Partially Observable Stochastic Games with Neural Perception Mechanisms

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

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

Abstract. Stochastic games are a well established model for multi-agent sequential decision making under uncertainty. In reality, though, agents have only partial observability of their environment, which makes the problem computationally challenging, even in the single-agent setting of partially observable Markov decision processes. Furthermore, in practice, agents increasingly perceive their environment using data-driven approaches such as neural networks trained on continuous data. To tackle this problem, we propose the model of neuro-symbolic partially-observable stochastic games (NS-POSGs), a variant of continuous-space concurrent stochastic games that explicitly incorporates perception mechanisms. We focus on a one-sided setting, comprising a partially-informed agent with discrete, data-driven observations and a fully-informed agent with continuous observations. We present a new point-based method, called one-sided NS-HSVI, for approximating values of one-sided NS-POSGs and implement it based on the popular particle-based beliefs, showing that it has closed forms for computing values of interest. We provide experimental results to demonstrate the practical applicability of our method for neural networks whose preimage is in polyhedral form.

1 Introduction

Strategic reasoning is essential to ensure stable multi-agent coordination in complex environments, as it allows the synthesis of optimal (or near-optimal) agent strategies and equilibria that guarantee expected outcomes, even in adversarial scenarios. Examples include coordination of autonomous road or underwater vehicles and robot motion planning. *Partially-observable stochastic games* (POSGs) are a natural model for real-world settings involving multiple agents, uncertainty and partial information, but pose significant challenges. Key problems are undecidable, already for the single-agent case of partially observable Markov decision processes (POMDPs) [22], and practical algorithms for computing or approximating optimal values and strategies are lacking.

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 [38,39]. This is well suited to a variety of applications, particularly when making worst-case assumptions about one agent; examples include the attacker in a security application, modelled, e.g., as a patrolling or pursuit-evasion game, or safety-critical settings, e.g., a pedestrian in an autonomous driving application.

From a computational perspective, one-sided POSGs avoid the need for nested beliefs [37], i.e., reasoning about beliefs not only over states but also over opponents' beliefs, since the fully informed agent can always reconstruct beliefs for the other agent from a full history of actions and observations. Recent computational advances for this model [17] have led to the first practical variant of heuristic search value iteration (HSVI) [31] for computing approximately optimal values and strategies in one-sided POSGs.

However, many realistic autonomous coordination scenarios involve agents perceiving *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, or NN-enabled vision in an airborne pursuit-evasion scenario.

Such perception mechanisms bring new challenges, notably continuous environments, which are inherently tied to NN-enabled perception because of standard training regimes. Discretising continuous models to finite-state representations, e.g, to leverage methods such as [17], is also difficult: decision boundaries obtained for data-driven perception are typically irregular and can be misaligned with gridding schemes for discretisation, affecting the precision of the computed strategies. In any case, discretisation may result in an exponential growth of the state space, depending on the granularity and the horizon.

So, in this paper, we work directly with the continuous state space of POSGs. It was shown in [26,17] that, under discrete observations and actions, continuous-state POMDPs and finite-state one-sided POSGs both have a piecewise linear and convex value function. In [35], this representation was generalised for continuous-state POMDPs with NN perception mechanisms (NS-POMDPs). The key idea is that ReLU neural network classifiers induce a finite decomposition of the continuous environment into polyhedra for each classification label. Building on this initial decomposition, a piecewise constant representation for the value, reward and perception functions, called α-functions, is developed. This forms the basis for a variant of HSVI, a point-based solution method that computes a lower and upper bound on the value function from a given belief, progressively subdividing the continuous state space over each iteration, and finally generating an (approximately) optimal strategy.

We extend these ideas from the single-agent (POMDP) setting [35] to zero-sum POSGs. This is significantly more challenging, even for the asymmetric one-sided case, because each value backup involves solving a normal form game and closure properties with respect to the minmax operator are needed to ensure that the polyhedral representation can be adapted to the game setting. Our approach also goes significantly beyond HSVI for finite POSGs [17] due to the use of α-functions and polyehdra to manage the continuous state space.

Contributions of the paper. We make the following contributions.

1. We introduce *one-sided neuro-symbolic POSGs (NS-POSGs)*, which generalise NS-POMDPs [35] to the two-agent zero-sum case, and extend one-sided POSGs in [17,38,39] to continuous state spaces. One-sided NS-POSGs are a subclass of continuous-state zero-sum POSGs with hybrid observations and discrete actions, in which the observation function of the partially-informed agent is discrete and synthesised in a data-driven fashion, and the other agent is fully informed with continuous observations.
2. We prove that the value function of one-sided NS-POSGs is continuous and convex and is a fixed point of a minimax operator, which has an equivalent maxsup formulation, motivated by [17], for discounted cumulative rewards.
3. We show that the piecewise constant α-function representation of the value function of [35], which admits a finite polyhedral representation, is closed with respect to the minimax operator.
4. We present a new point-based method, *one-sided NS-HSVI*, for solving one-sided NS-POSGs and implement it based on the popular particle-based beliefs, showing that it has closed forms for computing values of interest.
5. We provide experimental results showing the applicability of one-sided NS-HSVI in practice for neural networks whose preimage is in polyhedral form.

Related work. Solving POSGs is largely intractable. Methods based on exact dynamic programming [15] and approximations [21,11] exist but have high computational cost. Further approaches exist for *zero-sum* POSGs, including conversion to extensive-form games [3], counterfactual regret minimisation [40,19,20] and methods based on reinforcement learning and search [5,24]. [9] proposes an HSVI-like finite-horizon solver that provably converges to an ε-optimal solution; [32] provides convexity and concavity results but no algorithmic solution.

Methods exist for *one-sided* POSGs: a space partition approach when actions are public [38], a point-based approximate algorithm when observations are continuous [39] 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 [17], 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 [26,6,36] and discrete space approximation [4] can be used; the former also use α-functions to represent value functions but, unlike our approach, work with (approximate) Gaussian mixtures or dynamic Bayes nets. We use the same representations for lower/upper bounds as for NS-POMDPs [35], exploiting the underlying piecewise constant structure of the continuous-state model induced by the neural perception mechanism, but need stronger closure properties (under the minimax operator). A multi-agent model with perception, NS-CSGs, is proposed in [34,33], including a value iteration algorithm in [33], but partial observability is not considered, which is the main focus of this paper.

2 Background

POSGs. The semantics of our models are continuous-state *partially observable concurrent stochastic games* (POSGs) [19,5,16]. 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 2 agents; S a Borel measurable set of states; $A = A_1 \times A_2$ a finite set of joint actions where A_i are actions for agent $i \in N$; $\delta \colon (S \times A) \to \mathbb{P}(S)$ a probabilistic transition function; $\mathcal{O} = \mathcal{O}_1 \times \mathcal{O}_2$ a finite set of joint observations where $\mathcal{O}_i$ are observations for agent $i \in N$; 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 $\pi = s^0 \xrightarrow{a^0} \cdots \xrightarrow{a^{t-1}} s^t$ 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 history π from the perspective of agent i in terms of their knowledge about the current state: $\pi_i = o_{i,0} \xrightarrow{a_{i,0}} \cdots \xrightarrow{a_{i,t-1}} o_{i,t}$. If an agent has full information about the state, we assume that the agent is also informed of the last taken joint action. Let $FPaths_{\mathsf{G}}$ and $FPaths_{\mathsf{G},i}$ denote the sets of finite histories of G and AOHs of agent i, respectively.

A *(behaviour)* strategy of agent i is a mapping from its finite AOHs to probability distributions over actions $\sigma_i : FPaths_{\mathsf{G},i} \to \mathbb{P}(A_i)$. We denote by Σ_i the set of strategies of agent i. A *(strategy) 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 all profiles.

Objectives. We focus on infinite-horizon *discounted accumulated reward* objectives, where agents 1 and 2 aim to maximise and minimise the expected value, respectively. For state-action reward $r : (S \times A) \to \mathbb{R}$, the discounted reward for an infinite history π is $Y(\pi) = \sum_{k=0}^{\infty} \beta^k r(\pi(k), \pi[k])$ where $\beta \in (0, 1)$ is the discount factor. $\mathbb{E}_b^{\sigma}[Y]$ denotes the expected value of Y when starting from the state distribution $b \in \mathbb{P}(S)$ under profile $\sigma \in \Sigma$.

Values and minimax strategies. Given an objective Y and an initial state distribution b, the *upper value* $\overline{V}(b)$ equals $\inf_{\sigma_2 \in \Sigma_2} \sup_{\sigma_1 \in \Sigma_1} \mathbb{E}_b^{\sigma_1,\sigma_2}[Y]$ and the *lower value* $\underline{V}(b)$ equals $\sup_{\sigma_1 \in \Sigma_1} \inf_{\sigma_2 \in \Sigma_2} \mathbb{E}_b^{\sigma_1,\sigma_2}[Y]$. If $\underline{V}(b) = \overline{V}(b)$ for all $b \in \mathbb{P}(S)$, then the common function is called the *value* of G, denoted by $V^\star$. A profile $\sigma^\star = (\sigma_1^\star, \sigma_2^\star)$ is a *minimax strategy profile* if for all $b \in \mathbb{P}(S)$, $\mathbb{E}_b^{\sigma_1^\star,\sigma_2^\star}[Y] \leq \mathbb{E}_b^{\sigma_1^\star,\sigma_2}[Y]$ for all $\sigma_2 \in \Sigma_2$ and $\mathbb{E}_b^{\sigma_1^\star,\sigma_2^\star}[Y] \geq \mathbb{E}_b^{\sigma_1,\sigma_2^\star}[Y]$ for all $\sigma_1 \in \Sigma_1$.

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 acting 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 it makes (finite-valued) observations of the environment's state, stored locally as *percepts*. The second agent can observe the local state and percept of the first agent, as well as the state of the environment directly.

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

- $S_1 = Loc_1 \times Per_1$ *is a set of states for* Ag_1*, where* $Loc_1 \subseteq \mathbb{R}^{b_1}$ *and* $Per_1 \subseteq \mathbb{R}^{d_1}$ *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* $A \coloneqq A_1 \times A_2$ *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 probabilistic local 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 continuous-state POSGs with discrete observations (agent states S_1) and actions for Ag_1, and continuous observations (states $S_1 \times S_E$) and discrete actions for Ag_2. Thus, Ag_1 is partially informed, without access to the environment state, while Ag_2 is fully informed. Since Ag_2 needs no observations, we omit its local state (and transition function).

The game executes as follows. A global state of C comprises a state $s_1 = (loc_1, per_1)$ for the agent Ag_1 (a local-state-percept pair) 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 updates its state 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', observes s_E' to generate a new percept $per_1' = obs_1(loc_1', s_E')$ and C reaches the global state $s' = ((loc_1', per_1'), s_E')$.

We allow any (deterministic) function obs_1 from the continuous environment and discrete local states to percepts. However, we here focus on perception functions implemented via (trained) neural networks $f : \mathbb{R}^{b_1+e} \to \mathbb{P}(Per_1)$, yielding scores over different percepts, from which the percept with the maximum score is selected. The restriction to deterministic functions with discrete outputs is well aligned with NN classifiers in applications, e.g., object detection. A polyhedral decomposition of the continuous state space can be obtained by computing the preimage of the (ReLU or ReLU approximated) perception function [23].

Motivating example: Pedestrian-vehicle interaction. A key challenge for autonomous driving in urban environments is predicting the intentions or actions of pedestrians. One solution is NN models, e.g., trained on video datasets [28,27]. We consider decision making for an autonomous vehicle using an NN-based intention estimation model for a pedestrian at a crossing [27]. We use their simpler

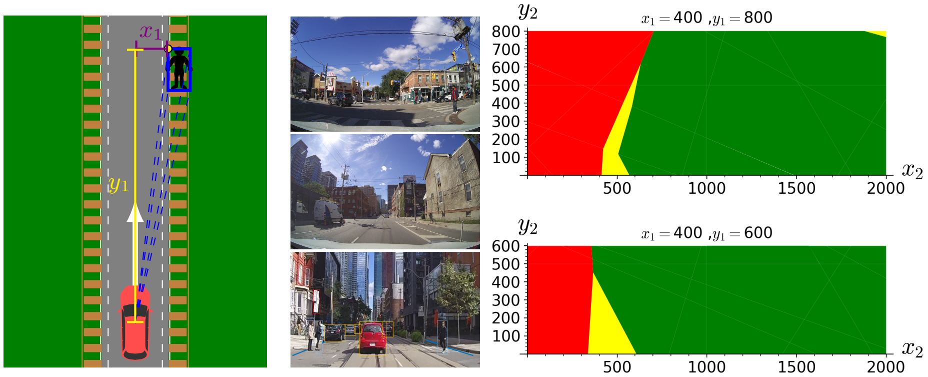

Fig. 1: Pedestrian-vehicle example. Left: Positions of two agents. Middle: Sample images from the PIE dataset [27]. Right: Slices of learnt perception function.

