# Surprise Minimizing Multi-Agent Learning with Energy-based Models

**Karush Suri**[1], **Xiao Qi Shi**[2], **Konstantinos Plataniotis**[1], **Yuri Lawryshyn**[1]
[1] University of Toronto, [2] RBC Capital Markets
karush.suri@mail.utoronto.ca

## Abstract

Multi-Agent Reinforcement Learning (MARL) has demonstrated significant success by virtue of collaboration across agents. Recent work, on the other hand, introduces *surprise* which quantifies the degree of change in an agent's environment. Surprise-based learning has received significant attention in the case of single-agent entropic settings but remains an open problem for fast-paced dynamics in multi-agent scenarios. A potential alternative to address surprise may be realized through the lens of free-energy minimization. We explore surprise minimization in multi-agent learning by utilizing the free energy across all agents in a multi-agent system. A temporal Energy-Based Model (EBM) represents an estimate of surprise which is minimized over the joint agent distribution. Our formulation of the EBM is theoretically akin to the minimum conjugate entropy objective and highlights suitable convergence towards minimum surprising states. We further validate our theoretical claims in an empirical study of multi-agent tasks demanding collaboration in the presence of fast-paced dynamics. Our implementation and agent videos are available at the Project Webpage.

## 1 Introduction

The rise of RL has led to an increasing interest in the study of multi-agent systems [34, 58], commonly known as Multi-Agent Reinforcement Learning (MARL). In the case of partially observable settings, MARL enables the learning of policies with centralised training and decentralised control [26]. This has proven to be useful for exploiting value-based methods which motivate collaboration across large number of agents. *But how do agents behave in the presence of sudden environmental changes?*

Consider the problem of autonomous driving wherein a *driver* (agent) autonomously operates a vehicle in real-time. The *driver* learns to optimize the reward function by maintaining constant speed and covering more distance in different traffic conditions. Whenever the vehicle approaches an obstacle, the *driver* acts to avoid it by utilizing the brake and directional steering commands. However, due to the fast-paced dynamics of the environment, say fast-moving traffic, the agent may abruptly encounter an obstacle (*a person running across the street*) which may result in a collision. Irrespective of the optimal action (*pushing of brakes*) executed by the agent, the vehicle may fail to evade the collision as a result of the abrupt temporal change.

The above arises as a consequence of *surprise*, which is defined as a statistical measure of uncertainty. Surprise minimization [3] is a recent phenomenon observed in the case of single-agent RL methods which deals with environments consisting of rapidly changing states. In the case of model-based RL [24], surprise minimization is used as an effective planning tool in the agent's model [3]. In the case of model-free RL, surprise minimization is witnessed as an intrinsic motivation [1, 36] or generalization problem [9]. On the other hand, MARL does not account for surprise across agents as a result of which agents remain unaware of drastic changes in the environment [35]. Thus, surprise minimization in the case of multi-agent settings requires attention from a critical standpoint.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

A potential pathway to treat surprising states may be realized in light of free-energy minimization. The free-energy principle depicts convergence to local niches and provides a general recipe for stability among agents. Through this lens, we unify surprise with free-energy in the multi-agent setting. We construct a temporal EBM which represents an estimate of surprise agents may face in the environment. All agents jointly minimize this estimate utilizing temporal difference learning upon their value functions and the EBM. Our formulation of free-energy minimization is theoretically akin to minimizing the entropy in conjugate gradient space. This insight provides a suitable convergence result towards minimum surprising states (or niches) of the agent state distributions. In an empirical study of multi-agent tasks which present significant collaboration bottlenecks and fast-paced dynamics, we validate our theoretical claims and motivate the practical usage of EBMs in MARL.

## 2 Related Work

**Surprise Minimization:** Despite the recent success of value-based methods [39, 22] RL agents suffer from spurious state spaces and encounter sudden changes in trajectories. Quantitatively, surprise has been studied as a measure of deviation [3, 9] among states encountered by the agent during its interaction with the environment. While exploring [7, 56] the environment, agents tend to have higher deviation among states which is gradually reduced by gaining a significant understanding of state-action transitions. In the case of model-based RL, agents can leverage spurious experiences [3] and plan effectively for future steps. On the other hand, in the case of model-free RL, surprise results in sample-inefficient learning [1]. This is primarily addressed by making use of rigorous exploration strategies [52, 31]. High-dimensional exploration further requires extrinsic feature engineering [27] and meta models [16]. A suitable way to tackle high-dimensional dynamics is by utilizing surprise as a penalty on the reward [9]. This leads to improved generalization for single-agent interactions [45]. Our proposed approach is parallel to the aforesaid methods.

**Energy-based Models:** EBMs have been successfully implemented in single-agent RL methods [42, 19]. These typically make use of Boltzmann distributions to approximate policies [32]. Such a formulation results in the minimization of free energy within the agent. While policy approximation depicts promise in the case of unknown dynamics, inference methods [57] play a key role in optimizing goal-oriented behavior.

A second type of usage of EBMs follows the maximization of entropy [65]. The maximum entropy framework [20] highlighted in Soft Q-Learning (SQL) [19] allows the agent to obey a policy which maximizes its reward and entropy concurrently. Maximization of agent's entropy results in diverse and adaptive behaviors [64] which may be difficult to accomplish using standard exploration techniques [7, 56]. The maximum entropy framework is akin to approximate inference in the case of policy gradient methods [49]. Such a connection between likelihood ratio gradient techniques and energy-based formulations leads to diverse and robust policies [17]. Furthermore, their hierarchical extensions [18] preserve the lower levels of hierarchies. In the case of MARL, EBMs have witnessed limited applicability as a result of the increasing number of agents and complexity within each agent [8]. While the framework is readily transferable to opponent-aware multi-agent systems [63], cooperative settings consisting of coordination between agents require a firm formulation of energy. This formulation must be scalable in the number of agents [15] and account for environments consisting of spurious states [62]. Our theoretical formulation is motivated by these methods in literature.

## 3 Preliminaries

**Multi-Agent Learning:** We review the cooperative MARL setup. The problem is modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [43] defined by the tuple $(\mathcal{S}, \mathcal{A}, r, N, P, Z, O, \gamma)$ where the state space $\mathcal{S}$ and action space $\mathcal{A}$ are discrete, $r : \mathcal{S} \times \mathcal{A} \rightarrow [r_{min}, r_{max}]$ presents the reward observed by agents $a \in N$ where $N$ is the set of all agents, $P : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ presents the unknown transition model consisting of the transition probability to the next state $s' \in \mathcal{S}$ given the current state $s \in \mathcal{S}$ and joint action $u \in \mathcal{A}$ (a combination of each agent's action $u^a \in \mathcal{A}^a$) at time step $t$ and $\gamma$ is the discount factor. Our setting consists of the finite-horizon discounted problem case where episodes terminate at timestep $T$ with the terminal state being $s_T$. As a result, task returns remain bounded for each episode. We consider a partially observable setting in which each agent $n$ draws individual observations $z \in Z$ according to the observation function $O(s, u) : \mathcal{S} \times \mathcal{A} \rightarrow Z$. We consider a joint policy $\pi_\theta(u|s)$ as a function of

model parameters $\theta$. Standard RL defines the agent's objective to maximize the expected discounted reward $\mathbb{E}_{\pi_\theta}[\sum_{t=0}^{T}\gamma^t r(s_t, u_t)]$ as a function of the parameters $\theta$. The joint action-value function for agents is represented as $Q(u, s; \theta) = \mathbb{E}_{\pi_\theta}[\sum_{t=1}^{T}\gamma^t r(s, u)|s = s_t, u = u_t]$ which is the expected sum of payoffs obtained in state $s$ upon performing action $u$ by following the policy $\pi_\theta$. We denote the optimal policy $\pi_{\theta^*}$( shorthand $\pi^*$) such that $Q(u, s; \theta^*) \geq Q(u, s; \theta)\forall s \in S, u \in A$. In the case of multiple agents, the joint optimal policy can be expressed as the Nash Equilibrium [40] of the Stochastic Markov Game as $\pi^* = (\pi^{1,*}, \pi^{2,*}, ...\pi^{N,*})$ such that $Q(u^a, s; \theta^*) \geq Q(u^a, s; \theta)\forall s \in S, u \in A, a \in N$. Q-Learning is an off-policy, model-free algorithm suitable for continuous and episodic tasks. The algorithm uses semi-gradient descent to minimize the Temporal Difference (TD) error in Equation 1.

$$\mathbb{L}(\theta) = \mathop{\mathbb{E}}_{s,u,s'\sim\mathcal{R}}\left[\left(r + \gamma \max_{u'\in A} Q(u', s'; \theta^-) - Q(u, s; \theta)\right)^2\right] \tag{1}$$

where $y = r + \gamma \max_{u'\in A} Q(u', s'; \theta^-)$ is the TD target consisting of $\theta^-$ as the target parameters and $\mathcal{R}$ denotes the replay buffer.

