Ensemble Value Functions for Efficient Exploration in Multi-Agent Reinforcement Learning

Lukas Schäfer*
University of Edinburgh
Edinburgh, United Kingdom
l.schaefer@ed.ac.uk

Oliver Slumbers
University College London
London, United Kingdom
o.slumbers@cs.ucl.ac.uk

Stephen McAleer
Carnegie Mellon University
Pittsburgh, USA
mcaleer.stephen@gmail.com

Yali Du
King's College London
London, United Kingdom
yali.du@kcl.ac.uk

Stefano V. Albrecht
University of Edinburgh
Edinburgh, United Kingdom
s.albrecht@ed.ac.uk

David Mguni
Queen Mary University London
London, United Kingdom
d.mguni@qmul.ac.uk

ABSTRACT

Multi-agent reinforcement learning (MARL) requires agents to explore within a vast joint action space to find joint actions that lead to coordination. Existing value-based MARL algorithms commonly rely on random exploration, such as ϵ-greedy, to explore the environment which is not systematic and inefficient at identifying effective actions in multi-agent problems. Additionally, the concurrent training of the policies of multiple agents during training can render the optimisation non-stationary. This can lead to unstable value estimates, highly variant gradients, and ultimately hinder coordination between agents. To address these challenges, we propose ensemble value functions for multi-agent exploration (EMAX). EMAX is a framework to seamlessly extend value-based MARL algorithms. EMAX leverages an ensemble of value functions for each agent to guide their exploration, reduce the variance of their optimisation, and makes their policies more robust to miscoordination. EMAX achieves these benefits by (1) systematically guiding the exploration of agents with a UCB policy towards parts of the environment that require multiple agents to coordinate. (2) EMAX computes average value estimates across the ensemble as target values to reduce the variance of gradients and make optimisation more stable. (3) During evaluation, EMAX selects actions following a majority vote across the ensemble to reduce the likelihood of miscoordination. We first instantiate independent DQN with EMAX and evaluate it in 11 general-sum tasks with sparse rewards. We show that EMAX improves final evaluation returns by 185% across all tasks. We then evaluate EMAX on top of IDQN, VDN and QMIX in 21 common-reward tasks, and show that EMAX improves sample efficiency and final evaluation returns across all tasks over all three vanilla algorithms by 60%, 47%, and 538%, respectively.

KEYWORDS

Multi-agent reinforcement learning; exploration; ensemble models

*Now at Microsoft Research Cambridge, contact at lukas.schaefer@microsoft.com.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

ACM Reference Format:
Lukas Schäfer, Oliver Slumbers, Stephen McAleer, Yali Du, Stefano V. Albrecht, and David Mguni. 2025. Ensemble Value Functions for Efficient Exploration in Multi-Agent Reinforcement Learning. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 21 pages.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) [2] is a common paradigm to concurrently train autonomous agents in decision-making tasks that require coordination between agents. To discover such coordinated actions, agents need to explore the state space and vast joint action space of the task. However, many value-based MARL algorithms still rely on random exploration, such as an ϵ-greedy policy [38, 43] that is not systematic and so can take many iterations to discover the optimal behaviour, in particular in states in which multiple agents need to coordinate their actions [37]. To better understand this inefficiency, we consider the following example (Figure 1) in which two agents have to navigate a grid-world to jointly pick up a heavy object. To learn to pick up the goal object, agents need to both select the pick-up action in a state where both agents are next to the object (Figure 1, right). Such coordination is highly unlikely when following a random exploration policy. In contrast, the exploration is not required to be coordinated if agents are not next to the object (Figure 1, left) so random exploration might not be highly inefficient in this case. This example illustrates that random exploration is particularly inefficient in states in which multiple agents are required to coordinate their actions, and demonstrates the need for more systematic exploration of such states. Furthermore, the concurrent training of multiple agents in MARL and the resulting non-stationarity of the policies of other agents makes it challenging for agents to robustly learn to solve the task, and can result in miscoordination between agents [36].

Motivated by these challenges in MARL, we propose *ensemble value functions for multi-agent exploration* (EMAX), a general framework to seamlessly extend any value-based MARL algorithms by training ensembles of value functions for each agent. EMAX systematically explores states and actions that may require multiple agents to coordinate by following an upper-confidence bound (UCB) policy [5] over the average and disagreement of value estimates across the ensemble. This exploration strategy prioritises the exploration of actions that appear promising (as measured by

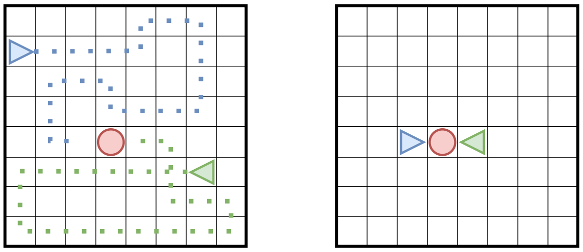


Figure 1: Motivational example: Two agents (triangles) can independently explore (left), but they must coordinate to pick up the object (circle) and complete the task (right).

high average value estimates) and that might require coordination between agents (as measured by high disagreement in value estimates). Beyond its exploration policy, EMAX computes average target values across the ensemble to reduce the variance of gradients and eliminate the need for additional target networks. Lastly, EMAX selects actions during evaluation by following a majority vote across the greedy actions of all value functions in the ensemble to reduce the likelihood of selecting sub-optimal actions.

To evaluate the efficacy of EMAX, we first extend independent DQN [28, 44] with EMAX and evaluate it in 11 mixed-objective tasks, in which agents receive individual rewards but must coordinate their actions in some states. In this setting, EMAX improves the final evaluation returns of IDQN by 185% across all tasks (Section 4.1). Afterwards, we focus on the cooperative MARL setting and extend IDQN as well as VDN [43] and QMIX [38] with EMAX, and conduct an extensive evaluation of EMAX in 21 common-reward tasks across four diverse environments. Across all common-reward tasks, EMAX improves sample efficiency and final achieved returns over all three vanilla algorithms (IDQN, VDN, QMIX) by 60%, 47%, and 538%, respectively (Section 4.1). Lastly, we empirically validate that all three major components of EMAX are essential for its performance in an ablation study, and further demonstrate the effects of the proposed exploration policy, target computation, and evaluation policy (Section 4.2).