"vanilla" model, which takes the (relative) location of a pair of successive fixed-size bounding boxes around the pedestrian, and classifies intention as: *unlikely to cross*; *likely to cross*; *very likely to cross*. We train a feed-forward NN with ReLU activation functions over the PIE dataset [27].

We build this perception mechanism into an NS-POSG model of a vehicle yielding at a pedestrian crossing, based on [12] (see Figure 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, who perceives the environment (successive pedestrian positions $(x_1, y_1), (x_2, y_2)$) using an NN, storing the three possible intentions as percepts, and picks an acceleration action. Its local state also includes its speed. 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. Figure 1 also shows selected slices of the state space decomposition obtained by computing the preimage [23] of the learnt NN: *green*, *yellow* and *red* corresponding to classifications *not likely*, *likely* and *very likely* to cross, respectively. The goal of the vehicle is to minimise likelihood of collision with the pedestrian, which is achieved using a positive reward for each step without a crash. More details are given in Appx. F.

One-sided NS-POSG semantics. The semantics of a one-sided NS-POSG C is a POSG $[\![\mathsf{C}]\!]$ over the product of the (discrete) states of Ag_1 and the (continuous) states of the environment, restricting 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 (NS-POSG 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* $s' = (s'_1, s'_E) \in S$.

Since δ_E has finite branching and S_1 is finite, the branching set $\Theta_s^a = \{s' \mid \delta(s,a)(s') > 0\}$ is finite for all $s \in S$ and $a \in A$. Note that, while one-sided NS-POSGs are finite branching, they are not discrete.

One-sided NS-POSG Strategies. As $[\![\mathsf{C}]\!]$ is a POSG, we consider *(behaviour) strategies* for two agents. To align with the perfect information view of Ag_2, we assume that Ag_2 also has full information about the joint actions taken, through which it can recover the beliefs of Ag_1, thus removing nested beliefs. Hence, the AOHs of Ag_2 are equal to the histories of C, i.e., $FPaths_{[\![\mathsf{C}]\!],2} = FPaths_{[\![\mathsf{C}]\!]}$.

We also consider the *stage strategies* at a single decision point, i.e., a history of C, which are required for solving the induced zero-sum normal-formal games in the minimax operator. For a history π of 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 tuple of the form $b = (s_1, b_1)$, 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.

Finally, 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 $(s'_1, b_1^{s_1,a_1,u_2,s'_1})$ (see closed-form belief updates and probability measures involved in Appx. A).

4 Values of One-Sided NS-POSGs

We establish the *value* of a one-sided NS-POSG, which is a function from initial beliefs to values. We first show the convexity and continuity of the value function. Next, to compute it, we introduce the minimax operator and a maxsup operator specialised for one-sided NS-POSGs, and prove their equivalence. Finally, we provide a fixed-point characterization of the value function.

The value function. The *value function* of C (see Section 2) represents the minimax expected reward in each possible initial belief of the game and is given by $V^\star : S_B \to \mathbb{R}$, where $V^\star(s_1, b_1) = \mathbb{E}_{(s_1,b_1)}^{\sigma^\star}[Y]$ for all $(s_1, b_1) \in S_B$ and $\sigma^\star$ is a minimax strategy profile. The value for zero-sum POSGs may not exist when the state space is uncountable [13,2,29] as in our case. In this paper, we only consider one-sided NS-POSGs that are determined.

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 $U = \max_{s \in S, a \in A} r(s,a)/(1-\beta)$. We prove the following (this and all other results are proved in Appx. E).

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* $b_1, b_1' \in \mathbb{P}(S_E)$: $|V^\star(s_1, b_1) - V^\star(s_1, b_1')| \leq K(b_1, b_1')$ *where* $K(b_1, b_1') = \frac{1}{2}(U - L)\int_{s_E \in S_E^{s_1}} \big|b_1(s_E) - b_1'(s_E)\big| \mathrm{d}s_E$.

Minimax and maxsup operators. Since the sup inf and inf sup do not provide a straightforward recipe for computing value function $V^\star$, we provide a fixed-point characterization. We introduce a minimax operator and then simplify it 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. Given $f : S \to \mathbb{R}$ 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 defined:*

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

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

Minimising over $\mathbb{P}(A_2 \mid S)$ in (1) is challenging as both $\mathbb{P}(A_2 \mid S)$ and S are uncountable sets. Motivated by [17], which proposed a comparable 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 any $s \in S$, $f_{u_1, \overline{\alpha}, a_2}(s)$ equals the backup value at s if Ag_1 selects u_1, Ag_2 selects a_2 at s and retrieves values from $\overline{\alpha}$, i.e., we have (the summation over s_E' is due to the finite branching of δ):

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

Definition 4 (Maxsup) *If there exists* $\Gamma \subseteq \mathbb{F}(S)$ *such that for* $(s_1, b_1) \in S_B$, $V(s_1, b_1) = \sup_{\alpha \in \Gamma} \langle \alpha, (s_1, b_1)\rangle$*, then the maxsup operator* $T : \mathbb{F}(S_B) \to \mathbb{F}(S_B)$ *is defined as:* $[TV](s_1, b_1) = \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \langle f_{u_1, \overline{\alpha}}, (s_1, b_1)\rangle$ *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 all* $s \in S$.

In the maxsup operator, u_1 and $\overline{\alpha}$ are aligned with Ag_1's goal and both are optimised to maximise the objective in Definition 4, where u_1 is over action distributions and $\overline{\alpha}$ is over the convex combinations of functions in Γ. The minimisation by Ag_2 is simplified to an optimiation over the finite action set and occurs in constructing 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. Note that, 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 $a_2' \in \arg\min_{a_2 \in A_2} f_{u_1,\overline{\alpha},a_2}(s)$. The equivalence between the maxsup and minimax operators and the fixed-point result are stated as follows, respectively.

Theorem 2 (Operator equivalence). *The maxsup and minimax operators are equivalent over functions $V \in \mathbb{F}(S_B)$ where there exists $\mathit{\Gamma} \subseteq \mathbb{F}(S)$ such that $V(s_1, b_1) = \sup_{\alpha \in \mathit{\Gamma}} \langle \alpha, (s_1, b_1) \rangle$ for all $(s_1, b_1) \in S_B$.*

Theorem 3 (Fixed point). *The unique fixed point of T is $V^\star$.*

5 P-PWLC Value Iteration

We next show that *piecewise constant* (PWC) representations for the perception, reward and transition functions originally introduced for NS-POMDPs [35] are closed with respect to the maxsup operator, and thus also sufficient for one-sided NS-POSGs under mild assumptions. This representation, called P-PLWC, extends the α-functions of [26,6,36], except that we work with polyhedral representations induced from NNs, not Gaussian mixtures as in [26]. Building on this representation, we give a (non-scalable) value iteration (VI) algorithm and then, in Section 6, a more practical point-based HSVI algorithm.

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

Definition 5 (PWC function) *A function $f : S \to \mathbb{R}$ is piecewise constant (PWC) if there exists an FCP $\mathit{\Phi}$ of S such that $f : \phi \to \mathbb{R}$ is constant for all $\phi \in \mathit{\Phi}$. Such an FCP $\mathit{\Phi}$ is called constant-FCP of S for f.*

Since we use an NN for Ag_1's perception function obs_1, it is PWC (as for the one-agent case [35]) 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 $\mathit{\Phi}_P$*, the smallest FCP of S such that all states in any $\phi \in \mathit{\Phi}_P$ are observationally equivalent, i.e., if $(s_1, s_E), (s_1', s_E') \in \phi$, then $s_1 = s_1'$ and we let $s_1^\phi = s_1$. We can use $\mathit{\Phi}_P$ to find the set $S_E^{s_1'}$ for any agent state $s_1' \in S_1$ over which we integrate beliefs in closed form, see e.g., beliefs in Section 3. Given an NN representation of obs_1, the corresponding FCP $\mathit{\Phi}_P$ can be extracted (or approximated) offline by analyzing its pre-image [23].

In addition to this, we need to make some mild assumptions about a one-sided NS-POSG's transitions and reward functions (in a similar style to [35]). We describe this informally below, and defer a precise definition to Appx. B.

Assumption 1 (Transition and reward functions) *The functions δ_1 and r induce decompositions of the state space into a finite set of regions, so that states in a given region transition to the same region and states in the same region have the same rewards. The function δ_E is represented by a probabilistic choice over a finite number of continuous (deterministic) functions.*

Assumption 1 does not necessarily imply that $V^\star$ itself is PWC, as the continuous-state space S is typically continually subdivided as the computation of V^* progresses. We now show, using results for continuous-state POMDPs [35,26], that $V^\star$ is the limit of a sequence of α-functions, called *piecewise linear and convex under PWC α-functions (P-PWLC)*. This representation was first introduced in [35] for NS-POMDPs. Let $\mathbb{F}_C(S)$ be the subset of PWC functions of $\mathbb{F}(S)$.

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 all $(s_1, b_1) \in S_B$ where the functions in Γ are called PWC* α-functions*.*

Definition 6 implies that, if $V \in \mathbb{F}(S_B)$ is P-PWLC, then it can be represented by a set Γ of PWC continuous-state functions over S (i.e., as a finite set of FCP regions and a value vector). For one-sided NS-POSGs, we demonstrate that, under Assumption 1, a P-PWLC representation of value functions is closed under the maxsup operator and the convergence of value iteration.

Closure property. We first show that if V is P-PWLC, the maxsup operator $[TV](s_1, b_1)$ at a belief (s_1, b_1) can be computed by solving an LP. We prove that $f_{u_1,\overline{\alpha},a_2}$ in (2) is PWC for any $u_1 \in \mathbb{P}(A_1), \overline{\alpha} \in \Gamma^{A_1 \times S_1}$ and $a_2 \in A_2$ (see Lemma 7 in Appx. E). Then, there exists an FCP Φ_Γ of S such that $f_{u_1,\overline{\alpha},a_2}$ is constant in each region of Φ_Γ for all $u_1 \in \mathbb{P}(A_1), \overline{\alpha} \in \Gamma^{A_1 \times S_1}$ and $a_2 \in A_2$.

Lemma 1 (LP for maxsup and P-PWLC) *If $V \in \mathbb{F}(S_B)$ is P-PWLC with PWC α-functions Γ, then 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 $(p^{a_1})_{a_1 \in A_1}$:*

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

$$v_\phi \leq \textstyle\sum_{a_1 \in A_1} p^{a_1} r((s_1, s_E), (a_1, a_2)) + \beta \sum_{a_1, s_1', s_E'} \delta((s_1, s_E), (a_1, a_2))(s_1', s_E')$$
$$\cdot \textstyle\sum_{\alpha \in \Gamma} \lambda_\alpha^{a_1,s_1'} \alpha(s_1', s_E'), \;\; \lambda_\alpha^{a_1,s_1'} \geq 0, \;\; p^{a_1} = \sum_{\alpha \in \Gamma} \lambda_\alpha^{a_1,s_1'} \text{ and } \sum_{a_1 \in A_1} p^{a_1} = 1 \quad (3)$$

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 $s_E \in \phi$.

If $(\overline{v}^\star, \overline{\lambda}_1^\star, \overline{p}_1^\star)$ is the optimal solution to the LP (3), 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. We can now show that the P-PWLC representation is closed under the maxsup operator.

Theorem 4 (P-PWLC closure). *If $V \in \mathbb{F}(S_B)$ is P-PWLC, then so is $[TV]$.*

The closure property from Theorem 4 enables iterative computation of a sequence of such functions to approximate $V^\star$ to within a convergence guarantee.

Lemma 2 (P-PWLC convergence) *If $V^0 \in \mathbb{F}(S_B)$ is P-PWLC, then the sequence $(V^t)_{t=0}^\infty$, such that $V^{t+1} = [TV^t]$ are 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 agent states S_1, it is not scalable in practice.

6 Heuristic Search Value Iteration for NS-POSGs

To provide a more practical approach to solving one-sided NS-POSGs, we now present a variant of HSVI (heuristic search value iteration) [31], an anytime algorithm that approximates the value function $V^\star$ via lower and upper bound functions, updated through heuristically generated beliefs. HSVI was proposed for NS-POMDPs in [35] using P-PWLC functions and belief-value induced functions, ideas which we build upon to tackle one-sided NS-POSGs.

The presence of two agents with opposite goals brings three main challenges to developing an HSVI algorithm. First, the value backups at a belief point require solving normal-formal games instead of maximising over the actions of one agent. Second, since the first agent is not informed of the joint action, uncountably many possible stage strategies by the second agent in the maxsup operator have to be considered in the value backups and belief updates, whereas, in the single-agent variant, the agent can decide the transition probabilistically on its own. Third, the forward exploration heuristic is more complicated as the largest difference between the lower and upper bounds at the next-step belief depends on the stage strategies of two agents in two stage games. We now introduce the key ingredients of our one-sided variant of the NS-HSVI algorithm.

6.1 Lower and Upper Bound Representations

Lower bound function. Selecting an appropriate representation for α-functions requires closure properties with respect to the maxsup operator. Motivated by [35], 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 4. The lower bound V_{lb}^{Γ} has a finite representation as each α-function is PWC, and is initialized as in [17].