**Energy-based Models:** EBMs [29, 30] have been successfully applied in the field of machine learning [55] and probabilistic inference [37]. A typical EBM $\mathcal{E}$ formulates the equilibrium probabilities [47] $P(v, h) = \frac{\exp(-\mathcal{E}(v,h))}{\sum_{\hat{v},\hat{h}}[\exp(-\mathcal{E}(\hat{v},\hat{h}))]}$ via a Boltzmann distribution [32] where $v$ and $h$ are the values of the visible and hidden variables and $\hat{v}$ and $\hat{h}$ are all the possible configurations of the visible and hidden variables respectively. The probability distribution over all the visible variables can be obtained by summing over all possible configurations of the hidden variables. This is mathematically expressed in Equation 2.

$$P(v) = \frac{\sum_h \exp(-\mathcal{E}(v, h))}{\sum_{\hat{v},\hat{h}} \exp(-\mathcal{E}(\hat{v}, \hat{h}))} \tag{2}$$

Here, $\mathcal{E}(v, h)$ is called the equilibrium free energy which is the minimum of the variational free energy and $\sum_{\hat{v},\hat{h}} \exp(-\mathcal{E}(\hat{v}, \hat{h}))$ is the partition function.

## 4 Energy-based Surprise Minimization

We begin by constructing surprise minimization as an energy-based problem in the temporal setting. The motivation behind an energy-based formulation stems from rapidly changing states as an undesired niche among agents in the case of partially-observed settings. To steer agents away from this niche, we further construct a method which incorporates the theoretical aspect of the study.

### 4.1 The Surprise Minimization Objective

To make analysis tractable towards valid function spaces and surprising states, we take into account two assumptions which form the central basis of surprise minimization among multiple agents.

**Assumption 1.** (Completeness of value function space) *The space $\Pi : \mathcal{S} \times \mathcal{A}$ of all $Q$ value functions $Q(s, u) \in \Pi,\ \forall s \in \mathcal{S},\ \forall u \in \mathcal{A}$ is a nonempty complete metric space.*

Assumption 1 restricts the formulation of individual agent value functions $Q_a$ to the nonempty complete metric space. A nonempty space confirms the presence of candidate functions $Q_a$ upper bounded by the optimal function $Q^*$, i.e.- $Q_a \leq Q^*,\ \forall a \in N$ [5]. The completeness counterpart, on the other hand, provisions a fixed interior **int** $\Pi$ for optimization [6].

**Assumption 2.** (Constant surprise at Equilibrium) *In the limit of convergence $\lim_{\pi_a \to \pi^*}$ to an optimal policy $\pi^*$, all agents $a \in N$ incur a finite surprise $\zeta > 0$ between consecutive states $s$ and $s'$ until termination state $s_T$.*

Assumption 2 is directly based on the constant and continuous temporal aspect of surprise minimization [50, 12]. Corresponding to the lifetime of each agent $a \in N$, a desired minima bakes in the optimal distribution of actions which correspond to minimum but finite instantaneous surprise.

We formulate the energy-based objective consisting of surprise as a function of states $s$, joint actions $u$ and standard deviation $\sigma$ of observations for each agent $a$. In the case of high-dimensional state spaces (such as multiple opponents), $\sigma$ informs agents of the abrupt statistical change that would take place upon executing action $u$. We formulate surprise as $\mathcal{T}V^a_{\text{surp}}(s, u, \sigma)$ which serves as an uncertainty quantifier Unc(s,a) of the state-action distribution. Here $V^a_{\text{surp}}(s, u, \sigma)$ denotes the surprise value function which serves as a mapping from agent and environment dynamics to surprise. Define an operator presented in Equation 3 which sums surprising configurations across all agents.

$$\mathcal{T}V^a_{\text{surp}}(s, u, \sigma) = \log \sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right) \tag{3}$$

**Remark 1.** *$\mathcal{T}V^a_{\text{surp}}(s, u, \sigma)$ intuitively provides a global estimate of surprise. If all agents are equally likely to face a surprising state, then $\mathcal{T}V^a_{\text{surp}}(s, u, \sigma)$ captures their individual contributions.*

The formulation makes use of the soft-maximum operator [2]. The operator $\mathcal{T}V^a_{\text{surp}}(s, u, \sigma)$ is similar to prior energy formulations [19] where the energy across different actions is evaluated. In our case, inference is carried out across all agents with actions as prior variables. However, in the special case of using an EBM as a $Q$-function, our approach suitably generalizes to the above methods (details in Appendix B).

Our choice of $\mathcal{T}V^a_{\text{surp}}(s, u, \sigma)$ is based on its unique mathematical properties which result in better convergence. Of these properties, the most useful result is that $\mathcal{T}$ forms a contraction on the surprise value function $V^a_{\text{surp}}(s, u, \sigma)$ indicating a guaranteed minimization of surprise within agents. This is formally stated in Theorem 1 while utilizing the completeness criterion of Assumption 1 which provides a tractable value function space. All proofs are deferred to Appendix A.

**Theorem 1.** *Given a surprise value function $V^a_{\text{surp}}(s, u, \sigma)\ \forall a \in N$, the energy operator $\mathcal{T}V^a_{\text{surp}}(s, u, \sigma) = \log \sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right)$ forms a contraction on $V^a_{\text{surp}}(s, u, \sigma)$.*

Theorem 1 provides a suitable guarantee of $\mathcal{T}V^a_{\text{surp}}(s, u, \sigma)$ converging to a fixed point niche. The contraction result is directly based on Banach's fixed point property and suggests the generalization of convergence in any nonempty complete metric space $(X, d)$ [5].

We now consider a weighted combination of $Q(s, u)$ with $\mathcal{T}V^a_{\text{surp}}(s, u, \sigma)$ wherein we denote $\beta$ as a temperature parameter,

$$\hat{Q}(u, s; \theta) = Q(u, s; \theta) + \beta \log \sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right)) \tag{4}$$

**Remark 2.** *Equation 4 is an instance of value function regularization wherein the $Q$ values are subject to a joint penalty while observing surprising states.*

Interestingly, upon considering the Legendre transform $f^*(x)$ [6, 14] (convex conjugate function corresponding to the conjugate space $\mathcal{X}$ of a differentiable function $f(\text{z})$) of $\mathcal{T}V^a_{\text{surp}}(s, u, \sigma)$, we obtain the following,

$$f^*(x) = \sup_{\text{z} \in \text{dom}\, f} \left(x^{\text{T}}\text{z} - f(\text{z})\right)\ ,\ f(\text{z}) = \mathcal{T}V^a_{\text{surp}}(s, u, \sigma) \tag{5}$$

$$f^*(x) = \sum_x x \log(x)\ ,\ x = \nabla_{\text{z}} f(\text{z}) \in \mathcal{X} \tag{6}$$

**Remark 3.** *The Legendre Transform of $\mathcal{T}V^a_{\text{surp}}(s, u, \sigma)$ given by $f^*(x) = \sum_x x \log(x)$ when utilized as value function regularization $\hat{Q} = Q - f^*(x)$ corresponds to the minimum entropy formulation in conjugate space $\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t (r(s_t, u_t) - \lambda \mathcal{H}(x))\right]$ for $x = \nabla_{\text{z}} f(\text{z}) \in \mathcal{X}$.*

Based on the above insight, minimizing entropy to express $\nabla_z f(z)$ in conjugate space is akin to minimizing uncertainty among all agents in the value function space $\Pi$. Intuitively, $\mathcal{H}(x)$ denotes the uncertainty for each agent $a \in N$ in the multi-agent population which is directly related to its

ability of accurately interpreting the environment. Minimizing $\mathcal{H}(x)$ leads to an increase in the expressiveness of value function. This in turn, induces an expressive state visitation distribution which steers the agent away from sudden changes in its environment. Note that the setting does not minimize entropy in value function space which would stand contrary to the maximum entropy formulation [20] (see Appendix B).

Figure 1 presents an illustration of the intuition behind surprise minimization using the energy-based scheme. Agents collaborate in partially-observed worlds to attain a joint niche. Interpreting the space of all surprising states as an energy landscape, MARL agents move from high energy states to low energy states which consist of minimum surprise. During training, agents train to find policies which not only provide rewarding actions, but also avoid risky states by minimizing $\mathcal{T}V^a_{\text{surp}}(s, u, \sigma)$. Seeking these states leads to finding the minima on the energy landscape. Thus, it is by virtue of regularized value estimates $\hat{Q}$ that the minimization scheme informs agents of joint surprise.

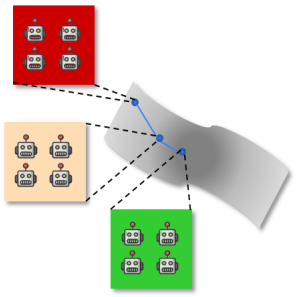

Figure 1: Agent populations (robots) traverse the energy landscape (in grey) during update steps (blue) to seek energy minima (darker shade at center). This results in surprise minimization from high (red) to low energy (green) niches.