2 BACKGROUND

Partially observable stochastic games: We formalise multi-agent environments as partially observable stochastic games (POSG) [18, 23] defined by $(I, S, \mathbf{O}, \mathbf{A}, \mathcal{T}, O, \{\mathcal{R}_i\}_{i\in I}, \gamma)$. Agents are indexed by $i \in I = \{1, \ldots, N\}$, S denotes the state space of the environment and $\mathbf{A} = A_1 \times \ldots \times A_N$ denotes the joint action space of all agents. Each agent has access to its local observations $o_i \in O_i$. The joint observation space is denoted $\mathbf{O} = O_i \times \ldots \times O_N$. $\mathcal{T} : S \times \mathbf{A} \mapsto \Delta(S)$ denotes the transition function of the environment and defines a distribution of successor states given the current state and the applied joint action. The observation transition function $O : S \times \mathbf{A} \times \Delta(\mathbf{O})$ defines a distribution of next joint observations received by agents given the current state and joint action of all agents. $\mathcal{R}_i : S \times \mathbf{A} \times S \mapsto \mathbb{R}$ denotes the reward function for each agent i. Each agent learns a policy $\pi_i(a_i^t \mid h_i^t)$ conditioned on its history of observations until time step t, i.e. $h_i^t = (o_i^0, o_i^1, \ldots, o_i^t)$. The objective of a POSG is for all agents to learn a joint policy $\pi = (\pi_1, \ldots, \pi_N)$ such that the expected discounted returns of each agent are maximised with respect to the policies of all other agents. The discounted returns for agent i can be written as

$$\mathbb{E}_{a_i^t \sim \pi_i(h_i^t);\mathbf{a}_{-i}^t \sim \pi_{-i}(\mathbf{h}_{-i}^t)}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}_i(s^t, \mathbf{a}^t, s^{t+1})\right] \tag{1}$$

where $\gamma \in [0; 1)$ denotes the discount factor, $\mathbf{a}^t = (a_1^t, \ldots, a_N^t)$ and $\mathbf{h}^t = (h_1^t, \ldots, h_N^t)$ denote the joint action and observation history, respectively, and subscript $-i$ denotes all agents but agent i. We also consider the special case of common-reward environments, often formalised as a Dec-POMDP [6, 30], in which agents collectively maximise the cumulative discounted sum of shared rewards.

Independent deep Q-networks: Independent deep Q-networks (IDQN) extends DQN [28] for MARL and independently learns a value function Q_i [44], parameterised by θ_i, for each agent i. Agents store tuples $(s^t, h^t, a^t, r^t, s^{t+1}, h^{t+1})$ of experience consisting of state s^t, joint observation history h^t, applied joint action a^t, received reward r^t, next state s^{t+1}, and next joint observation history h^{t+1}, respectively, in a replay buffer. The value function of agent i is then optimised by minimising the average loss across sampled batches of experience:

$$\mathcal{L}(\theta_i) = \left[r_i^t + \gamma \max_{a_i \in A_i} \bar{Q}_i(h_i^{t+1}, a_i; \bar{\theta}_i) - Q_i(h_i^t, a_i^t; \theta_i)\right]^2 \tag{2}$$

with $\bar{\theta}_i$ denoting the parameters of the target network $\bar{Q}$ which are periodically copied from θ_i.

Value decomposition: Independent learning serves as an effective baseline in many cooperative MARL tasks [37] but suffers from the multi-agent credit assignment problem, i.e. agents need to identify their individual contribution to received rewards [15, 38]. Value decomposition algorithms extend IDQN by learning a decomposed centralised state-action value function Q_{tot}, conditioned on the state and joint action of all agents.[1] Directly learning such a value function is often computationally infeasible due to the exponential growth of the joint action space with the number of agents, so the centralised value function is approximated with an aggregation of individual utility functions of all agents conditioned on the local observation history. These individual utility functions of agents estimate their contribution to the centralised state-action value function and, thus, address the multi-agent credit assignment problem. All functions are jointly optimised by minimising the following loss with y_{tot} denoting centralised target values:

$$\mathcal{L}(\theta) = \left[Q_{\text{tot}}(s^t, a^t; \theta) - y_{\text{tot}}\right]^2 \tag{3}$$

Two common value decomposition algorithms are VDN [43] and QMIX [38] that assume a linear and monotonic decomposition of the centralised value function, respectively.

3 ENSEMBLE VALUE FUNCTIONS FOR MULTI-AGENT EXPLORATION

In this section, we present *ensemble value functions for multi-agent exploration* (EMAX), a general framework that trains an ensemble of value functions for each agent in value-based MARL. Formally, each agent i trains an ensemble of K value functions $\{Q_i^k\}_{k=1}^K$ with

[1]In environments, where the state is not available during training, we use the joint observation as a proxy for the state.

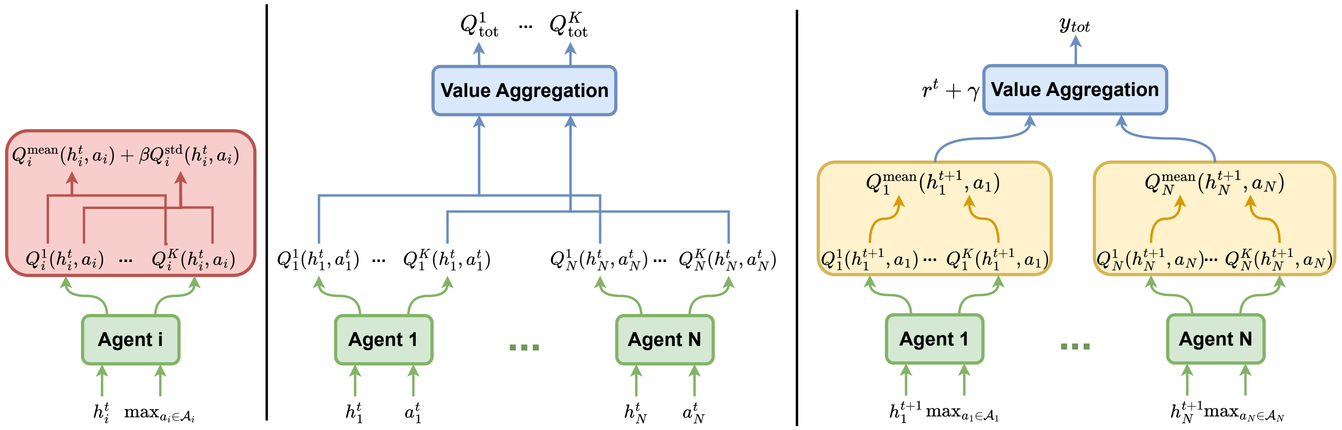


Figure 2: Illustration of the components of the EMAX algorithm. Left: UCB exploration strategy for agent i. Middle: value estimation with value decomposition. Right: target computation with value decomposition. The value functions of individual agents are highlighted in green, the exploration policy in red, value decomposition in blue, and target computation in orange.

