only depends on a_1 and s_1' and using the same arguments as [19, Proposition 4.11] we have:

$$\sup_{\alpha \in \Gamma} \langle \alpha, (s_1, b_1) \rangle = \sup_{\alpha \in \mathrm{Conv}(\Gamma)} \langle \alpha, (s_1, b_1) \rangle$$

for $(s_1, b_1) \in S_B$. We next define the game with strategy spaces $\Gamma^{A_1 \times S_1}$ and $\mathbb{P}(A_2 \mid S)$ and payoff function $J_{u_1} : \Gamma^{A_1 \times S_1} \times \mathbb{P}(A_2 \mid S) \to \mathbb{R}$ where for $\overline{\alpha} \in \Gamma^{A_1 \times S_1}$ and $u_2 \in \mathbb{P}(A_2 \mid S)$:

$$J_{u_1}(\overline{\alpha}, u_2) = \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] + \beta \sum_{a_1 \in A_1, s_1' \in S_1} u_1(a_1) \langle \alpha^{a_1,u_2,s_1'}, (s_1, b_1) \rangle$$

$$= \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] + \beta \sum_{a_1 \in A_1, s_1' \in S_1} u_1(a_1) \int_{s_E \in S_E} \Big(\sum_{a_2 \in A_2} u_2(a_2 \mid s_1, s_E) \cdot \sum_{s_E' \in S_E} \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \alpha^{a_1, s_1'}(s_1', s_E') \Big) b_1(s_E) \mathrm{d}s_E \quad \text{by (24).} \quad (28)$$

Substituting (27) and (28) into (18) we have:

$$\max_{u_1 \in \mathbb{P}(A_1)} \min_{u_2 \in \mathbb{P}(A_2|S)} J(u_1, u_2) = \max_{u_1 \in \mathbb{P}(A_1)} \min_{u_2 \in \mathbb{P}(A_2|S)} \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} J_{u_1}(\overline{\alpha}, u_2)\,. \quad (29)$$

We next show that Sion's Minimax Theorem [32] applies to the game with strategy spaces $\Gamma^{A_1 \times S_1}$ and $\mathbb{P}(A_2 \mid S)$ and payoff function J_{u_1}. Sion's Minimax Theorem requires that:

- $\Gamma^{A_1 \times S_1}$ is convex;
- $\mathbb{P}(A_2 \mid S)$ is compact and convex;
- for any $u_2 \in \mathbb{P}(A_2 \mid S)$ the function $J_{u_1}(\cdot, u_2) : \Gamma^{A_1 \times S_1} \to \mathbb{R}$ is upper semicontinuous and quasi-concave;
- for any $\overline{\alpha} \in \Gamma^{A_1 \times S_1}$ the function $J_{u_1}(\overline{\alpha}, \cdot) : \mathbb{P}(A_2 \mid S) \to \mathbb{R}$ is lower semicontinuous and quasi-convex.

The first properties clearly hold and the second to follow from (28) which demonstrate that both $J_{u_1}(\cdot, u_2)$ and $J_{u_1}(\overline{\alpha}, \cdot)$ are linear.

Therefore using Sion's Minimax Theorem, we have:

$$\min_{u_2 \in \mathbb{P}(A_2|S)} \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} J_{u_1}(\overline{\alpha}, u_2) = \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \min_{u_2 \in \mathbb{P}(A_2|S)} J_{u_1}(\overline{\alpha}, u_2)$$

and combining with (29) it follows that $\max_{u_1 \in \mathbb{P}(A_1)} \min_{u_2 \in \mathbb{P}(A_2|S)} J(u_1, u_2)$ equals:

$$\begin{aligned}
&\max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \min_{u_2 \in \mathbb{P}(A_2|S)} J_{u_1}(\overline{\alpha}, u_2) \\
&= \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \min_{u_2 \in \mathbb{P}(A_2|S)} \int_{s_E \in S_E} \sum_{a_2} u_2(a_2 \mid s_1, s_E) \sum_{a_1} u_1(a_1) \\
&\quad \cdot r((s_1, s_E), (a_1, a_2)) b_1(s_E) \mathrm{d}s_E + \beta \int_{s_E \in S_E} \Big(\sum_{a_2} u_2(a_2 \mid s_1, s_E) \sum_{a_1, s_1'} u_1(a_1) \\
&\quad \cdot \sum_{s_E' \in S_E} \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \alpha(s_1', s_E') \Big) b_1(s_E) \mathrm{d}s_E \qquad \text{by (28)} \\
&= \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \int_{s_E \in S_E} \min_{u_2 \in \mathbb{P}(A_2|S)} \sum_{a_2} u_2(a_2 \mid s_1, s_E) \\
&\quad \Big(\sum_{a_1} u_1(a_1) r((s_1, s_E), (a_1, a_2)) + \beta \sum_{a_1, s_1'} u_1(a_1) \\
&\quad \sum_{s_E' \in S_E} \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \alpha(s_1', s_E') \Big) b_1(s_E) \mathrm{d}s_E \qquad \text{rearranging} \\
&= \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \int_{s_E \in S_E} \min_{a_2 \in A_2} \Big(\sum_{a_1} u_1(a_1) r((s_1, s_E), (a_1, a_2)) \\
&\quad + \beta \sum_{a_1, s_1'} u_1(a_1) \sum_{s_E' \in S_E} \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \alpha(s_1', s_E') \Big) b_1(s_E) \mathrm{d}s_E \\
&\qquad \text{since } \mathsf{Ag}_2 \text{ is fully informed} \\
&= \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \int_{s_E \in S_E} \big(\min_{a_2 \in A_2} f_{u_1, \overline{\alpha}, a_2}(s_1, s_E) \big) b_1(s_E) \mathrm{d}s_E \\
&\qquad \text{by (2)} \\
&= \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \langle f_{u_1, \overline{\alpha}}, (s_1, b_1) \rangle \qquad \text{by Definition 4}
\end{aligned}$$

which demonstrates that (18) and (20) are equal, i.e., T and T_Γ are equivalent.

Fixed point. To show the unique fixed point of T and T_Γ is $V^\star$. We first prove that $V^\star$ is a fixed point of the operator T, i.e., $V^\star = [TV^\star]$. According to the proof of Theorem 1, for $(s_1, b_1) \in S_B$ the value function $V^\star$ can be represented by:

$$\begin{aligned} V^\star(s_1,b_1) &= \sup_{\sigma_1\in\Sigma_1} V_{\sigma_1}(s_1,b_1) \\ &= \sup_{\sigma_1\in\Sigma_1} \int_{s_E\in S_E} b_1(s_E)\big(\inf_{\sigma_2\in\Sigma_2} \mathbb{E}^{\sigma_1,\sigma_2}_{(s_1,s_E)}[Y]\big)\mathrm{d}s_E \qquad \text{by (15)} \\ &= \sup_{\sigma_1\in\Sigma_1} \langle \inf_{\sigma_2\in\Sigma_2} \mathbb{E}^{\sigma_1,\sigma_2}_{(s_1,s_E)}[Y], (s_1,b_1)\rangle \\ &= \sup_{\alpha\in\Gamma} \langle \alpha, (s_1,b_1)\rangle \end{aligned}$$

where $\Gamma \triangleq \{\inf_{\sigma_2\in\Sigma_2} \mathbb{E}^{\sigma_1,\sigma_2}_{(s_1,s_E)}[Y] \mid \sigma_1 \in \Sigma_1\}$. According to the operator equivalence above, we have:

$$[TV^\star](s_1,b_1) = \max_{u_1\in\mathbb{P}(A_1)} \sup_{\overline{\alpha}\in\Gamma^{A_1\times S_1}} \langle f_{u_1,\overline{\alpha}}, (s_1,b_1)\rangle \tag{30}$$

for all $(s_1, b_1) \in S_B$, where $\Gamma^{A_1\times S_1} \triangleq \{\{\alpha^{a_1,s_1'}\}_{a_1\in A_1, s_1'\in S_1} \mid \alpha^{a_1,s_1'} \in \mathrm{Conv}(\Gamma)\}$ and Γ is given above. Now, by following the same argument as in the proof of [19, Lemma 6.7], we can show that $V^\star(s_1,b_1) = [TV^\star](s_1,b_1)$ for all $(s_1,b_1) \in S_B$, i.e., $V^\star = [TV^\star]$.