### 4.2 Surprise Minimization with Function Approximation

We utilize the above insights as surprise-based regularization in the TD learning setting. Upon replacing $Q(u, s; \theta)$ with $\hat{Q}(u, s, ; \theta)$ in the RL construction of Equation 1 one obtains the following,

$$L(\theta) = \mathop{\mathbb{E}}_{s,u,s' \sim \mathcal{R}} \left[ \frac{1}{2} \left( \hat{y} - (Q(u, s; \theta) + \beta \log \sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right)) \right)^2 \right]$$

where $\hat{y}$ is given by the following expression,

$$\hat{y} = r + \gamma \max_{u'} Q(u', s'; \theta^-) + \beta \log \sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s', u', \sigma')\right) \tag{7}$$

Collecting the log terms yields the following,

$$L(\theta) = \mathop{\mathbb{E}}_{s,u,s' \sim \mathcal{R}} \left[ \frac{1}{2} \left( r + \gamma \max_{u'} Q(u', s'; \theta^-) \right. \right. \\ \left. \left. + \beta \log \left( \frac{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s', u', \sigma')\right)}{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right)} \right) - Q(u, s; \theta) \right)^2 \right]$$

$$L(\theta) = \mathop{\mathbb{E}}_{s,u,s' \sim \mathcal{R}} \left[ \frac{1}{2} \left( r + \gamma \max_{u'} Q(u', s'; \theta^-) + \beta E - Q(u, s; \theta) \right)^2 \right] \tag{8}$$

Here, $E$ is defined as the *surprise ratio*. The surprise value function $V^a_{\text{surp}}(s', u', \sigma')$ is expressed as the negative free energy and $\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right)$ as the partition function of a conventional EBM described in Equation 2. Alternatively, $V^a_{\text{surp}}(s, u, \sigma)$ can be formulated as the negative free energy with $\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s', u', \sigma')\right)$ as the partition function. The TD objective incorporates the minimization of surprise across all agents as minimizing the energy in rapidly changing states.

**Remark 4.** *The above formulation of $\beta E$ can be realized as intrinsic motivation steering the agent towards subgoals with reduced surprise.*

The energy formulation $E$ provides a tractable distribution over all surprising configurations in the state space $\mathcal{S}$. This guarantees convergence to minimum surprise at optimal policy $\pi^*$ and is formally expressed in Theorem 2 (see Appendix C for a detailed convergence analysis).

**Theorem 2.** *Upon agent's convergence to an optimal policy $\pi^*$, total energy of $\pi^*$, expressed by $E^*$ will reach a thermal equilibrium consisting of minimum surprise among consecutive states $s$ and $s'$.*

Theorem 2 demonstrates an intuitive convergence result of agent populations collaborating to reside in a mutual ecological niche [12]. The multi-agent population with minimum surprise exhibits the optimal policy $\pi^*$ which results in minimum energy corresponding to each surprising state in the state distribution $\mathcal{S}$. Orthogonally, agents may continue to experience finite and constant surprise in the long-horizon while acting optimally to visit non-surprising and rewarding states. This presents surprise minimization as a secondary surrogate objective in MARL.

### 4.3 Energy-based MIXer (EMIX)

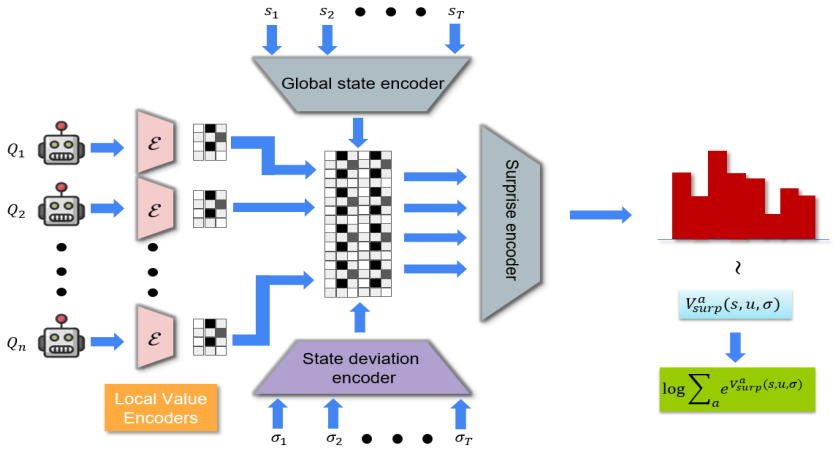


Figure 2: The EMIX architecture for learning surprise across global states.

Based on our theoretical analysis, we incorporate learning of surprise as global intrinsic motivation across all agents in the multi-agent system. A global estimate of surprise, following the energy operator $\mathcal{T}V^a_{\text{surp}}(s,u,\sigma)$, is befitting from a computational perspective as well. An individual estimate of surprise for each agent may be intractable to obtain due to the non-stationarity of the environment. Instead, we seek to minimize surprise jointly across all agents using an expressive Energy-based MIXer (EMIX) architecture which is compatible with any multi-agent RL algorithm. Figure 2 illustrates our learning scheme.

Learning of surprise in the high-dimensional value function space is cumbersome with the number of actions scaling linearly in the number of agents. This imposes an inherent restriction to learn global surprise efficaciously across all agents at a given timestep. Towards this goal, EMIX encodes individual value functions $Q_1,\ Q_2,\ \ldots\ Q_n$ corresponding to each agent using local value encoders. These encoders capture the local change in value functions arising over subsequent TD learning iterations [60]. A global state encoder maps environment states $s_1,\ s_2,\ \ldots\ s_T$ to a low dimensional representation. Further, a state deviation encoder encodes deviations across all states $s_1,\ s_2,\ \ldots\ s_T$ within the given batch. Akin to a model-based method [23], the state deviation encoder accounts for uncertainty in an agent's state visitation distribution. Note that the encoder does not construct an explicit model of states, but only represents their variation in the agent's environment. This insight is essential to account for abrupt dynamics encountered by agents. Representations obtained from state and value function encoders are concatenated and compressed using a final surprise encoder which estimates a distribution of surprise values. The distribution implicitly represents the density of states wherein an agent may encounter most surprise. A value estimate $V^a_{\text{surp}}(s,u,\sigma)$ sampled from the surprise distribution depicts the variational free energy configuration upon application of $\mathcal{T}$ which serves as global intrinsic motivation. Practical training of EMIX proceeds with backpropagation [46] using gradient descent and the reparameterization trick [25] for sampling of $V^a_{\text{surp}}(s,u,\sigma)$.

### 4.4 Practical Implementation

Algorithm 1 presents the EMIX framework (in **green**) combined with QMIX [44], an off-the-shelf MARL algorithm. The total $Q$-value $Q^{\theta}_{tot}$ is computed by the mixer network with its inputs as the $Q$-values of all the agents conditioned on $s$ via the hypernetworks. Similarly, the target mixers approximate $Q^{\theta^-}_i$ conditioned on $s'$. In order to evaluate surprise within agents, we compute the standard deviations $\sigma$ and $\sigma'$ across all observations $z$ and $z'$ for each agent using $s$ and $s'$ respectively. The surprise value function, called the Surprise-Mixer, estimates surprise $V^a_{\text{surp}}(s, u, \sigma)$ conditioned on $s$, $u$ and $\sigma$. The same computation is repeated using the Target-Surprise-Mixer for estimating surprise $V^a_{\text{surp}}(s', u', \sigma')$ within next-states in the batch. Application of the energy operator along the non-singleton agent dimension for $V^a_{\text{surp}}(s, u, \sigma)$ and $V^a_{\text{surp}}(s', u', \sigma')$ yields the energy ratio $E$ which is used in Equation 8 to evaluate $L(\theta)$. We then use batch gradient descent to update parameters of the mixer $\theta$. Target parameters $\theta^-_i$ are updated every $update - interval$ steps.

**Algorithm 1** Energy-based MIXer (EMIX)

Initialize $\phi$, $\theta$, $\theta^-_1 ..., \theta^-_m$, agent and hypernetwork parameters.
Initialize learning rate $\alpha$, temperature $\beta$ and replay buffer $\mathcal{R}$.
**for** environment step **do**
  $u \leftarrow (u_1, u_2 ..., u_N)$
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{(s, u, r, s')\}$
  **if** $|\mathcal{R}| >$ batch-size **then**
    **for** random batch **do**
      $Q^{\theta}_{tot} \leftarrow$ *Mixer-Network*$(Q_1, Q_2 ..., Q_N, s)$
      $Q^{\theta^-}_i \leftarrow$ *Target-Mixer*$_i(Q_1, Q_2 ..., Q_N, s'), \forall i = 1, 2.., m$
      **Calculate** $\sigma$ **and** $\sigma'$ **using** $s$ **and** $s'$
      $V^a_{\text{surp}}(s, u, \sigma) \leftarrow$ ***Surprise-Mixer***$(s, u, \sigma)$
      $V^a_{\text{surp}}(s', u', \sigma') \leftarrow$ ***Target-Mixer***$(s', u', \sigma')$
      $E \leftarrow \log \left( \frac{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s', u', \sigma')\right)}{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right)} \right)$
      **Calculate** $L(\theta)$ **using** $E$ **in Equation 8**
      $\theta \leftarrow \theta - \alpha \nabla_{\theta} L(\theta)$
    **end for**
  **end if**
  **if** update-interval steps have passed **then**
    $\theta^-_i \leftarrow \theta, \forall i = 1, 2.., m$
  **end if**
**end for**

## 5 Experiments

Our experiments aim to evaluate the theoretical claims presented by EMIX along with its performance to prior MARL methods. Specifically, we aim to answer the following questions;

**(1)** How does the provision of an EBM for surprise minimization compare to current MARL methods?

**(2)** Does the algorithm validate the theoretical claims corresponding to its components?

### 5.1 Energy-based Surprise Minimization

We assess the validity of EMIX, when combined with QMIX, on multi-agent StarCraft II micromanagement scenarios [48] as these consist of a larger number of agents with different action spaces. This in turn motivates a greater deal of coordination. Additionally, micromanagement scenarios in StarCraft II consist of multiple opponents which introduce a greater degree of surprise within consecutive states.

