Analyzing Probabilistic Logic Shields for Multi-Agent Reinforcement Learning

Satchit Chatterji[a,*] **and Erman Acar**[a]

[a]IvI & ILLC, University of Amsterdam

Abstract.

Safe reinforcement learning (RL) is crucial for real-world applications, and multi-agent interactions introduce additional safety challenges. While Probabilistic Logic Shields (PLS) has been a powerful proposal to enforce safety in single-agent RL, their generalizability to multi-agent settings remains unexplored. In this paper, we address this gap by conducting extensive analyses of PLS within decentralized, multi-agent environments, and in doing so, propose **Shielded Multi-Agent Reinforcement Learning (SMARL)** as a general framework for steering MARL towards norm-compliant outcomes. Our key contributions are: (1) a novel Probabilistic Logic Temporal Difference (PLTD) update for shielded, independent Q-learning, which incorporates probabilistic constraints directly into the value update process; (2) a probabilistic logic policy gradient method for shielded PPO with formal safety guarantees for MARL; and (3) comprehensive evaluation across symmetric and asymmetrically shielded n-player game-theoretic benchmarks, demonstrating fewer constraint violations and significantly better cooperation under normative constraints. These results position SMARL as an effective mechanism for equilibrium selection, paving the way toward safer, socially aligned multi-agent systems.

1 Introduction

Recent years have witnessed significant progress in multi-agent reinforcement learning (MARL), with sophisticated algorithms tackling increasingly complex problems in various domains including autonomous vehicles [29], distributed robotics [5], algorithmic trading [14], energy grid management [34] and healthcare [27]. The ultimate success of RL in this diverse collection of domains and the deployment in the real-world, however, demands overcoming a difficult key challenge: *safety*.

This has naturally resulted in the research direction of *Safe RL* which aims to learn optimal policies that are, by some measure, 'safe'. Gu et al. [16] present an up-to-date overview of the field. A number of proposals focus on the application of formal methods [19, 11], within which the notion of *shielding* is used, a technique that is inspired by formal verification through temporal logic specifications to avoid unsafe actions during the agent's learning process [4, 2, 20, 6].

One recent proposal that uses shielding to represent safety constraints is *probabilistic logic shields* [PLS, 38] whose semantics are based on probabilistic logic (PL) programming [9]. PLS constrains an agent's policy to comply with formal specifications *probabilistically*. The specification of constraints, (the *shield*), is defined within the exploration and learning pipeline. PLS offers policy-level evaluation of safety in contrast to action-level hard-rejection shields [e.g., 19, 20, 6], differentiability, modest requirements on knowing the MDP, and safety guarantees within single-agent RL.

* Corresponding Author. Email: s.chatterji@uva.nl

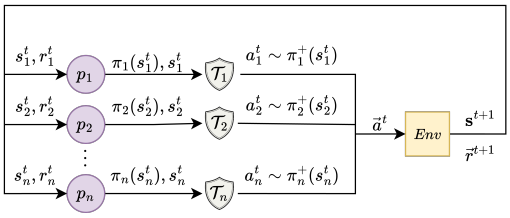
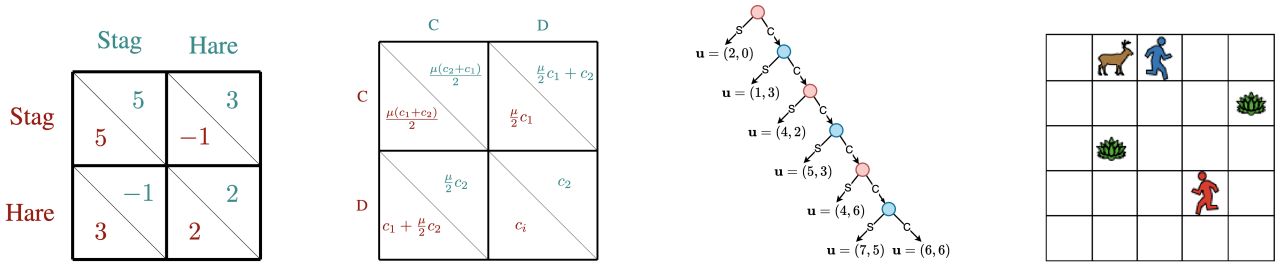


Figure 1. Shielded MARL (SMARL) interaction schematic. At time t, each agent p_i passes their respective policy $\pi_i^t(s_i^t)$ and a safety-specific state s_i^t to their shield $\mathcal{T}_i$, resulting in a safe policy $\pi_i^+(s_i^t)$ from which an action a_i^t is sampled and returned to the environment.

However, safety is inherently a *multi-agent* concept, as real-world environments often involve multiple agents interacting simultaneously, leading to hard-to-control complex systems. While there exist a few approaches tackling safety in multi-agent settings [15, 12, 23, 30], to the best of our knowledge, PLS has not been adopted, extended or analyzed in the context of MARL. In this paper, we address this gap with the following contributions:

1. We present a framework that extends shielding in RL to multi-agent settings, named *Shielded MARL* (SMARL). We provide a theoretical guarantee that SMARL with PLS produces safer joint policies than unshielded counterparts. Using decentralized techniques, we introduce (a) *Probabilistic Logic Temporal Difference Learning* (PLTD), with convergence guarantees, allowing for shielded independent Q-learning (SIQL), and (b) shielded independent PPO (SIPPO) using probabilistic logic policy gradients;
2. We show that PLS can be used as an equilibrium selection mechanism, a key challenge in MARL, under various game-theoretic settings, such as coordination, spatio-temporal coordination, social dilemmas, cooperation under uncertainty, and mixed-motive problems. We provide strong empirical evidence across various n-player environments including an extensive-form game (Centipede), a stochastic game (Extended Public Goods Game), a simultaneous game (Stag-Hunt), and its grid-world extension (Markov Stag-Hunt). Moreover, we investigate the impact of smaller (weak, less specific) and larger (strong, more specific) shields;
3. We investigate asymmetric shielding, i.e. the ability of shielded agents to influence unshielded peers. Results show that partial shielding can significantly enhance safety, highlighting SMARL's effectiveness in both cooperative and non-cooperative settings.



(a) Payoff matrix for *Stag-Hunt* **(b)** Expected EPGG utilities **(c)** *Centipede* instance **(d)** *Markov Stag-Hunt* [26]

Figure 2. Representations of the games that are experimented within this paper. For details please refer to Section 2.3.

2 Preliminaries

2.1 Safety Definition

We expand the interpretation of 'safety' in RL beyond that of works like ElSayed-Aly et al. [12] and Yang et al. [38] (i.e. informally, *the agent should not perform actions that may lead to something 'harmful'*), and more towards an alternative conception provided in Gu et al. [16, pg. 3], namely, *they act, reason, and generalize obeying human desire*. We formalize this in the vein of Alshiekh et al. [2]:

Definition 2.1. Formally-Constrained Safe RL is the process of learning an optimal policy π^* while maximally satisfying a set of formal specifications $\mathcal{T}$ during learning and deployment.