Next we demonstrate that the operator T is a contraction mapping on the space $\mathbb{F}(S_B)$ with respect to the supremum norm $\|J\| = \sup_{(s_1,b_1)\in S_B} |J(s_1,b_1)|$. Therefore consider any $J_1, J_2 \in \mathbb{F}(S_B)$ and for any belief $(s_1,b_1) \in S_B$, let $(u_1^{1\star}, u_2^{1\star})$ and $(u_1^{2\star}, u_2^{2\star})$ be the minimax strategy profiles in the stage games $[TJ_1](s_1,b_1)$ and $[TJ_2](s_1,b_1)$, respectively. Also, let $\bar{J}_1(u_1,u_2)$ and $\bar{J}_2(u_1,u_2)$ be the values of state (s_1,b_1) of the stage game under the strategy pair $(u_1,u_2) \in \mathbb{P}(A_1)\times\mathbb{P}(A_2 \mid S)$ when computing the backup values in (22) for J_1 and J_2, respectively. Without loss of generality, we assume $[TJ_1](s_1,b_1) \le [TJ_2](s_1,b_1)$, and thus since $(u_1^{1\star}, u_2^{1\star})$ is minimax strategy profile for $[TJ_1](s_1,b_1)$:

$$\begin{aligned} \bar{J}_1(u_1^{2\star}, u_2^{1\star}) &\le \bar{J}_1(u_1^{1\star}, u_2^{1\star}) \\ &= [TJ_1](s_1,b_1) && \text{by definition of } \bar{J}_1 \\ &\le [TJ_2](s_1,b_1) && \text{without loss of generality} \\ &= \bar{J}_2(u_1^{2\star}, u_2^{2\star}) && \text{by definition of } \bar{J}_2 \\ &\le \bar{J}_2(u_1^{2\star}, u_2^{1\star}) && \text{since } (u_1^{2\star}, u_2^{2\star}) \text{ is minimax strategy.} \end{aligned} \tag{31}$$

Now using (31) for any $(s_1,b_1) \in S_B$ we have

$$\begin{aligned} &|[TJ_2](s_1,b_1) - [TJ_1](s_1,b_1)| \le \bar{J}_2(u_1^{2\star}, u_2^{1\star}) - \bar{J}_1(u_1^{2\star}, u_2^{1\star}) \\ &= \beta \textstyle\sum_{a_1,s_1'} P(a_1, s_1' \mid (s_1,b_1), u_1^{2\star}, u_2^{1\star})\big(J_2(s_1', b_1^{s_1,a_1,u_2^{1\star},s_1'}) - J_1(s_1', b_1^{s_1,a_1,u_2^{1\star},s_1'})\big) \\ &\qquad\qquad \text{by (22)} \\ &\le \beta \textstyle\sum_{a_1,s_1'} P(a_1, s_1' \mid (s_1,b_1), u_1^{2\star}, u_2^{1\star}) \|J_2 - J_1\| \quad \text{by definition of } \|\cdot\| \end{aligned}$$

$$= \beta\|J_2 - J_1\| \qquad \text{since } P(\cdot \mid (s_1, b_1), u_1^{2\star}, u_2^{1\star}) \text{ is a distribution.} \quad (32)$$

Now by definition of the supremum norm:

$$\begin{aligned}\|[TJ_2] - [TJ_1]\| &= \sup_{(s_1,b_1)\in S_B} |[TJ_2](s_1, b_1) - [TJ_1](s_1, b_1)| \\ &\leq \sup_{(s_1,b_1)\in S_B} \beta\|J_2 - J_1\| && \text{by (32)} \\ &= \beta\|J_2 - J_1\| && \text{rearranging}\end{aligned}$$

and hence, since $\beta \in (0, 1)$, we have that T is a contraction mapping. Thus, the fact that the value function $V^\star$ is the unique fixed point of T now follows directly from Banach's fixed point theorem. □

Lemma 5 (PWC function) *For any $a \in A$, $s_1' \in S_1$ and $\alpha \in \mathbb{F}_C(S)$, if $\alpha^{a,s_1'} : S \to \mathbb{R}$ is the function where for any $s \in S$:*

$$\alpha^{a,s_1'}(s) = \textstyle\sum_{(s_1',s_E')\in\Theta_s^a} \delta(s, a)(s_1', s_E')\alpha(s_1', s_E')$$

then $\alpha^{a,s_1'}$ is PWC.

Proof. Let $a = (a_1, a_2)$. Since α is PWC, there exists an FCP Φ of S such that α is constant in each region of Φ. According to Assumption 1, there exists a pre-image FCP Φ' of $\Phi + \Phi_P$ for joint action a, where Φ_P is the perception FCP for Ag_1. Consider any region $\phi' \in \Phi'$ and let ϕ be any region of $\Phi + \Phi_P$ such that $\Theta_s^a \cap \phi \neq \varnothing$ for all $s \in \phi'$. Since Φ_P is the perception FCP for Ag_1, there exists $s_1' \in S_1$ such that if $s' \in \phi$, then $s' = (s_1', s_E')$ for some $s_E' \in S_E$ and let $\phi_E = \{s_E \in S_E \mid (s_1', s_E) \in \phi\}$. If $s, \tilde{s} \in \phi'$ such that $s = (s_1, s_E)$ and $\tilde{s} = (\tilde{s}_1, \tilde{s}_E)$, then using Assumption 1 we have $\sum_{s'\in\Theta_s^a\cap\phi} \delta(s, a)(s') = \sum_{\tilde{s}'\in\Theta_{\tilde{s}}^a\cap\phi} \delta(\tilde{s}, a)(\tilde{s}')$ and $s_1 = \tilde{s}_1$. Now combining this fact with Definition 2, it follows that:

$$\textstyle\sum_{(s_1',s_E')\in\Theta_s^a\wedge s_E'\in\phi_E} \delta(s, a)(s_1', s_E') = \sum_{(s_1',\tilde{s}_E')\in\Theta_{\tilde{s}}^a\wedge \tilde{s}_E'\in\phi_E} \delta(\tilde{s}, a)(s_1', \tilde{s}_E')\,.$$

Since $\alpha^{a_1,s_1'}(s_1', s_E') = \alpha^{a_1,s_1'}(s_1', \tilde{s}_E')$ for any $(s_1', s_E'), (s_1', \tilde{s}_E') \in \phi$ and $S_E^{s_1'} = \{s_E' \in S_E \mid obs_1(loc_1', s_E') = per_1'\}$ is equal to $\{\phi_E \mid \phi \in \Phi^{s_1'}\}$ for some finite set of regions $\Phi^{s_1'} \subseteq \Phi + \Phi_P$, it follows that

$$\begin{aligned}&\textstyle\sum_{(s_1',s_E')\in\Theta_s^a\wedge s_E'\in S_E^{s_1'}} \delta(s, a)(s_1', s_E')\alpha^{a_1,s_1'}(s_1', s_E') \\ &= \textstyle\sum_{(s_1',\tilde{s}_E')\in\Theta_{\tilde{s}}^a\wedge \tilde{s}_E'\in S_E^{s_1'}} \delta(\tilde{s}, a)(s_1', \tilde{s}_E')\alpha^{a_1,s_1'}(s_1', \tilde{s}_E')\end{aligned}$$

and therefore $\alpha^{a,s_1'}(s) = \alpha^{a,s_1'}(\tilde{s})$, implying that $\alpha^{a,s_1'}$ is constant in each region of Φ'. □

Lemma 6 (LP for minimax and P-PWLC, extended Lemma 1) *If $V \in \mathbb{F}(S_B)$ is P-PWLC with PWC α-functions Γ, for any $(s_1, b_1) \in S_B$, $[TV](s_1, b_1)$ is given by the LP over the real-valued variables $(v_\phi)_{\phi\in\Phi_\Gamma}$, $(\lambda_\alpha^{a_1,s_1'})_{(a_1,s_1')\in A_1\times S_1,\alpha\in\Gamma}$ and $(p^{a_1})_{a_1\in A_1}$:*

$$\text{maximise } \textstyle\sum_{\phi\in\Phi_\Gamma} v_\phi \int_{(s_1,s_E)\in\phi} b_1(s_E)\mathrm{d}s_E \text{ subject to}$$

$$
\begin{aligned}
&v_\phi \leq \textstyle\sum_{a_1 \in A_1} p^{a_1} r((s_1, s_E), (a_1, a_2)) + \beta \sum_{a_1, s_1', s_E'} \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \\
&\quad \cdot \textstyle\sum_{\alpha \in \Gamma} \lambda_\alpha^{a_1, s_1'} \alpha(s_1', s_E') \\
&\lambda_\alpha^{a_1, s_1'} \geq 0, \\
&p^{a_1} = \textstyle\sum_{\alpha \in \Gamma} \lambda_\alpha^{a_1, s_1'} \\
&\textstyle\sum_{a_1 \in A_1} p^{a_1} = 1
\end{aligned} \tag{33}
$$

for all $\phi \in \Phi_\Gamma$, $a_2 \in A_2$, $(a_1, s_1') \in A_1 \times S_1$ and $\alpha \in \Gamma$ where $s_E \in \phi$. Moreover, if $(\overline{v}^\star, \overline{\lambda}_1^\star, \overline{p}_1^\star)$ is the optimal solution to the LP (33), then the maximiser of the maxsup operator in Definition 4 is $(\overline{p}_1^\star, \overline{\alpha}^\star)$, where $\overline{\alpha}^\star \in \Gamma^{A_1 \times S_1}$ is such that for $(a_1, s_1') \in A_1 \times S_1$, if $a_1 \in A_1$ and $p^{\star a_1} > 0$, then $\alpha^{\star a_1, s_1'} = \sum_{\alpha \in \Gamma} (\lambda_\alpha^{\star a_1, s_1'} / p^{\star a_1}) \alpha$ and $\alpha^{\star a_1, s_1'}(s) = L$ for all $s \in S$ otherwise.