Upper bound function. The upper bound $V_{ub}^{\Upsilon} \in \mathbb{F}(S_B)$ is represented by a finite set of belief-value points $\Upsilon = \{((s_1^i, b_1^i), y_i) \in (S_1 \times \mathbb{P}(S_E)) \times \mathbb{R} \mid i \in I\}$ where y_i is an upper bound of $V^\star(s_1^i, b_1^i)$. Similarly to [35], for any $(s_1, b_1) \in S_1 \times \mathbb{P}(S_E)$ 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

$$\textstyle\sum_{i \in I_{s_1}} \lambda_i y_i + K_{ub}(b_1, \sum_{i \in I_{s_1}} \lambda_i b_1^i) \text{ subject to } \lambda_i \geq 0 \text{ and } \sum_{i \in I_{s_1}} \lambda_i = 1 \qquad (4)$$

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 $b_1, b_1', b_1'' \in \mathbb{P}(S_E)$: $K_{ub}(b_1, b_1) = 0$,

$$K_{ub}(b_1, b_1') \geq K(b_1, b_1') \quad \text{and} \quad |K_{ub}(b_1, b_1') - K_{ub}(b_1, b_1'')| \leq K_{ub}(b_1', b_1'') \, . \qquad (5)$$

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

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

Note that (4) 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 [35], K_{ub} here needs to satisfy an additional triangle property in (5) to ensure the continuity of V_{ub}^{Υ}, for the convergence of the point-based algorithm below. The properties on K_{ub} imply that (4) is an upper bound after a value backup, as stated in Lemma 4 below. The upper bound V_{ub}^{Υ} is initialized as in [17].

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:

$$\langle \alpha^\star, (s_1, b_1) \rangle = [TV_{lb}^{\Gamma}](s_1, b_1) = \langle f_{\overline{p}_1^\star, \overline{\alpha}^\star}, (s_1, b_1) \rangle \tag{6}$$

where the second equality follows from the operator equivalence in Theorem 2 and the LP (3), $(\overline{v}^\star, \overline{\lambda}_1^\star, \overline{p}_1^\star)$ is an optimal solution to the LP (3) at (s_1, b_1) and $\overline{\alpha}^\star \in \Gamma^{A_1 \times S_1}$ is the vector of PWC α-functions based on $\overline{\lambda}_1^\star$ and $\overline{p}_1^\star$.

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 (6) and improve the efficiency, we only compute the backup values for regions $\phi \in \Phi_P$ over which (s_1, b_1) has positive probabilities, i.e., $s_1^\phi = s_1$ (recall 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: $\alpha^\star(\hat{s}_1, \hat{s}_E) = f_{\overline{p}_1^\star, \overline{\alpha}^\star}(\hat{s}_1, \hat{s}_E)$ 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 5 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 [35]. The details of the ISPP backup for one-sided NS-POSGs are in Appx. C. The lemma also shows that the lower bound function increases and is valid after each update.

Lemma 3 (Lower bound) *The function $\alpha^\star$ generated by Algorithm 1 is a PWC α-function satisfying* (6)*, and if $\Gamma' = \Gamma \cup \{\alpha^\star\}$, then $V_{lb}^{\Gamma} \leq V_{lb}^{\Gamma'} \leq V^\star$.*

Upper bound updates. For the upper bound V_{ub}^{Υ}, due to representation (4), 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

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

```
1: while $V_{ub}^{\varUpsilon}(s_1^{init}, b_1^{init}) - V_{lb}^{\varGamma}(s_1^{init}, b_1^{init}) > \varepsilon$ do Explore$((s_1^{init}, b_1^{init}), 0)$
2: return $V_{lb}^{\varGamma}$ and $V_{ub}^{\varUpsilon}$ via sets $\varGamma$ and $\varUpsilon$
3: function Explore$((s_1, b_1), t)$
4:     $(u_1^{lb}, u_2^{lb}) \leftarrow$ minimax strategy profile in $[TV_{lb}^{\varGamma}](s_1, b_1)$
5:     $(u_1^{ub}, u_2^{ub}) \leftarrow$ minimax strategy profile in $[TV_{ub}^{\varUpsilon}](s_1, b_1)$
6:     Update$(s_1, b_1)$                                        ▷ Algorithm 1
7:     $(\hat{a}_1, \hat{s}_1) \leftarrow$ select according to forward exploration heuristic
8:     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
9:         Explore$((\hat{s}_1, b_1^{s_1, \hat{a}_1, u_2^{lb}, \hat{s}_1}), t + 1)$
10:        Update$(s_1, b_1)$                                    ▷ Algorithm 1
```

(1) and (4) requires the concrete formula for K_{ub} and the belief representations. Thus, we will show how to compute $[TV_{ub}^{\varUpsilon}](s_1, b_1)$ when introducing belief representations below. The following lemma shows that $y^\star \geq V^\star(s_1, b_1)$ required by (4), and the upper bound function is decreasing and is valid after each update.

Lemma 4 (Upper bound) *Given belief $(s_1, b_1) \in S_B$, if $y^\star = [TV_{ub}^{\varUpsilon}](s_1, b_1)$, 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 $\varUpsilon' = \varUpsilon \cup \{((s_1, b_1), y^\star)\}$, then $V_{ub}^{\varUpsilon} \geq V_{ub}^{\varUpsilon'} \geq V^\star$.*

6.2 One-Sided NS-HSVI Algorithm

Algorithm 2 presents the 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 [17], 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 $excess_t(s_1, b_1) = V_{ub}^{\varUpsilon}(s_1, b_1) - V_{lb}^{\varGamma}(s_1, b_1) - \rho(t)$, where $\rho(0) = \varepsilon$ and $\rho(t+1) = (\rho(t) - 2(U - L)\bar{\varepsilon})/\beta$, and $\bar{\varepsilon} \in (0, (1-\beta)\varepsilon/(2U - 2L))$. Then, the next action-observation pair $(\hat{a}_1, \hat{s}_1)$ for exploration is selected from:

$$\mathrm{argmax}_{(a_1, s_1') \in A_1 \times S_1} P(a_1, s_1' \mid (s_1, b_1), u_1^{ub}, u_2^{lb}) excess_{t+1}(s_1', b_1^{s_1, a_1, u_2^{lb}, s_1'}) . \quad (7)$$

To compute the next belief via lines 8 and 9, the minimax strategy profiles in stage games $[TV_{lb}^{\varGamma}](s_1, b_1)$ and $[TV_{ub}^{\varUpsilon}](s_1, b_1)$, i.e., (u_1^{ub}, u_2^{lb}), are required. Since $V_{lb}^{\varGamma}$ is P-PWLC, then using Lemma 1, the strategy u_2^{lb} is obtained by solving the dual of the LP (3). 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 5 (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: $V_{ub}^{\varUpsilon}(s_1^{init}, b_1^{init}) - V_{lb}^{\varGamma}(s_1^{init}, b_1^{init}) \leq \varepsilon$ and $V_{lb}^{\varGamma}(s_1^{init}, b_1^{init}) \leq V^\star(s_1^{init}, b_1^{init}) \leq V_{ub}^{\varUpsilon}(s_1^{init}, b_1^{init})$.*

6.3 Belief Representation and Computations

Implementing one-sided NS-HSVI depends on belief representations, as closed forms are needed. We consider the popular particle-based representation [35,26,10], which can approximate arbitrary beliefs and handle non-Gaussian systems.

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 normalized weights, 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 centered at 0. Let $P(s_E; b_1)$ be the probability of particle s_E under b_1.

To implement one-sided NS-HSVI using particle-based beliefs, we must demonstrate that V_{lb}^{Γ} and V_{ub}^{Υ} are eligible representations for particle-based beliefs, i.e., that closed forms exist for the quantities of interest. For a particle-based belief (s_1, b_1), we can compute $b_1^{s_1,a_1,u_2,s_1'}$, $\langle \alpha, (s_1, b_1) \rangle$, $\langle r, (s_1, b_1) \rangle$ and $P(a_1, s_1' \mid (s_1, b_1), u_1, u_2)$ as simple summations (see Appx. A).

Lower bound and stage game. Since V_{lb}^{Γ} 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, $V_{lb}^{\Gamma}(s_1, b_1) = \max_{\alpha \in \Gamma} \sum_{i=1}^{N_b} \kappa_i \alpha(s_1, s_E^i)$. Using Lemma 1, the stage game $[TV_{lb}^{\Gamma}](s_1, b_1)$ equals the optimal value of the LP (3). Solving (3) and its dual LP (see Appx. D), we obtain the minimax strategy profile (u_1^{lb}, u_2^{lb}).

Upper bound and stage game. To compute V_{ub}^{Υ} in (4), we need to design a function K_{ub} that measures belief differences that satisfy (5). We take $K_{ub} = K$. By the definition of K, K_{ub} satisfies (5) and $K_{ub}(b_1, b_1')$ is equal to:

$$K_{ub}(b_1, b_1') = \tfrac{1}{2}(U - L) \textstyle\sum_{b_1(s_E) + b_1'(s_E) > 0} |P(s_E; b_1) - P(s_E; b_1')| \, . \qquad (8)$$

Given $\Upsilon = \{((s_1^i, b_1^i), y_i) \mid i \in I\}$, the upper bound can be computed by solving an LP as demonstrated by the following lemma.

Lemma 5 (LP for upper bound) *Given the function K_{ub} from (8), and for particle-based belief (s_1, b_1), $V_{ub}^{\Upsilon}(s_1, b_1)$ is the optimal value of the LP:*

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

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

The minimax strategy profile (u_1^{ub}, u_2^{ub}) in the stage game $[TV_{ub}^{\Upsilon}](s_1, b_1)$ is obtained by solving an LP and its dual (see Appx. D), as demonstrated below.

Theorem 6 (LP for maxsup over upper bound). *For K_{ub} (see (8)) and particle-based belief (s_1, b_1), $[TV_{ub}^{\Upsilon}](s_1, b_1)$ is the optimal value of an LP.*

7 Experimental Evaluation

We have built a prototype implementation in Python, using Gurobi [14] to solve the LPs needed for computing lower and upper bound values, and the minimax

Model	Initial pts.	β	$\lvert\varGamma\rvert$	Lower bound		$\lvert\varUpsilon\rvert$	Upper bound		Iter.	Time (min)
				init.	final		init.	final		
Pursuit-evasion (3x3, 1 pursuer)	1	0.7	184	0	5.065266	265	333.33	9.181894	169	15
	1	0.7	515	0	5.279798	788	333.33	6.631739	264	120
	2	0.7	413	0	4.529885	998	333.33	11.570381	299	120
	1	0.8	468	0	9.882658	731	500	16.288952	170	120
	1	0.9	331	0	22.386704	731	1000	58.906245	130	120
Pursuit-evasion (3x3, 2 pursuers)	1	0.7	509	0	14.134097	790	333.33	39.943246	274	120
Pedestrian-vehicle	1	0.7	1928	0	620.537	4936	666.666	666.666	297	120
	2	0.7	2783	0	526.344	8532	666.666	666.666	363	120
	1	0.8	2089	0	805.924	5708	1000	1000	330	120

Table 1: Statistics for a set of one-sided NS-POSG solution instances.

values and strategies of one-shot games. We use the Parma Polyhedra Library [1] to operate over polyhedral preimages of NNs, α-functions and reward structures. The α-functions and reward functions are represented by associating values to polyhedra described as linear constraints over the continuous variables.

We developed two one-sided NS-POSG case studies for evaluation, a *pursuit-evasion* game and the *pedestrian-vehicle* scenario from Section 3. Table 1 shows statistics for solving various instances, varying the number of points in the initial belief and discount factor β. We show the initial/final values of the bounds, the number $|\varGamma|$ of α-functions generated, number $|\varUpsilon|$ of belief points for the upper bound computation, and iterations and time required (with a timeout of 2 hours)

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 $|\varGamma|$), as well as expensive operations over polyhedra and the probabilistic branching of mixed strategies to guide exploration. We note that HSVI for *finite* one-sided POSGs, in [17], is already computationally very expensive, even with multiple optimisations (they use a timeout of 10 hours, versus 2 hours here).

Pursuit-evasion. A pursuit-evasion game models a number of centrally controlled (pursuer) agents trying to capture an evader, aiming to avoid capture. We develop a continuous-space variant of the (discrete) model from [17] inspired by mobile robotics applications [8,18]. The pursuing agents use NNs as perception functions to determine their positions, while the evader is fully informed.