$\mathcal{T}$ may represent any set of constraints, (e.g. arbitrary temporal logic formulae). These may not *necessarily* be 'harmful', but may instead ascribe normative or desirable behavior, such as social norms or equilibrium conditions. In our case, $\mathcal{T}$ is a ProbLog program.

2.2 Multi-Agent Reinforcement Learning

Multi-Agent Reinforcement Learning (MARL) generalizes RL to environments with multiple interacting agents, making the learning problem non-stationary since each agent's dynamics depend on the evolving policies of others ([1] provides a detailed introduction). Formally, MARL is often modeled as a *stochastic game* in which each agent i selects actions according to a local policy π_i, and inducing a joint policy $\vec{\pi} = \langle \pi_1, \ldots, \pi_n \rangle$ determining the next state and rewards. A common baseline is *Independent Q-Learning* [IQL, 31], where each agent maintains its own Q-function and updates it independently using local observations and rewards. Similarly, *Independent Proximal Policy Optimization* [IPPO, 10] adapts PPO to the multi-agent case by training each one with a separate policy-gradient update. In practice, MARL algorithms may employ *parameter sharing*, where agents use a common neural architecture for policies and/or critics, differing only through their inputs (e.g., observations). This has been shown to improve sample efficiency and facilitates coordination, but may reduce policy diversity and lead to homogenized behaviors [1, 17].

2.3 Game-Theoretic Environments

A normal-form game (NFG) is a tuple $\langle N, \mathbf{A}, \mathbf{u} \rangle$ where $N = \{1, ..., n\}$ is a finite set of agents (or players), $\mathbf{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ is a finite set of *action profiles* $\mathbf{a} = (a_1, ...a_n)$ with $\mathcal{A}_i$ being the set of player i's actions, and $\mathbf{u} = (u_1, ..., u_n)$ is a profile of utility functions $u_i : \mathbf{A} \to \mathbb{R}$. A strategy profile $\mathbf{s} = (s_1, ...s_n)$ is called a *Nash equilibrium* if for each player $k \in N$, the strategy s_k is a best response to the strategies of all the other players $s_{i \in N \setminus \{k\}}$ [21, 22].

► **Stag-Hunt** is a two-player NFG in which each player has the actions $\mathcal{A}_1 = \mathcal{A}_2 = \{Stag, Hare\}$. If they coordinate and both play *Stag*, each will get a large reward. However, if only one plays *Stag*, it gets a penalty (low/negative utility). Alternatively, either agent may unilaterally play *Hare* to receive a small positive utility regardless of the actions of the other agent. An example Stag-Hunt *payoff matrix* (a representation of the agents' utility functions) is seen in Figure 1a.

► **Extended Public Goods Game** (EPGG) is a mixed-motive simultaneous game that represents the *cooperation vs. competition* social dilemma [25]. An EPGG is a tuple $\langle N, c, \mathbf{A}, f, u \rangle$, where $N = \{1, \ldots, n\}$ is the set of players, $c_i \in \mathbb{R}_{\geq 0}$ is the amount of coin each player i is endowed with and collected in $c = (c_1, \ldots, c_n)$, and $f \in \mathbb{R}_{\geq 0}$ is the multiplication factor for the lump sum endowment (hence the name 'extended', as opposed to the case $f \leq n$). Each player $i \in N$ decides whether to invest in the public good (*cooperate*) or not (*defect*), i.e., $\mathcal{A}_i = \{C, D\}$. The resulting quantity is then evenly distributed among all agents. The utility function for i is defined as $u_i : \mathbf{A} \times \mathbb{R}_{\geq 0} \times \mathbb{R}^n_{\geq 0} \to \mathbb{R}$, with: $r_i(\mathbf{a}, f, c) = \frac{1}{n} \sum_{j=1}^{n} c_j I(a_j) \cdot f + c_i(1 - I(a_i))$ where $\mathbf{a}$ is the action profile, a_j is the $j-$th entry of $\mathbf{a}$, $I(a_j)$ is the indicator function returning 1 if the action of the agent j is cooperative and 0 otherwise, and c_j denotes the $j-$th entry of c. Here, EPGG is formulated as a partially observable stochastic game in which $f := f_t \sim \mathcal{N}(\mu, \sigma)$, sampled every time step t, where $\mathcal{N}(\mu, \sigma)$ is a normal distribution with mean μ and variance σ. Depending on the value of μ, the game is expected to be non-cooperative ($\mu < 1$), cooperative ($\mu > n$) or a mix of both (i.e., mixed-motive for $1 < \mu < n$). The *expected* Nash equilibrium is determined by $\mu = \mathbb{E}_{f_t \sim \mathcal{N}(\mu, \sigma)}[f_t]$, and can be estimated empirically ($\hat{\mu}$) by taking the mean of several observations of f_t. A generic two-player EPGG payoff matrix is shown in Figure 1b.

► **Centipede** is a two-player extensive-form game (i.e., the game has several states) in which two agents take turns deciding whether to continue the game (increasing the potential rewards for both) or defect (ending the game and collecting a short-term reward). The game structure incentivizes short-term defection, as each agent risks being defected upon by their partner. Formally, the game begins with a 'pot' p_0 (a set amount of utility) where each player has two actions, $\mathcal{A}_i = \{Continue, Stop\}$. After each round t, the pot increases, e.g., linearly (such as $p_{t+1} \leftarrow p_t + 1$) or exponentially (such as $p_{t+1} \leftarrow p_t^2$). If a player plays $Stop$, then it receives $p_t/2 + 1$ while the other player receives $p_t/2 - 1$. If both players play $Continue$, they split the pot equally after a certain amount of rounds t_{max}, each receiving $p_{t_{max}}/2$. An instance of the game ($t_{max} = 6$, $n = 2$, $p_0 \leftarrow 2$, $p_{t+1} = p_t + 2$) is given in Figure 1c.

► **Markov Stag-Hunt** is a grid-world environment inspired by Peysakhovich and Lerer [26], where agents move through a grid and must decide whether to hunt a stag (with risk of penalty if hunting

alone) or harvest a plant, with an underlying reward structure similar to the *Stag-Hunt* game. Figure 1d visualizes this environment.

2.4 Probabilistic Logic Shielding

We summarize Yang et al. [38]'s approach known as *Probabilistic Logic Shielding* (PLS), a method for incorporating probabilistic safety in RL, and refer the reader to the original work for a detailed treatment. Unlike earlier rejection-based shields that fully block unsafe actions and limit exploration [2], PLS reduces the probability of unsafe actions proportional to their risk, allowing them to occasionally occur. Under soft constraints, this lets agents learn from unsafe actions.