Proof. Since V is P-PWLC, then according to Definitions 4 and 6 and Theorem 2:

$$
\begin{aligned}
&[TV](s_1, b_1) = \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \langle f_{u_1, \overline{\alpha}}, (s_1, b_1) \rangle \\
&= \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \textstyle\int_{s_E \in S_E} \big(\min_{a_2} f_{u_1, \overline{\alpha}, a_2}(s_1, s_E) \big) b_1(s_E) \mathrm{d}s_E
\end{aligned} \tag{34}
$$

which can be formulated as the following optimization problem:

$$
\begin{aligned}
[TV](s_1, b_1) = \max_{u_1 \in \mathbb{P}(A_1), \overline{\alpha} \in \Gamma^{A_1 \times S_1}, \overline{v}} & \textstyle\sum_{\phi \in \Phi_\Gamma} v_\phi \int_{(s_1, s_E) \in \phi} b_1(s_E) \mathrm{d}s_E \\
\text{subject to } v_\phi \leq f_{u_1, \overline{\alpha}, a_2}(s_1, s_E) & \quad \text{for all } \phi \in \Phi_\Gamma \text{ and } a_2 \in A_2
\end{aligned}
$$

where $\overline{v} = (v_\phi)_{\phi \in \Phi_\Gamma}$, $f_{u_1, \overline{\alpha}, a_2}$ is constant over ϕ and $(s_1, s_E) \in \phi$. Using (2), the constraint $v_\phi \leq f_{u_1, \overline{\alpha}, a_2}(s_1, s_E)$ can be written as:

$$
\begin{aligned}
v_\phi \leq & \textstyle\sum_{a_1 \in A_1} u_1(a_1) r((s_1, s_E), (a_1, a_2)) \\
& + \beta \textstyle\sum_{(a_1, s_1') \in A_1 \times S_1, s_E' \in S_E} u_1(a_1) \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \alpha^{a_1, s_1'}(s_1', s_E').
\end{aligned}
$$

Since $\alpha^{a_1, s_1'} \in \mathrm{Conv}(\Gamma)$, we have $\alpha^{a_1, s_1'} = \sum_{\alpha \in \Gamma} \lambda_\alpha^{a_1, s_1'} \alpha$ for some vector of real-values $(\lambda_\alpha^{a_1, s_1'})_{(a_1, s_1) \in A_1 \times S_1}$ such that $\sum_{\alpha \in \Gamma} \lambda_\alpha^{a_1, s_1'} = 1$, and therefore:

$$
\begin{aligned}
v_\phi \leq & \textstyle\sum_{a_1 \in A_1} u_1(a_1) r((s_1, s_E), (a_1, a_2)) + \beta \sum_{(a_1, s_1') \in A_1 \times S_1, s_E' \in S_E} \\
& u_1(a_1) \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \textstyle\sum_{\alpha \in \Gamma} \lambda_\alpha^{a_1, s_1'} \alpha(s_1', s_E') \\
= & \textstyle\sum_{a_1 \in A_1} p_{a_1} r((s_1, s_E), (a_1, a_2)) + \\
& + \beta \textstyle\sum_{(a_1, s_1') \in A_1 \times S_1, s_E' \in S_E} \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \textstyle\sum_{\alpha \in \Gamma} \lambda_\alpha^{a_1, s_1'} \alpha(s_1', s_E')
\end{aligned}
$$

where $p_{a_1} = u_1(a_1)$ for all $a_1 \in A_1$ and in the equality we scale $\lambda_\alpha^{a_1, s_1'} = p_{a_1} \lambda_\alpha^{a_1, s_1'}$ for all $a_1 \in A_1$, $s_1' \in S_1$ and $\alpha \in \Gamma$, which gives the constraints:

$$
\lambda_\alpha^{a_1, s_1'} \geq 0
$$

$$\begin{aligned} p_{a_1} &= \textstyle\sum_{\alpha\in\Gamma}\lambda_\alpha^{a_1,s_1'} \\ \textstyle\sum_{a_1\in A_1} p_{a_1} &= 1 \end{aligned}$$

and hence the fact we can solve the LP problem (33) to compute $[TV](s_1, b_1)$ follows directly. □

*Proof (**Proof of Theorem 3**).* **P-PWLC closure.** Consider the LP in Lemma 1, i.e., (33) in the extended Lemma 6, which computes the minimax or maxsup backup $[TV](s_1, b_1)$ when V is P-PWLC. The polytope of feasible solutions of the LP defined by the constraints is independent of the environment belief b_1, because b_1 only appears in the objective. Therefore, the set Q_{s_1} of vertices of this polytope is also independent of b_1. For each $b_1 \in \mathbb{P}(S_E)$, the optimal value of an LP representing $[TV](s_1, b_1)$ can be found with the vertices Q_{s_1}, as the objective is linear in $\hat{V}$ for any given b_1. There is a finite number of vertices $q \in Q_{s_1}$, and each vertex $q \in Q_{s_1}$ corresponds to some assignment of variables u_1^q and $\overline{\alpha}^q$ (u_1^q and $\overline{\alpha}^q$ are computed by (33)). Since Q_{s_1} is finite, then letting $Q = \{q \in Q_{s_1} \mid s_1 \in S_1\}$, which is finite, we have:

$$[TV](s_1, b_1) = \max_{q\in Q} \langle f_{u_1^q, \overline{\alpha}^q}, (s_1, b_1)\rangle \,.$$

Moreover, since $f_{u_1,\overline{\alpha},a_2}$ is PWC for any $u_1 \in \mathbb{P}(A_1), \overline{\alpha} \in \Gamma^{A_1\times S_1}$ and $a_2 \in A_2$, then it follows from Definition 4, the function $f_{u_1^p,\overline{\alpha}^p}$ is PWC. This implies that $[TV] \in \mathbb{F}(S_B)$ and P-PWLC.

Convergence. Using the fixed point in Theorem 2, the conclusion directly follows from Banach's fixed point theorem and the fact we have proved in Theorem 3 that if $V \in \mathbb{F}(S_B)$ and P-PWLC, so is $[TV]$. □

*Proof (**Proof of Lemma 2**).* By following the proof of Theorem 3 and how $\overline{p}_1^\star$ and $\overline{\alpha}^\star$ are constructed, we can easily verify that in Algorithm 1 $\alpha^\star$ is a PWC α-function satisfying (5).

For $V_1, V_2 \in \mathbb{F}(S_B)$, we use the notation $V_1 \le V_2$ if $V_1(\hat{s}_1, \hat{b}_1) \le V_2(\hat{s}_1, \hat{b}_1)$ for all $(\hat{s}_1, \hat{b}_1) \in S_B$. Since $\Gamma' = \Gamma \cup \{\alpha^\star\}$, then it follows from Definition 6 that $V_{lb}^{\Gamma} \le V_{lb}^{\Gamma'}$.

In Algorithm 1, if the backup at line 4 is executed, then the maxsup operator is applied to some states in ϕ which may result in non-optimal minimax backup for other states in ϕ, and if the backup at line 5 is executed, $\alpha^\star$ is assigned the lower bound L over ϕ. Therefore we have for any $(\hat{s}_1, \hat{b}_1) \in S_B$:

$$\begin{aligned} \langle \alpha^\star, (\hat{s}_1, \hat{b}_1)\rangle &\le [TV_{lb}^{\Gamma}](\hat{s}_1, \hat{b}_1) \\ &\le [TV^\star](\hat{s}_1, \hat{b}_1) && \text{since } V_{lb}^{\Gamma} \le V^\star \\ &= V^\star(\hat{s}_1, \hat{b}_1) && \text{by Theorem 2.} \end{aligned} \tag{35}$$

Combining this inequality with $V_{lb}^{\Gamma} \le V^\star$, we have $V_{lb}^{\Gamma'} \le V^\star$ as required. □

*Proof (**Proof of Lemma 3**).* Combining Theorem 1, (3) and (4), the conclusion can be obtained by following the argument in the proof of [37, Lemma 4] for NS-POMDPs. □

The following lemma is required to prove the convergence of the algorithm.