Q_i^k being parameterised by θ_i^k and conditioned on agent i's local observation history. EMAX leverages these ensembles of value functions to guide the exploration of agents and stabilise their optimisation. Figure 2 illustrates the exploration policy as well as the value and target estimation of EMAX. We provide pseudocode for EMAX in Appendix A.[2] In the following, we denote the average and standard deviation across the ensemble of value functions of agent i with parameters $\theta_i = \{\theta_i^k\}_{k=1}^K$ as follows:

$$Q_i^{\text{mean}}(h_i^t, a_i^t; \theta_i) = \frac{1}{K} \sum_{k=1}^{K} Q_i^k(h_i^t, a_i^t; \theta_i^k) \tag{4}$$

$$Q_i^{\text{std}}(h_i^t, a_i^t; \theta_i) = \sqrt{\frac{\sum_{k=1}^K \left(Q_i^k(h_i^t, a_i^t; \theta_i^k) - Q_i^{\text{mean}}(h_i^t, a_i^t; \theta_i)\right)^2}{K}} \tag{5}$$

Exploration policy: EMAX follow a UCB policy using the average and standard deviation of value estimates across the ensemble:

$$\pi_i^{\text{expl}}(h_i^t; \theta_i) \in \arg\max_{a_i \in A_i} Q_i^{\text{mean}}(h_i^t, a_i; \theta_i) + \beta Q_i^{\text{std}}(h_i^t, a_i; \theta_i) \tag{6}$$

with $\beta > 0$ denoting a weighting hyperparameter. As measured by the mean value estimate, this policy guides agents to explore actions that are deemed promising. Prior work in single-agent RL already established that the disagreement across an ensemble of value functions can indicate epistemic uncertainty and the need for exploration [5, 20, 22]. In this work, we argue that in MARL this disagreement of value estimates can additionally indicate whether state-action pairs require agents to coordinate their actions. To see why, consider states in which multiple agents have to coordinate, i.e. multiple agents need to select specific actions, to receive a large reward (as in our motivational example in Figure 1, right). If any agent deviates from this joint action, the agents receive no reward. In such states, received rewards for a given action of agent i will vary significantly whenever other agents follow stochastic policies, since the reward depends on the stochastic actions of other agents. In contrast, in states where agent i receives identical rewards independent of the actions of other agents (as in Figure 1, left), no such variability of rewards is experienced. Due to this variability of rewards (or lack thereof), value estimates across the ensemble will exhibit high disagreement in states that might require coordination, and little disagreement in states that might require no or limited coordination. Therefore, the EMAX exploration policy systematically focuses the exploration of agents on state-action pairs that might require coordination in contrast to common random exploration for value-based MARL such as ϵ-greedy policies. We note that the disagreement diminishes throughout training as value functions and policies converge. Once agents always succeed at coordinating in a state with such potential, returns will no longer be variable, and the disagreement of value estimates will reduce. Furthermore, the disagreement of value estimates also incentivises the exploration of states with potential for future rather than just immediate coordination since value functions estimate expected returns over entire episodes. We empirically validate these effects and benefits of the EMAX exploration policy in Section 4.2.

Optimisation: To extend IDQN with EMAX, we optimise the k-th value function of agent i by minimising the following loss:

$$\mathcal{L}(\theta_i^k) = \mathbb{E}_{(h_i^t, a_i^t, r_i^t, h_i^{t+1}) \sim \mathcal{D}} \left[\left(r_i^t + \gamma \max_{a_i \in A_i} Q_i^{\text{mean}}(h_i^{t+1}, a_i; \theta_i) - Q_i^k(h_i^t, a_i^t; \theta_i^k) \right)^2 \right] \tag{7}$$

Computing target values as the average across all value estimates of the ensemble [22] reduces the computational and memory cost of training ensemble networks by eliminating the need for target networks. Additionally, as we empirically show in Section 4.2, these target values reduce the variability of gradients and improve the stability of training. Such improved stability is particularly valuable in MARL where the non-stationarity of the policies of other agents can otherwise result in unstable or inefficient training [36, 37].

Evaluation policy: Value-based MARL algorithms typically exploit using the greedy policy with respect to their value function. In

[2] Appendices are available at https://arxiv.org/abs/2302.03439.

EMAX, agent i selects its action during evaluation using a majority vote across the greedy actions of all models in its ensemble [31]:

$$\pi_i^{\text{eval}}(h_i^t; \theta_i) \in \arg\max_{a_i \in A_i} \sum_{k=1}^{K} [1]_{\mathcal{A}^k_{\text{opt},i}}(a_i)$$
$$\mathcal{A}^k_{\text{opt},i} = \{a_i \in A_i \mid a_i \in \arg\max_{a'_i} Q_i^k(h_i^t, a'_i; \theta_i^k)\} \quad (8)$$

with indicator function $[1]_{\mathcal{A}^k_{\text{opt},i}}(a) = 1$ for the greedy action(s) of the k-th value function of agent i and 0 otherwise. Such a policy decreases the likelihood of taking poor actions because any individual value function preferring a poor action due to errors in value estimates does not impact the action selection as long as the majority of models agree on the optimal action. We empirically demonstrate the benefits of such an evaluation policy in Section 4.2.

Ensemble value functions: All aforementioned techniques rely on value functions within the ensemble to be sufficiently diverse early in training. To ensure such diversity, we employ three techniques: (1) Ensemble models share no parameters and are randomly initialised. (2) Each model in the ensemble is trained on separately sampled batches of experiences [22]. (3) Each model is trained on bootstrapped samples of the entire experience collected [31]. We provide more details on the bootstrapping procedure in Appendix B.

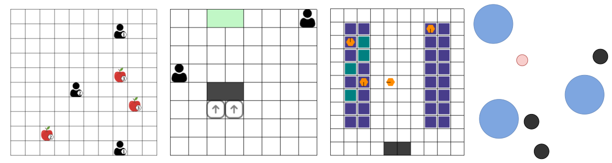
Value decomposition: So far, we presented EMAX as an extension of IDQN. We now discuss the application of EMAX to value decomposition algorithms for common-reward tasks in which agents suffer from the multi-agent credit assignment problem. In EMAX, this problem has the additional implication that the exploration policy defined in Equation (6) does not distinguish which agents need to coordinate their actions in a particular state. To make sure that each agent explores states and actions in which that particular agent's coordination, rather than any agents' coordination, is required, we integrate EMAX into value decomposition algorithms such as VDN [43] and QMIX [38]. These algorithms enable agents to learn individual value functions that identify their contribution to received common rewards. In EMAX, we train an ensemble of these utility functions for each agent. The parameters of the k-th utility function of all agents $\theta^k = \{\theta_i^k\}_{i \in I}$ will be optimised to minimise Equation (3). For VDN, the decomposition and target value are defined as follows:

$$Q_{\text{tot}}(s^t, a^t; \theta^k) = \sum_{i \in I} Q_i^k(h_i^t, a_i^t; \theta_i^k) \quad (9)$$
$$y_{\text{tot}} = r^t + \gamma \sum_{i \in I} \max_{a_i \in A_i} Q_i^{\text{mean}}(h_i^{t+1}, a_i; \theta_i) \quad (10)$$

and for QMIX is defined as follows:

$$Q_{\text{tot}}(s^t, a^t; \theta^k) = f_{\text{mix}}\left(Q_1^k(h_1^t, a_1^t; \theta_1^k), \ldots, Q_N^k(h_N^t, a_N^t; \theta_N^k); \theta_{\text{mix}}\right)$$
$$y_{\text{tot}} = r^t + \gamma f_{\text{mix}} \begin{pmatrix} \max_{a_1 \in A_1} Q_1^{\text{mean}}(h_1^{t+1}, a_1; \theta_1), \\ \ldots \\ \max_{a_N \in A_N} Q_N^{\text{mean}}(h_N^{t+1}, a_N; \theta_N) \end{pmatrix} ; \bar{\theta}_{\text{mix}} \quad (11)$$