Assume a model P such that $P(\texttt{safe}|s,a)$ represents the likelihood that taking action a in state s is safe. The safety of π is the sum of the disjoint probabilities of the safety of taking each action:

$$P_\pi(\texttt{safe} \mid s) = \sum_{a \in \mathcal{A}} P(\texttt{safe} \mid s, a) \cdot \pi(a \mid s) \tag{1}$$

By marginalizing the base policy over the safety model, we obtain a reweighted policy that favors safer actions – this is called *probabilistic shielding*. Formally, given a base policy π and a probabilistic safety model P, the shielded policy π^+ is constructed as:

$$\pi^+(a \mid s) = P_\pi(a \mid s, \texttt{safe}) = \frac{P(\texttt{safe} \mid s, a)}{P(\texttt{safe} \mid s)} \pi(a \mid s) \tag{2}$$

Succinctly, PLS re-normalizes the action distribution to make unsafe actions less likely. Yang et al. [38] chose to implement the model P in ProbLog [9] for a number of reasons, including that it is easily differentiable and allows easy modeling and planning. This means our safety constraints are to be specified *symbolically* through PL predicates and relations. Within this paper, a *shield* is thus synonymous with a ProbLog program $\mathcal{T}$ (further details in Section 3.3), which induces a probabilistic measure $\mathbf{P}_\mathcal{T}$. We can (conditionally) query this program to give us various information including the action safety (for each action a) as $P(\texttt{safe}|s,a) = \mathbf{P}_\mathcal{T}(\texttt{safe}|a)$, the safety of the whole policy under s as $P_\pi(\texttt{safe}|s) = \mathbf{P}_\mathcal{T}(\texttt{safe})$ and the safe/shielded policy π^+ as $P_\pi(a|s,\texttt{safe}) = \mathbf{P}_\mathcal{T}(a|\texttt{safe})$.

3 Probabilistic Logic Shields for Multi-Agent Reinforcement Learning

We present a general SMARL framework in Section 3.1, applying (probabilistic) shielding to multi-agent algorithms. Next, in Section 3.2, we introduce PLTD, which incorporates logical constraints directly into the TD-learning process. Finally, Section 3.3 outlines the encoding of safety constraints as ProbLog programs.

3.1 General Framework Description

A schematic diagram of a parallel SMARL setup is in Figure 1. Each agent p_i (indexed with $i \in \{1, ..., n\}$) observes a possibly subjective state s_i^t at time t, and sends both the computed policy $\pi_i(s_i^t)$ and relevant safety-related information about s_i^t to its shield $\mathcal{T}_i$. Each $\mathcal{T}_i$ is a ProbLog program that is used to compute a safe policy $\pi_i^+(s_i^t)$ according to Eq. 2 from which an action a_i^t is sampled. Finally, the joint action profile $\vec{a}^t$ is sent to the environment Env. After updating its state, Env sends new observations $\mathbf{s}^{t+1}$ and rewards $\vec{\mathbf{r}}^{t+1}$ to the agents, and the cycle continues with $t \leftarrow t+1$ until a terminal state is reached. The *agent-environment-cycle interaction* scheme, used often in turn-based games [32] can likewise be extended naturally to the SMARL framework by having each agent interact with their own shield before sampling an action.

Though SMARL is agnostic to the underlying MARL algorithm, we experiment with two: IQL and IPPO (introduced in Section 2.2). IQL with parameter sharing will be called *Parameter Sharing IQL (PSQL)*, and IPPO will be called either *Critic Sharing IPPO (CSPPO)* or *Actor-Critic Sharing IPPO (ACSPPO)*, depending on whether the actors and/or critics are shared. In **Probabilistic Logic SMARL (PL-SMARL)**, at least one of the RL agents uses PLS, optionally sharing policies and parameter updates. PL-SMARL algorithms will be denoted *'Shielded X' (SX)*; e.g., *'Shielded IQL', (SIQL)* when $X =$ IQL, or *'Shielded CSPPO', (SCSPPO)* when $X =$ CSPPO.

3.1.1 Safety Guarantees

Following Definition 2.1, we define relative safety for SMARL:

Definition 3.1 (Relative SMARL Safety)**.** A joint policy $\vec{\pi}^+ = \langle \pi_1^+, \pi_2^+, \ldots, \pi_n^+ \rangle$ is defined to be **at least as safe** as another joint policy $\vec{\pi} = \langle \pi_1, \pi_2, \ldots, \pi_n \rangle$ if and only if for all agents $i \in N$ with respective shields $\mathcal{T}_i$, it is the case that $P_{\pi_i^+}(\texttt{safe}|s) \geq P_{\pi_i}(\texttt{safe}|s)$ for all reachable states $s \in S$; and is **strictly safer** if *additionally* for at least one $j \in N$ and $s \in S$, $P_{\pi_j^+}(\texttt{safe}|s) > P_{\pi_j}(\texttt{safe}|s)$.

Based on Definition 3.1, we attain the following proposition:

Proposition 3.2. *A PL-SMARL algorithm with agents $i \in N$ with respective policies π_i^+ and shields $\mathcal{T}_i$ has a joint shielded policy $\vec{\pi}^+$ **at least as safe** as their base joint policy $\vec{\pi}$, and **strictly safer** whenever any individual base policy π_j violates $\mathcal{T}_j$.*

A proof is provided in the supplementary materials [7]. This is similar to the first case of ElSayed-Aly et al. [12]'s correctness proof of the safety of factored shields. However, unlike factored shields, PLS does not guarantee *per-step* safety; rather, it ensures that the shielded algorithm's safety is higher than that of the base algorithm *in expectation*.

3.2 PL-Shielded Temporal Difference Learning (PLTD)

Yang et al. [38] define their shielded algorithm, probabilistic logic policy gradients (PLPG), as an extension of policy gradient methods, specifically, PPO [28]. Thus, to use it with DQN [24], SARSA [40] (or other TD-learning methods), formalize the use of a ProbLog shield in conjunction with TD-learning. An important consideration for extending PLS to TD-learning is whether the algorithm is *on-policy* or *off-policy*. Yang et al. [38] note that for off-policy algorithms to converge, the exploration and learned policies must cover the same state-action space – this assumption must be handled with care in the design of exploration strategies. We propose both on- and off-policy algorithms that incorporate safety constraints into Q-learning.

3.2.1 Objective Function

PLPG includes a *shielded policy gradient* $[\nabla_\theta \log P_{\pi^+}(\texttt{safe}|s)]$ and a *safety gradient penalty* $[-\nabla_\theta \log P_{\pi^+}(\texttt{safe}|s)]$. Since TD-methods do not use policy gradients, we must rely on a modified version of the latter to introduce safety constraints into the objective function.

Let $\mathcal{D}$ be the distribution of $d = \langle s^t, a^t, r^t, s^{t+1}, a^{t+1} \rangle$ tuples extracted from a history buffer of the agent interacting with the MDP and s^t, a^t, r^t be the state, taken action, and reward at time t. Let the Q-value approximator $Q_\theta(s,a)$ be parameterized by some parameters θ. The off-policy loss (Q-learning based) and on-policy loss (SARSA-based) are augmented using the aforementioned safety penalty term. We refer to these methods as *Probabilistic Logic TD Learning* (PLTD).