We compare our method to prior methods namely; (1) QMIX [44], constituting of nonlinear value function factorization with monotonicity constraints; (2) Value Decomposition Networks (VDN)

| Scenarios | EMIX | SMiRL-QMIX | QMIX | VDN | COMA | IQL |
|---|---|---|---|---|---|---|
| 3m | **94.90±0.39** | 93.94±0.22 | 93.43±0.20 | 94.58±0.58 | 84.75±7.93 | 94.79±0.50 |
| 3s_vs_4z | **97.22±0.73** | 0.24±0.11 | 96.01±3.93 | 94.29±2.13 | 0.00±0.00 | 59.75±12.22 |
| 8m_vs_9m | **71.03±2.69** | 69.90±1.94 | **68.28±2.30** | 58.81±4.68 | 4.17±0.58 | 28.48±22.38 |
| 10m_vs_11m | 75.35±2.30 | **77.85±2.02** | 70.36±2.87 | 71.81±6.50 | 4.55±0.73 | 32.27±25.68 |
| so_many_baneling | **95.87±0.16** | 93.61±0.94 | 93.35±0.78 | 92.26±1.06 | 91.65±2.26 | 74.97±6.52 |
| 5m_vs_6m | **37.07±2.42** | 33.27±2.79 | 34.42±2.63 | 35.63±3.32 | 0.52±0.13 | 14.78±2.72 |

Table 1: Comparison of success rate percentages between EMIX and prior MARL methods on StarCraft II micromanagement scenarios. EMIX is comparable to or improves over QMIX agent. In comparison to SMiRL-QMIX, EMIX demonstrates improved minimization of surprise. Results are averaged over 5 random seeds.

[53], consisting of linear additive factorization of $Q$ function; (3) Counterfactual Multi-Agent Policy Gradients (COMA) [11], which consist of counterfactual actor-critic updates in a centralized critic; and (4) Independent $Q$ Learning (IQL) [54], wherein each agent acts independent of other agents. (5) In order to compare our surprise minimization scheme against pre-existing mechanisms, we compare EMIX additionally to a model-free implementation of SMiRL [3] in QMIX. We use the generalized version of SMiRL as it demonstrates reduced variance across batches [9]. This implementation is denoted as SMiRL-QMIX for comparisons. Details related to the implementation of EMIX are presented in Appendix D.

Table 5 presents the comparison of success rate percentages between EMIX and prior MARL algorithms on 6 StarCraft II micromanagement scenarios. Corresponding to each scenario, algorithms demonstrating higher success rate values in comparison to other methods have their entries highlighted in **bold** (see Appendix E.1 for a statistical analysis). Out of the 6 scenarios considered, EMIX presents higher success rates on 5 of these scenarios depicting the suitability of the proposed approach. In cases of *so_many_baneling* and *5m_vs _6m* having large number of opponents and a greater level of surprise, EMIX aptly improves over prior methods. When compared to QMIX, EMIX depicts improved success rates on all of the 6 scenarios. On comparing EMIX with SMiRL-QMIX, EMIX demonstrates higher average success rates indicating surprise robust policies.

### 5.2 Ablation Study

We now present the ablation study for the various components of EMIX. Our experiments aim to determine the effectiveness of the energy-based surprise minimization method. Additionally, we also aim to evaluate the utility of dual approximators for surprise estimation in accordance with the precept from RL literature [21, 13, 20].

**EMIX Objective:** To weigh the effectiveness of energy-based scheme, we ablate the energy operator $\mathcal{T}$ and only utilize $V^a_{\text{surp}}$. Since this implementation employs dual approximators $V^a_{\text{surp},(i)}$ $i \in \{1, 2\}$ for stability, we call this implementation as TwinQMIX. Thus, we compare between QMIX, TwinQMIX and EMIX to assess the contributions of each of the proposed methods.

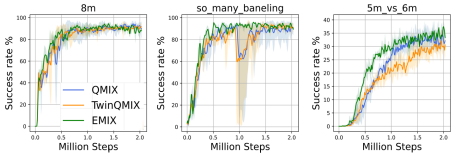


Figure 3: Ablations for each of EMIX's component. When compared to QMIX, EMIX and TwinQMIX depict improvements in performance and sample efficiency.

Figure 3 presents the comparison of average success rates for QMIX, TwinQMIX and EMIX on 3 different scenarios. In comparison to QMIX, TwinQMIX adds stability to the original objective by incorporating surprising estimates. On comparing TwinQMIX to EMIX we note that dual approximators play little role in improving convergence. Thus, the energy-based surprise minimization scheme is the main facet for significant performance improvement. This is demonstrated in the *5m_vs_6m* scenario wherein the EMIX implementation improves the performance of TwinQMIX in comparison to QMIX by utilizing a surprise-robust policy. In the case of *so_many _baneling* scenario which consists of a large number of opponents (27 banelings), EMIX tackles surprise effectively by preventing a significant drop in performance which is observed in cases of QMIX and TwinQMIX.

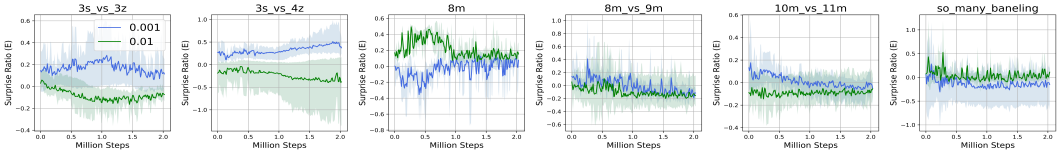


Figure 4: Variation of surprise minimization with temperature $\beta$. Learning of surprise is achieved by making use of a suitable value of temperature parameter ($\beta = 0.01$) which controls the stability in surprise minimization by utilizing $E$ as intrinsic motivation.

**Surprise Minimization with Temperature:** The importance of $\beta$ can be validated by assessing its usage in surprise minimization. We observe the variation of $E$ as it is a collection of surprise-based sample estimates across the batch. Additionally, $E$ consists of prior samples $V^a_{\text{surp}}(s, u, \sigma)$ for $V^a_{\text{surp}}(s', u', \sigma')$ which makes inference tractable.

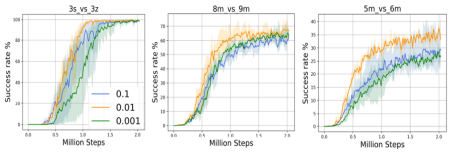


Figure 5: Variation in success rates with temperature $\beta$. A value of $\beta = 0.01$ is found to work best.

Figure 4 presents the variation of Energy ratio $E$ with the temperature parameter $\beta$ during learning. We compare two stable variations of $E$ at $\beta = 0.001$ and $\beta = 0.01$. The objective minimizes $E$ over the course of learning and attains thermal equilibrium with minimum energy. Intuitively, equilibrium corresponds to convergence to optimal policy $\pi^*$ which validates the claim in Theorem 2. With $\beta = 0.01$, EMIX presents improved convergence and surprise minimization for 5 out of the 6 considered scenarios, hence validating the suitable choice of $\beta$. The choice of $\beta$ is further validated in Figure 5 wherein $\beta = 0.01$ provides consistent stable improvements over other values. Lower values of $\beta$, such as $\beta = 0.001$, do little to minimize surprise or improve performance.

## 6 Qualitative Analysis

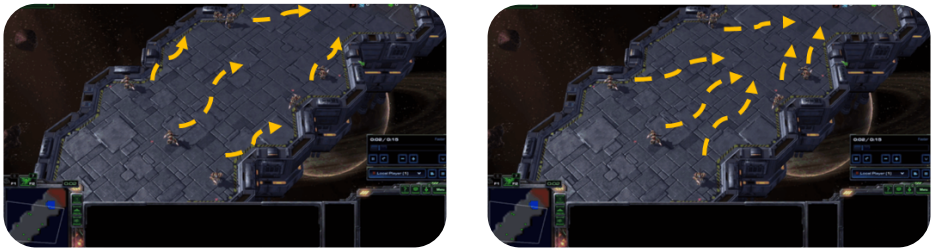

Figure 6: Task- *so_many_baneling*, **(left)** Behaviors learned by EMIX agents, **(right)** Behaviors learned by QMIX agents

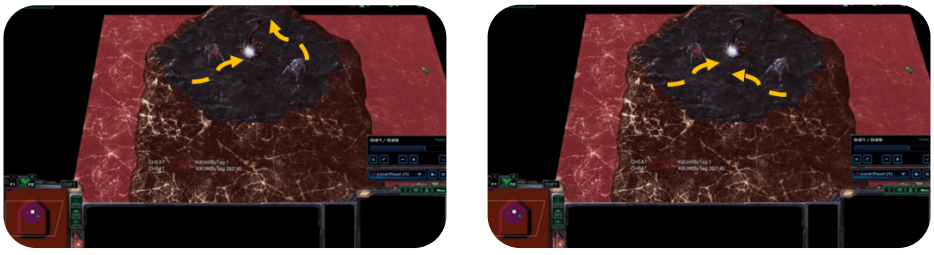

Figure 7: Task- *2s_vs_1sc*, **(left)** Behaviors learned by EMIX agents, **(right)** Behaviors learned by QMIX agents

We visualize and compare behaviors learned by surprise minimizing agents to the prior method of QMIX. Fig. 6 presents the comparison of EMIX and QMIX agent trajectories (in yellow arrows) on

the challenging *so_many_baneling* task. The task consists of 27 baneling opponents which rapidly attack the agent team on a bridge. QMIX agents naively move to the central alley of the bridge and start attacking enemies early on. While QMIX agents naively maximize returns, EMIX agents learn a different strategy. EMIX agents rearrange themselves first at the corners of the bridge. Note that these corners provide cover from enemy's fire. Thus, EMIX agents learn to take cover before approaching the enemy head-on. This indicates that the surprise-robust policy is aware of the incoming fast-paced assault.