Lemma 7 (Finite terminal belief points) *For any $t \geq 0$, if $\Psi_t \subseteq S_B$ of belief points where the trials performed by the procedure Explore of Algorithm 2 terminated at exploration depth t, then Ψ_t is a finite set.*

Proof. Consider any $t \geq 0$ and suppose that $\Psi_t \subseteq S_B$ is the set of belief points where the trials performed by the procedure *Explore* terminated at depth t. In order to prove that Ψ_t is a finite set, we first need to show the following continuity of the lower and upper bounds. Using the same argument in the proof Theorem 1, we can prove that the lower bound V_{lb}^{Γ} also has the continuity property of Theorem 1, i.e., for any $(s_1, b_1), (s_1, b_1') \in S_B$:

$$|V_{lb}^{\Gamma}(s_1, b_1) - V_{lb}^{\Gamma}(s_1, b_1')| \leq K(b_1, b_1')\,. \tag{36}$$

We still consider two beliefs $(s_1, b_1), (s_1, b_1') \in S_B$. Let $(\lambda_i^{\star\prime})_{i \in I_{s_1}}$ be the solution for $V_{ub}^{\Upsilon}(s_1, b_1')$ in (3), i.e.,

$$V_{ub}^{\Upsilon}(s_1, b_1') = \textstyle\sum_{i \in I_{s_1}} \lambda_i^{\star\prime} y_i + K_{ub}(b_1', \sum_{i \in I_{s_1}} \lambda_i^{\star\prime} b_1^i)\,. \tag{37}$$

Now since $(\lambda_i^{\star\prime})_{i \in I_{s_1}}$ satisfies the constraints in (3) for I_{s_1}, it follows that:

$$\begin{aligned}
&V_{ub}^{\Upsilon}(s_1, b_1) \leq \textstyle\sum_{i \in I_{s_1}} \lambda_i^{\star\prime} y_i + K_{ub}(b_1, \sum_{i \in I_{s_1}} \lambda_i^{\star} b_1^i) \\
&= \big(V_{ub}^{\Upsilon}(s_1, b_1') - K_{ub}(b_1', \textstyle\sum_{i \in I_{s_1}} \lambda_i^{\star} b_1^i)\big) + K_{ub}(b_1, \textstyle\sum_{i \in I_{s_1}} \lambda_i^{\star\prime} b_1^i) && \text{by (37)} \\
&= V_{ub}^{\Upsilon}(s_1, b_1') + \big(K_{ub}(b_1, \textstyle\sum_{i \in I_{s_1}} \lambda_i^{\star\prime} b_1^i) - K_{ub}(b_1', \textstyle\sum_{i \in I_{s_1}} \lambda_i^{\star} b_1^i)\big) && \text{rearranging} \\
&\leq V_{ub}^{\Upsilon}(s_1, b_1') + K_{ub}(b_1, b_1') && \text{by (4).}
\end{aligned}$$

Using similar steps we can also show that:

$$V_{ub}^{\Upsilon}(s_1, b_1') \leq V_{ub}^{\Upsilon}(s_1, b_1) + K_{ub}(b_1, b_1')$$

and hence:

$$|V_{ub}^{\Upsilon}(s_1, b_1) - V_{ub}^{\Upsilon}(s_1, b_1')| \leq K_{ub}(b_1, b_1')\,. \tag{38}$$

Let a belief point $(s_1^t, b_1^t) \in \Psi_t$. Since the procedure *Explore* terminates at (s_1^t, b_1^t) with exploration depth t, then the action-observation pair $(\hat{a}_1, \hat{s}_1)$ computed by (6) (from line 7 of Algorithm 2) satisfies

$$P(\hat{a}_1, \hat{s}_1 \mid (s_1^t, b_1^t), u_1^{ub}, u_2^{lb}) excess_{t+1}(\hat{s}_1, b_1^{s_1^t, \hat{a}_1, u_2^{lb}, \hat{s}_1}) \leq 0\,.$$

Thus, for any $(a_1, s_1') \in A_1 \times S_1$, if $P(a_1, s_1' \mid (s_1^t, b_1^t), u_1^{ub}, u_2^{lb}) > 0$, then we have $excess_{t+1}(s_1', b_1^{s_1^t, a_1, u_2^{lb}, s_1'}) \leq 0$, i.e.,

$$V_{ub}^{\Upsilon}(s_1', b_1^{s_1^t, a_1, u_2^{lb}, s_1'}) - V_{lb}^{\Gamma}(s_1', b_1^{s_1^t, a_1, u_2^{lb}, s_1'}) \leq \rho(t+1)\,. \tag{39}$$

Let (u_1^{lb}, u_2^{lb}) and (u_1^{ub}, u_2^{ub}) be the minimax strategy profiles in stage games $[TV_{lb}^{\Gamma}](s_1^t, b_1^t)$ and $[TV_{ub}^{\Upsilon}](s_1^t, b_1^t)$, respectively. Then, we denote by $J^{lb}(u_1, u_2)$

and $J^{ub}(u_1, u_2)$ the value of the stage game at (s_1^t, b_1^t) under the strategy pair $(u_1, u_2) \in \mathbb{P}(A_1) \times \mathbb{P}(A_2 \mid S)$ when computing the backup values in (22) via V_{lb}^{Γ} and V_{ub}^{Υ}, respectively. Thus, since (u_1^{lb}, u_2^{lb}) is a minimax strategy profile:

$$\begin{aligned} J^{lb}(u_1^{ub}, u_2^{lb}) &\leq J^{lb}(u_1^{lb}, u_2^{lb}) \\ &= [TV_{lb}^{\Gamma}](s_1^t, b_1^t) && \text{by definition of } J^{lb} \\ &\leq [TV_{ub}^{\Upsilon}](s_1^t, b_1^t) && \text{by Lemmas 2 and 3} \\ &= J^{ub}(u_1^{ub}, u_2^{ub}) && \text{by definition of } J^{ub} \\ &\leq J^{ub}(u_1^{ub}, u_2^{lb}) && (u_1^{ub}, u_2^{ub}) \text{ is a minimax strategy profile.} \end{aligned} \tag{40}$$

Now using (40) we have:

$$\begin{aligned} &[TV_{ub}^{\Upsilon}](s_1^t, b_1^t) - [TV_{lb}^{\Gamma}](s_1^t, b_1^t) \leq J^{ub}(u_1^{ub}, u_2^{lb}) - J^{lb}(u_1^{ub}, u_2^{lb}) \\ &= \beta \textstyle\sum_{a_1, s_1' \in A_1 \times S_1} P(a_1, s_1' \mid (s_1^t, b_1^t), u_1^{ub}, u_2^{lb}) \\ &\quad (V_{ub}^{\Gamma}(s_1', b_1^{s_1^t, a_1, u_2^{lb}, s_1'}) - V_{lb}^{\Gamma}(s_1', b_1^{s_1^t, a_1, u_2^{lb}, s_1'})) && \text{by (22)} \\ &\leq \beta \textstyle\sum_{a_1, s_1' \in A_1 \times S_1} P(a_1, s_1' \mid (s_1^t, b_1^t), u_1^{ub}, u_2^{lb}) \rho(t+1) && \text{by (39)} \\ &= \beta \rho(t+1) && \text{since } P \text{ is a distribution.} \end{aligned} \tag{41}$$

Substituting (41) into the excess gap $excess_t(s_1^t, b_1^t)$ we have that the excess gap after performing the point-based update at (s_1^t, b_1^t) in line 10 of Algorithm 2:

$$\begin{aligned} excess_t(s_1^t, b_1^t) &\leq \beta\rho(t+1) - \rho(t) \\ &= \rho(t) - 2(U-L)\bar{\varepsilon} - \rho(t) && \text{by definition of } \rho(t+1) \\ &= -2(U-L)\bar{\varepsilon} && \text{rearranging.} \end{aligned}$$

Due to the continuity (36) and (38), for any $(s_1, b_1), (s_1, b_1') \in S_B$, we have

$$V_{ub}^{\Upsilon}(s_1, b_1) - V_{lb}^{\Gamma}(s_1, b_1) \leq V_{ub}^{\Upsilon}(s_1, b_1') - V_{lb}^{\Gamma}(s_1, b_1') + 2K_{ub}(b_1, b_1')\,. \tag{42}$$

Now, for every belief $(s_1^t, b_1) \in S_B$ satisfying $K_{ub}(b_1, b_1^t) \leq (U-L)\bar{\varepsilon}$, substituting (42) into the excess gap $excess_t(s_1^t, b_1)$:

$$\begin{aligned} &excess_t(s_1^t, b_1) \leq V_{ub}^{\Upsilon}(s_1^t, b_1^t) - V_{lb}^{\Gamma}(s_1^t, b_1^t) + 2K_{ub}(b_1, b_1^t) - \rho(t) \\ &\beta\rho(t+1) + 2K_{ub}(b_1, b_1^t) - \rho(t) && \text{by (41)} \\ &\leq \rho(t) - 2(U-L)\bar{\varepsilon} + 2K_{ub}(b_1, b_1^t) - \rho(t) && \text{by definition of } \rho(t+1) \\ &\leq -2(U-L)\bar{\varepsilon} + 2(U-L)\bar{\varepsilon} && \text{since } K_{ub}(b_1, b_1^t) \leq (U-L)\bar{\varepsilon} \\ &= 0 && \text{rearranging} \end{aligned}$$

which means that $(s_1^t, b_1) \notin \Psi_t$. Since $\mathbb{P}(S_E)$ is compact and thus totally bounded, we can conclude that Ψ_t is finite. □

*Proof (**Proof of Theorem 4**).* By the choice of $\bar{\varepsilon}$, the sequence $(\rho(t))_{t \in \mathbb{N}}$ is monotonically increasing and unbounded. Since $L \leq V_{lb}^{\Gamma}(s_B) \leq V_{ub}^{\Upsilon}(s_B) \leq U$ for

all $s_B \in S_B$, the difference between V_{lb}^{Γ} and V_{ub}^{Υ} is bounded by $U-L$. Therefore, there exists $T_{\max}$ such that $\rho(T_{\max}) \geq U-L \geq V_{ub}^{\Upsilon}(s_B) - V_{lb}^{\Gamma}(s_B)$ for all $s_B \in S_B$, and therefore the recursive procedure *Explore* always terminates.