For QMIX, we use a single mixing network and target mixing network with parameters θ_{mix} and $\bar{\theta}_{\text{mix}}$, respectively, to aggregate the utility estimates for all utility functions in the ensemble.



(a) LBF (b) BPUSH (c) RWARE (d) MPE

Figure 3: Visualisations of four multi-agent environments.

4 EVALUATION

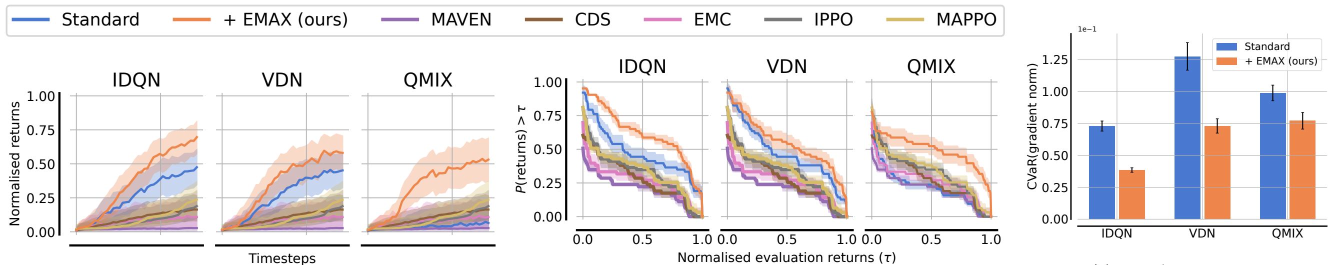
We evaluate EMAX in 11 mixed-objective tasks, in which all agents receive individual rewards, and in 21 common-reward tasks across four multi-agent environments shown in Figure 3: level-based foraging (LBF) [3, 37], multi-robot warehouse (RWARE) [12, 37], boulder-push (BPUSH) [10], and multi-agent particle environment (MPE) [25, 29]. In mixed-objective tasks, we evaluate IDQN with and without EMAX in LBF and RWARE tasks that require a mixture of cooperation in competition, represented by agents picking up food either by themselves or collectively in LBF and by agents delivering shelves and avoiding to block each others' path in RWARE. In common-reward tasks, we evaluate EMAX as an extension of IDQN, VDN and QMIX. All considered common-reward tasks require agents to cooperate to achieve high rewards. Additionally, many of these tasks feature sparse rewards and, thus, are challenging for exploration, making them particularly suited to evaluate the benefits of the systematic exploration of EMAX. We provide detailed descriptions of all environments in Appendix D.1. In the common-reward setting, we compare EMAX to additional baselines in three value-based exploration algorithms with MAVEN [26], CDS [21], and EMC [48], as well as independent and multi-agent PPO (IPPO and MAPPO) that have been shown to exhibit strong MARL performance [37, 47]. Lastly, we provide an analysis of each component of EMAX to investigate our hypotheses on their benefits and effects, and provide an ablation study (Section 4.2).

Evaluation metrics: We report the mean evaluation returns as well as 95% confidence intervals computed over five runs in all individual tasks across both settings. In common-reward tasks, we report the returns computed over the common rewards, and in mixed-objective tasks we report the sum of all agents' evaluation returns. Following the methodology of Agarwal et al. [1], we report aggregated normalised evaluation returns[3] and performance profiles with the interquartile mean (IQM) and 95% confidence intervals computed over all tasks in each setting. The learning curves indicate the sample efficiency of agents, and performance profiles allow to compare the distribution of final evaluation returns indicating the robustness of the final policies learned by each algorithm.

To evaluate the training stability of algorithms, we would like to capture how variable and noisy gradients are during training. To measure this variability, we detrend gradient norms by deducting each gradient norm from its subsequent norm, and compute the conditional value at risk (CVaR) of detrended gradient norms:

$$\text{CVaR}(g') = \mathbb{E}\left[g' \mid g' \geq \text{VaR}_{95\%}(g')\right] \text{ and } g'_t = |\nabla_{t+1}| - |\nabla_t| \quad (12)$$

[3]We follow the task-based normalisation procedure of Papoudakis et al. [37].

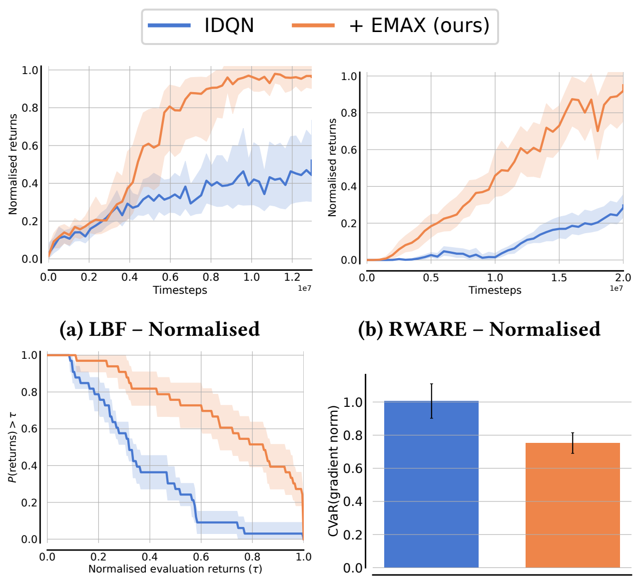


(a) Normalised evaluation returns

(b) Performance profile

(c) Gradient CVaR

Figure 4: Common-reward evaluation across 21 tasks. (a) Normalised evaluation returns and (b) performance profile of all algorithms aggregated across all tasks. (c) Average and standard error of gradient stability (Equation (12)).



(a) LBF – Normalised

(b) RWARE – Normalised

(c) Performance profile

(d) Gradient CVaR

Figure 5: Mixed-objective evaluation across 11 tasks. Evaluation returns in (a) LBF and (b) RWARE, (c) performance profiles across all tasks, and (d) average and standard error of gradient stability (Equation (12)).

where the value at risk (VaR) corresponds to the value at the 95% quantile of all detrended gradient norm values. This metric corresponds to the short-term risk across time suggested by Chan et al. [7]. A larger CVaR value indicates more variability in gradients which can indicate unstable training, while a smaller CVaR value indicates less variability in gradients and more stable training.