Definition 3.3 (Probabilistic Logic TD Learning)**.** The PLTD minimization objective is:

$$\mathcal{L}^{Q^+}(\theta) = \mathbb{E}_{d\sim\mathcal{D}}\Big[\Big(r^t + \gamma\mathcal{X} - Q_\theta(s^t, a^t)\Big)^2 - \alpha S_P\Big] \quad (3)$$

where $\mathcal{X} = \max_{a'} Q_\theta(s^{t+1}, a')$ for off-policy, and $\mathcal{X} = Q_\theta(s^{t+1}, a^{t+1})$ for on-policy DQN; $S_P = \log P_{\pi^+}(\texttt{safe}|s)$ is the safety penalty; and $\alpha \in \mathbb{R}_{\geq 0}$ is the safety coefficient, or the weight of the safety penalty.

The quantity $[-\log P_{\pi^+}(\texttt{safe}|s)]$ is interpreted as *the probability that π^+ satisfies the safety constraints* [38]. Under perfect sensor information, PLTD reduces to vanilla TD-learning within the reachable set of safe states. We thus get the following convergence property (following Tsitsiklis and Van Roy [33]):

Proposition 3.4. *PLTD, i.e. Eq. 3 converges to an optimal safe policy given perfect safety information in all states in tabular and linear function approximation settings.*

We prove this in [7]. For neural Q-networks, global convergence is currently unsolved, though we can adopt common DQN stabilization tricks (summarized in [18]). While we observe empirical stability of PLTD, a complete theoretical analysis is left for future work.

3.3 Shield Construction using ProbLog

A shield in PLS is defined using a ProbLog program [9], denoted $\mathcal{T}$. Each consists of three components: (i) an annotated disjunction Π_s of predicates whose values are set to a base action distribution π, (ii) a set of constraint-related inputs describing the current state $\mathbf{H}_s$, and (iii) a set of sentences $\mathcal{KB}$ (for *knowledge base*) that specify the agent's constraints, including a definition for a predicate '`safe_next`' (the predicted safety of the next state under π). Again, 'safety' broadly refers to any probabilistic constraint satisfaction goal (ref. Sec 2.1).

An exemplary shield (Shield 1) for constraining agent behavior towards the mixed Nash equilibrium in Stag-Hunt games is described as such: The probabilistic valuations of the '`action(·)`' predicates are inferred from π. The '`sensor(·)`' predicates are set to the normalized absolute difference between the precomputed mixed Nash strategy and the mean of the agent's historical actions. The predicate '`safe_next`' is inferred via ProbLog semantics [8] and determines the safety of the next state under π – safer states are those where the stated normalized absolute differences are low. For brevity, we describe subsequent shields in-line rather than displaying its full ProbLog source. For details on how we designed the shields for each experiment, we kindly direct the reader to [7].

```
% actions
action(0)::action(stag);
action(1)::action(hare).
% sensors
sensor_value(0)::sensor(stag_diff).
sensor_value(1)::sensor(hare_diff).
% safety constraints
unsafe_next :- action(stag),
               sensor(stag_diff).
unsafe_next :- action(hare),
               sensor(hare_diff).
safe_next :- \+unsafe_next.
```

Shield 1. Shield ($\mathcal{T}_{mixed}$) for mixed *Stag-Hunt* equilibrum.

3.3.1 Deterministic Shields

Using PLS, agents may be made to be deterministic which can be useful when training agents in environments requiring hard constraints. This can be formulated as the following (proof in [7]):

Proposition 3.5. *For any game there exists a shield $\mathcal{T}$ such that the resulting shielded agent deterministically selects a single action $a \in \mathcal{A}$ for each state $s \in S$ if the shielded policy π^+ computes all other actions $a' \in \mathcal{A} \setminus \{a\}$ as perfectly unsafe.*

4 Results

We organize our experiments around key multi-agent phenomena relevant to several open MARL challenges. Rather than focusing on games individually, each section centers on a particular strategic challenge (e.g., equilibrium selection, social dilemmas) and uses one or more games as illustrative testbeds. For the sake of clarity, details are moved to the supplementary material [7], such as implementation details, shield programs for each experiment, a validation of PLTD in a single-agent experiment, all hyperparameters and training curves. All displayed results are averaged over 5 random seeds to account for variability in training. All reported errors correspond to standard deviations across runs and respective episodes/steps. Our code is open-sourced at https://github.com/satchitchatterji/ShieldedMARL.

4.1 Coordination and Equilibrium Selection (Stag-Hunt)

We use *Stag-Hunt* to test the ability of PLS to guide agents towards normative behaviors in games with multiple Nash equilibria. IPPO serves as the baseline, while SIPPO agents are equipped with a shield constraining the agents towards either the pure cooperative ($\mathcal{T}_{pure}$) or mixed Nash equilibrium ($\mathcal{T}_{mixed}$) – play $Stag/Hare$ 60%/40% of the time with an expected utility of 2.6. $\mathcal{T}_{pure}$ simply tells the agent to not play $Hare$, making it a deterministic shield according to Proposition 3.5. $\mathcal{T}_{mixed}$, used as an example in Section 3.3, defines safety as the absolute difference between the precomputed mixed Nash strategy and the mean of the agent's historical actions.

Table 1. Results for *Stag-Hunt* with PPO-based agents over the last 50 training episodes. $\bar{r}$ is the mean return and *cooperation* is defined as $t_{max}^{-1}\sum_{t=0}^{t_{max}} \mathbf{P}_{\mathcal{T}_{pure}}(\texttt{safe}|s^t)$.

Algorithm	$\bar{r}$ (train)	$\bar{r}$ (eval)	cooperation
IPPO	1.99±0.03	1.99±0.02	0.01±0.01
SIPPO ($\mathcal{T}_{pure}$)	5.00±0.00	5.00±0.00	1.00±0.00
SIPPO ($\mathcal{T}_{mixed}$)	2.57±0.48	2.63±0.43	0.58±0.08

Table 1 shows the results this setting. The IPPO agents converge to the non-cooperative Nash equilibrium, consistently playing *Hare* with a reward of 2 per step. In contrast, agents using $\mathcal{T}_{pure}$ quickly and reliably play *Stag*, earning a higher reward of 5. The unshielded agents never converge to the mixed Nash equilibrium, even though PPO is capable of learning stochastic policies – small deviations push these agents toward playing pure strategies, often resulting in the non-cooperative equilibrium. However, *shielded* PPO agents using $\mathcal{T}_{mixed}$ successfully adopt the mixed strategy, though with high variability due to this attracting nature of the pure equilibria.

4.2 Temporal Coordination (Centipede)

We use *Centipede* to test how well SMARL guides agents to cooperate to achieve large collective long-term rewards, even in scenarios where the dominant strategy suggests defection. A shield was constructed ($\mathcal{T}_{continue}$) to encourage agents to play the game by constraining against early defection. Table 2 shows results for pairs of MARL

agents learning to play the *Centipede* game. IPPO agents occasionally learn to play *Continue* through all $t_{max} = 50$ steps within 500 episodes. In contrast, we see the SIPPO agents playing the full game from the start of training.