As another example, Fig. 7 presents behaviors on the *2s_vs_1sc* task wherein two agents must collaborate together to defeat a SpineCrawler enemy. The enemy, having a long tentacle pointing to the front, chooses to attack any one of the agents **randomly** in front of it. Additionally, the tentacle has a fixed length and cannot extend beyond this range. Random intermittent attacks indicate that the agents face a greater degree of surprise with no prior knowledge of the enemy's movement. We observe that QMIX agents take turns to attack the enemy by moving back and forth to minimize damage. EMIX agents, on the other hand, learn a different strategy. One of the EMIX agents stands at a distance to attack th enemy while the other agent goes around to attack from behind. This indicates that the policy is aware of enemy's limited movement.

### 6.1 Predator-Prey Benchmark

We extend our comparison of EMIX on the Predator-Prey (particle world) tasks. In addition to the difficulty of task, we vary the number of opponents. This helps quantify the variation in performance against increasing level of surprise under fixed dynamics. Table 2 presents average returns. While all agents present comparable performance on the easier tasks, EMIX improves over QMIX and TwinQMIX on the more challenging *punish* and *hard* tasks. In the case of *punish*, EMIX is the only method to achieve greater than 20 returns. Additional results can be found in Appendix E.3.

| Scenarios | EMIX | TwinQMIX | SMiRL-QMIX | QMIX | VDN | COMA | IQL |
|---|---|---|---|---|---|---|---|
| predator_prey_easy | **40.00 ± 0.13** | 40.00 ± 0.34 | 40.00 ± 0.98 | 40.00 ± 0.22 | 38.74 ± 0.64 | 27.49 ± 4.26 | 34.73 ± 2.92 |
| predator_prey | 40.00 ± 0.72 | 40.00 ± 1.92 | 40.00 ± 0.27 | **40.00 ± 0.16** | 36.23 ± 3.19 | 25.13 ± 0.92 | 31.59 ± 0.74 |
| predator_prey_punish | **24.17 ± 3.29** | 20.32 ± 4.15 | 19.31 ± 1.12 | 14.33 ± 3.81 | 17.21 ± 2.31 | 10.92 ± 4.35 | 7.86 ± 3.21 |
| predator_prey_hard | **12.34 ± 3.11** | 10.19 ± 1.15 | 10.47 ± 0.83 | 8.76 ± 4.33 | 5.19 ± 3.97 | -4.37 ± 1.53 | -9.26 ± 4.84 |

Table 2: Comparison of average returns between EMIX, its ablations and prior MARL methods on Predator-Prey tasks. EMIX improves over QMIX and SMiRL-QMIX.

## 7 Discussion

**Conclusion:** In this paper, we presented an energy-based perspective towards surprise minimization in multi-agent RL. Towards this goal we introduce EMIX, an energy-based intrinsic motivation framework for surprise minimization in MARL algorithms. EMIX utilizes a temporal EBM to estimate and minimize surprise jointly across all agents. Our theoretical claims on the formulation of minimization of temporal energy with surprise are corroborated upon utilizing EMIX on a suite of challenging MARL tasks requiring significant collaboration under fast-paced dynamics.

**Future Work:** While EMIX serves as a practical example of EBMs in cooperative MARL, it presents several new avenues for future work. We shed light on 2 such aspects,

(1) Provision of an energy-based model naturally raises the question of *how can we efficiently sample from the surprise distribution?* Advances in sampling methods depict promise towards this aspect.

(2) Although suitable for lower dimensions, the scalability of EBMs towards high dimensional action spaces remains an open question. We conjecture that the utility of density-based methods and generative models can address the scalability gap. These directions are left for future work.

## References


[1] J. Achiam and S. Sastry. Surprise-based intrinsic motivation for deep reinforcement learning, 2017.

[2] K. Asadi and M. L. Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, 2017.

[3] G. Berseth, D. Geng, C. Devin, D. Jayaraman, C. Finn, and S. Levine. Smirl: Surprise minimizing rl in entropic environments. 2019.

[4] D. P. Bertsekas. *Abstract dynamic programming*. Athena Scientific Nashua, NH, USA, 2018.

[5] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-dynamic Programming*, volume 1. Athena Scientific, 1995.

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[7] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros. Large-scale study of curiosity-driven learning. In *ICLR*, 2019.

[8] L. Buşoniu, R. Babuška, and B. De Schutter. Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications-1*. 2010.

[9] J. Z. Chen. Reinforcement learning generalization with surprise minimization, 2020.

[10] L. Chenghao, T. Wang, C. Wu, Q. Zhao, J. Yang, and C. Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[11] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. Counterfactual multi-agent policy gradients, 2017.

[12] K. Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127–138, 2010.

[13] S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods, 2018.

[14] B. Gao and L. Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

[15] J. Grau-Moya, F. Leibfried, and H. Bou-Ammar. Balancing two-player stochastic games with soft q-learning. *arXiv preprint arXiv:1802.03216*, 2018.

[16] A. Gupta, R. Mendonca, Y. Liu, P. Abbeel, and S. Levine. Meta-reinforcement learning of structured exploration strategies. In *Advances in Neural Information Processing Systems 31*. 2018.

[17] T. Haarnoja. *Acquiring Diverse Robot Skills via Maximum Entropy Deep Reinforcement Learning*. PhD thesis, UC Berkeley, 2018.

[18] T. Haarnoja, K. Hartikainen, P. Abbeel, and S. Levine. Latent space policies for hierarchical reinforcement learning. *arXiv preprint arXiv:1804.02808*, 2018.

[19] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017.

[20] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[21] H. v. Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[22] 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. *arXiv preprint arXiv:1710.02298*, 2017.

[23] M. Janner, J. Fu, M. Zhang, and S. Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, 2019.

[24] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, A. Mohiuddin, R. Sepassi, G. Tucker, and H. Michalewski. Model-based reinforcement learning for atari, 2019.

[25] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations*, 2014.

[26] L. Kraemer and B. Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190, 02 2016.

[27] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems*, 2016.

[28] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. *NeurIPS 2020*, 2020.

[29] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. *Predicting structured data*, 1, 2006.

[30] Y. LeCun, S. Chopra, M. Ranzato, and F.-J. Huang. Energy-based models in document recognition and computer vision. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 1, 2007.

[31] L. Lee, B. Eysenbach, E. Parisotto, E. Xing, S. Levine, and R. Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

[32] S. Levine and P. Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems*, 2014.

[33] M. A. Lones. How to avoid machine learning pitfalls: a guide for academic researchers. *arXiv preprint arXiv:2108.02497*, 2021.

[34] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments, 2017.

[35] L. Macedo and A. Cardoso. The role of surprise, curiosity and hunger on exploration of unknown environments populated with entities. In *2005 portuguese conference on artificial intelligence*, 2005.

[36] L. Macedo, R. Reisezein, and A. Cardoso. Modeling forms of surprise in artificial agents: empirical and theoretical study of surprise functions. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 26, 2004.

[37] D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

[38] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18, 1947.

[39] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 2016.

[40] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1), 1950.

[41] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[42] B. O'Donoghue, R. Munos, K. Kavukcuoglu, and V. Mnih. Combining policy gradient and q-learning. *arXiv preprint arXiv:1611.01626*, 2016.

[43] F. A. Oliehoek and C. Amato. *A concise introduction to decentralized POMDPs*. Springer, 2016.

[44] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, and S. Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *ICML 2018: Proceedings of the Thirty-Fifth International Conference on Machine Learning*, 2018.

[45] W. Ren, R. W. Beard, and E. M. Atkins. A survey of consensus problems in multi-agent coordination. In *Proceedings of the 2005, American Control Conference, 2005.*, 2005.

[46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Representations by Back-propagating Errors. *Nature*, 323:533–536, 1986.

[47] B. Sallans and G. E. Hinton. Reinforcement learning with factored states and actions. *Journal of Machine Learning Research*, 5, 2004.