Implementation details: In all experiments, agents share parameters with each other to improve sample efficiency [11, 37]. To allow for agent specialisation, shared networks receive one-hot vectors that indicate agents' identity as additional inputs. Unless stated otherwise, EMAX trains an ensemble of $K = 5$ value functions. For more details on chosen hyperparameters, see Appendix D.3.

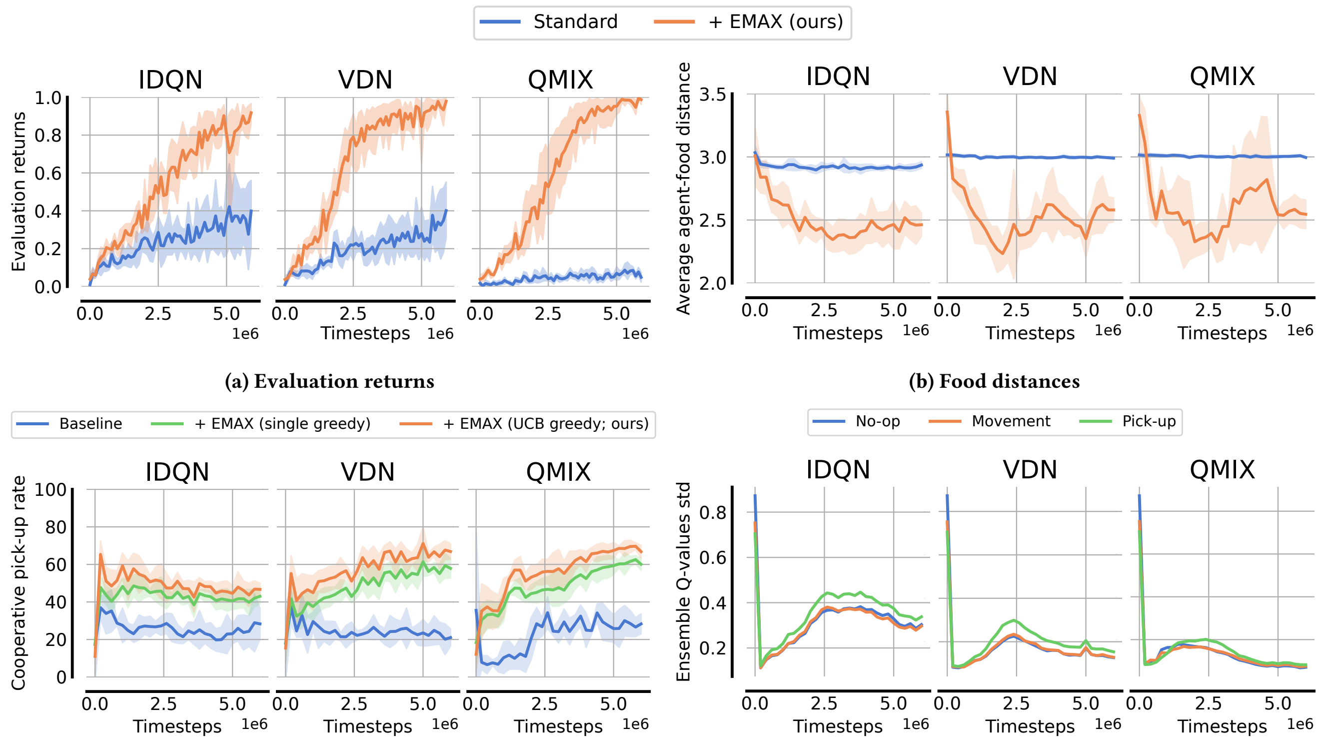
4.1 Evaluation Results

Figure 5 shows the learning curves of IDQN with and without EMAX in 11 mixed-objective tasks in LBF and RWARE with normalised evaluation returns, and a performance profile at the end of training. Across all 11 tasks, EMAX improves final evaluation returns of IDQN by 189%, with 105% and 275% improvement in LBF and RWARE, respectively. The performance profile also shows that EMAX significantly improves the robustness of IDQN. Appendix E provides learning curves in each individual task.

Following the same evaluation protocol, we evaluate EMAX on top of IDQN, VDN, and QMIX across 21 common-reward tasks. Figure 4 visualises the learning curve and performance profile of evaluation returns of all algorithms. Similar to mixed-objective tasks, EMAX substantially improves final evaluation returns of IDQN, VDN, and QMIX in common-reward tasks, shown in Figure 4a, by 60%, 47%, and 538%, leading to higher final returns compared to their vanilla baselines in 18, 16, and 20 out of 21 tasks, respectively. These results arise from EMAX improving the sample efficiency and learning stability of the vanilla algorithms, as we will show in Section 4.2. Additionally, QMIX-EMAX is able to learn effective policies in several hard exploration tasks where QMIX fails to achieve any reward. From the performance profile in Figure 4b we also see that algorithms with EMAX achieve higher returns with a higher probability at the end of training. We provide normalised evaluation returns for each environment and learning curves in individual tasks in Appendix F and Appendix G, respectively.

In LBF, EMAX significantly improves the performance of QMIX whereas minor improvements can be seen for IDQN and VDN in the common-reward setting (Figure 10a), and significant gains are observed in the mixed-objective setting (Figure 5a). Learning curves of individual tasks (Appendix E) show that QMIX, MAVEN, CDS and EMC fail to achieve any rewards in several LBF tasks with particularly sparse rewards. A similar trend can be observed in BPUSH where, most notably, VDN-EMAX and QMIX-EMAX learn to solve a BPUSH task in which four agents need to cooperate and no baseline demonstrates any positive rewards (see Figure 12d).

In RWARE, prior work found that no value-based algorithm was able to achieve notable rewards within four million time steps of training due to the significant sparsity of rewards, and on-policy algorithms like IPPO and MAPPO vastly outperformed value-based



(a) Evaluation returns

(b) Food distances

(c) Cooperation percentage

(d) Ensemble Q-value std

Figure 6: Exploration policy analysis in LBF 10x10-3p-5f with common rewards. (a) Mean and 95% confidence intervals of evaluation returns, mean and standard deviation of (b) average food distances across rollouts, (c) percentages of agents selecting the pick-up action in states that require coordination, and (d) the standard deviation of value estimates for the no-op, movement, and pick-up actions in states that require coordination.

algorithms [37]. In contrast, IDQN-EMAX is able to achieve notable rewards in all RWARE tasks and outperforms all baselines, including both IPPO and MAPPO, in four out of six RWARE tasks. To the best of our knowledge, IDQN-EMAX is the first value-based algorithm that outperforms on-policy algorithms such as IPPO and MAPPO in RWARE tasks. IDQN and VDN achieve 330% and 252% higher final evaluation returns with EMAX than their vanilla algorithms, respectively, whereas QMIX with and without EMAX fail to learn (Figure 10c). Similarly significant improvements can be seen for IDQN in the mixed-objective setting (Figure 5b).