Figure 2 shows consecutive steps of the strategies synthesised for a 3×3 game with a single pursuer and $\beta = 0.7$, with the NN-induced polyhedral decomposition indicated in the top row. The strategies of the pursuer (red) and evader (green) are indicated by probabilistic transitions showing the direction of movement, and the pursuer's beliefs are shaded in green. Analysing these highlights interesting subtleties in both agents' behaviour. For instance, in the third step, the pursuer's strategy is to move to the bottom-right regions with equal probability since, not only do they account for most of the probability in the belief, but also the evader could still be in one of the three in the next step. The evader, however, is fully informed and knows where the pursuer is. Thus, its strategy in

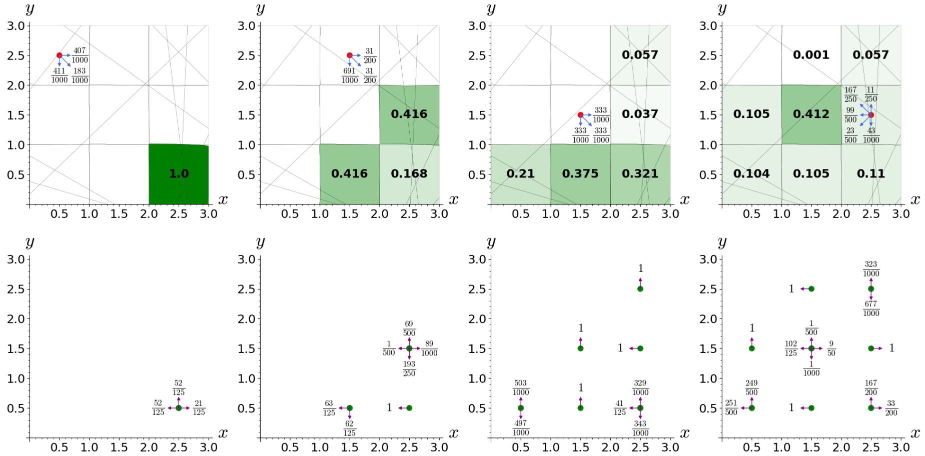

Fig. 2: Strategy and beliefs for the pursuer (top) and the evader (bottom).

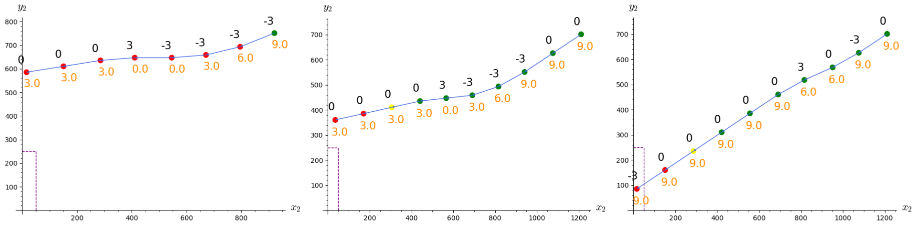

Fig. 3: Paths generated from strategies for the pedestrian-vehicle example.

those regions is to move to the position where the pursuer was is in the previous step or, if in the corner, to move up, left or stay with similar probabilities.

Pedestrian-vehicle interaction. Figure 3 shows paths generated from different strategies for the pedestrian-vehicle example, aiming to minimise the likelihood of a crash. We plot (x_2, y_2), the current relative distances between the vehicle and pedestrian. To generate these paths, we fix the pedestrian's action to progressively get closer to the vehicle so as to simulate a *crossing* scenario. Observations made by the vehicle are marked in *green*, *yellow* or *red* (predicted intentions *not likely*, *likely* and *very likely* to cross). Below and above each circle, we indicate the current speed and acceleration action taken, respectively. The crash area is the rectangle between the axes and the dashed lines.

We see that the synthesised strategies mostly produce safe paths, where the vehicle reduces its speed as it nears the pedestrian. However, there are paths where it does not and a crash occurs (see the rightmost plot in Figure 3). In this instance, the computation had not converged within the timeout, yielding strategies with residual probabilities associated to *unsafe* actions. We plan to consider finite-horizon objectives to try and address this.

8 Conclusions

We proposed one-sided neuro-symbolic POSGs, generalising NS-POMDPs [35] to the two-agent zero-sum case, and extending one-sided POSGs [17,38,39] to continuous state spaces. We characterised the value function for discounted infinite-horizon rewards, and are the first to design, implement and evaluate a practical HSVI algorithm for computing (approximately) optimal strategies for this model, and prove the algorithm's convergence. The computational complexity is high due to expensive polyhedra operations. Nevertheless, the techniques provide an important baseline that accounts for true decision boundaries for game models with neural perception mechanisms. As future work, we will consider restricted two-sided NS-POSGs, e.g., with public observations [16].

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 preprint 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. 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)
13. Ghosh, M.K., McDonald, D., Sinha, S.: Zero-sum stochastic games with partial information. Journal of optimization theory and applications **121**, 99–118 (2004)
14. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2021), gurobi.com
15. 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)
16. Horák, K., Bošanskỳ, B.: Solving partially observable stochastic games with public observations. In: Proc. AAAI'19. vol. 33, pp. 2029–2036 (2019)
17. Horák, K., Bošanskỳ, B., Kovařík, V., Kiekintveld, C.: Solving zero-sum one-sided partially observable stochastic games. Artificial Intelligence **316**, 103838 (2023)
18. Isler, V., Nikhil, K.: The role of information in the cop-robber game. Theoretical Computer Science **399**(3), 179–190 (2008)
19. Kovařík, V., Schmid, M., Burch, N., Bowling, M., Lisỳ, V.: Rethinking formal models of partially observable multiagent decision making. Artificial Intelligence **303**, 103645 (2022)

20. 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)
21. Kumar, A., Zilberstein, S.: Dynamic programming approximations for partially observable stochastic games. In: Pro. FLAIRS'09. vol. 147, pp. 547–552 (2009)
22. 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)
23. Matoba, K., Fleuret, F.: Computing preimages of deep neural networks with applications to safety (2020), openreview.netforum?id=FN7_BUOG78e
24. 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)
25. v. Neumann, J.: Zur theorie der gesellschaftsspiele. Mathematische annalen **100**(1), 295–320 (1928)
26. Porta, J.M., Vlassis, N., Spaan, M.T., Poupart, P.: Point-based value iteration for continuous POMDPs. JMLR **7**, 2329–2367 (2006)
27. 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)
28. 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)
29. Saha, S.: Zero-sum stochastic games with partial information and average payoff. Journal of Optimization Theory and Applications **160**(1), 344–354 (2014)
30. Sion, M.: On general minimax theorems. Pacific J. Math. **8**(1), 171–176 (1958)
31. Smith, T., Simmons, R.: Heuristic search value iteration for POMDPs. In: Proc. UAI'04. p. 520–527. AUAI (2004)
32. Wiggers, A.J., Oliehoek, F.A., Roijers, D.M.: Structure in the value function of two-player zero-sum games of incomplete information. Frontiers in Artificial Intelligence and Applications **285**, 1628 – 1629 (2016)
33. Yan, R., Santos, G., Norman, G., Parker, D., Kwiatkowska, M.: Strategy synthesis for zero-sum neuro-symbolic concurrent stochastic games. arXiv.2202.06255 (2022)
34. Yan, R., Santos, G., Duan, X., Parker, D., Kwiatkowska, M.: Finite-horizon equilibria for neuro-symbolic concurrent stochastic games. In: Proc. UAI'22. pp. 2170–2180. AUAI Press (2022)
35. Yan, R., Santos, G., Norman, G., Parker, D., Kwiatkowska, M.: Point-based value iteration for neuro-symbolic POMDPs. arXiv.2306.17639 (2023)
36. Zamani, Z., Sanner, S., Poupart, P., Kersting, K.: Symbolic dynamic programming for continuous state and observation POMDPs. Adv. Neural Inf. Process. Syst. **25** (2012)
37. Zettlemoyer, L., Milch, B., Kaelbling, L.: Multi-agent filtering with infinitely nested beliefs. Advances in neural information processing systems **21** (2008)
38. Zheng, W., Jung, T., Lin, H.: The Stackelberg equilibrium for one-sided zero-sum partially observable stochastic games. Automatica **140**, 110231 (2022)
39. 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)
40. 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 the details below.

Belief updates. Section 3 (p. 7) 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 $s_E' \in S_E$:

$$b_1^{s_1,a_1,u_2,s_1'}(s_E') = \frac{P((s_1', s_E') \mid (s_1, b_1), a_1, u_2)}{P(s_1' \mid (s_1, b_1), a_1, u_2)} \text{ if } s_E' \in S_E^{s_1'} \text{ and 0 otherwise.} \tag{9}$$

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 $s_E' \in S_E$:

$$b_1^{s_1,a,s_1'}(s_E') = \frac{P((s_1', s_E') \mid (s_1, b_1), a)}{P(s_1' \mid (s_1, b_1), a)} \text{ if } s_E' \in S_E^{s_1'} \text{ and 0 otherwise.} \tag{10}$$

Then, we show how to compute the probability measures in the belief updates (9) and (10). Recalling that $s_1 = (loc_1, per_1)$, for (9), using the syntax in Definition 1, $P(s_1' \mid (s_1, b_1), a_1, u_2)$ equals

$$\textstyle\int_{s_E \in S_E} b_1(s_E) \sum_{a_2 \in A_2} u_2(a_2 \mid s_1, s_E) \sum_{s_E' \in S_E} \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \mathrm{d}s_E \tag{11}$$

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

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

For (10), $P(s_1' \mid (s_1, b_1), a)$ equals

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

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

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

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 (9) 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:

$$\frac{\sum_{i=1}^{N_b} \kappa_i \sum_{a_2} u_2(a_2 \mid s_1, s_E^i) \delta((s_1, s_E^i), (a_1, a_2))(s_1', s_E')}{\sum_{i=1}^{N_b} \kappa_i \sum_{a_2} u_2(a_2 \mid s_1, s_E^i) \sum_{s_E''} \delta((s_1, s_E^i), (a_1, a_2))(s_1', s_E'')} \tag{12}$$

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.

B Assumptions on One-Sided NS-POSGs

We provide here formal definitions of our assumptions about the structure of one-sided NS-POSGs, captured informally in the main paper as Assumption 1.

Assumption 2 (Transitions) *For $a \in A$ and FCP Φ of S, there exists an FCP Φ' of S, called the* pre-image FCP *of Φ for a, where for $\phi \in \Phi$ and $\phi' \in \Phi'$ either $\Theta_s^a \cap \phi = \varnothing$ for all $s \in \phi'$ or $\Theta_s^a \cap \phi \neq \varnothing$ for all $s \in \phi'$, and if $s, \tilde{s} \in \phi'$, then $\sum_{s' \in \Theta_s^a \cap \phi} \delta(s,a)(s') = \sum_{\tilde{s}' \in \Theta_{\tilde{s}}^a \cap \phi} \delta(\tilde{s},a)(\tilde{s}')$. Furthermore, $\delta_E = \sum_{i=1}^{N_e} \mu_i \delta_E^i$ where $\delta_E^i : (Loc_1 \times S_E \times A) \to S_E$ is piecewise continuous, $\mu_i \geq 0$ and $\sum_{i=1}^{N_e} \mu_i = 1$.*

Assumption 3 (Rewards) *The reward function $r(\cdot, a) \to \mathbb{R}$ is bounded PWC for all $a \in A$. Therefore, For each joint action $a \in A$, there exists a smallest FCP of S, called the reward FCP under joint action a and denoted Φ_R^a, such that all states in any $\phi \in \Phi_R^a$ have the same rewards, i.e., if $s, s' \in \phi$, then $r(s,a) = r(s',a)$.*

C 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 [35], as used for a region-by-region backup in line 5 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. Recall from Assumption 1 (formally, from Assumption 2) 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 3, 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 *preimage* 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 5 of Algorithm 1 can be computed by finite backups.