Table 2. Results for *Centipede* with PPO- and DQN-based agents over the last 50 training episodes. **Safety** is defined as $\frac{1}{T}\sum_{t=0}^{T}\mathbf{P}_{\mathcal{T}_{cent}}(\texttt{safe}|s^t)$ and $R_{\text{ep}} = \sum_{t=0}^{t_{max}} r^t$.

Algorithm	R_{ep} (train)	R_{ep} (eval)	Safety
IPPO	42.35±36.62	42.83±35.81	0.83±0.23
SIPPO	100.50±0.00	100.50±0.00	1.00±0.00
IQL (ϵ-greedy)	34.62±46.59	34.70±46.53	0.68±0.23
SIQL (ϵ-greedy)	100.50±0.00	100.50±0.00	1.00±0.00
IQL (softmax)	1.73±1.01	30.10±38.61	0.73±0.21
SIQL (softmax)	100.50±0.00	100.50±0.00	1.00±0.00

For DQN-based agents, ϵ-greedy and softmax exploration policies were tested. Unshielded ϵ-greedy agents fail to progress far, as early on in exploration, $\epsilon \approx 1$ (π is uniform), and thus a 25% probability of the agents going to the second stage of the game. The probability of reaching some early stage ℓ is $\approx 0.25^{\ell-1}$. This discourages exploration, as stopping early provides a small but consistent reward, aligning with the Nash/subgame-perfect equilibrium. Unshielded softmax agents perform slightly better, but with high variance – some agent pairs cooperate while most exit early. This highlights the influence of the choice of exploration strategy, and how using shield can guide desirable behavior – all SIQL variations learn to play *Continue* consistently.

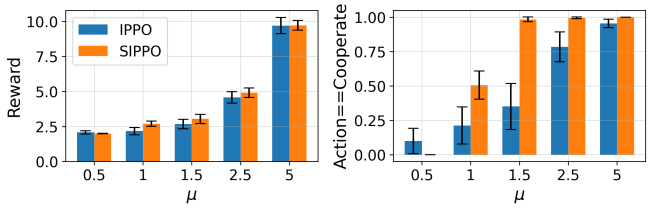


Figure 4. Results for the final 10 episodes for the two-player *Extended Public Goods Game* for PLPG-based agents.

4.3 *Social Dilemmas in Stochastic, Mixed-Motive Environments (Extended Public Goods Game)*

We use *EPGG* to investigate how PLS can promote cooperation in environments with stochastic social incentives. Our experiments address two central questions: (i) *can agents be made to learn strategies that reflect long-term expectations rather than short-term payoffs?*, and (ii) *how does the effectiveness of shielding scale when only a subset of agents is shielded, and under different degrees of parameter sharing?* In the first experiment, we assess how shielding can steer agents toward the *expected* Nash equilibrium of the EPGG. In the second, we evaluate how the degree of parameter sharing affects emergent cooperation.

4.3.1 *Learning Long-Term Normative Behavior*

We begin with a 2-player setting to isolate the effect of shielding on equilibrium selection under payoff uncertainty. This controlled setup enables a clear comparison between shielded agents learning long-term behavior versus unshielded agents reacting to stochastic payoffs. Limiting the population size removes confounding factors such as social influence and allows for direct game-theoretic analysis, providing a clean baseline before scaling to larger populations.

To test whether PLS can guide agents toward long-term cooperation, we designed a shield ($\mathcal{T}_{EPGG}$) that directs agents to follow the expected Nash equilibrium, based on an empirical estimate $\hat{\mu}$ of the stochastic payoff multiplier $f_t \sim \mathcal{N}(\mu, 1)$. This contrasts with the unshielded baseline, where agents maximize the instantaneous payoff.

Figure 4 reports results for $\mu \in \{0.5, 1, 1.5, 2.5, 5.0\}$. Unshielded IPPO agents reliably converge to the instantaneous equilibrium – defecting when $f_t < 1$ and cooperating when $f_t > 2$. In contrast, SIPPO agents quickly converge to the desired behavior, leading to more prosocial behavior under uncertainty. We observe that SIPPO agents exhibit lower reward variance and more stable cooperation compared to the baseline, especially in mixed-motive cases ($1 < \mu < 2$). For $\mu = 0.5$, the shielded agents correctly learn to defect, but earn slightly less due to foregoing occasional cooperative gains from high f_t. Overall, shielded agents match or exceed the baseline in reward and demonstrate more consistent cooperation.

4.3.2 *Partial Shielding and Parameter Sharing*

To focus on how shield coverage and parameter sharing influence group behavior, we use a simplified shield that always selects the cooperative action. Unlike the previous experiment, this avoids computing expected equilibria and isolates the effect of shielding structure. Importantly, even though the shield is deterministic, shielded agents still receive gradient-based learning signals. This allows safety information to influence shared parameters, enabling unshielded agents to adopt cooperative behavior indirectly.

We run EPGG with $n = 5$ agents, where a subset $k \in \{0, \ldots, 5\}$ of agents are shielded with a deterministic policy that always cooperates (reported as a *shielded ratio*=k/n). We compare SIPPO, SCSPPO, and SACSPPO to test how different levels of parameter sharing modulate the influence of the shielded agents on the unshielded population.

Figure 3 shows grouped bar charts for $\mu \in \{0.5, 1, 1.5, 2.5, 5, 7.5\}$. Results show a clear trend: increasing the number of shielded agents leads to higher overall cooperation and greater episodic reward in the mixed-motive range ($1 < \mu < 5$). Furthermore, this effect is amplified by parameter sharing: SACSPPO consistently achieves the same or higher cooperation than SCSPPO and SIPPO at the same shielded ratio. In fully cooperative settings ($\mu = 7.5$), all configurations converge to high cooperation, but SACSPPO still exhibits the lowest variance. In competitive settings ($\mu = 0.5$), cooperation remains low for the baseline algorithms, as expected, increasing with the number of shielded agents – this results in an increasingly lower mean reward with the benefit of the agents instead cooperating. Notably, in the competitive regime, SACSPPO exhibits the highest level of cooperation among the methods, indicating that shared learning with shielded agents can drive prosocial behavior in unshielded agents – even when it reduces their individual rewards.

These results demonstrate that probabilistic shields can influence unshielded agents via shared parameters, serving as an effective conduit for prosocial behavior in multi-agent social dilemmas.