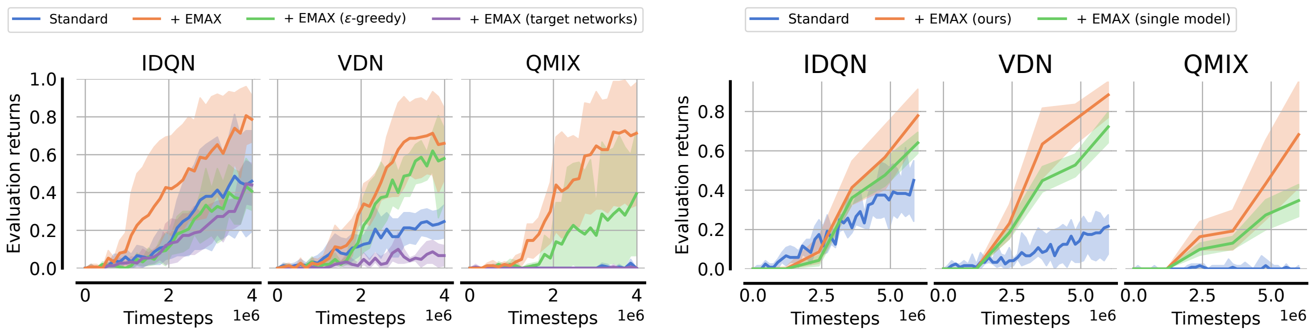
Lastly, we evaluate in three common-reward tasks of the MPE environment. In contrast to other environments, MPE features continuous observations and dense rewards, and the adversary and predator-prey tasks contain stochastic transitions due to the adversarial agent being controlled by a pre-trained policy. In all three MPE tasks, we see improvements in sample efficiency and final performance for algorithms with EMAX compared to vanilla algorithms, even if the improvements are less severe than in the other environments that feature sparse rewards. These improvements are particularly notable in the predator-prey and adversary tasks that feature stochastic transitions due to pretrained policy of adversary agents, indicating that EMAX is able to effectively guide the exploration even in environments with such stochasticity.

4.2 Analysis

We now further investigate the efficacy of all components of EMAX to study our hypotheses that (1) EMAX targets reduce the variability of gradients during training, (2) the EMAX exploration policy leads to more exploration of states and actions with the potential for coordination, and (3) the EMAX evaluation policy reduces the likelihood of selecting sub-optimal actions.

Training stability: To demonstrate that EMAX target computation reduces the variability of gradients during training, and, thus, improves stability of the optimisation, we visualise the average and standard error of the stability of gradients, as defined in Equation (12). We observe that for IDQN with and without EMAX across 11 mixed-objective tasks (Figure 5d) as well as for IDQN, VDN, QMIX with and without EMAX across all 21 common-reward tasks (Figure 4c), EMAX significantly reduces the CVaR of gradient norms for all algorithms. These results indicate more stable optimisation and confirm our hypothesis. The difference for QMIX in the common-reward setting is less pronounced since the base algorithm fails to learn in several tasks, leading to training with low gradient variability independent of target values.

Exploration policy: To validate our hypothesis that the EMAX exploration policy leads to more exploration of states and actions with the potential for coordination (Section 3), we train IDQN,



(a) Exploration policy and target ablations

(b) Evaluation policy ablation

Figure 7: Evaluation returns for all vanilla and EMAX algorithms with ablations of (a) the exploration policy and target computation in LBF 10x10-4p-3f-coop, and (b) of the evaluation policy in LBF 10x10-4p-4f-coop task. We ablate the EMAX exploration policy with an ϵ-greedy policy (green in a), the EMAX target computation with target networks (purple in a), and the EMAX evaluation policy by greedily following any single value function within the ensemble (b).

VDN, and QMIX with and without EMAX in the LBF 10x10-3p-5f task with common rewards where agents need to cooperate to pick up some of the food items. Figure 6 shows the evaluation returns throughout training, the average distances of agents to the closest food, and the percentage of agents selecting the pick-up action in states where multiple agents need to coordinate their actions to pick up food.[4] These results validate our hypotheses about the EMAX exploration policy in the tested task. We observe that agents following the EMAX exploration policy (1) seek out states with the potential for coordination more often compared to the baseline following a random exploration policy, as indicated by the lower average distance of EMAX agents to food items compared to the baseline in Figure 6b, and (2) are more likely to select the cooperative pick-up action in states with potential for coordination, as shown in Figure 6c. Together, these effects lead to EMAX agents learning significantly more efficiently and achieving higher evaluation returns compared to the baseline (Figure 6a).

To separate of the exploration policy and other components of EMAX, we also compare to the percentage of choosing the pick-up action in states that require coordination by greedily following any individual value functions in the ensemble instead of following the UCB policy. While this ablation leads to a significant improvement over the vanilla algorithms, it still exhibits a lower rate of coordinating compared to the EMAX exploration policy (Figure 6c).

Lastly, Figure 6d visualises the standard deviation of action-value estimates across the ensemble for the no-op action, movement actions, and the pick-up action in states with the potential for cooperation between agents. This plot shows that the value estimate deviations across the ensemble are similar for all actions early in training but once agents sometimes cooperate successfully and sometimes fail to cooperate, the deviation for the pick-up action with potential for cooperation rises higher than the deviation for other actions in states with the potential for cooperation. Furthermore, alongside Figure 6a we can see that once agents successfully cooperate most of the time (indicated by high returns), the standard deviation for the pick-up action starts to reduce. For QMIX with EMAX, we can see that this reduction ends in the standard deviation of action values for the cooperative pick-up action and non-cooperative actions reaching similar levels once close-to-optimal performance is reached since now agents almost always cooperative successfully. This further indicates that, as desired, EMAX incentivises exploration of cooperative actions as long as such cooperation is not reliably achieved yet, but this bias towards cooperative actions diminishes as the policy starts to reliably cooperate successfully.

Ablations: To demonstrate the importance of all components of the EMAX algorithm to its performance, we provide ablations of its main components. First, we ablate the exploration policy and target computation and evaluate them in the LBF 10x10-4p-3f-coop task with common rewards (Figure 7a). In these ablations, we replace the EMAX exploration policy with an ϵ-greedy policy, and substitute the target computation with target networks in which each value function in the ensemble has its own target network. We observe that both components significantly improve the performance of all algorithms. Second, we ablate the EMAX evaluation policy in the LBF 10x10-4p-4f-coop task (Figure 7b) by following the greedy policy with respect to any of the individual value functions within the ensemble. We highlight that no value functions were trained for this ablation so the only difference between the ablation and EMAX is the followed policy, not the underlying value functions. This experiment indicates the improved robustness in performance resulting from the EMAX evaluation policy.