To demonstrate that Algorithm 2 terminates, we reason about the sets $\Psi_t \subseteq S_B$ of belief points where the trials performed by the procedure *Explore* terminated at exploration depth t. Initially, $\Psi_t = \varnothing$ for every $0 \leq t < T_{max}$. Whenever the *Explore* recursion terminates at exploration depth t (i.e., the condition on line 9 does not hold), the belief s_B^t (which was the last belief considered during the trial) is added into the set Ψ_t, i.e., $\Psi_t \triangleq \Psi_t \cup \{s_B^t\}$. Since the agent state space S_1 is finite and the number of possible termination depth is finite ($0 \leq t < T_{max}$) and the set Ψ_t is finite by Lemma 7, the algorithm has to terminate. Then, combining Lemmas 2 and 3, the conclusion follows directly. □

Lemma 8 (LP for upper bound) *For particle-based belief (s_1, b_1), let $P(s_E; b_1)$ be the probability of particle s_E under b_1. Consider the function $K_{ub} = K$, i.e.,*

$$K_{ub}(b_1, b_1') = \tfrac{1}{2}(U-L)\textstyle\sum_{b_1(s_E)+b_1'(s_E)>0} |P(s_E; b_1) - P(s_E; b_1')| \,. \tag{43}$$

Then, $V_{ub}^{\Upsilon}(s_1, b_1)$ is the optimal value of the LP:

$$\begin{aligned} &\text{minimise } \textstyle\sum_{k \in I_{s_1}} \lambda_k y_k + 1/2(U-L)\sum_{s_E \in S_E^+} c_{s_E} \text{ subject to} \\ &\quad c_{s_E} \geq |P(s_E; b_1) - \textstyle\sum_{k \in I_{s_1}} \lambda_k P(s_E; b_1^k)|,\ \lambda_k \geq 0 \ \text{ and } \ \textstyle\sum_{k \in I_{s_1}} \lambda_k = 1 \end{aligned}$$

for $s_E \in S_E^+$ and $k \in I_{s_1}$, where $S_E^+ = \{s_E \in S_E \mid b_1(s_E) + \sum_{k \in I_{s_1}} b_1^k(s_E) > 0\}$.

Proof. The result follows directly from (3) and (43). □

Theorem 7 (LP for minimax operator over upper bound, extended version of Theorem 5). *For the function K_{ub}, see (43), and particle-based belief (s_1, b_1) represented by $\{(s_E^i, \kappa_i)\}_{i=1}^{N_b}$, we have that $[TV_{ub}^{\Upsilon}](s_1, b_1)$ is the optimal value of the LP (13).*

Proof. We first prove that given any $s_1 \in S_1$, $V_{ub}^{\Upsilon}(s_1, \cdot)$ is a convex function. Consider any two beliefs $b_1, b_1' \in \mathbb{P}(S_E)$ and $\tau, \tau' \geq 0$ such that $\tau + \tau' = 1$. Let $(\lambda_k^\star)_{k \in I_{s_1}}$ and $(\lambda_k'^\star)_{k \in I_{s_1}}$ be optimal solutions of (3) for $V_{ub}^{\Upsilon}(s_1, b_1)$ and $V_{ub}^{\Upsilon}(s_1, b_1')$ respectively, i.e.,

$$\begin{aligned} V_{ub}^{\Upsilon}(s_1, b_1) &= \textstyle\sum_{k \in I_{s_1}} \lambda_k^\star y_k + K_{ub}(b_1, \sum_{k \in I_{s_1}} \lambda_k^\star b_1^k) \\ V_{ub}^{\Upsilon}(s_1, b_1') &= \textstyle\sum_{k \in I_{s_1}} \lambda_k'^\star y_k + K_{ub}(b_1', \sum_{k \in I_{s_1}} \lambda_k'^\star b_1^k) \,. \end{aligned} \tag{44}$$

From the constraints of (3) it follows that:

$$\tau\lambda_k^\star + \tau'\lambda_k'^\star \geq 0 \text{ for all } k \in I_{s_1} \text{ and } \textstyle\sum_{k \in I_{s_1}} (\tau\lambda_k^\star + \tau'\lambda_k'^\star) = 1. \tag{45}$$

Also let:

$$S_E^1 = \{s_E \in S_E \mid b_1(s_E) + b_1'(s_E) + \textstyle\sum_{k \in I_{s_1}} b_1^k(s_E) > 0\} \tag{46}$$

$$S_E^2 = \{s_E \in S_E \mid b_1(s_E) + \textstyle\sum_{k\in I_{s_1}} b_1^k(s_E) > 0\} \tag{47}$$
$$S_E^3 = \{s_E \in S_E \mid b_1'(s_E) + \textstyle\sum_{k\in I_{s_1}} b_1^k(s_E) > 0\}\,. \tag{48}$$

Now using (43) and (46) we have:

$$\begin{aligned}
&K_{ub}(\tau b_1 + \tau' b_1', \textstyle\sum_{k\in I_{s_1}} (\tau\lambda_k^\star + \tau'\lambda_k'^\star) b_1^k)\\
&= \tfrac{1}{2}(U-L)\textstyle\sum_{s_E\in S_E^1} |\tau b_1(s_E) + \tau' b_1'(s_E) - \sum_{k\in I_{s_1}} (\tau\lambda_k^\star + \tau'\lambda_k'^\star) b_1^k(s_E)|\\
&\le \tfrac{1}{2}(U-L)\textstyle\sum_{s_E\in S_E^1} \Big(\Big| \tau\big(b_1(s_E) - \sum_{k\in I_{s_1}} \lambda_k^\star b_1^k(s_E)\big)\\
&\qquad + \tau'\big(b_1'(s_E) - \textstyle\sum_{k\in I_{s_1}} \lambda_k'^\star b_1^k(s_E)\big)\Big|\Big) && \text{rearranging}\\
&= \tfrac{1}{2}(U-L)\textstyle\sum_{s_E\in S_E^1} \Big(\tau|b_1(s_E) - \sum_{k\in I_{s_1}} \lambda_k^\star b_1^k(s_E)|\\
&\qquad + \tau'|b_1'(s_E) - \textstyle\sum_{k\in I_{s_1}} \lambda_k'^\star b_1^k(s_E)|\Big) && \text{since } \tau,\tau' \ge 0\\
&= \tfrac{1}{2}(U-L)\tau\textstyle\sum_{s_E\in S_E^2} \big|b_1(s_E) - \sum_{k\in I_{s_1}} \lambda_k^\star b_1^k(s_E)\big|\\
&\quad + \tfrac{1}{2}(U-L)\tau'\textstyle\sum_{s_E\in S_E^3} \big|b_1'(s_E) - \sum_{k\in I_{s_1}} \lambda_k'^\star b_1^k(s_E)\big| && \text{by (47) and (48)}\\
&= \tau K_{ub}(b_1, \textstyle\sum_{k\in I_{s_1}} \lambda_k^\star b_1^k) + \tau' K_{ub}(b_1', \textstyle\sum_{k\in I_{s_1}} \lambda_k'^\star b_1^k) && (49)
\end{aligned}$$

Next, from (3) we have:

$$\begin{aligned}
&V_{ub}^\Upsilon(s_1, \tau b_1 + \tau' b_1') = \min_{(\lambda_k)_{k\in I_{s_1}}} \textstyle\sum_{k\in I_{s_1}} \lambda_k y_k + K_{ub}(\tau b_1 + \tau' b_1', \sum_{k\in I_{s_1}} \lambda_k b_1^k)\\
&\le \textstyle\sum_{k\in I_{s_1}} (\tau\lambda_k^\star + \tau'\lambda_k'^\star) y_k + K_{ub}(\tau b_1 + \tau' b_1', \sum_{k\in I_{s_1}} (\tau\lambda_k^\star + \tau'\lambda_k'^\star) b_1^k) && \text{by (45)}\\
&\le \textstyle\sum_{k\in I_{s_1}} (\tau\lambda_k^\star + \tau'\lambda_k'^\star) y_k + \tau K_{ub}(b_1, \sum_{k\in I_{s_1}} \lambda_k^\star b_1^k)\\
&\qquad + \tau' K_{ub}(b_1', \textstyle\sum_{k\in I_{s_1}} \lambda_k'^\star b_1^k) && \text{by (49)}\\
&= \tau V_{ub}^\Upsilon(s_1, b_1) + \tau' V_{ub}^\Upsilon(s_1, b_1') && \text{by (44)}
\end{aligned}$$

and hence $V_{ub}^\Upsilon(s_1, \cdot)$ is convex in $\mathbb{P}(S_E)$.