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

```
Input: region $\phi$, action $\overline{p}_1^\star$, PWC functions $\overline{\alpha}^\star$
$\bar{A}_1 \leftarrow \{a_1 \in A_1 \mid \overline{p}_1^\star(a_1) > 0\}$
$Loc'_a \leftarrow \{loc'_1 \in Loc_1 \mid \delta_1(s_1^\phi, a)(loc'_1) > 0\}$ for $a \in \bar{A}_1 \times A_2$, $\varPhi_{\text{product}} \leftarrow \phi$
for $a = (a_1, a_2) \in \bar{A}_1 \times A_2, loc'_1 \in Loc'_a, i = 1, \dots, N_e$ do
    $\phi'_E \leftarrow \{\delta_E^i(s_E, a) \mid (s_1^\phi, s_E) \in \phi\}$                                   ▷ Image
    $\varPhi_{\text{image}} \leftarrow$ divide $\phi'_E$ into regions over $S$ by $obs_1(loc'_1, \cdot)$
    $\varPhi_{\text{split}} \leftarrow \varnothing$                                                          ▷ Split
    for $\phi_{\text{image}} \in \varPhi_{\text{image}}$ do
        $\varPhi_\alpha \leftarrow$ a constant-FCP of $S$ for the PWC function $\alpha^{\star a_1, s_1^{\phi_{\text{image}}}}$
        $\varPhi_{\text{split}} \leftarrow \varPhi_{\text{split}} \cup \{\phi_{\text{image}} \cap \phi' \mid \phi' \in \varPhi_\alpha\}$
    $\varPhi_{\text{pre}} \leftarrow \varnothing$                                                            ▷ Preimage
    for $\phi_{\text{image}} \in \varPhi_{\text{split}}$ do
        $\varPhi_{\text{pre}} \leftarrow \varPhi_{\text{pre}} \cup \{(s_1^\phi, s_E) \in \phi \mid \delta_E^i(s_E, a) \in \phi_{\text{image}}\}$
    $\varPhi_{\text{product}} \leftarrow \{\phi_1 \cap \phi_2 \mid \phi_1 \in \varPhi_{\text{pre}} \wedge \phi_2 \in \varPhi_{\text{product}}\}$        ▷ Product
$\varPhi_{\text{product}} \leftarrow \{\phi_1 \cap \phi_2 \mid \phi_1 \in \varPhi_{\text{product}} \wedge \phi_2 \in \sum_{a \in \bar{A}_1 \times A_2} \varPhi_R^a\}$
for $\phi_{\text{product}} \in \varPhi_{\text{product}}$ do                                                 ▷ Value backup
    Take one state $(\hat{s}_1, \hat{s}_E) \in \phi_{\text{product}}$
    $\alpha^\star(\phi_{\text{product}}) \leftarrow f_{\overline{p}_1^\star, \overline{\alpha}^\star}(\hat{s}_1, \hat{s}_E)$
return: $(\varPhi_{\text{product}}, \alpha^\star)$
```

Lemma 6 (ISPP backup) *The FCP $\varPhi_{\text{product}}$ returned by Algorithm 3 is a constant-FCP of ϕ for $\alpha^\star$ and the region-by-region backup for α^* satisfies the line 5 of Algorithm 1.*

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

According to Assumption 1, there exists a preimage-FCP of $\varPhi$ for each joint action a. Through the image, split, preimage and product operations of Algorithm 3, all the states in any region $\phi' \in \varPhi_{\text{product}}$ reach the same regions of $\varPhi$. Since each α-function α^{a_1, s'_1} is constant over each region in $\varPhi$, all states in ϕ' have the same backup value from α^{a_1, s'_1} for $a_1 \in \bar{A}_1$ and $s'_1 \in S_1$. This implies that $\varPhi_{\text{product}}$ is the product of the preimage-FCPs of $\varPhi$ for all $a \in \bar{A}_1 \times A_2$. Since the value backup in line 5 of Algorithm 1 is used for each region in $\varPhi_{\text{product}}$ and the image is from the region ϕ, then $\varPhi_{\text{product}}$ is a constant-FCP of ϕ for $\alpha^\star$, and thus the value backup in line 5 of Algorithm 1 for $\alpha^\star$ is achieved by considering the regions of $\varPhi_{\text{product}}$.

D 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 $[TV_{lb}^{\mathit{\Gamma}}](s_1, b_1)$ and $[TV_{ub}^{\mathit{\Upsilon}}](s_1, b_1)$. 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, the LP (3) for the stage game $[TV_{lb}^{\mathit{\Gamma}}](s_1, b_1)$ is simplified to the LP over the variables:

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

and is given by

$$
\begin{aligned}
&\text{maximise } \textstyle\sum_{i=1}^{N_b}\kappa_i v_{s_E^i} \text{ subject to}\\
&\quad v_{s_E^i} \le \textstyle\sum_{a_1\in A_1} p^{a_1} r((s_1, s_E^i),(a_1,a_2)) + \beta \textstyle\sum_{(a_1,s_1')\in A_1\times S_1, s_E'\in S_E}\\
&\qquad \delta((s_1,s_E^i),(a_1,a_2))(s_1',s_E')\textstyle\sum_{\alpha\in\mathit{\Gamma}}\lambda_\alpha^{a_1,s_1'}\alpha(s_1',s_E')\\
&\quad \lambda_\alpha^{a_1,s_1'} \ge 0\\
&\quad p^{a_1} = \textstyle\sum_{\alpha\in\mathit{\Gamma}}\lambda_\alpha^{a_1,s_1'}\\
&\quad \textstyle\sum_{a_1\in A_1} p^{a_1} = 1
\end{aligned}
\tag{13}
$$

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

The dual of LP problem (13) 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:

$$
\begin{aligned}
&\text{minimise } v \text{ subject to}\\
&\quad v \ge \textstyle\sum_{i=1}^{N_b}\sum_{a_2\in A_2} p_{a_2}^{s_1,s_E^i} r((s_1,s_E^i),(a_1,a_2)) + \beta\textstyle\sum_{s_1'\in S_1} v_{a_1,s_1'}\\
&\quad v_{a_1,s_1'} \ge \textstyle\sum_{i=1}^{N_b}\sum_{a_2\in A_2} p_{a_2}^{s_1,s_E^i}\delta((s_1,s_E^i),(a_1,a_2))(s_1',s_E')\alpha(s_1',s_E')\\
&\quad \textstyle\sum_{a_2\in A_2} p_{a_2}^{s_1,s_E^i} = \kappa_i
\end{aligned}
\tag{14}
$$

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

By solving (13) and (14), we obtain the minimax strategy profile in the stage game $[TV_{lb}^{\mathit{\Gamma}}](s_1, b_1)$: $u_1^{lb}(a_1) = p^{\star a_1}$ 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 $[TV_{ub}^{\mathit{\Upsilon}}](s_1, b_1)$ is over the variables:

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

and is given by

$$
\begin{aligned}
&\text{minimise } \ v \ \text{ subject to}\\
&\quad v \ge \textstyle\sum_{i=1}^{N_b}\sum_{a_2\in A_2}\kappa_i p^{s_1,s_E^i}_{a_2} r((s_1,s_E^i),(a_1,a_2))\\
&\qquad\quad + \beta\textstyle\sum_{s_1'\in S_1}\sum_{k\in I_{s_1'}}\lambda^{a_1,s_1'}_k y_k + \frac{1}{2}\beta(U-L)\sum_{s_1'\in S_1}\sum_{s_E'\in S_E^{a_1,s_1'}} c^{a_1,s_1'}_{s_E'}\\
&\quad c^{a_1,s_1'}_{s_E'} \ge \Big|\textstyle\sum_{i=1}^{N_b}\sum_{a_2\in A_2}\kappa_i p^{s_1,s_E^i}_{a_2}\delta((s_1,s_E^i),(a_1,a_2))(s_1',s_E')\\
&\qquad\quad - \textstyle\sum_{k\in I_{s_1'}}\lambda^{a_1,s_1'}_k P(s_E';b_1^k)\Big|\\
&\quad \textstyle\sum_{k\in I_{s_1'}}\lambda^{a_1,s_1'}_k = \sum_{i=1}^{N_b}\sum_{a_2\in A_2, s_E'\in S_E}\kappa_i p^{s_1,s_E^i}_{a_2}\delta((s_1,s_E^i),(a_1,a_2))(s_1',s_E')\\
&\quad \lambda^{a_1,s_1'}_k \ge 0\\
&\quad p^{s_1,s_E^i}_{a_2} \ge 0\\
&\quad \textstyle\sum_{a_2\in A_2} p^{s_1,s_E^i}_{a_2} = 1
\end{aligned}
\tag{15}
$$

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 (15) is the following LP problem over the variables:

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

and is given by:

$$
\begin{aligned}
&\text{maximise } \ \textstyle\sum_{i=1}^{N_b}\kappa_i v_{s_E^i} \ \text{ subject to}\\
&\quad v_{s_E^i} \le \textstyle\sum_{a_1\in A_1} p^{a_1} r((s_1,s_E^i),(a_1,a_2)) + \beta\sum_{a_1\in A_1, s_1'\in S_1, s_E'\in S_E^{a_1,s_1'}}\\
&\qquad\quad \delta((s_1,s_E^i),(a_1,a_2))(s_1',s_E')(v_{a_1,s_1'} + d_{a_1,s_1',s_E'} - e_{a_1,s_1',s_E'})\\
&\quad v_{a_1,s_1'} \le y_k p^{a_1} - \textstyle\sum_{s_E'\in S_E^{a_1,s_1'}}(d_{a_1,s_1',s_E'} - e_{a_1,s_1',s_E'})P(s_E';b_1^k)\\
&\quad d_{a_1,s_1',s_E'} - e_{a_1,s_1',s_E'} \le \tfrac{1}{2}(U-L)\\
&\quad d_{a_1,s_1',s_E'} \ge 0
\end{aligned}
$$

$$
\begin{aligned}
&e_{a_1,s_1',s_E'} \geq 0\\
&p^{a_1} \geq 0\\
&\textstyle\sum_{a_1 \in A_1} p^{a_1} = 1
\end{aligned} \tag{16}
$$

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 (15) and (16), 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 $a_2 \in A_2$.

E 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 $V^\star(s_1, b_1) = \sup_{\sigma_1 \in \Sigma_1} \inf_{\sigma_2 \in \Sigma_2} \mathbb{E}_{(s_1,b_1)}^{\sigma_1,\sigma_2}[Y]$. 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 $V_{\sigma_1}(s_1, b_1) = \inf_{\sigma_2 \in \Sigma_2} \mathbb{E}_{(s_1,b_1)}^{\sigma_1,\sigma_2}[Y]$. 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:

$$
V_{\sigma_1}(s_1, b_1) = \textstyle\int_{s_E \in S_E} b_1(s_E) \big(\inf_{\sigma_2 \in \Sigma_2} \mathbb{E}_{(s_1,s_E)}^{\sigma_1,\sigma_2}[Y] \big) \mathrm{d}s_E \,. \tag{17}
$$

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 [17, 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:

$$
\textstyle\int_{s_E \in S_E^{s_1}} b_1(s_E) \mathrm{d}s_E = \int_{s_E \in S_E^{s_1}} b_1'(s_E) \mathrm{d}s_E = 1 \,. \tag{18}
$$

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 (18) and using the fact that $S_E^{>} \cup S_E^{\leq} = S_E^{s_1}$ it follows that:

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

from which we have:

$$
\textstyle\int_{s_E \in S_E^{s_1}} |b_1(s_E) - b_1'(s_E)| \mathrm{d}s_E = \int_{s_E \in S_E^{>} \cup S_E^{\leq}} |b_1(s_E) - b_1'(s_E)| \mathrm{d}s_E
$$

$$
\begin{aligned}
&= \textstyle\int_{s_E \in S_E^{>}} (b_1(s_E) - b_1'(s_E))\mathrm{d}s_E - \int_{s_E \in S_E^{\leq}} (b_1(s_E) - b_1'(s_E))\mathrm{d}s_E \\
&= 2\textstyle\int_{s_E \in S_E^{>}} (b_1(s_E) - b_1'(s_E))\mathrm{d}s_E \qquad (19)
\end{aligned}
$$

and thus, using (19) and [35, Theorem 2], the inequality in Theorem 1 holds.

Theorem 7 (Operator equivalence - extended version of Theorem 2). *Given a function $V \in \mathbb{F}(S_B)$, if there exist a set $\mathit{\Gamma}$ of functions in $\mathbb{F}(S)$ such that $V(s_1, b_1) = \sup_{\alpha \in \mathit{\Gamma}} \langle \alpha, (s_1, b_1) \rangle$ for all $(s_1, b_1) \in S_B$, then the maxsup and minimax operators are equivalent, i.e., for $(s_1, b_1) \in S_B$ we have:*

$$
\begin{aligned}
[TV](s_1, b_1) &= \max\nolimits_{u_1 \in \mathbb{P}(A_1)} \min\nolimits_{u_2 \in \mathbb{P}(A_2 \mid S)} \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] \\
&\quad + \beta \textstyle\sum_{a_1 \in A_1} \sum_{s_1' \in S_1} P((a_1, s_1') \mid (s_1, b_1), u_1, u_2) V(s_1', b_1^{s_1,a_1,u_2,s_1'}) \qquad (20) \\
&= \min\nolimits_{u_2 \in \mathbb{P}(A_2 \mid S)} \max\nolimits_{u_1 \in \mathbb{P}(A_1)} \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] \\
&\quad + \beta \textstyle\sum_{a_1 \in A_1} \sum_{s_1' \in S_1} P((a_1, s_1') \mid (s_1, b_1), u_1, u_2) V(s_1', b_1^{s_1,a_1,u_2,s_1'}) \qquad (21) \\
&= \max\nolimits_{u_1 \in \mathbb{P}(A_1)} \sup\nolimits_{\overline{\alpha} \in \mathit{\Gamma}^{A_1 \times S_1}} \langle f_{u_1, \overline{\alpha}}, (s_1, b_1) \rangle \, . \qquad (22)
\end{aligned}
$$

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

$$
V(s_1, b_1) = \sup\nolimits_{\alpha \in \mathit{\Gamma}} \langle \alpha, (s_1, b_1) \rangle \quad \text{for all } (s_1, b_1) \in S_B. \qquad (23)
$$

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 (20) and (21) such that for $u_1 \in \mathbb{P}(A_1)$ and $u_2 \in \mathbb{P}(A_2 \mid S)$:

$$
\begin{aligned}
J(u_1, u_2) \;=\; & \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] + \\
& \beta \textstyle\sum_{a_1 \in A_1} \sum_{s_1' \in S_1} P(a_1, s_1' \mid (s_1, b_1), u_1, u_2) V(s_1', b_1^{s_1,a_1,u_2,s_1'}) \, . \qquad (24)
\end{aligned}
$$