[48] M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C.-M. Hung, P. H. S. Torr, J. Foerster, and S. Whiteson. The starcraft multi-agent challenge, 2019.

[49] J. Schulman, X. Chen, and P. Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017.

[50] P. Schwartenbeck, T. FitzGerald, R. Dolan, and K. Friston. Exploration, novelty, surprise, and free energy minimization. *Frontiers in psychology*, 4:710, 2013.

[51] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, 2019.

[52] B. C. Stadie, S. Levine, and P. Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

[53] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 2085–2087, 2018.

[54] M. Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *In Proceedings of the Tenth International Conference on Machine Learning*, 1993.

[55] Y. W. Teh, M. Welling, S. Osindero, and G. E. Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4, 2003.

[56] S. B. Thrun. Efficient exploration in reinforcement learning. 1992.

[57] M. Toussaint. Robot trajectory optimization using approximate inference. In *Proceedings of the 26th annual international conference on machine learning*, 2009.

[58] O. Vinyals, I. Babuschkin, W. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. Agapiou, M. Jaderberg, and D. Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575, 11 2019.

[59] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang. Qplex: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2021.

[60] T. Wang, T. Gupta, A. Mahajan, B. Peng, S. Whiteson, and C. Zhang. {RODE}: Learning roles to decompose multi-agent tasks. In *International Conference on Learning Representations*, 2021.

[61] Y. Wang, B. Han, T. Wang, H. Dong, and C. Zhang. Dop: Off-policy multi-agent decomposed policy gradients. In *International Conference on Learning Representations*, 2021.

[62] E. Wei, D. Wicke, D. Freelan, and S. Luke. Multiagent soft q-learning. *arXiv preprint arXiv:1804.09817*, 2018.

[63] Y. Wen, Y. Yang, R. Luo, J. Wang, and W. Pan. Probabilistic recursive reasoning for multi-agent reinforcement learning. *arXiv preprint arXiv:1901.09207*, 2019.

[64] B. D. Ziebart. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. 2010.

[65] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008.

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

# A Proofs

**Theorem 1.** *Given a surprise value function $V^a_{\text{surp}}(s,u,\sigma)\,\forall a \in N$, the energy operator $\mathcal{T}V^a_{\text{surp}}(s,u,\sigma) = \log\sum_{a=1}^{N}\exp\left(V^a_{\text{surp}}(s,u,\sigma)\right)$ forms a contraction on $V^a_{\text{surp}}(s,u,\sigma)$.*

*Proof.* We follow the process of [2]. Let us first define a norm on surprise values $||V_1 - V_2|| \equiv \max_{s,u,\sigma}|V_1(s,u,\sigma) - V_2(s,u,\sigma)|$. Suppose $\epsilon = ||V_1 - V_2||$,

$$\begin{aligned}
\log\sum_{a=1}^{N}\exp\left(V_1(s,u,\sigma)\right) &\leq \log\sum_{a=1}^{N}\exp\left(V_2(s,u,\sigma)+\epsilon\right)\\
= \log\sum_{a=1}^{N}\exp\left(V_1(s,u,\sigma)\right) &\leq \log\exp\left(\epsilon\right)\sum_{a=1}^{N}\exp\left(V_2(s,u,\sigma)\right)\\
= \log\sum_{a=1}^{N}\exp\left(V_1(s,u,\sigma)\right) &\leq \epsilon + \log\sum_{a=1}^{N}\exp\left(V_2(s,u,\sigma)\right)\\
= \log\sum_{a=1}^{N}\exp\left(V_1(s,u,\sigma)\right) - \log\sum_{a=1}^{N}\exp\left(V_2(s,u,\sigma)\right) &\leq ||V_1 - V_2|| \qquad (9)
\end{aligned}$$

Similarly, using $\epsilon$ with $\log\sum_{a=1}^{N}\exp\left(V_1(s,u,\sigma)\right)$,

$$\begin{aligned}
\log\sum_{a=1}^{N}\exp\left(V_1(s,u,\sigma)+\epsilon\right) &\geq \log\sum_{a=1}^{N}\exp\left(V_2(s,u,\sigma)\right)\\
= \log\exp\left(\epsilon\right)\sum_{a=1}^{N}\exp\left(V_1(s,u,\sigma)\right) &\geq \log\sum_{a=1}^{N}\exp\left(V_2(s,u,\sigma)\right)\\
= \epsilon + \log\sum_{a=1}^{N}\exp\left(V_1(s,u,\sigma)\right) &\geq \log\sum_{a=1}^{N}\exp\left(V_2(s,u,\sigma)\right)\\
= ||V_1 - V_2|| \geq \log\sum_{a=1}^{N}\exp\left(V_2(s,u,\sigma)\right) - \log\sum_{a=1}^{N}\exp\left(V_1(s,u,\sigma)\right) &\qquad (10)
\end{aligned}$$

Results in Equation 9 and Equation 10 prove that the energy operation is a contraction. □

**Theorem 2.** *Upon agent's convergence to an optimal policy $\pi^*$, total energy of $\pi^*$, expressed by $E^*$ will reach a thermal equilibrium consisting of minimum surprise among consecutive states $s$ and $s'$.*

*Proof.* We begin by initializing a set of $M$ policies $\{\pi_1, \pi_2..., \pi_M\}$ having energy ratios $\{E_1, E_2..., E_M\}$. Consider a policy $\pi_1$ with surprise value function $V_1$. $E_1$ can then be expressed as

$$E_1 = \log\left[\frac{\sum_{a=1}^{N}\exp\left(V_1^a(s',u',\sigma')\right)}{\sum_{a=1}^{N}\exp\left(V_1^a(s,u,\sigma)\right)}\right]$$

Invoking Assumption 2 for $s$ and $s'$, we can express $V_1^a(s',u',\sigma') = V_1^a(s,u,\sigma) + \zeta_1$ where $\zeta_1$ is a constant. Using this expression in $E_1$ we get,

$$\begin{aligned}
E_1 &= \log\left[\frac{\sum_{a=1}^{N}\exp\left(V_1^a(s,u,\sigma)+\zeta_1\right)}{\sum_{a=1}^{N}\exp\left(V_1^a(s,u,\sigma)\right)}\right]\\
E_1 &= \log\left[\frac{\exp\left(\zeta_1\right)\sum_{a=1}^{N}\exp\left(V_1^a(s,u,\sigma)\right)}{\sum_{a=1}^{N}\exp\left(V_1^a(s,u,\sigma)\right)}\right]\\
E_1 &= \zeta_1
\end{aligned}$$

Similarly, $E_2 = \zeta_2$,$E_3 = \zeta_3$...,$E_M = \zeta_M$. Thus, the energy residing in policy $\pi$ is proportional to the surprise between consecutive states $s$ and $s'$. Clearly, an optimal policy $\pi^*$ is the one with minimum surprise. Mathematically,

$$\pi^* \geq \pi_1, \pi_2..., \pi_M \implies \zeta^* \leq \zeta_1, \zeta_2..., \zeta_M$$
$$= \pi^* \geq \pi_1, \pi_2..., \pi_M \implies E^* \leq E_1, E_2..., E_M$$

Thus, proving that the optimal policy consists of minimum surprise at thermal equilibrium. □

# B Relation to Maximum Entropy Framework

## B.1 Similarities & Differences

We conceptually compare EMIX to the maximum entropy framework.

**Similarities:** Both methods utilize an auxilary objective as intrinsic motivation to tackle uncertainty. While the maximum entropy formulation assigns low energy to uncertain actions, our method assigns low energy to uncertain encoded representations od states (as presented in Fig. 2).

**Differences:** Our method differs from maximum entropy in its optimization process and learning scheme. The maximum entropy formulation aims to maximize entropy in the value function space so as to motivate exploration. Our proposed scheme, on the other hand, aims to minimize surprise in the low-dimensional representation space to obtain dynamics-aware robust policies.