The inequality (38) shows that $V_{ub}^\Upsilon(s_1, \cdot)$ is continuous in $\mathbb{P}(S_E)$. By following the proof of [19, Proposition 4.12], we can prove that there exists a set Γ' of functions $\mathbb{F}(S)$ such that $V_{ub}^\Upsilon(s_1, b_1) = \sup_{\alpha\in\Gamma'} \langle \alpha, (s_1, b_1)\rangle$ for all $(s_1, b_1) \in S_B$. Therefore, according to Theorem 2, for any $(s_1, b_1) \in S_B$:

$$\begin{aligned}
[TV_{ub}^\Upsilon](s_1, b_1) &= \max_{u_1\in\mathbb{P}(A_1)} \min_{u_2\in\mathbb{P}(A_2|S)} \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)]\\
&\quad + \beta \textstyle\sum_{a_1, s_1'} P(a_1, s_1' \mid (s_1, b_1), u_1, u_2) V_{ub}^\Upsilon(s_1', b_1^{s_1,a_1,u_2,s_1'})\\
&= \min_{u_2\in\mathbb{P}(A_2|S)} \max_{u_1\in\mathbb{P}(A_1)} \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)]\\
&\quad + \beta \textstyle\sum_{a_1, s_1'} P(a_1, s_1' \mid (s_1, b_1), u_1, u_2) V_{ub}^\Upsilon(s_1', b_1^{s_1,a_1,u_2,s_1'})\,. \qquad (50)
\end{aligned}$$

We now define a payoff function $J : \mathbb{P}(A_1) \times \mathbb{P}(A_2 \mid S) \to \mathbb{R}$ to be the objective of the maximin and minimax optimisation in (50) such that for $u_1 \in \mathbb{P}(A_1)$ and $u_2 \in \mathbb{P}(A_2 \mid S)$, letting $E_1 = \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)]$, $p^{a_1} = u_1(a_1)$, $p^{a_1,u_2,s_1'} =$

$P(s_1' \mid (s_1, b_1), a_1, u_2)$ then we have:

$$\begin{aligned} J(u_1,u_2) &= E_1 + \beta \textstyle\sum_{a_1,s_1'} p^{a_1} p^{a_1,u_2,s_1'} V_{ub}^{\Upsilon}(s_1', b_1^{s_1,a_1,u_2,s_1'}) \\ &= E_1 + \beta \textstyle\sum_{a_1,s_1' \in A_1 \times S_1} p^{a_1} p^{a_1,u_2,s_1'} \min_{(\lambda_k)_{k \in I_{s_1'}}} \\ &\quad \left(\textstyle\sum_{k \in I_{s_1'}} \lambda_k y_k + K_{ub}\big(b_1^{s_1,a_1,u_2,s_1'}, \textstyle\sum_{k \in I_{s_1'}} \lambda_k b_1^{s_1,a_1,u_2,s_1'}\big) \right) \qquad \text{by (3).} \end{aligned}$$

Now combining this with (43) we have:

$$\begin{aligned} &J(u_1,u_2) = E_1 + \beta \textstyle\sum_{a_1,s_1'} p^{a_1} p^{a_1,u_2,s_1'} V_{ub}^{\Upsilon}(s_1', b_1^{s_1,a_1,u_2,s_1'}) \\ &= E_1 + \beta \textstyle\sum_{a_1,s_1' \in A_1 \times S_1} p^{a_1} p^{a_1,u_2,s_1'} \min_{\overline{\nu},\overline{d}} \big(\textstyle\sum_{k \in I_{s_1'}} \nu_k y_k + \frac{1}{2}(U-L) \textstyle\sum_{s_E \in S_E^+} d_{s_E} \big) \end{aligned}$$

where

$$\overline{\nu} = (\nu_k^{a_1,s_1'})_{(a_1,s_1') \in A_1 \times S_1, k \in I_{s_1'}} \quad \text{and} \quad \overline{c} = (d_{s_E'}^{a_1,s_1'})_{(a_1,s_1') \in A_1 \times S_1, s_E' \in S_E^{a_1,s_1'}}$$

are real-valued vectors of variables subject to the following linear constraint

$$\begin{aligned} d_{s_E'}^{a_1,s_1'} &\geq |P(s_E'; b_1^{s_1,a_1,u_2,s_1'}) - \textstyle\sum_{k \in I_{s_1'}} \nu_k^{a_1,s_1'} P(s_E'; b_1^k)| \\ \nu_k^{a_1,s_1'} &\geq 0 \text{ for } k \in I_{s_1'} \text{ and } \textstyle\sum_{k \in I_{s_1'}} \nu_k^{a_1,s_1'} = 1 \end{aligned} \tag{51}$$

and $S_E^{a_1,s_1'} = \{s_E' \in S_E \mid \sum_{a_2 \in A_2} b_1^{s_1,a_1,a_2,s_1'}(s_E') + \sum_{k \in I_{s_1'}} b_1^k(s_E') > 0\}$. Letting

$$C^{a_1,s_1'} = \tfrac{1}{2}(U-L) \textstyle\sum_{s_E' \in S_E^{a_1,s_1'}} d_{s_E'}^{a_1,s_1'}$$

it follows that $J(u_1,u_2)$ equals:

$$\min_{\overline{\nu},\overline{c}} \big(E_1 + \beta \textstyle\sum_{(a_1,s_1') \in A_1 \times S_1} p^{a_1} p^{a_1,u_2,s_1'} \big(\sum_{k \in I_{s_1'}} \nu_k^{a_1,s_1'} y_k + C^{a_1,s_1'} \big) \big) \tag{52}$$

Now, given any $u_2 \in \mathbb{P}(A_2 \mid S)$, let Λ be the feasible set for $(\overline{\nu},\overline{c})$, which is convex using (51). We then define a game with strategy spaces Λ and $\mathbb{P}(A_1)$ and payoff function $J_{u_2} : \Lambda \times \mathbb{P}(A_1) \to \mathbb{R}$ which is the objective of (52), i.e., for $(\overline{\nu},\overline{c}) \in \Lambda$ and $u_1 \in \mathbb{P}(A_1)$:

$$J_{u_2}((\overline{\nu},\overline{c}), u_1) = E_1 + \beta \textstyle\sum_{a_1,s_1'} p^{a_1} p^{a_1,u_2,s_1'} \big(\sum_{k \in I_{s_1'}} \nu_k^{a_1,s_1'} y_k + C^{a_1,s_1'} \big) . \tag{53}$$

Combining (50), (52) and (53) we have:

$$\begin{aligned} [TV_{ub}^{\Upsilon}](s_1,b_1) &= \min_{u_2 \in \mathbb{P}(A_2|S)} \max_{u_1 \in \mathbb{P}(A_1)} J(u_1,u_2) \\ &= \min_{u_2 \in \mathbb{P}(A_2|S)} \max_{u_1 \in \mathbb{P}(A_1)} \min_{(\overline{\nu},\overline{c}) \in \Lambda} J_{u_2}((\overline{\nu},\overline{c}), u_1) . \end{aligned} \tag{54}$$

We next show that the von Neumann's Minimax Theorem [27] applies to the game with payoff function J_{u_2} and strategy spaces Λ and $\mathbb{P}(A_1)$. This theorem requires that:

- Λ and $\mathbb{P}(A_1)$ are compact convex sets;
- J_{u_2} is a continuous function that is concave-convex, i.e., $J_{u_2}((\overline{\nu},\overline{c}),\cdot)$ is concave for fixed $(\overline{\nu},\overline{c})$ and $J_{u_2}(\cdot,u_1)$ is convex for fixed u_1.

Clearly Λ and $\mathbb{P}(A_1)$ are compact convex sets and by (53), J_{u_2} is bilinear in $\overline{\nu},\overline{c}$ and u_1, and thus concave in $\mathbb{P}(A_1)$ and convex in Λ. Hence we can apply von Neumann's Minimax Theorem, which gives us:

$$\max_{u_1\in\mathbb{P}(A_1)}\min_{(\overline{\nu},\overline{c})\in\Lambda} J_{u_2}((\overline{\nu},\overline{c}),u_1) = \min_{(\overline{\nu},\overline{c})\in\Lambda}\max_{u_1\in\mathbb{P}(A_1)} J_{u_2}((\overline{\nu},\overline{c}),u_1)\,.$$

Therefore, using this result and (54) we have that:

$$\begin{aligned}
[TV^{\Upsilon}_{ub}](s_1,b_1) &= \min_{u_2\in\mathbb{P}(A_2|S)}\min_{(\overline{\nu},\overline{c})\in\Lambda}\max_{u_1\in\mathbb{P}(A_1)} J_{u_2}((\overline{\nu},\overline{c}),u_1)\\
&= \min_{u_2\in\mathbb{P}(A_2|S)}\min_{(\overline{\nu},\overline{c})\in\Lambda}\max_{u_1\in\mathbb{P}(A_1)}\big(E_1+\\
&\quad+\beta\textstyle\sum_{a_1,s_1'} p^{a_1}p^{a_1,u_2,s_1'}\big(\sum_{k\in I_{s_1'}}\nu_k^{a_1,s_1'}y_k + C^{a_1,s_1'}\big)\big) \qquad \text{by (53)}\\
&= \min_{u_2\in\mathbb{P}(A_2|S)}\min_{(\overline{\nu},\overline{c})\in\Lambda}\max_{a_1\in A_1}\big(E_1+\\
&\quad+\beta\textstyle\sum_{s_1'\in S_1} p^{a_1,u_2,s_1'}\big(\sum_{k\in I_{s_1'}}\nu_k^{a_1,s_1'}y_k + C^{a_1,s_1'}\big)\big)
\end{aligned}$$

where the final equality follows from the fact that, for fixed u_2 and $\overline{\nu}$ and $\overline{c}$, the objective is linear in u_1, from which $[TV^{\Upsilon}_{ub}](s_1,b_1)$ can be formulated as the LP problem given by minimise v subject to:

$$v \;\geq E_1 + \beta\textstyle\sum_{s_1'\in S_1} p^{a_1,u_2,s_1'}\big(\sum_{k\in I_{s_1'}}\nu_k^{a_1,s_1'}y_k + C^{a_1,s_1'}\big) \;\text{ for all } a_1\in A_1. \quad (55)$$