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 $P_1 \coloneqq P(s_1' \mid (s_1, b_1), a_1, u_2)$ by (23) we have:

$$
\begin{aligned}
V(s_1', b_1^{s_1,a_1,u_2,s_1'}) \;&=\; \sup\nolimits_{\alpha \in \mathit{\Gamma}} \langle \alpha, (s_1', b_1^{s_1,a_1,u_2,s_1'}) \rangle \\
= \sup\nolimits_{\alpha \in \mathit{\Gamma}} & \textstyle\int_{s_E' \in S_E} \alpha(s_1', s_E') b_1^{s_1,a_1,u_2,s_1'}(s_E') \mathrm{d}s_E' && \text{rearranging} \\
= \sup\nolimits_{\alpha \in \mathit{\Gamma}} & \textstyle\int_{s_E' \in S_E} \alpha(s_1', s_E') \dfrac{P((s_1', s_E') \mid (s_1, b_1), a_1, u_2)}{P(s_1' \mid (s_1, b_1), a_1, u_2)} \mathrm{d}s_E' && \text{by (9)} \\
= \frac{1}{P_1} & \sup\nolimits_{\alpha \in \mathit{\Gamma}} \textstyle\int_{s_E' \in S_E} \alpha(s_1', s_E') P((s_1', s_E') \mid (s_1, b_1), a_1, u_2) \mathrm{d}s_E' && \text{rearranging} \\
= \frac{1}{P_1} & \sup\nolimits_{\alpha \in \mathit{\Gamma}} \big(\textstyle\int_{s_E' \in S_E} \alpha(s_1', s_E') \int_{s_E' \in S_E^{s_1'} \wedge s_E \in S_E} b_1(s_E) \sum_{a_2 \in A_2} u_2(a_2 \mid s_1, s_E) \\
& \qquad \cdot \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \mathrm{d}s_E \big) \mathrm{d}s_E' && \text{by (11)} \\
= \frac{1}{P_1} & \sup\nolimits_{\alpha \in \mathit{\Gamma}} \big(\textstyle\int_{s_E \in S_E} \big(\int_{s_E' \in S_E^{s_1'}} \alpha(s_1', s_E') \sum_{a_2 \in A_2} u_2(a_2 \mid s_1, s_E)
\end{aligned}
$$

$$\cdot\, \delta((s_1,s_E),(a_1,a_2))(s_1',s_E')\mathrm{d}s_E'\big)b_1(s_E)\mathrm{d}s_E \qquad \text{rearranging.} \quad (25)$$

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 $s = ((loc_1, per_1), s_E) \in S$:

$$\begin{aligned}\alpha^{a_1,u_2,s_1'}(s) &= \textstyle\int_{s_E' \in S_E^{s_1'}} \alpha(s_1',s_E') \sum_{a_2} u_2(a_2 \mid s)\delta(s,(a_1,a_2))(s_1',s_E')\mathrm{d}s_E' \\ &= \textstyle\sum_{a_2} u_2(a_2 \mid s) \sum_{s_E'} \delta(s,(a_1,a_2))(s_1',s_E')\alpha(s_1',s_E') \qquad (26)\end{aligned}$$

and the summation in s_E' is due to the finite branching of δ. Combining (25) and (26) we have:

$$\begin{aligned}V(s_1', b_1^{s_1,a_1,u_2,s_1'}) &= \frac{1}{P_1} \sup\nolimits_{\alpha \in \Gamma} \textstyle\int_{s_E \in S_E} \alpha^{a_1,u_2,s_1'}(s_1,s_E)b_1(s_E)\mathrm{d}s_E \\ &= \frac{1}{P(s_1' \mid (s_1,b_1),a_1,u_2)} \sup\nolimits_{\alpha \in \Gamma} \langle \alpha^{a_1,u_2,s_1'}, (s_1,b_1)\rangle \qquad (27)\end{aligned}$$

by definition of P_1. Substituting (27) into (24), the payoff function $J(u_1,u_2)$ equals:

$$\begin{aligned}&\mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] + \beta \textstyle\sum_{a_1,s_1'} u_1(a_1)P(s_1' \mid (s_1,b_1),a_1,u_2)V(s_1', b_1^{s_1,a_1,u_2,s_1'}) \\ &= \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] + \beta \textstyle\sum_{a_1,s_1'} u_1(a_1) \sup_{\alpha \in \Gamma} \langle \alpha^{a_1,u_2,s_1'}, (s_1,b_1)\rangle \,. \qquad (28)\end{aligned}$$

We next show that the von Neumann's Minimax Theorem [25] 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:

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

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 (26) we have that $f(u_2)$ equals:

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

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 $u_1(a_1)\sup_{\alpha \in \mathit{\Gamma}}\langle \alpha^{a_1,u_2,s_1'}, (s_1,b_1)\rangle$ 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 (20) and (21) holds.

Next we prove the equality of (20) and (22). Letting $\mathrm{Conv}(\mathit{\Gamma})$ be the convex hull of $\mathit{\Gamma}$, recall that $\mathit{\Gamma}^{A_1 \times S_1}$ is the set of vectors of functions in $\mathrm{Conv}(\mathit{\Gamma})$ indexed by the elements of $A_1 \times S_1$. The function $J(u_1,u_2)$ in (28) can be rewritten as follows:

$$\sup_{\overline{\alpha} \in \mathit{\Gamma}^{A_1 \times S_1}} \Big(\mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] \\ + \beta \textstyle\sum_{a_1 \in A_1, s_1' \in S_1} u_1(a_1)\langle \alpha^{a_1,u_2,s_1'}, (s_1,b_1)\rangle \Big) \qquad (29)$$

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 $\mathit{\Gamma}$ only depends on a_1 and s_1' and using the same arguments as [17, Proposition 4.11] we have:

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

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

$$\begin{aligned} J_{u_1}(\overline{\alpha},u_2) &= \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] + \beta \textstyle\sum_{a_1 \in A_1, s_1' \in S_1} u_1(a_1)\langle \alpha^{a_1,u_2,s_1'}, (s_1,b_1)\rangle \\ &= \mathbb{E}_{(s_1,b_1),u_1,u_2}[r(s,a)] + \beta \textstyle\sum_{a_1 \in A_1, s_1' \in S_1} u_1(a_1) \int_{s_E \in S_E} \big(\sum_{a_2 \in A_2} u_2(a_2 \mid s_1, s_E) \\ &\quad \cdot \textstyle\sum_{s_E' \in S_E} \delta((s_1,s_E),(a_1,a_2))(s_1',s_E')\alpha^{a_1,s_1'}(s_1',s_E')\big) b_1(s_E) \mathrm{d}s_E \quad \text{by (26).} \end{aligned} \qquad (30)$$

Substituting (29) and (30) into (20) we have:

$$\max_{u_1 \in \mathbb{P}(A_1)} \min_{u_2 \in \mathbb{P}(A_2|S)} J(u_1,u_2) \\ = \max_{u_1 \in \mathbb{P}(A_1)} \min_{u_2 \in \mathbb{P}(A_2|S)} \sup_{\overline{\alpha} \in \mathit{\Gamma}^{A_1 \times S_1}} J_{u_1}(\overline{\alpha},u_2)\,. \qquad (31)$$

We next show that Sion's Minimax Theorem [30] applies to the game with strategy spaces $\mathit{\Gamma}^{A_1 \times S_1}$ and $\mathbb{P}(A_2 \mid S)$ and payoff function J_{u_1}. Sion's Minimax Theorem requires that:

- $\mathit{\Gamma}^{A_1 \times S_1}$ 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 $J_{u_1}(\cdot,u_2) : \mathit{\Gamma}^{A_1 \times S_1} \to \mathbb{R}$ is upper semicontinuous and quasi-concave;
- for any $\overline{\alpha} \in \mathit{\Gamma}^{A_1 \times S_1}$ 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 (30) 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:

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

and combining with (31) 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:

$$\begin{aligned}
&\max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \varGamma^{A_1 \times S_1}} \min_{u_2 \in \mathbb{P}(A_2|S)} J_{u_1}(\overline{\alpha}, u_2) \\
&= \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \varGamma^{A_1 \times S_1}} \min_{u_2 \in \mathbb{P}(A_2|S)} \int_{s_E \in S_E} \sum\nolimits_{a_2} u_2(a_2 \mid s_1, s_E) \sum\nolimits_{a_1} u_1(a_1) \\
&\quad \cdot r((s_1, s_E), (a_1, a_2)) b_1(s_E) \mathrm{d}s_E + \beta \int_{s_E \in S_E} \Big(\sum\nolimits_{a_2} u_2(a_2 \mid s_1, s_E) \sum\nolimits_{a_1, s_1'} u_1(a_1) \\
&\quad \cdot \sum\nolimits_{s_E'} \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \alpha(s_1', s_E') \Big) b_1(s_E) \mathrm{d}s_E && \text{by (30)} \\
&= \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \varGamma^{A_1 \times S_1}} \int_{s_E \in S_E} \min_{u_2 \in \mathbb{P}(A_2|S)} \sum\nolimits_{a_2} u_2(a_2 \mid s_1, s_E) \\
&\quad \Big(\sum\nolimits_{a_1} u_1(a_1) r((s_1, s_E), (a_1, a_2)) + \beta \sum\nolimits_{a_1, s_1'} u_1(a_1) \\
&\quad \sum\nolimits_{s_E'} \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \alpha(s_1', s_E') \Big) b_1(s_E) \mathrm{d}s_E && \text{rearranging} \\
&= \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \varGamma^{A_1 \times S_1}} \int_{s_E \in S_E} \min_{a_2 \in A_2} \Big(\sum\nolimits_{a_1} u_1(a_1) r((s_1, s_E), (a_1, a_2)) \\
&\quad + \beta \sum\nolimits_{a_1, s_1'} u_1(a_1) \sum\nolimits_{s_E'} \delta((s_1, s_E), (a_1, a_2))(s_1', s_E') \alpha(s_1', s_E') \Big) b_1(s_E) \mathrm{d}s_E \\
& && \text{since } \mathsf{Ag}_2 \text{ is fully informed} \\
&= \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \varGamma^{A_1 \times S_1}} \int_{s_E \in S_E} \big(\min_{a_2 \in A_2} f_{u_1, \overline{\alpha}, a_2}(s_1, s_E) \big) b_1(s_E) \mathrm{d}s_E \\
& && \text{by (2)} \\
&= \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \varGamma^{A_1 \times S_1}} \langle f_{u_1, \overline{\alpha}}, (s_1, b_1) \rangle && \text{by Definition 4}
\end{aligned}$$

which demonstrates that (20) and (22) are equal and completes the proof.

*Proof (**Proof of Theorem 3**).* 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:

$$\begin{aligned}
V^\star(s_1, b_1) &= \sup_{\sigma_1 \in \varSigma_1} V_{\sigma_1}(s_1, b_1) \\
&= \sup_{\sigma_1 \in \varSigma_1} \int_{s_E \in S_E} b_1(s_E) \big(\inf_{\sigma_2 \in \varSigma_2} \mathbb{E}^{\sigma_1, \sigma_2}_{(s_1, s_E)}[Y] \big) \mathrm{d}s_E && \text{by (17)} \\
&= \sup_{\sigma_1 \in \varSigma_1} \langle \inf_{\sigma_2 \in \varSigma_2} \mathbb{E}^{\sigma_1, \sigma_2}_{(s_1, s_E)}[Y], (s_1, b_1) \rangle \\
&= \sup_{\alpha \in \varGamma} \langle \alpha, (s_1, b_1) \rangle
\end{aligned}$$

where $\varGamma \coloneqq \{ \inf_{\sigma_2 \in \varSigma_2} \mathbb{E}^{\sigma_1, \sigma_2}_{(s_1, s_E)}[Y] \mid \sigma_1 \in \varSigma_1 \}$. According to the operator equivalence in Theorem 2, we have:

$$[TV^\star](s_1, b_1) = \max_{u_1 \in \mathbb{P}(A_1)} \sup_{\overline{\alpha} \in \varGamma^{A_1 \times S_1}} \langle f_{u_1, \overline{\alpha}}, (s_1, b_1) \rangle \tag{32}$$

for all $(s_1, b_1) \in S_B$, where $\varGamma^{A_1 \times S_1} \coloneqq \{ \{\alpha^{a_1, s_1'}\}_{a_1 \in A_1, s_1' \in S_1} \mid \alpha^{a_1, s_1'} \in \mathrm{Conv}(\varGamma) \}$ and $\varGamma$ is given above. Now, by following the same argument as in the proof of [17, 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 (24) for J_1 and J_2, respectively. Without loss of generality, we assume $[TJ_1](s_1, b_1) \leq [TJ_2](s_1, b_1)$, and thus since $(u_1^{1\star}, u_2^{1\star})$ is minimax strategy profile for $[TJ_1](s_1, b_1)$:

$$\begin{aligned}
\bar{J}_1(u_1^{2\star}, u_2^{1\star}) &\leq \bar{J}_1(u_1^{1\star}, u_2^{1\star}) && \\
&= [TJ_1](s_1, b_1) && \text{by definition of } \bar{J}_1 \\
&\leq [TJ_2](s_1, b_1) && \text{without loss of generality} \\
&= \bar{J}_2(u_1^{2\star}, u_2^{2\star}) && \text{by definition of } \bar{J}_2 \\
&\leq \bar{J}_2(u_1^{2\star}, u_2^{1\star}) && \text{since } (u_1^{2\star}, u_2^{2\star}) \text{ is minimax strategy.} \qquad (33)
\end{aligned}$$