## B.2 Connection to Soft Q-Learning

The Soft Q-Learning objective with $V^{\theta^-}_{\text{soft}}(s')$ and $Q_{\text{soft}}(u, s; \theta)$ as state and action value functions respectively is given by-

$$J_Q(\theta) = \mathbb{E}_{s,u\sim R}\left[\frac{1}{2}\left(r + \gamma \mathbb{E}_{s'\sim R}[V^{\theta^-}_{\text{soft}}(s')] - Q_{\text{soft}}(u, s; \theta)\right)^2\right]$$

$$= J_Q(\theta) = \mathbb{E}_{s,u\sim R}\left[\frac{1}{2}\left(r + \gamma \mathbb{E}_{s'\sim R}\left[\log \sum_{u\in A} \exp Q_{\text{soft}}(u', s'; \theta^-)\right] - Q_{\text{soft}}(u, s; \theta)\right)^2\right]$$

The gradient of this objective can be expressed as-

$$\nabla_\theta J_Q(\theta) = \mathbb{E}_{s,u\sim R}\left[\left(r + \gamma \mathbb{E}_{s'\sim R}\left[\log \sum_{u\in A} \exp Q(u', s'; \theta^-)\right] - Q_{\text{soft}}(u, s; \theta)\right)\right] \nabla_\theta Q_{\text{soft}}(u, s; \theta) \tag{11}$$

And the gradient of the EMIX objective is obtained as-

$$L(\theta) = \mathbb{E}_{s,u,s'\sim R}\left[\frac{1}{2}\left(r + \gamma \max_{u'} Q(u', s'; \theta^-) + \beta \log\left(\frac{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s', u', \sigma')\right)}{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right)}\right) - Q(u, s; \theta)\right)^2\right]$$

$$\nabla_\theta L(\theta) = \mathbb{E}_{s,u,s'\sim R}\Bigg[\Bigg(r + \gamma \max_{u'} Q(u', s'; \theta^-) + \beta \log\left(\frac{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s', u', \sigma')\right)}{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right)}\right) - Q(u, s; \theta)\Bigg)\Bigg] \nabla_\theta Q(u, s; \theta) \tag{12}$$

Comparing Equation 11 to Equation 12 we notice that Soft Q-Learning and EMIX are related to each other as they utilize EBMs. Soft Q-Learning makes use of a discounted energy function which downweights the energy values over longer horizons. Actions consisting of lower energy configurations are given preference by making use of $Q_{\text{soft}}(u, s; \theta)$ as the negative energy. On the other hand, EMIX makes use of a constant energy function weighed by $\beta$ which minimizes surprise-based energy between consecutive states. Both the objectives can be thought of as energy minimizing models which search for an optimal energy configuration. Soft Q-Learning searches for an optimal configuration in the action space whereas EMIX favours optimal behavior on spurious states. In fact, EMIX can be realized as a special case of Soft Q-Learning if the mixer agent utilizes an energy-based policy and attains thermal equilibrium. This leads us to express Theorem 3.

**Theorem 3.** *Given an energy-based policy $\pi$ with its target function $V(s') = \log \sum_{u \in A} \exp Q(u', s'; \theta^-)$, the surprise minimization objective $L(\theta)$ reduces to the Soft Q-Learning objective $L(\theta_{\text{soft}})$ in the special case surprise absent between consecutive states, $\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s', u', \sigma')\right) = \sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right)$.*

*Proof.* We know that the EMIX objective is given by-

$$L(\theta) = \mathbb{E}_{s,u,s' \sim R}\left[\frac{1}{2}\left(r + \gamma \max_{u'} Q(u'; s', \theta^-) + \beta \log\left(\frac{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s', u', \sigma')\right)}{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right)}\right) - Q(u, s; \theta)\right)^2\right] \tag{13}$$

Replacing the greedy policy term $\max_{u'} Q(u', s'; \theta^-)$ with the energy-based value function $V(s') = \log \sum_{u' \in A} \exp Q(u', s'; \theta^-)$, we get,

$$L(\theta) = \mathbb{E}_{s,u,s' \sim R}\left[\frac{1}{2}\left(r + \gamma \mathbb{E}_{s' \sim R}[V(s')] + \beta \log\left(\frac{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s', u', \sigma')\right)}{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right)}\right) - Q(u, s; \theta)\right)^2\right] \tag{14}$$

$$= L(\theta) = \mathbb{E}_{s,u,s' \sim R}\left[\frac{1}{2}\left(r + \gamma \mathbb{E}_{s' \sim R}\left[\log \sum_{u' \in A} \exp Q(u', s'; \theta^-)\right] + \beta \log\left(\frac{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s', u', \sigma')\right)}{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right)}\right) - Q(u, s; \theta)\right)^2\right]$$

At thermal equilibrium, $\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s, u, \sigma)\right) = \sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s', u', \sigma')\right)$,

$$= L(\theta) = \mathbb{E}_{s,u,s' \sim R}\left[\frac{1}{2}\left(r + \gamma \mathbb{E}_{s' \sim R}\left[\log \sum_{u' \in A} \exp Q(u', s'; \theta^-)\right] + \beta \log\left(\frac{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s', u', \sigma')\right)}{\sum_{a=1}^{N} \exp\left(V^a_{\text{surp}}(s', u', \sigma')\right)}\right) - Q(u, s; \theta)\right)^2\right]$$

$$= L(\theta) = \mathbb{E}_{s,u,s' \sim R}\left[\frac{1}{2}\left(r + \gamma \mathbb{E}_{s' \sim R}\left[\log \sum_{u' \in A} \exp Q(u', s'; \theta^-)\right] + \beta \log(1) - Q(u, s; \theta)\right)^2\right] \tag{15}$$

$$= L(\theta) = \mathbb{E}_{s,u,s' \sim R}\left[\frac{1}{2}\left(r + \gamma \mathbb{E}_{s' \sim R}\left[\log \sum_{u' \in A} \exp Q(u', s'; \theta^-)\right] - Q(u, s; \theta)\right)^2\right] \tag{16}$$

Equation 16 represents the Soft Q-Learning objective, hence proving the result. □

## C Convergence Analysis

We now analyze convergence of the surprise minimization scheme during policy optimization. Our notation denotes $\mathcal{B}V_{k-1} = r + \gamma V_{k-1}$ as the Bellman operator which obeys monotonicity and contraction.

Monotonicity: $V_1 \leq V_2 \implies \mathcal{B}V_1 \leq \mathcal{B}V_2$ ; Contraction: $\|\mathcal{B}V_1 - \mathcal{B}V_2\|_2 \leq \gamma \|V_1 - V_2\|_2$

We now denote $\hat{V}_k = r_k + \gamma \hat{V}_{k-1} + \beta \log \sum_{a=1}^{N} \exp(V^a_{\text{surp},(k)}(s, u, \sigma))$ as the total value at step $k$. As per the definition of $\mathcal{B}$, this gives us $\hat{V}_k = \mathcal{B}\hat{V}_{k-1} + \beta \log \sum_{a=1}^{N} \exp(V^a_{\text{surp},(k)}(s, u, \sigma))$.

Consider $\left\|\hat{V}_k - V^*\right\|_2$ with $V^*$ being the optimal value at convergence,

$$\left\|\hat{V}_k - V^*\right\|_2 \leq \left\|\mathcal{B}\hat{V}_{k-1} + \beta \log \sum_{a=1}^{N} \exp(V^a_{\text{surp},(k)}) - V^*\right\|_2 \tag{17}$$

Where $\beta \log \sum_{a=1}^{N} \exp(V^a_{\text{surp},(k)}) > 0$ as per constant positive surprise $\zeta > 0$ in 2. We impose this constraint by adding ReLU nonlinearities in surprise encoder to obtain positive $V^a_{\text{surp},(k)}$ values.

$$\leq \left\|\mathcal{B}^2\hat{V}_{k-2} + \beta \log \sum_{a=1}^{N} \exp(V^a_{\text{surp},(k-1)}) + \beta \log \sum_{a=1}^{N} \exp(V^a_{\text{surp},(k)}) - V^*\right\|_2 \tag{18}$$

$$\leq \left\|\mathcal{B}^2\hat{V}_{k-2} + \beta \left( \log \sum_{a=1}^{N} \exp(V^a_{\text{surp},(k-1)}) + \log \sum_{a=1}^{N} \exp(V^a_{\text{surp},(k)}) \right) - V^*\right\|_2 \tag{19}$$

$$\leq \left\|\mathcal{B}^2\hat{V}_{k-2} + \beta \left( \log \left[ \sum_{a=1}^{N} \exp(V^a_{\text{surp},(k-1)}) \right] \left[ \sum_{a=1}^{N} \exp(V^a_{\text{surp},(k)}) \right] \right) - V^*\right\|_2 \tag{20}$$

Thus, for $k$ iterations, we have,

$$\leq \left\|\mathcal{B}^k V_0 + \beta \left( \log \prod_{i=1}^{k} \left[ \sum_{a=1}^{N} \exp(V^a_{\text{surp},(i)}) \right] \right) - V^*\right\|_2 \tag{21}$$

$$= \left\|\mathcal{B}^k V_0 + \beta \left( \log \sum_{a=1}^{N} \left[ \prod_{i=1}^{k} \exp(V^a_{\text{surp},(i)}) \right] \right) - V^*\right\|_2 \tag{22}$$

$$= \left\|\mathcal{B}^k V_0 + \beta \left( \log \sum_{a=1}^{N} \left[ \exp(\sum_{i=1}^{k} V^a_{\text{surp},(i)}) \right] \right) - V^*\right\|_2 \tag{23}$$

We now absorb the sum of surprise values from time index $i = 1, .., k$ in a single variable $V^a_{\text{tot}}$. Thus, using $V^a_{\text{tot}} = \sum_{i=1}^{k} V^a_{\text{surp},(i)}$ and utilizing the Triangle Inequality, we get,

$$= \left\|\mathcal{B}^k V_0 - V^*\right\|_2 + \left\|\beta \left( \log \sum_{a=1}^{N} [\exp(V^a_{\text{tot}})] \right)\right\|_2 \tag{24}$$

We now bound the two terms separately. Considering the first term and following the results of value iteration convergence [5],

$$\left\|\mathcal{B}^k V - V^*\right\|_2 \leq \gamma^k \left\|V - V^*\right\|_2 \tag{25}$$

$$\left\|\mathcal{B}^k V_0 - V^*\right\|_2 \leq \gamma^k \left\|V + V_\mu - V_\mu - V^*\right\|_2 \tag{26}$$

wherein $V_\mu$ denotes an approximation to $V$. Utilizing the triangle inequality yields,

$$\left\|\mathcal{B}^k V_0 - V^*\right\|_2 \leq \gamma^k \left\|V - V_\mu\right\|_2 + \gamma^k \left\|V_\mu - V^*\right\|_2 \tag{27}$$