Ensemble size: Training ensemble models is expensive and its cost scales with the ensemble size K. We report the training cost for all algorithms and varying K in Appendix H.1. To identify how many models are needed for the benefits of EMAX, we train all algorithms with varying K in the RWARE 11x10 task with four agents and common rewards (Section 4.2), in which EMAX led to substantial improvements for IDQN and VDN. We observe that the benefits of EMAX saturate at $K = 5$, and larger ensemble sizes ($K = 8$) can result in worse performance. We hypothesise that larger ensembles

[4] Average distances to food and cooperation rates are determined over 50 rollout episodes of the exploration policy of baseline algorithms and EMAX every 200,000 time steps of training.

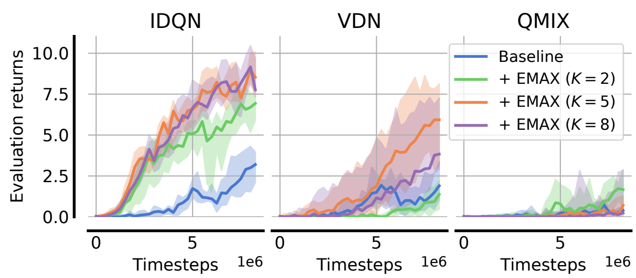


Figure 8: Evaluation returns for varying ensemble sizes $K \in \{2, 5, 8\}$ in the RWARE 11×10 4ag task.

might require more data to train, thus leading to diminishing benefits for ensembles of many value functions. Lastly, we compare EMAX to the baselines with larger non-ensemble networks and find that even at comparable or larger computational budget, the baselines perform significantly worse than EMAX (Appendix H.2).

5 RELATED WORK

Uncertainty for exploration in RL: Using uncertainty to guide exploration is a well-established idea in RL . One family of algorithms that leverages this idea are randomised value functions [34] that build on the idea of Thompson sampling [45] from the multi-armed bandits literature [8, 40]. Posterior sampling RL extends Thompson sampling by maintaining a distribution of plausible tasks, computes optimal policies for sampled tasks, and continually updates its distribution of tasks from the collected experience [32]. This approach has extensive theoretical guarantees [33] but is difficult to apply to complex tasks [35]. This limitation has been addressed in subsequent works [19, 31, 35], most notably in bootstrapped DQN [31] which approximates randomised value functions with an ensemble of value functions. SUNRISE [20] and MeanQ [22] also leverage an ensemble of value functions but instead of sampling value functions to explore, they follow a UCB policy using the average and standard deviation of value estimates across the ensemble to explore. Additionally, SUNRISE computes a weighting of values loss terms based on the variance of target values, and MeanQ stabilises the optimisation by computing lower variance target values as the average value estimate across the ensemble [4]. Separately, Fu et al. [17] extend posterior sampling to model-based RL by learning a probabilistic model of the environment, and Dearden et al. [14] applied these ideas to tabular Q-learning to learn distributions over Q-values and approximate the value of information of actions. Related to all these ideas, optimistic value estimates in the face of uncertainty can be used to promote exploration for actor-critic [13] and model-based RL [41]. All these approaches leverage uncertainty to guide their exploration, similar to EMAX. However, in contrast to discussed approaches, EMAX focuses on exploration of agent coordination in environments with multiple concurrently learning agents.

Multi-agent exploration: There exist a plethora of exploration methods in the multi-agent setting. Several approaches provide agents with intrinsic incentives to explore, e.g. by rewarding them for interacting with each other as measured by influences on their transitions or value estimates [46] or for reaching identified goal states [24]. However, intrinsic rewards for exploration have to be carefully balanced for each task due to the modified optimisation objective [39]. To address this challenge, LIGS [27] trains an agent to determine when and which intrinsic reward should be given to each agents. Orthogonally to this line of work, experience and parameter sharing have been leveraged to greatly improve sample efficiency for MARL by synchronising agents' learning and make use of more data [11, 12]. However, there is little research using distributional and ensemble-based techniques for MARL exploration. Zhou et al. [49] extend posterior sampling [32] for MARL, but are limited to two-player zero-sum extensive games. We aim to close this gap by proposing EMAX, an ensemble-based technique for efficient exploration in MARL. We further highlight that EMAX is a plug-and-play algorithm that can enhance any value-based MARL algorithm, including most existing MARL exploration techniques described in this paragraph.

6 CONCLUSION

In this paper, we proposed EMAX, a general framework to seamlessly extend any value-based MARL algorithm using ensembles of value functions. EMAX leverages the disagreement of value estimates across the ensemble to systematically guide the exploration of agents towards parts of the environment that might require them to coordinate with other agents. Additionally, EMAX computes low-variance target values across the ensemble to stabilise MARL training that is otherwise prone to be unstable due to the non-stationarity of the policies of other agents, and reduces the risk of miscoordination by computing optimal actions through a majority vote across the ensemble. We empirically demonstrated the benefits of EMAX for sample efficiency, final performance, and training stability as an extension of IDQN, VDN, and QMIX across 11 mixed-objective tasks and 21 common-reward tasks across four environments. Further analysis and ablations established the efficacy of the EMAX exploration policy, target computation, and evaluation policy of EMAX, and we discussed the computational cost introduced by EMAX with experiments indicating that comparably small ensemble models are sufficient to achieve the demonstrated improvements. We believe that EMAX is a promising approach to improve exploration in MARL due to its plug-and-play nature and demonstrated efficacy in complex tasks. In this work, we design EMAX as an extension of value-based MARL algorithms but future work could investigate the application of EMAX to actor-critic MARL algorithms such as IPPO and MAPPO by training an ensemble of value functions and policies. A further limitation of the current EMAX approach is the considerable computation cost of training ensembles of value functions. Future work could consider the application of hypernetworks [16] or latent-conditioned models [42] to approximate the ensemble and, thereby, reduce the computational cost of EMAX.

ACKNOWLEDGMENTS

We would like to thank Jun Wang, Matthieu Zimmer, Alexandre Maraval, and Juliusz Ziomek for providing helpful feedback during the ideation of this project. Their comments have helped to focus and strengthen this work.

REFERENCES

[1] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc Bellemare. 2021. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*.

[2] Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. 2024. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches.* MIT Press. https://www.marl-book.com

[3] Stefano V. Albrecht and Subramanian Ramamoorthy. 2013. A Game-Theoretic Model and Best-Response Learning Method for Ad Hoc Coordination in Multiagent Systems. In *International Conference on Autonomous Agents and Multi-Agent Systems.*

[4] Oron Anschel, Nir Baram, and Nahum Shimkin. 2017. Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning. In *International Conference on Machine Learning.*

[5] Peter Auer. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3 (2002).