Letting $\lambda_k^{a_1,s_1'} = p^{a_1,u_2,s_1'}\nu_k^{a_1,s_1'}$ and $c_{s_E'}^{a_1,s_1'} = p^{a_1,u_2,s_1'}d_{s_E'}^{a_1,s_1'}$, we can reformulate (55) as minimise v subject to:

$$\begin{aligned}
v \;&\geq \textstyle\sum_{i=1}^{N_b}\sum_{a_2}\kappa_i p_{a_2}^{s_1,s_E^i} r((s_1,s_E^i),(a_1,a_2)) + \beta\sum_{s_1'\in S_1} v_{a_1,s_1'}\\
v_{a_1,s_1'} \;&= \textstyle\sum_{k\in I_{s_1'}}\lambda_k^{a_1,s_1'}y_k + \frac{1}{2}(U-L)\sum_{s_E'\in S_E^{a_1,s_1'}}\hat{c}_{s_E'}^{a_1,s_1'} \quad .
\end{aligned}$$

for all $a_1\in A_1$ and $s_1'\in S_1$, where $u_2(a_2|s_1,s_E^i) = p_{a_2}^{s_1,s_E^i}$. We next compute the constraints for $\lambda_k^{a_1,s_1'}$ and $\hat{c}_{s_E'}^{a_1,s_1'}$. According to the belief update (7):

$$\begin{aligned}
p^{a_1,u_2,s_1'}b_1^{s_1,a_1,u_2,s_1'}(s_E') &= P(s_1' \mid (s_1,b_1),a_1,u_2)\frac{P(s_1',s_E' \mid (s_1,b_1),a_1,u_2)}{P(s_1' \mid (s_1,b_1),a_1,u_2)}\\
&= P(s_1',s_E' \mid (s_1,b_1),a_1,u_2) \qquad \text{rearranging}\\
&= \textstyle\sum_{i=1}^{N_b}\sum_{a_2}\kappa_i p_{a_2}^{s_1,s_E^i}\delta((s_1,s_E^i),(a_1,a_2))(s_1',s_E')
\end{aligned}$$

where the final equality follows from the definition of a particle-based belief. Since $\nu_k^{a_1,s_1'}$ and $d_{s_E'}^{a_1,s_1'}$ are subject to the linear constraints (51), it follows that:

$$c_{s_E'}^{a_1,s_1'} \;\geq \Big|\textstyle\sum_{i=1}^{N_b}\sum_{a_2}\kappa_i p_{a_2}^{s_1,s_E^i}\delta((s_1,s_E^i),(a_1,a_2))(s_1',s_E')$$

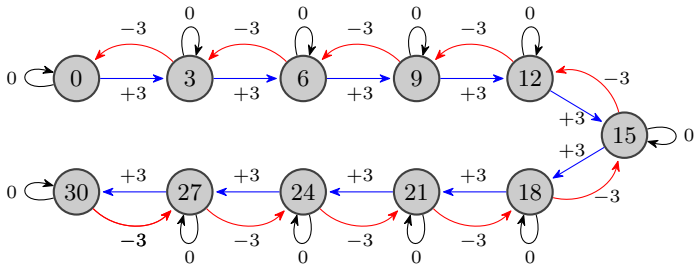


Fig. 4: Pedestrian-vehicle interaction: local transition diagram over the local states, i.e., vehicle speeds (m/s), with actions corresponding to the possible accelerations of the vehicle, i.e., +3, 0 and −3 (m/s^2).

$$\begin{aligned} & - \sum_{k \in I_{s_1'}} \lambda_k^{a_1, s_1'} P(s_E'; b_1^k) \Big| \\ \sum_{k \in I_{s_1'}} \lambda_k^{a_1, s_1'} &= \sum_{i=1}^{N_b} \sum_{a_2, s_E'} \kappa_i p_{a_2}^{s_1, s_E^i} \delta((s_1, s_E^i), (a_1, a_2))(s_1', s_E') \\ \lambda_k^{a_1, s_1'} &\geq 0 \end{aligned} \tag{56}$$

for all $(a_1, s_1') \in A_1 \times S_1$, $1 \leq i \leq N_b$ and $s_E' \in S_E$, $k \in I_{s_1'}$. Thus, the optimization problem can be reformulated as the LP problem in (13). □

E Further Case Study Details and Statistics

Finally, we give some additional details and statistics for the models developed for the two case studies used for evaluation in Section 7.

Pedestrian-vehicle interaction. The one-sided NS-POSG for the pedestrian-vehicle scenario is defined as follows:

- $S_1 = Loc_1 \times Per_1$, where:

$$\begin{aligned} Loc_1 &= \{30, 27, 24, 21, 18, 15, 12, 9, 6, 3, 0\} \\ Per_1 &= \{\text{“}\textit{unlikely to cross}\text{”}, \text{“}\textit{likely to cross}\text{”}, \text{“}\textit{very likely to cross}\text{”}\} \end{aligned}$$

  are the vehicle's discrete speeds (km/h) and perceived pedestrian intentions.
- $S_E = \{((x_1, y_1), (x_2, y_2)) \in (\mathbb{R}^2)^2 \mid 0 \leq x_1, x_2 \leq 20 \wedge 0 \leq y_1, y_2 \leq 10\}$, where (x_1, y_1) and (x_2, y_2) are the top-left coordinates of the 2D fixed-size bounding boxes of size 0.5×1.5 (m^2) around the pedestrian at the previous and current steps, respectively.
- $A = A_1 \times A_2$, where $A_1 = \{-3, 0, 3\}$ (m/s^2) are the possible accelerations of the vehicle, and $A_2 = \{cross, back\}$ are the possible directions the pedestrian can choose to move.
- For each local state $v_1 \in Loc_1$ and environment state $((x_1, y_1), (x_2, y_2)) \in S_E$, we let $obs_1(v_1, ((x_1, y_1), (x_2, y_2))) = f_{ped}^{\max}((x_1, y_1), (x_2, y_2))$, where $f_{ped} : S_E \to \mathbb{P}(Per_1)$ is a data-driven pedestrian intention classifier implemented

via a feed-forward NN with ReLU activation functions and trained over the PIE dataset in [29]. Note here obs_1 is independent of the local state of Ag_1.
- For $(v_1, per_1) \in Loc_1 \times Per_1$, $v_1' \in Loc_1$ and $(a_1, a_2) \in A$ we have:

$$\delta_1((v_1, per_1), (a_1, a_2))(v_1') = \begin{cases} 1 \text{ if } v_1' = g_{next}(v_1, a_1) \\ 0 \text{ otherwise} \end{cases}$$

where $g_{next} : Loc_1 \times A_1 \to Loc_1$ is the speed update function of the vehicle with the transition diagram in Fig. 4.
- For $v_1 \in Loc_1$, $((x_1, y_1), (x_2, y_2)), ((x_1', y_1'), (x_2', y_2')) \in S_E$ and $(a_1, a_2) \in A$ we have $\delta_E(v_1, ((x_1, y_1), (x_2, y_2)), (a_1, a_2))((x_1', y_1'), (x_2', y_2')) = 1$ where

$$x_1' = x_2, \qquad y_1' = y_2,$$
$$x_2' = x_2 + move(a_2) v_2 \Delta t, \qquad y_2' = y_2 - v_1 \Delta t - \frac{a_1}{2} \Delta t^2,$$

$v_2 = 4.5$ (m/s) is the speed of the pedestrian, $move(a_2)$ is the direction of the movement of the pedestrian action, i.e., $move(cross)$=−1 and $move(back)$=1, and $\Delta t = |g_{next}(v_1, a_1) - v_1|/|a_1|$ if $a_1 \neq 0$ and 0.3 (s) otherwise.

A crash occurs if the environment state is in the set:

$$\mathcal{R}_{crash} = \{((x_1, y_1), (x_2, y_2)) \in S_E \mid 0 \leq x_2 \leq 0.5 \wedge 0 \leq y_2 \leq 2.5\}$$

i.e., the current bounding box around the pedestrian has a distance of no more than 0.5 and 1.0 (m) along the x and y coordinates to the vehicle, respectively (recall the bounding box has size 0.5×1.5 (m^2)). The reward structure is such that, for any $(s_1, s_E) \in S$ and $a \in A$, $r((s_1, s_E), a) = -200$ if $s_E \in \mathcal{R}_{crash}$ and 0 otherwise.

Pursuit-evasion game. We modify the example presented in [19] by considering a continuous environment $\mathcal{R} \triangleq \{(x, y) \in \mathbb{R}^2 \mid 0 \leq x, y \leq 3\}$ that is partitioned into multiple cells by perception functions. In this game, we have a pursuer agent Ag_p that tries to catch an evader agent Ag_e. In each step, the evader moves by picking from the set of actions $A_e \triangleq \{up, down, left, right\}$. The pursuer moves in a similar manner with additional diagonal movements, and thus has the action set $A_p \triangleq \{up, down, left, right, upleft, upright, downleft, downright\}$.