4.4 *Spatio-Temporal Coordination with Imperfect Information (Markov Stag-Hunt)*

We use the *Markov Stag-Hunt* environment [26] to investigate how PLS-SMARL enables coordination in sequential, stochastic, and partially observable multi-agent settings. This environment extends the normal-form *Stag-Hunt* to a spatial domain where agents must learn

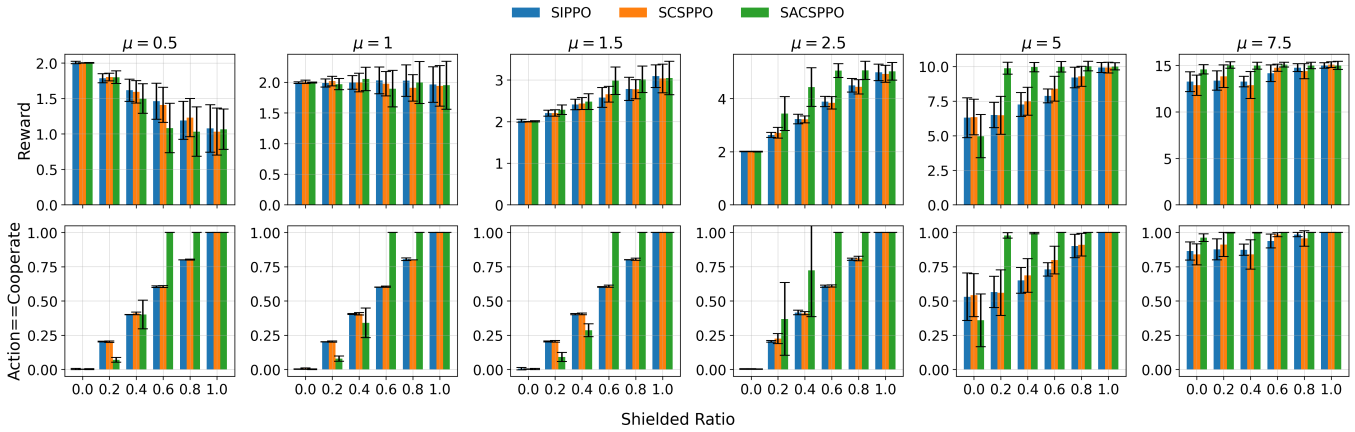


Figure 3. Results for the final 10 episodes of the 5-player *EPGG* for SIPPO, SCSPPO, and SACSPPO agents with varying μ and ratios of shielded agents.

when and how to cooperate. Three experiments are conducted: (i) we study the effect of shield constraint strength on agent behavior, (ii) we evaluate how cooperation scales with population size, (iii) finally we explore how cooperation emerges when only a subset of agents is shielded, highlighting the benefits of shielded parameter sharing.

4.4.1 *Variation in Shield Strength*

To isolate the effect of constraint strength on emergent cooperation, we compare two shields: $\mathcal{T}_{weak}$ and $\mathcal{T}_{strong}$. These were constructed such that $\mathcal{T}_{weak} \subset \mathcal{T}_{strong}$ in terms of their safety constraints. The former constrains agents to cooperate only when both are near a stag, while the latter more strictly promotes cooperation regardless of global position. Both shields are applied in a 2-agent setting using the SIPPO algorithm, with unshielded IPPO as a baseline.

In this experiment, analyzing just reward or safety does not fully capture cooperative behavior, so the key metrics displayed in Table 3 are: total plant harvests per episode, successful cooperative stag hunts, and unsuccessful attempts to hunt a stag alone. Additionally, we show the total return for each episode, and mean safety for shielded agents.

As expected, IPPO agents avoid cooperation (thereby avoiding risk) and steadily harvest plants for a small, reliable reward. SIPPO agents using $\mathcal{T}_{weak}$ show moderate improvement, achieving occasional stag hunts while maintaining similar plant collection. Their increased cooperation also results in more penalties, as agents are willing to risk solo hunts. Under $\mathcal{T}_{strong}$, cooperation improves dramatically: agents prioritize stag hunting and harvest fewer plants, yet earn more than five times the episodic reward compared to SIPPO with $\mathcal{T}_{weak}$. This suggests that well-designed stronger shields not only induce safer behavior, but may lead to better coordination and long-term returns.

4.4.2 *Scaling Cooperation with Population Size*

We next examine how cooperation and safety scale with group size when all agents are shielded using $\mathcal{T}_{strong}$. We vary the number of agents $n \in \{2, 3, 4, 5\}$ and compare three unshielded baselines (IPPO, CSPPO, ACSPPO) and three shielded variants (SIPPO, SCSPPO, SACSPPO). Metrics include total plants harvested, stags hunted, solo hunt penalties, mean episodic safety, and per-agent reward.

Figure 5 shows that unshielded agents (IPPO, CSPPO, ACSPPO) consistently avoid cooperation, harvesting plants for low-risk, low-reward outcomes. Notably, ACSPPO – despite its fully shared architecture – performs worst overall, with near-zero safety and minimal rewards, underscoring that parameter sharing alone is insufficient to induce prosocial behavior.

In contrast, all shielded variants show increasing cooperation as population size grows. SACSPPO agents, in particular, demonstrate the strongest cooperative intent: they achieve the highest number of stag hunts, while also incurring more penalties, reflecting increased willingness to take coordination risks. Despite these penalties, SACSPPO yields the highest rewards and very high adherence to the shield's definition of safety constraints. While per-agent rewards tend to decrease with larger populations, this is expected given that key environment parameters – such as the number of stags and plants, grid size, and episode length – remain fixed across all settings.

These results demonstrate that PLS scales robustly with population size. Notably, with full parameter sharing, SACSPPO leverages shared gradients and consistent safety signals to generalize cooperation more effectively across agents, even in complex multi-agent settings.

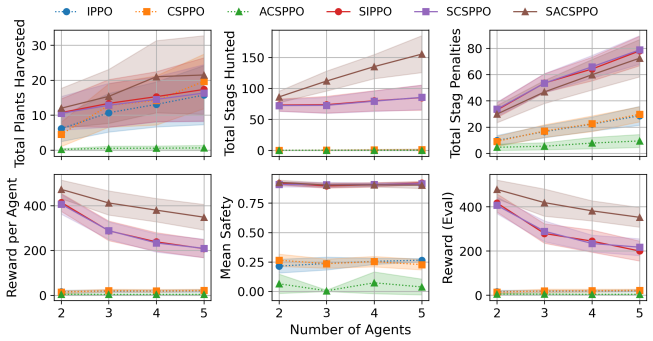


Figure 5. Results for the final 50 episodes of *Markov Stag-Hunt* with all agents shielded with $\mathcal{T}_{strong}$, for $n \in \{2, 3, 4, 5\}$.

4.4.3 *Partial Shielding and Parameter Sharing*

Finally, we investigate cooperation in partially shielded populations. Fixing the population size at $n = 5$, we vary the number of shielded agents $k \in \{0, 1, \ldots, 5\}$ and compare SIPPO, SCSPPO, and SACSPPO. All shielded agents use the strong shield $\mathcal{T}_{strong}$, and unshielded agents remain unconstrained. This experiment tests how shielded behavior propagates through shared parameters in populations where only a subset of agents receive explicit safety constraints.

Figure 6 shows that cooperation, as measured by total stags hunted, increases monotonically with the number of shielded agents across

Table 3. Results for 2-player *Markov Stag-Hunt* with shielded and unshielded PPO-based agents. Columns show: total plants harvested, stags killed, penalties from solo hunts, episodic reward ($R_{\text{ep}} = \sum_{t=0}^{t_{max}} r^t$), and mean episodic safety ($t_{max}^{-1} \sum_{t=0}^{t_{max}} \mathbf{P}_{\mathcal{T}_{(\cdot)}}(\texttt{safe}|s^t)$).