[6] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research* (2002).

[7] Stephanie C.Y. Chan, Samuel Fishman, John Canny, Anoop Korattikara, and Sergio Guadarrama. 2020. Measuring the reliability of reinforcement learning algorithms. In *International Conference on Learning Representations.*

[8] Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems.*

[9] Kyunghyun Cho, Bart Van Merriënboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing.*

[10] Filippos Christianos, Georgios Papoudakis, and Stefano V. Albrecht. 2023. Pareto Actor-Critic for Equilibrium Selection in Multi-Agent Reinforcement Learning. *Transactions on Machine Learning Research* (2023).

[11] Filippos Christianos, Georgios Papoudakis, Muhammad A. Rahman, and Stefano V. Albrecht. 2021. Scaling multi-agent reinforcement learning with selective parameter sharing. In *International Conference on Machine Learning.*

[12] Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. 2020. Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems.*

[13] Kamil Ciosek, Quan Vuong, Robert Loftin, and Katja Hofmann. 2019. Better Exploration with Optimistic Actor Critic. In *Advances in Neural Information Processing Systems.*

[14] Richard Dearden, Nir Friedman, and Stuart Russell. 1998. Bayesian Q-learning. *AAAI* (1998).

[15] Yali Du, Lei Han, Meng Fang, Ji Liu, Tianhong Dai, and Dacheng Tao. 2019. LIIR: Learning Individual Intrinsic Reward in Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems.*

[16] Vikranth Dwaracherla, Xiuyuan Lu, Morteza Ibrahimi, Ian Osband, Zheng Wen, and Benjamin van Roy. 2020. Hypermodels for exploration. In *International Conference on Learning Representations.*

[17] Haotian Fu, Shangqun Yu, Michael Littman, and George Konidaris. 2022. Model-based Lifelong Reinforcement Learning with Bayesian Exploration. In *Advances in Neural Information Processing Systems.*

[18] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. 2004. Dynamic programming for partially observable stochastic games. In *AAAI Conference on Artificial Intelligence.*

[19] David Janz, Jiri Hron, Przemysław Mazur, Katja Hofmann, José Miguel Hernández-Lobato, and Sebastian Tschiatschek. 2019. Successor uncertainties: Exploration and uncertainty in temporal difference learning. In *Advances in Neural Information Processing Systems.*

[20] Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. 2021. SUNRISE: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning.*

[21] Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. 2021. Celebrating diversity in shared multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems.*

[22] Litian Liang, Yaosheng Xu, Stephen McAleer, Dailin Hu, Alexander Ihler, Pieter Abbeel, and Roy Fox. 2022. Reducing Variance in Temporal-Difference Value Estimation via Ensemble of Deep Networks. In *International Conference on Machine Learning.*

[23] Michael L. Littman. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning*. Elsevier, 157–163.

[24] Iou-Jen Liu, Unnat Jain, Raymond A. Yeh, and Alexander Schwing. 2021. Cooperative exploration for multi-agent deep reinforcement learning. In *International Conference on Machine Learning.*

[25] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems.*

[26] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. 2019. MAVEN: Multi-Agent Variational Exploration. In *Advances in Neural Information Processing Systems.*

[27] David Henry Mguni, Taher Jafferjee, Jianhong Wang, Nicolas Perez-Nieves, Oliver Slumbers, Feifei Tong, Yang Li, Jiangcheng Zhu, Yaodong Yang, and Jun Wang. 2022. LIGS: Learnable Intrinsic-Reward Generation Selection for Multi-Agent Learning. In *International Conference on Learning Representations.*

[28] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[29] Igor Mordatch and Pieter Abbeel. 2018. Emergence of grounded compositional language in multi-agent populations. In *AAAI Conference on Artificial Intelligence.*

[30] Frans A. Oliehoek and Christopher Amato. 2016. *A Concise Introduction to Decentralized POMDPs.* Vol. 1. Springer.

[31] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin van Roy. 2016. Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems.*

[32] Ian Osband, Daniel Russo, and Benjamin van Roy. 2013. (More) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems.*

[33] Ian Osband and Benjamin van Roy. 2017. Why is posterior sampling better than optimism for reinforcement learning?. In *International Conference on Machine Learning.*

[34] Ian Osband, Benjamin van Roy, Daniel J. Russo, and Zheng Wen. 2019. Deep Exploration via Randomized Value Functions. *Journal of Machine Learning Research* 20, 124 (2019).

[35] Ian Osband, Benjamin van Roy, and Zheng Wen. 2016. Generalization and exploration via randomized value functions. In *International Conference on Machine Learning.*

[36] Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V. Albrecht. 2019. Dealing with Non-Stationarity in Multi-Agent Deep Reinforcement Learning. *arXiv preprint 1906.04737* (2019).

[37] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. 2021. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. In *Advances in Neural Information Processing Systems, Track on Datasets and Benchmarks.*

[38] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research* 21, 1 (2020).

[39] Lukas Schäfer, Filippos Christianos, Josiah P. Hanna, and Stefano V. Albrecht. 2022. Decoupled Reinforcement Learning to Stabilise Intrinsically-Motivated Exploration. In *International Conference on Autonomous Agents and Multiagent Systems.*

[40] Steven L. Scott. 2010. A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry* 26, 6 (2010).

[41] Pier Giuseppe Sessa, Maryam Kamgarpour, and Andreas Krause. 2022. Efficient Model-based Multi-agent Reinforcement Learning via Optimistic Equilibrium Computation. In *International Conference on Machine Learning.*

[42] Macheng Shen and Jonathan P. How. 2023. Implicit Ensemble Training for Efficient and Robust Multiagent Reinforcement Learning. *Transactions on Machine Learning Research* (2023).

[43] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech M. Czarnecki, Vinícius Flores Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 2018. Value-Decomposition networks for cooperative multi-agent learning. In *International Conference on Autonomous Agents and Multi-Agent Systems.*

[44] Ming Tan. 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *International Conference on Machine Learning.*

[45] William R. Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3/4 (1933), 285–294.

[46] Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. 2020. Influence-based multi-agent exploration. In *International Conference on Learning Representations.*

[47] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Advances in Neural Information Processing Systems, Track on Datasets and Benchmarks.*

[48] Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. 2021. Episodic multi-agent reinforcement learning with curiosity-driven exploration. In *Advances in Neural Information Processing Systems.*

[49] Yichi Zhou, Jialian Li, and Jun Zhu. 2020. Posterior sampling for multi-agent reinforcement learning: Solving extensive games with imperfect information. In *International Conference on Learning Representations.*