The evader has full observation and knows the exact location of both players. The pursuer has partial observation, that is, it knows which cell it is in, but does not know its exact location and does not know which cell the evader is in. The perception function of the pursuer employs an NN classifier $f_{\mathcal{R}} : \mathcal{R} \to \mathbb{P}(Grid)$, where $Grid \triangleq \{(i, j) \mid 1 \leq i, j \leq 3\}$, which takes the location (coordinates) of the pursuer as input and outputs a probability distribution over nine abstract grid points (cells), thus partitioning the environment as illustrated by Fig. 5. This is modelled as a one-sided NS-POSG as follows.

- $S_1 = Loc_1 \times Per_1$, where the local state $Loc_1 = \{\perp\}$ is a dummy state and $Per_1 = Grid$.

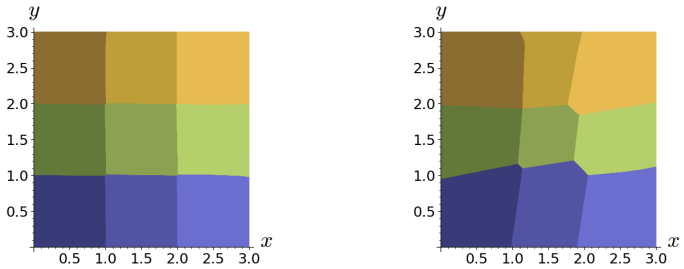


Fig. 5: Representation of a regular (left) and coarse (right) perception function for the pursuit-evasion example. Each graph depicts the boundaries of the nine abstract grid cells learnt by the classifiers. The pre-images of the regular and coarse perception functions are composed of 48 and 50 polytopes, respectively.

- $S_E = \{((x_p, y_p), (x_e, y_e)) \in \mathcal{R}^2\}$.
- $A = A_1 \times A_2$, where $A_1 = A_p$ and $A_2 = A_e$.
- $obs_i(\perp, (x_p, y_p)) = f^{\max}_{\mathcal{R}}(x_p, y_p)$ and the NN classifier $f_{\mathcal{R}} : \mathcal{R} \to \mathbb{P}(Grid)$ is implemented via a feed-forward NN with one hidden ReLU layer and 14 neurons.
- For environment states $s_E = ((x_p, y_p), (x_e, y_e))$, $s'_E = ((x'_p, y'_p), (x'_e, y'_e))$, local state loc_1 and joint action $a = (a_1, a_2)$:

$$\delta_E(\perp, s_E, a)(s'_E) = \prod_{i \in \{p,e\}} \delta_{E_i}((x_i, y_i), a_i)(x'_i, y'_i)$$

where for $i \in \{p, e\}$:

$$\delta_{E_i}((x_i, y_i), a_i)(x'_i, y'_i) = \begin{cases} 1 \text{ if } x'_i = x_i + d^x_{a_i} \text{ and } y'_i = y_i + d^y_{a_i} \\ 0 \text{ otherwise} \end{cases}$$

and the pairs $(d^x_{a_i}, d^y_{a_i})$ indicate the direction of movement under action a_i, e.g., $(d^x_{up}, d^y_{up}) = (0, 1)$ and $(d^x_{left}, d^y_{left}) = (-1, 0)$.

The capture condition in [19] is also used, that is, the evader is captured if it is in the same cell as the pursuer, which means the set of capture states $\mathcal{R}_{capture}$ is given by:

$$\{((x_p, y_p), (x_e, y_e)) \in S_E \mid \exists (i, j) \in Grid,\ (i{-}1 \leq x_p, x_e < i) \wedge (j{-}1 \leq y_p, y_e < j)\}.$$

Differently from [19], the game does not end once the evader is captured, allowing for the possibility of multiple captures. In case the pursuer is successful, that is, it enters the same cell as the evader, it receives a reward of 100. This can be modelled by assigning that value to any state-action pair with the state in $\mathcal{R}_{capture}$.

A model where the pursuer agent Ag_p has two pursuers under its control was also developed. For that model, the actions available to the pursuer agent are pairs corresponding to a chosen direction of movement for each pursuer, where each pursuer can now only move horizontally or vertically, i.e., the actions available to the pursuer agent are given by $A_p \triangleq (\{up, down, left, right\})^2$.

Model	Initial pts.	β	$\lvert\Gamma\rvert$	Lower bound		$\lvert\Upsilon\rvert$	Upper bound		Iter.	Time (min)
				init.	final		init.	final		
Pursuit-evasion (one pursuer)	1	0.7	184	0	5.0653	265	333.33	9.1819	169	15
	1	0.7	515	0	5.2798	788	333.33	6.6317	264	120
	2	0.7	413	0	4.5299	998	333.33	11.570	299	120
	1	0.8	468	0	9.8827	731	500.00	16.289	170	120
	1	0.9	331	0	22.387	731	1000.0	58.906	130	120
	1	**0.99**	**5**	**0**	**34.973**	**128**	**10000**	**35.972**	**44**	**3**
Pursuit-evasion (two pursuers)	1	0.7	509	0	14.134	790	333.33	39.943	274	120
Pedestrian-vehicle	1	0.7	1,928	0	620.54	4936	666.67	666.67	297	120
	2	0.7	2,783	0	526.34	8532	666.67	666.67	363	120
	1	0.8	2,089	0	805.92	5708	1000.0	1000.0	330	120

Table 1: Statistics for a set of one-sided NS-POSG solution instances. The bold entries for the pursuit-evasion model correspond to that with the coarser perception function (see Fig. 5).

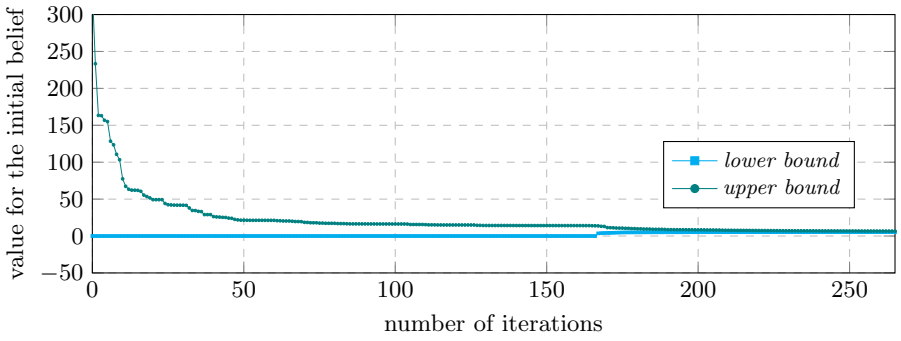


Fig. 6: Lower and upper bound values for a pursuit-evasion game with one pursuer when $\beta = 0.7$.

Additionally, the perception function of the pursuer agent is modified to take the coordinates of both pursuers and output the perceived grid cell for each of them. In this scenario, a capture happens if the evader is in the same grid cell as either pursuer.

Statistics. Table 1 shows statistics for solving various instances, varying the number of points in the initial belief and discount factor β. The table presents the initial and final values of the upper and lower bounds, the number of α-functions generated for the lower bound computation ($|\Gamma|$), the number of belief points for the upper bound computation ($|\Upsilon|$), and the number of iterations and the time required. In the experiments, we have set a timeout of 120 minutes (except for the first row where the timeout was reduced to 15 minutes). In addition, Figure 6 shows how the lower and upper bound values change for the initial belief as the number of iterations increases for one instance of the pursuit-evasion game.

Since our algorithm is *anytime*, lower and upper bounds hold throughout computations and we successfully generate meaningful strategies (discussed further below) on a range of models. However, computation is generally slow due to the number of LP problems to solve (whose size increases with $|\Gamma|$ and $|\Upsilon|$),

as well as expensive operations over polyhedra and the probabilistic branching of mixed strategies to guide exploration.

Both Table 1 and Figure 6 illustrate the impact of these factors. In the first two rows of Table 1 we observe the difference between a 15 and 120 minute timeout for the same instance of pursuit-evasion game with a single pursuer. As can be seen, the increase in the timeout causes the lower bound to improve (increase) by 0.2145, while the upper bound improves (decreases) by 2.55. With a timeout of 15 minutes we see that 169 iterations are performed; however, due to the number of α-functions growing from 184 to 515, increasing the timeout to 120 minutes only allows 95 more iterations to be performed.

Considering Figure 6, we initially see a sharp decrease of the upper bound, but improvement to either bound becomes progressively harder as computation progresses. The entry for the pursuit-evasion game with a single pursuer with a coarser perception function in Table 1 (highlighted in bold) is the only instance that converges before the timeout due to the fact that the number of reachable regions is smaller.

The table also demonstrates that, as expected, larger discount factors lead to larger lower and upper bounds. For the pursuit-evasion model with two pursuers, the larger difference between the lower and upper bound values reached at the timeout is a consequence of a higher branching factor during exploration slowing down the computation, as we have 64 joint actions in each state. For the entries related to the pedestrian-vehicle interaction model in Table 1, the final upper bound values match their initial values due to the fact that the initial beliefs were selected so that it should be possible to avoid a crash if an optimal strategy was played.

We note that HSVI for *finite* one-sided POSGs, in [19], is already computationally very expensive, even with multiple optimisations ([19] uses a timeout of 10 hours, versus 2 hours here).