Algorithm	∑plants	∑stags	∑penalties	R_{ep} (train)	R_{ep} (eval)	Mean safety
IPPO	18.74±4.98	0.09±0.11	5.50±2.04	13.71±4.97	16.07±5.80	–
SIPPO ($\mathcal{T}_{weak}$)	18.73±4.26	12.80±7.41	14.25±3.57	68.48±35.19	79.80±38.33	0.96±0.01
SIPPO ($\mathcal{T}_{strong}$)	11.33±4.08	77.32±8.82	11.07±3.25	386.86±46.50	393.07±65.27	0.95±0.01

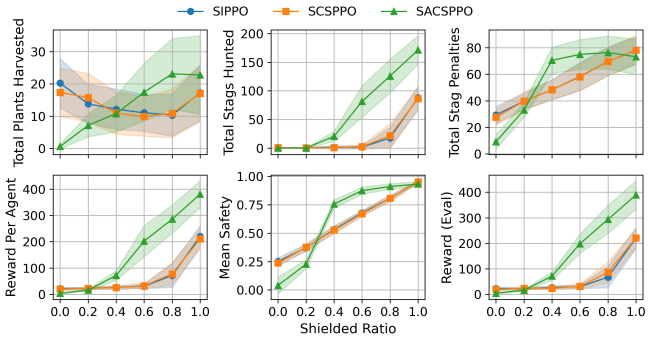


Figure 6. Results for the final 50 episodes of the *Markov Stag-Hunt* with 5 agents, varying the number of shielded agents using the strong shield $\mathcal{T}_{strong}$.

all methods. SACSPPO shows the strongest effect: even with just two shielded agents, it significantly outperforms the other methods in cooperative behavior and safety adherence. As more agents are shielded, SACSPPO reaches a high level of cooperation, but also accumulates the most stag penalties – suggesting increased risk-taking and cooperative intent, even when coordination occasionally fails.

Reward and evaluation performance trend similarly. While all methods improve with greater shield coverage, SACSPPO consistently yields the highest returns, particularly at higher shielded ratios. This indicates that the safety-driven cooperative strategies learned via PLS are not only prosocial but also effective (reward-maximizing). In contrast, SIPPO and SCSPPO agents remain more conservative: they hunt fewer stags, incur fewer penalties, and achieve lower overall rewards.

Plant harvesting does not uniformly decrease with increasing shielded ratio across all methods. While SIPPO and SCSPPO show a modest decline – suggesting a shift from self-interested strategies to coordinated risk-taking – SACSPPO exhibits an unexpected increase in plant harvesting as more agents are shielded. Taken together with SACSPPO's high shield adherence, this may indicate that stronger parameter sharing drives homogenization that improves reward maximization, encouraging agents to opportunistically exploit lower-risk strategies. Alternatively, the behavior could reflect an emergent division of labor, where agents already near the stag focus on hunting while others stabilize returns via plant harvesting. Distinguishing between these hypotheses would require further analysis of agent roles and policy diversity. Importantly, mean safety steadily increases with shield coverage in all methods, but SACSPPO achieves the highest safety levels – highlighting its superior ability to align unshielded agents with shielded behavior through shared learning.

Together, these results demonstrate that PLS can induce prosociality even under partial shielding, and that shared parameters are a key conduit for propagating safety-aligned behavior to unshielded agents.

5 Related Work

In safe (MA)RL, several recent frameworks address safety during learning and execution [16]. Gu et al. [15], propose **constrained Markov games** and **MACPO**, but without formal language semantics. Subramanian et al. [30] integrate **PL neural networks** as the function approximator within agents to affect interpretable top-down control; however, in our method, the agents may use any function approximator, maximizing potential learning flexibility. Waga et al. [35] use **Mealy machines** to approximate a world model to ensure safety, leading to some inherent symbolic interpretability. Other related concepts include **dynamic shielding**, for instance, ElSayed-Aly et al. [12] who introduce two frameworks that dynamically enforce safety constraints – our own method is similar to their **factored shields**. Banerjee et al. [3], Zhang et al. [39] and Xiao et al. [37] are recent works that use the notion of (dynamic) **model predictive shielding**. Though these show promising results, these do not benefit from the interpretability of formal language semantics. Melcer et al. [23] explore **decentralized shielding** (similar to factored shields), i.e. allowing each agent to have its own shield, reducing computational overhead and enhancing scalability. [23] and [12] use **LTL as formal language**, though with a high knowledge requirement of the underlying MDP – this is in contrast to our use of PLS which requires only safety-related parts of the states. Finally, some methods, such as Wang et al. [36], enforce safety through **natural language constraints**; in contrast, our approach builds on PLS to provide stronger formal guarantees.

6 Discussion and Conclusion

In this article, we introduced Probabilistic Logic Shielded MARL, a novel set of DQN- and PPO-based methods that extend PLS to MARL. Several classes of games were analyzed and shown to benefit from SMARL in terms of safety and cooperation, and theoretical guarantees of these methods were presented.

Despite the benefits implied by our results, a few limitations and extensions must be reflected upon: ▶ First, current methods for solving PL programs become computationally expensive as the state and action spaces grow [13]. However, as with factored shields [12], our method scales only linearly with the number of agents. ▶ The use of *a-priori* constraints in PLS provides *predefined* safety knowledge, which is straightforward for simple environments but becomes complex in dynamic, real-world systems. This approach limits scalability, relying on expert-defined safety measures. Most existing dynamic shielding methods do not use formal language semantics, and thus a balance must be struck between autonomous shield updating and formal verification. ▶ The complexity of the ProbLog program in PLS impacts performance and behavior, as seen in the Markov Stag-Hunt experiments where the stronger shield promoted more cooperation but at the cost of increased complexity w.r.t. hand-designed rules. It may be interesting to examine how varying shield complexity affects computational load and safety outcomes. ▶ Partial shielding was explored where shielded agents update their parameters with safety gradients, while unshielded agents focus on maximizing rewards, optionally sharing parameters. A weighted shared parameter update may balance safety and reward optimization dynamically. ▶ Finally, while PL-SMARL aims to enhance safety, its implementation must carefully consider ethical and societal impacts, as hand-designed constraints may introduce biases, under-specified definitions, or system vulnerabilities that current research has yet to fully address.

Acknowledgements

This publication is part of the project 'Hybrid Intelligence: augmenting human intellect' (https://hybrid-intelligence-centre.nl,) with project number 024.004.022 of the research programme 'Gravitation' which is (partly) financed by the Dutch Research Council (NWO).

References

[1] S. V. Albrecht, F. Christianos, and L. Schäfer. Multi-agent Reinforcement Learning: Foundations and Modern Approaches. *Massachusetts Institute of Technology: Cambridge, MA, USA*, 2023.

[2] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu. Safe Reinforcement Learning via Shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[3] A. Banerjee, K. Rahmani, J. Biswas, and I. Dillig. Dynamic Model Predictive Shielding for Provably Safe Reinforcement Learning. *Advances in Neural Information Processing Systems*, 37:100131–100159, 2024.