Now using (33) for any $(s_1, b_1) \in S_B$ we have

$$\begin{aligned}
&|[TJ_2](s_1, b_1) - [TJ_1](s_1, b_1)| \leq \bar{J}_2(u_1^{2\star}, u_2^{1\star}) - \bar{J}_1(u_1^{2\star}, u_2^{1\star}) \\
&= \beta \textstyle\sum_{a_1, s_1'} P(a_1, s_1' \mid (s_1, b_1), u_1^{2\star}, u_2^{1\star}) \big(J_2(s_1', b_1^{s_1, a_1, u_2^{1\star}, s_1'}) - J_1(s_1', b_1^{s_1, a_1, u_2^{1\star}, s_1'})\big) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{by (24)} \\
&\leq \beta \textstyle\sum_{a_1, s_1'} P(a_1, s_1' \mid (s_1, b_1), u_1^{2\star}, u_2^{1\star}) \|J_2 - J_1\| \quad \text{by definition of } \|\cdot\| \\
&= \beta \|J_2 - J_1\| \qquad\qquad \text{since } P(\cdot \mid (s_1, b_1), u_1^{2\star}, u_2^{1\star}) \text{ is a distribution.} \qquad (34)
\end{aligned}$$

Now by definition of the supremum norm:

$$\begin{aligned}
\|[TJ_2] - [TJ_1]\| &= \sup_{(s_1,b_1)\in S_B} |[TJ_2](s_1, b_1) - [TJ_1](s_1, b_1)| && \\
&\leq \sup_{(s_1,b_1)\in S_B} \beta \|J_2 - J_1\| && \text{by (34)} \\
&= \beta \|J_2 - J_1\| && \text{rearranging}
\end{aligned}$$

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 7 (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 (***Proof of Lemma 7***). 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 (formally, Assumption 2), there exists a preimage 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 $\Theta^a_s \cap \phi \neq \varnothing$ 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 2 we have $\sum_{s' \in \Theta^a_s \cap \phi} \delta(s,a)(s') = \sum_{\tilde{s}' \in \Theta^a_{\tilde{s}} \cap \phi} \delta(\tilde{s},a)(\tilde{s}')$ and $s_1 = \tilde{s}_1$. Now combining this fact with Definition 2, it follows that:

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

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 $S_E^{s_1'} = \{s_E' \in S_E \mid \mathit{obs}_1(\mathit{loc}_1', s_E') = \mathit{per}_1'\}$ 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

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

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'.

Proof (***Proof of Lemma 1***). Since V is P-PWLC, then according to Definitions 4 and 6 and Theorem 2:

$$\begin{aligned}[TV](s_1, b_1) &= \max\nolimits_{u_1 \in \mathbb{P}(A_1)} \sup\nolimits_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \langle f_{u_1,\overline{\alpha}}, (s_1, b_1) \rangle \\ &= \max\nolimits_{u_1 \in \mathbb{P}(A_1)} \sup\nolimits_{\overline{\alpha} \in \Gamma^{A_1 \times S_1}} \textstyle\int_{s_E \in S_E} \big(\min_{a_2} f_{u_1,\overline{\alpha},a_2}(s_1, s_E)\big) b_1(s_E) \mathrm{d}s_E \qquad (35)\end{aligned}$$

which can be formulated as the following optimization problem:

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

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:

$v_\phi \leq \sum_{a_1 \in A_1} u_1(a_1) r((s_1, s_E), (a_1, a_2)) + \beta \sum_{(a_1, s_1') \in A_1 \times S_1, s_E' \in S_E}$

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

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:

$$
\begin{aligned}
&\lambda_\alpha^{a_1,s_1'}\geq 0\\
&p_{a_1}=\textstyle\sum_{\alpha\in\Gamma}\lambda_\alpha^{a_1,s_1'}\\
&\textstyle\sum_{a_1\in A_1}p_{a_1}=1
\end{aligned}
$$

and hence the fact we can solve the LP problem (3) to compute $[TV](s_1,b_1)$ follows directly.

*Proof (**Proof of Theorem 4**).* Consider the LP in Lemma 1, 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 (3)). 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:

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

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.

*Proof (**Proof of Lemma 2**).* Using Theorem 3, the conclusion directly follows from Banach's fixed point theorem and the fact we have proved in Theorem 4 that if $V\in\mathbb{F}(S_B)$ and P-PWLC, so is $[TV]$.

*Proof (**Proof of Lemma 3**).* By following the proof of Theorem 4 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 (6).

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

In Algorithm 1, if the backup at line 5 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 6 is executed, $\alpha^\star$ is assigned the lower bound L over ϕ. Therefore we have for any $(\hat{s}_1,\hat{b}_1)\in S_B$:

$$\langle\alpha^\star,(\hat{s}_1,\hat{b}_1)\rangle\leq[TV_{lb}^\Gamma](\hat{s}_1,\hat{b}_1)$$

$$
\begin{aligned}
&\le [TV^\star](\hat{s}_1, \hat{b}_1) && \text{since } V_{lb}^{\Gamma} \le V^\star \\
&= V^\star(\hat{s}_1, \hat{b}_1) && \text{by Theorem 3.} \qquad (36)
\end{aligned}
$$

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

*Proof (**Proof of Lemma 4**).* Combining Theorem 1, (4) and (5), the conclusion can be obtained by following the argument in the proof of [35, Lemma 4] for NS-POMDPs.

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

Lemma 8 (Finite terminal belief points) *For any $t \ge 0$, 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 (**Proof of Lemma 8**).* Consider any $t \ge 0$ 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 $(s_1, b_1), (s_1, b_1') \in S_B$:

$$
|V_{lb}^{\Gamma}(s_1, b_1) - V_{lb}^{\Gamma}(s_1, b_1')| \le K(b_1, b_1') \,. \qquad (37)
$$

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 (4), i.e.,

$$
V_{ub}^{\Upsilon}(s_1, b_1') = \textstyle\sum_{i \in I_{s_1}} \lambda_i^{\star\prime} y_i + K_{ub}(b_1', \sum_{i \in I_{s_1}} \lambda_i^{\star\prime} b_1^i) \,. \qquad (38)
$$

Now since $(\lambda_i^{\star\prime})_{i \in I_{s_1}}$ satisfies the constraints in (4) for I_{s_1}, it follows that:

$$
\begin{aligned}
&V_{ub}^{\Upsilon}(s_1, b_1) \le \textstyle\sum_{i \in I_{s_1}} \lambda_i^{\star\prime} y_i + K_{ub}(b_1, \sum_{i \in I_{s_1}} \lambda_i^{\star} b_1^i) \\
&= \big(V_{ub}^{\Upsilon}(s_1, b_1') - K_{ub}(b_1', \textstyle\sum_{i \in I_{s_1}} \lambda_i^{\star} b_1^i)\big) + K_{ub}(b_1, \sum_{i \in I_{s_1}} \lambda_i^{\star\prime} b_1^i) && \text{by (38)} \\
&= V_{ub}^{\Upsilon}(s_1, b_1') + \big(K_{ub}(b_1, \textstyle\sum_{i \in I_{s_1}} \lambda_i^{\star\prime} b_1^i) - K_{ub}(b_1', \sum_{i \in I_{s_1}} \lambda_i^{\star} b_1^i)\big) && \text{rearranging} \\
&\le V_{ub}^{\Upsilon}(s_1, b_1') + K_{ub}(b_1, b_1') && \text{by (5).}
\end{aligned}
$$

Using similar steps we can also show that:

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

and hence:

$$
|V_{ub}^{\Upsilon}(s_1, b_1) - V_{ub}^{\Upsilon}(s_1, b_1')| \le K_{ub}(b_1, b_1') \,. \qquad (39)
$$

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 (7) (from line 7 of Algorithm 2) satisfies

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

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.,

$$V_{ub}^{\Upsilon}(s_1', b_1^{s_1^t, a_1, u_2^{lb}, s_1'}) - V_{lb}^{\Gamma}(s_1', b_1^{s_1^t, a_1, u_2^{lb}, s_1'}) \leq \rho(t+1)\,. \tag{40}$$

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 (24) via V_{lb}^{Γ} and V_{ub}^{Υ}, respectively. Thus, since (u_1^{lb}, u_2^{lb}) is a minimax strategy profile:

$$\begin{aligned}
J^{lb}(u_1^{ub}, u_2^{lb}) &\leq J^{lb}(u_1^{lb}, u_2^{lb}) \\
&= [TV_{lb}^{\Gamma}](s_1^t, b_1^t) && \text{by definition of } J^{lb} \\
&\leq [TV_{ub}^{\Upsilon}](s_1^t, b_1^t) && \text{by Lemmas 3 and 4} \\
&= J^{ub}(u_1^{ub}, u_2^{ub}) && \text{by definition of } J^{ub} \\
&\leq J^{ub}(u_1^{ub}, u_2^{lb}) && (u_1^{ub}, u_2^{ub}) \text{ is a minimax strategy profile.}
\end{aligned} \tag{41}$$

Now using (41) we have:

$$\begin{aligned}
&[TV_{ub}^{\Upsilon}](s_1^t, b_1^t) - [TV_{lb}^{\Gamma}](s_1^t, b_1^t) \leq J^{ub}(u_1^{ub}, u_2^{lb}) - J^{lb}(u_1^{ub}, u_2^{lb}) \\
&= \beta \textstyle\sum_{a_1, s_1' \in A_1 \times S_1} P(a_1, s_1' \mid (s_1^t, b_1^t), u_1^{ub}, u_2^{lb}) \\
&\quad (V_{ub}^{\Gamma}(s_1', b_1^{s_1^t, a_1, u_2^{lb}, s_1'}) - V_{lb}^{\Gamma}(s_1', b_1^{s_1^t, a_1, u_2^{lb}, s_1'})) && \text{by (24)} \\
&\leq \beta \textstyle\sum_{a_1, s_1' \in A_1 \times S_1} P(a_1, s_1' \mid (s_1^t, b_1^t), u_1^{ub}, u_2^{lb}) \rho(t+1) && \text{by (40)} \\
&= \beta \rho(t+1) && \text{since } P \text{ is a distribution.}
\end{aligned} \tag{42}$$

Substituting (42) 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 (37) and (39), for any $(s_1, b_1), (s_1, b_1') \in S_B$, we have

$$V_{ub}^{\Upsilon}(s_1, b_1) - V_{lb}^{\Gamma}(s_1, b_1) \leq V_{ub}^{\Upsilon}(s_1, b_1') - V_{lb}^{\Gamma}(s_1, b_1') + 2K_{ub}(b_1, b_1')\,. \tag{43}$$

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 (43) into the excess gap $excess_t(s_1^t, b_1)$:

$$\begin{aligned}
&excess_t(s_1^t, b_1) \leq V_{ub}^{\Upsilon}(s_1^t, b_1^t) - V_{lb}^{\Gamma}(s_1^t, b_1^t) + 2K_{ub}(b_1, b_1^t) - \rho(t) \\
&\beta\rho(t+1) + 2K_{ub}(b_1, b_1^t) - \rho(t) && \text{by (42)} \\
&\leq \rho(t) - 2(U-L)\bar{\varepsilon} + 2K_{ub}(b_1, b_1^t) - \rho(t) && \text{by definition of } \rho(t+1) \\
&\leq -2(U-L)\bar{\varepsilon} + 2(U-L)\bar{\varepsilon} && \text{since } K_{ub}(b_1, b_1^t) \leq (U-L)\bar{\varepsilon}
\end{aligned}$$

$= 0$ rearranging

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 5***). 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., $\Psi_t := \Psi_t \cup \{s_B^t\}$. 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 8, the algorithm has to terminate. Then, combining Lemmas 3 and 4, the conclusion follows directly.

Proof (***Proof of Lemma 5***). The result follows directly from (4) and (8).