[4] R. Bloem, B. Könighofer, R. Könighofer, and C. Wang. Shield Synthesis: Runtime Enforcement for Reactive Systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 533–548. Springer, 2015.

[5] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo. Swarm Robotics: A Review from the Swarm Engineering Perspective. *Swarm Intelligence*, 7:1–41, 2013.

[6] S. Carr, N. Jansen, S. Junges, and U. Topcu. Safe Reinforcement Learning via Shielding Under Partial Observability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14748–14756, 2023.

[7] S. Chatterji and E. Acar. Think Smart, Act SMARL! Analyzing Probabilistic Logic Shields for Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2411.04867*, 2024. Full version of this paper.

[8] F. G. Cozman and D. D. Mauá. On the Semantics and Complexity of Probabilistic Logic Programs. *Journal of Artificial Intelligence Research*, 60:221–262, 2017.

[9] L. De Raedt, A. Kimmig, and H. Toivonen. Problog: A Probabilistic Prolog and Its Application in Link Discovery. In *IJCAI*, volume 7, pages 2462–2467, 2007.

[10] C. S. De Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. Torr, M. Sun, and S. Whiteson. Is Independent Learning All You Need in the Starcraft Multi-agent Challenge? *arXiv preprint arXiv:2011.09533*, 2020.

[11] F. Den Hengst, V. François-Lavet, M. Hoogendoorn, and F. van Harmelen. Planning for Potential: Efficient Safe Reinforcement Learning. *Machine Learning*, 111(6):2255–2274, 2022.

[12] I. ElSayed-Aly, S. Bharadwaj, C. Amato, R. Ehlers, U. Topcu, and L. Feng. Safe Multi-Agent Reinforcement Learning via Shielding. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 483–491, 2021.

[13] D. Fierens, G. Van den Broeck, J. Renkens, D. Shterionov, B. Gutmann, I. Thon, G. Janssens, and L. De Raedt. Inference and Learning in Probabilistic Logic Programs using Weighted Boolean Formulas. *Theory and Practice of Logic Programming*, 15(3):358–401, 2015.

[14] S. Ganesh, N. Vadori, M. Xu, H. Zheng, P. Reddy, and M. Veloso. Reinforcement Learning for Market Making in a Multi-Agent Dealer Market. *arXiv preprint arXiv:1911.05892*, 2019.

[15] S. Gu, J. G. Kuba, Y. Chen, Y. Du, L. Yang, A. Knoll, and Y. Yang. Safe Multi-Agent Reinforcement Learning for Multi-Robot Control. *Artificial Intelligence*, 319:103905, 2023.

[16] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, and A. Knoll. A Review of Safe Reinforcement Learning: Methods, Theories and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[17] J. K. Gupta, M. Egorov, and M. Kochenderfer. Cooperative Multi-Agent Control using Deep Reinforcement Learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 66–83. Springer, 2017.

[18] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver. Rainbow: Combining Improvements in Deep Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[19] N. Hunt, N. Fulton, S. Magliacane, T. N. Hoang, S. Das, and A. Solar-Lezama. Verifiably Safe Exploration for End-to-End Reinforcement Learning. In *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, pages 1–11, 2021.

[20] N. Jansen, B. Könighofer, S. Junges, A. Serban, and R. Bloem. Safe Reinforcement Learning using Probabilistic Shields. In *31st International Conference on Concurrency Theory (CONCUR 2020)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2020.

[21] K. Leyton-Brown and Y. Shoham. *Essentials of Game Theory: A Concise Multidisciplinary Introduction*. Springer Nature, 2022.

[22] M. Maschler, S. Zamir, and E. Solan. *Game Theory*. Cambridge University Press, 2020.

[23] D. Melcer, C. Amato, and S. Tripakis. Shield Decentralization for Safe Multi-Agent Reinforcement Learning. *Advances in Neural Information Processing Systems*, 35:13367–13379, 2022.

[24] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing Atari with Deep Reinforcement Learning. *arXiv preprint arXiv:1312.5602*, 2013.

[25] N. Orzan, E. Acar, D. Grossi, and R. Rădulescu. Emergent Cooperation under Uncertain Incentive Alignment. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, page 1521–1530, Richland, SC, 2024. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400704864.

[26] A. Peysakhovich and A. Lerer. Prosocial Learning Agents Solve Generalized Stag Hunts Better than Selfish Ones. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 2043–2044. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[27] L. D. Riek. Healthcare Robotics. *Communications of the ACM*, 60(11): 68–78, 2017.

[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[29] S. Shalev-Shwartz, S. Shammah, and A. Shashua. Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. *arXiv preprint arXiv:1610.03295*, 2016.

[30] C. Subramanian, M. Liu, N. Khan, J. Lenchner, A. Amarnath, S. Swaminathan, R. Riegel, and A. Gray. A Neuro-Symbolic Approach to Multi-Agent RL for Interpretability and Probabilistic Decision Making. *arXiv preprint arXiv:2402.13440*, 2024.

[31] M. Tan. Multi-agent Reinforcement Learning: Independent vs. Cooperative agents. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 330–337, 1993.

[32] J. Terry, B. Black, N. Grammel, M. Jayakumar, A. Hari, R. Sullivan, L. S. Santos, C. Dieffendahl, C. Horsch, R. Perez-Vicente, et al. PettingZoo: Gym for Multi-Agent Reinforcement Learning. *Advances in Neural Information Processing Systems*, 34:15032–15043, 2021.

[33] J. Tsitsiklis and B. Van Roy. Analysis of Temporal-Difference Learning with Function Approximation. *Advances in Neural Information Processing Systems*, 9, 1996.

[34] E. van der Sar, A. Zocca, and S. Bhulai. Multi-Agent Reinforcement Learning for Power Grid Topology Optimization. *arXiv preprint arXiv:2310.02605*, 2023.

[35] M. Waga, E. Castellano, S. Pruekprasert, S. Klikovits, T. Takisaka, and I. Hasuo. Dynamic Shielding for Reinforcement Learning in Black-box Environments. In *International Symposium on Automated Technology for Verification and Analysis*, pages 25–41. Springer, 2022.

[36] Z. Wang, M. Fang, T. Tomilin, F. Fang, and Y. Du. Safe Multi-Agent Reinforcement Learning with Natural Language Constraints. *arXiv preprint arXiv:2405.20018*, 2024.

[37] W. Xiao, Y. Lyu, and J. Dolan. Model-based Dynamic Shielding for Safe and Efficient Multi-agent Reinforcement Learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 1587–1596, 2023.

[38] W. Yang, G. Marra, and L. De Raedt. Safe Reinforcement Learning via Probabilistic Logic Shields. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5739–5749, 2023.

[39] W. Zhang, O. Bastani, and V. Kumar. MAMPS: Safe Multi-agent Reinforcement Learning via Model Predictive Shielding. *arXiv preprint arXiv:1910.12639*, 2019.

[40] D. Zhao, H. Wang, K. Shao, and Y. Zhu. Deep Reinforcement Learning with Experience Replay Based on SARSA. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6. IEEE, 2016.