Theorem 8 (LP for minimax operator over upper bound – extended version of Theorem 6). *For the function K_{ub}, see (8), 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 (15).*

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 (4) 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 (4) 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 (8) and (46) we have:

$$K_{ub}(\tau b_1 + \tau' b_1', \textstyle\sum_{k\in I_{s_1}} (\tau\lambda_k^\star + \tau'\lambda_k'^\star) b_1^k)$$

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

Next, from (4) we have:

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

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

The inequality (39) shows that $V_{ub}^{\Upsilon}(s_1, \cdot)$ is continuous in $\mathbb{P}(S_E)$. By following the proof of [17, 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 $(s_1, b_1) \in S_B$:

$$\begin{aligned}
[T V_{ub}^{\Upsilon}](s_1, b_1) &= \max_{u_1 \in \mathbb{P}(A_1)} \min_{u_2 \in \mathbb{P}(A_2 | S)} \mathbb{E}_{(s_1, b_1), u_1, u_2}[r(s, a)] \\
&\qquad + \beta {\textstyle\sum}_{a_1, s_1'} P(a_1, s_1' \mid (s_1, b_1), u_1, u_2) V_{ub}^{\Upsilon}(s_1', b_1^{s_1, a_1, u_2, s_1'}) \\
&= \min_{u_2 \in \mathbb{P}(A_2 | S)} \max_{u_1 \in \mathbb{P}(A_1)} \mathbb{E}_{(s_1, b_1), u_1, u_2}[r(s, a)] \\
&\qquad + \beta {\textstyle\sum}_{a_1, s_1'} P(a_1, s_1' \mid (s_1, b_1), u_1, u_2) V_{ub}^{\Upsilon}(s_1', b_1^{s_1, a_1, u_2, s_1'}) \, . \qquad (50)
\end{aligned}$$

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:

$$\begin{aligned}
&J(u_1, u_2) = E_1 + \beta {\textstyle\sum}_{a_1, s_1'} p^{a_1} p^{a_1, u_2, s_1'} V_{ub}^{\Upsilon}(s_1', b_1^{s_1, a_1, u_2, s_1'}) \\
&= E_1 + \beta {\textstyle\sum}_{a_1, s_1' \in A_1 \times S_1} p^{a_1} p^{a_1, u_2, s_1'} \min_{(\lambda_k)_{k \in I_{s_1'}}} \\
&\quad \Big({\textstyle\sum}_{k \in I_{s_1'}} \lambda_k y_k + K_{ub}\big(b_1^{s_1, a_1, u_2, s_1'}, {\textstyle\sum}_{k \in I_{s_1'}} \lambda_k b_1^{s_1, a_1, u_2, s_1'}\big) \Big) \qquad \text{by (4).}
\end{aligned}$$

Now combining this with (8) 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 $\overline{\nu} = (\nu_k^{a_1,s_1'})_{(a_1,s_1') \in A_1 \times S_1, k \in I_{s_1'}}$ and $\overline{c} = (d_{s_E'}^{a_1,s_1'})_{(a_1,s_1') \in A_1 \times S_1, s_E' \in S_E^{a_1,s_1'}}$ are real-valued vectors of variables subject to the following linear constraints

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

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:

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

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 $u_1 \in \mathbb{P}(A_1)$:

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

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

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

We next show that the von Neumann's Minimax Theorem [25] 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:

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

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 $[TV_{ub}^{\Upsilon}](s_1, b_1)$ can be formulated as the following LP problem:

$$\begin{aligned}
&\text{minimise } \ v \ \text{ subject to} \\
&v \geq E_1 + \beta {\textstyle\sum_{s_1'}} p^{a_1, u_2, s_1'} \big({\textstyle\sum_{k \in I_{s_1'}}} \nu_k^{a_1, s_1'} y_k + C^{a_1, s_1'} \big) \ \text{ for all } a_1 \in A_1.
\end{aligned} \tag{55}$$

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 follows:

$$\begin{aligned}
&\min_{u_2, \lambda, \hat{c}, \hat{v}, v} v \ \text{ such that} \\
&v \geq {\textstyle\sum_{i=1}^{N_b} \sum_{a_2}} \kappa_i p_{a_2}^{s_1, s_E^i} r((s_1, s_E^i), (a_1, a_2)) + \beta {\textstyle\sum_{s_1'}} v_{a_1, s_1'} \\
&v_{a_1, s_1'} = {\textstyle\sum_{k \in I_{s_1'}}} \lambda_k^{a_1, s_1'} y_k + \tfrac{1}{2}(U - L) {\textstyle\sum_{s_E' \in S_E^{a_1, s_1'}}} \hat{c}_{s_E'}^{a_1, s_1'}
\end{aligned}$$

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 (9):

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

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:

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

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 (15).

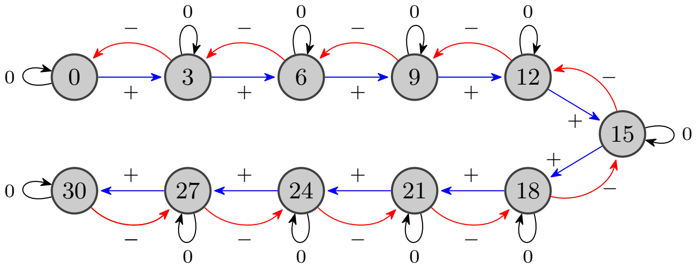

Fig. 4: Pedestrian-vehicle interaction: local transition diagram over the vehicle speeds with + for positive acceleration, − for negative acceleration, and 0 for zero acceleration.

F Further Case Study Details

Finally, we give some additional details 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 $Loc_1 = \{30, 27, 24, 21, 18, 15, 12, 9, 6, 3, 0\}$ (local states) are the speeds (km/h) of the vehicle and $Per_1 = \{1, 2, 3\}$ are the perceived pedestrian intentions with 1 representing *unlikely to cross*, 2 *likely to cross* and 3 *very likely to cross*.
- $S_E = \{(x_1, y_1, x_2, y_2) \in \mathbb{R}^4 \mid 0 \leq x_1, x_2 \leq 20, 0 \leq y_1, y_2 \leq 10\}$ (m), where $[(x_1, y_1), (x_1 + L_x, y_1 - L_y)]$ and $[(x_2, y_2), (x_2 + L_x, y_2 - L_y)]$ are the top-left and bottom-right points of the 2D bounding boxes (of fixed size L_x by L_y) around the pedestrian at the last 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 to choose to move.
- The perception function $obs_1 : S_E \to Per_1$ is a data-driven pedestrian intention estimation model implemented via a feed-forward NN with ReLU activation functions and trained over the PIE dataset in [27].
- For $(v_1, per_1) \in Loc_1 \times Per_1$, $v_1' \in Loc_1$ and $(a_1, a_2) \in A$,

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

  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 $(a_1, a_2) \in A$, if

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

  then $\delta_E(v_1, (x_1, y_1, x_2, y_2), (a_1, a_2))(x_1', y_1', x_2', y_2') = 1$, where $v_2 = 4.5$ (m/s) is the speed of the running pedestrian, a_2 is the direction of the movement of the pedestrian action, e.g., $a_2 = -1$ for *cross* and $a_2 = 1$ for *back*, and $\Delta t = 0.3$ (s)

A crash occurs if the environment state is in the set

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

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 (the bounding box has size $L_x = 0.5$ and $L_y = 1.5$ (m)). In the reward structure, all action rewards are zero and the state reward function is such that for any $(s_1, s_E) \in S$: $r_S(s_1, s_E) = 0$ if $s_E \in \mathcal{R}_{crash}$ and 200 otherwise.

Pursuit-evasion game. We modify the example presented in [17] by considering a continuous environment $\mathcal{R} = \{(x, y) \in \mathbb{R}^2 \mid 0 \leq x, y \leq 3\}$ that is partitioned into multiple cells by their perception functions. In this game, we have a pair of centrally controlled pursuers $\{P_1, P_2\}$ that try to catch an evader E. In each step, the evader moves by picking from the set of actions $A_e = \{\mathit{up}, \mathit{down}, \mathit{left}, \mathit{right}\}$. The pursuers move in a similar manner, but as we consider them to be a centrally controlled entity, they can be modelled as a single agent with action set $A_p = A_e \times A_e$. The perception function of the pursuers uses an NN classifier $f : \mathcal{R} \to Per$, where $Per = \{(i, j) \mid i \in \{1, \dots, 3\}, j \in \{1, \dots, 3\}\}$, which takes the location (coordinates) of a player as input and outputs one of the 9 abstract grid points (cells), thus partitioning the environment. The pursuers are partially observable, that is, they know which cell they are in, but do not know their exact location and do not know which cell the evader is in as well as its exact location. However, the evader is fully observable and knows the exact locations of all players. The capture condition in [17] is also used, that is, the evader is captured if it is in the same regression cell with at least one pursuer, which means the capture states $\mathcal{R}_{capture}$ are given by

$$\begin{aligned}\{(x_{p_1}, y_{p_1}, x_{p_2}, y_{p_2}, x_e, y_e) \in & S_E \mid \exists k \in \{1, 2\}, \exists (i, j) \in Per, \\ & \text{subject to } i - 1 \leq x_{p_k}, x_e < i, j - 1 \leq y_{p_k}, y_e < j\}\,.\end{aligned}$$

This is modelled as a one-sided NS-POSG as follows:

- $S_1 = Loc_1 \times Per_1$, where $Loc_1 = \varnothing$ and $Per_1 = Per \times Per$.
- $S_E = \mathcal{R}^3 = \{(x_{p_1}, y_{p_1}, x_{p_2}, y_{p_2}, x_e, y_e) \in \mathbb{R}^6 \mid (x_i, y_i) \in \mathcal{R}, i \in \{p_1, p_2, e\}\}$.
- $A = A_1 \times A_2$, where $A_1 = A_p$ and $A_2 = A_e$.
- The perception function $obs_1 : S_E \to Per_1$ is implemented via a feed-forward NN f with one hidden ReLU layer and 14 neurons, takes the coordinate vector of the pursuers as input and then outputs a pair of the 9 abstract grid points.
- For $s_E = (x_{p_1}, y_{p_1}, x_{p_2}, y_{p_2}, x_e, y_e), s'_E = (x'_{p_1}, y'_{p_1}, x'_{p_2}, y'_{p_2}, x'_e, y'_e) \in S_E$, $loc_1 \in Loc_1$ and $a \in A$, $\delta_E(loc_1, s_E, a)(s'_E)$ is equal to

$$\begin{cases} 1 & \text{if } s_E \in \mathcal{R}_{capture} \text{ and } s_E = s'_E \\ \prod_{i \in \{p_1, p_2, e\}} \delta_{Ei}((x_i, y_i), d_{ai})(x'_i, y'_i) & \text{if } s_E \notin \mathcal{R}_{capture} \\ 0 & \text{otherwise} \end{cases}$$

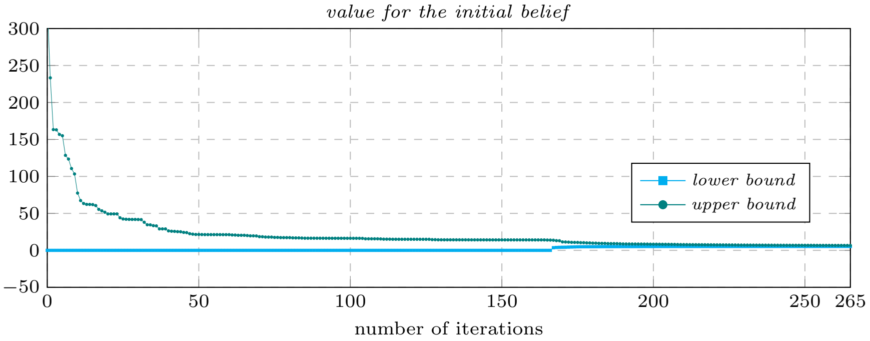

Fig. 5: Lower and upper bound values for a pursuit-evasion game (3 × 3, one pursuer, $\beta = 0.7$.

where for $i \in \{p_1, p_2, e\}$, if $x''_i = x_i + d^x_{ai}\mathit{\Delta} t$ and $y''_i = y_i + d^y_{ai}\mathit{\Delta} t$, then

$$\delta_{Ei}((x_i, y_i), d_{ai})(x'_i, y'_i) = \begin{cases} 1 \text{ if } (x''_i, y''_i) \in \mathcal{R} \text{ and } (x'_i, y'_i) = (x''_i, y''_i) \\ 1 \text{ if } (x''_i, y''_i) \notin \mathcal{R} \text{ and } (x'_i, y'_i) = (x_i, y_i) \\ 0 \text{ otherwise} \end{cases}$$

where $d_a = (d_{ap_1}, d_{ap_2}, d_{ae})$ indicates the direction of movement of a for each agent and $d_{ai} = (d^x_{ai}, d^y_{ai})$, e.g., $d_{(up,up,up)} = ((0,1),(0,1),(0,1))$, and $\mathit{\Delta} t$ is the time step.

As the environment transition δ_E indicates, the evader is captured if at any point the environment state is in the set $\mathcal{R}_{capture}$ and then the game ends by keeping the state consistent afterwards. In case the pursuers are successful, that is, if at least one of them enters the same regression cell as the evader, the team receives a reward of 100. The reward for all other states is zero. All action rewards are zero. For the model with a single pursuer, in contrast to [17], as well as being able to move vertically or horizontally, it can also move diagonally. The evader, however, cannot move diagonally but has the option of staying still when in one of the border cells, which the pursuer is not allowed to do. Instead of stopping when capture happens as in [17], the game continues indefinitely in all models. Figure 5 shows in more detail how the computed values for lower and upper bounds change as more iterations are performed.