The two terms are bounded using the convergence result of [4].

$$\left\|\mathcal{B}^k V_0 - V^*\right\|_2 \leq \gamma^k \sqrt{r_{\max}} + \gamma^k \sqrt{\frac{r_{\max}|\mathcal{S}|}{1-\gamma}} \tag{28}$$

Now, considering the second term in Equation 24 and denoting $V^*_{\text{tot}} = \sum_{i=1}^{k} V^*_{\text{surp},(i)}$ as the sum of optimal surprise values,

$$\beta \left\| \log \sum_{a=1}^{N} \exp(V^a_{\text{tot}}) \right\|_2 = \beta \left\| \log \sum_{a=1}^{N} \exp(V^a_{\text{tot}}) - \log \sum_{a=1}^{N} \exp(V^*_{\text{tot}}) + \log \sum_{a=1}^{N} \exp(V^*_{\text{tot}}) \right\|_2 \tag{29}$$

using the triangle inequality,

$$\leq \beta \left\| \log \sum_{a=1}^{N} \exp(V_{\text{tot}}^{a}) - \log \sum_{a=1}^{N} \exp(V_{\text{tot}}^{*}) \right\|_2 + \beta \left\| \log \sum_{a=1}^{N} \exp(V_{\text{tot}}^{*}) \right\|_2 \tag{30}$$

Since $\mathcal{T} = \log \sum_{a=1}^{N} \exp(V_{\text{tot}}^{a})$ is a contraction following Theorem 1, for the first term we have,

$$\leq \beta\gamma \left\| V_{\text{tot}}^{a} - V_{\text{tot}}^{*} \right\|_2 + \beta \left\| \log \sum_{a=1}^{N} \exp(V_{\text{tot}}^{*}) \right\|_2 \tag{31}$$

The second term in the above relation is bounded due to the completeness assumption, $\left\| \log \sum_{a=1}^{N} \exp(V_{\text{tot}}^{*}) \right\|_2$. The first term, on the other hand, is simplified by applying Jensen's Inequality on the definitions of $V_{\text{tot}}^{a}$ and $V_{\text{tot}}^{*}$, $\left\| \sum_{i=1}^{k} V_{\text{surp},(i)}^{a} - \sum_{i=1}^{k} V_{\text{surp},(i)}^{*} \right\|_2 = \left\| \sum_{i=1}^{k} \left( V_{\text{surp},(i)}^{a} - V_{\text{surp},(i)}^{*} \right) \right\|_2 \leq \sum_{i=1}^{k} \left\| V_{\text{surp},(i)}^{a} - V_{\text{surp},(i)}^{*} \right\|_2$. Denoting $\left\| V_{\text{surp},(i)}^{a} - V_{\text{surp},(i)}^{*} \right\|_2 = \text{RMSE}(V_{\text{surp},(i)}^{a})$, we obtain the following result,

$$\leq \beta\gamma \sum_{i=1}^{k} \text{RMSE}(V_{\text{surp},(i)}^{a}) + \beta\zeta \ , \ \zeta > 0 \tag{32}$$

Finally, combining Equation 28 and Equation 32 in Equation 24, we obtain the desired convergence bound.

$$\left\| V_k - V^* \right\|_2 \leq \gamma^k \left( \sqrt{r_{\max}} + \sqrt{\frac{r_{\max}|\mathcal{S}|}{1-\gamma}} \right) + \beta \left( \gamma \sum_{i=1}^{k} \text{RMSE}(V_{\text{surp},(i)}^{a}) + \zeta \right) \tag{33}$$

While the first term in Equation 33 denotes the convergence of policy optimization, the second term indicates the bounded convergence of surprise to ecological niches with finite (yet nonzero) surprising elements. The policy optimization process converges at a geometric rate $\mathcal{O}(\gamma^k)$ towards its stable fixed points. The surprise minimization process, on the other hand, demonstrates an annealing behavior which depends on the temperature parameter $\beta$. Furthermore, convergence to stable fixed point $V_{\text{tot}}^{a}$ is bounded in respect to each agents individual surprise values $V_{\text{tot}}^{a}$. This insight indicates that different agents converge towards different locally optimal values of surprise. Finally, the presence of constant $\zeta$ corroborates prior claims [50, 12] that agents continue to experience surprise irrespective of their convergence to minimum energy niches. To further develop intuition for this claim, consider the special case wherein $\sum_{i=1}^{k} \text{RMSE}(V_{\text{surp},(i)}^{a}) \to 0$, i.e.- surprise estimation error for all iterations goes to zero. Irrespective of global convergence among all agents, a finite yet small $\zeta$ continues to contribute to the upper bound of $\left\| V_k - V^* \right\|_2$.

**Role of $\beta$:** We further discuss the role of $\beta$ which is of balancing the terms at successive iterations. While the first term geometrically decays with $\mathcal{O}(\gamma^k)$ rate, the second term approaches a finite constant $\beta\zeta$ as $V_{tot}^{a} \to V_{tot}^{*}$. Irrespective of our choice of $\beta$, the LHS $\|V_k - V^*\|_2$ is upper bounded by a constant which validates the claims of minimum yet finite surprise values. We do note that a small $\beta$ is still desirable to remove any approximation errors in order to push $V_k \to V^*$. However, this comes at the cost of increased surprise if $\beta$ is not selected appropriately.

# D Implementation Details

## D.1 Model Specifications

**Architecture:** This section highlights model architecture for the surprise value function. At the lower level, the architecture consists of 3 independent networks called *state_net*, *q_net* and *surp_net*.

Each of these networks consist of a single layer of 256 units with ReLU non-linearity as activations. Similar to the mixer-network, we use the ReLU non-linearity in order to provide monotonicity constraints across agents. Using a modular architecture in combination with independent networks leads to a richer extraction of joint latent transition space. Outputs from each of the networks are concatenated and are provided as input to the *main_net* consisting of 256 units with ReLU activations. The *main_net* yields a single output as the surprise value $V^a_{\text{surp}}(s, u, \sigma)$ which is reduced along the agent dimension by the energy operator. Alternatively, deeper versions of networks can be used in order to make the extracted embeddings increasingly expressive. However, increasing the number of layers does little in comparison to additional computational expense.

**Computation of $\sigma$:** The deviation $\sigma$ corresponds to the standard deviation across each dimension of the state $s$. Considering the state as a tensor of size $B \times A \times M$ with $B$ as the batch size, $A$ as the number of agents and $M$ as the observation dimension, we compute $\sigma$ by calculating the standard deviation across the $M$ dimension. This yields $\sigma$ as a $B \times A \times 1$ dimensional array.

**Computation of surprise estimates:** $V_{\text{surp}}$ denotes the surprise value function which quantifies the amount of surprise experienced by agents. Analogous to a $Q$ value function which provides estimates of returns, $V_{\text{surp}}$ provides estimate of surprise. Our framework learns $V_{surp}$ much like any other value function (using a neural network), but by additionally undergoing a $\log \sum \exp$ transformation to obey the fixed point property. This is achieved by realizing log-sum-exp as an energy operator $\mathcal{T} = \log \sum \exp$ which can be computed using standard computation libraries. Since our code is implemented in PyTorch, we implement this as `T_V = torch.logsumexp(V_surp, dim=1)`.

**Global State Encoder:** The global state encoder serves as a mapping from the state space to a low dimensional representation space $\mathcal{S} \rightarrow \mathcal{Z}$. The encoder takes in a sequence of states $\{s_1, s_2, ..., s_T\}$ as input and outputs a latent representation $z_{\text{state}}$. We use a standard pyramid MLP network consisting of 2 hidden layers of 256 units each with ReLU non-linearity. Embeddings obtained from the encoder are concatenated with other latent embeddings before being passed to the final surprise encoder.

**Standard Deviation Encoder:** The standard deviation encoder serves as a mapping from standard deviations across state dimensions to a low dimensional representation space. Each standard deviation $\sigma$ is computed across dimensions of the state $s_t$. These deviations are then packed in a sequence $\{\sigma_1, \sigma_2, ..., \sigma_T\}$ and passed as inputs to the standard deviation encoder. Intuitively, the encoder learns changes across states in a batch of observations. This is similar to a dynamics model predicting future states, except that we map these states to a low dimensional embedding. We use a standard pyramid MLP network consisting of 2 hidden layers of 256 units each with ReLU non-linearity. Embeddings obtained from the encoder are concatenated with other latent representations and used by the final surprise encoder to estimate the surprise distribution.

### D.2 Hyperparameters

Table 3 presents hyperparameter values for EMIX. A total of 2 target $Q$-functions were used as the model is found to be robust to any greater values.

| Hyperparameters | Values |
|---|---|
| batch size | $b = 32$ |
| learning rate | $\alpha = 0.0005$ |
| discount factor | $\gamma = 0.99$ |
| target update interval | 200 episodes |
| gradient clipping | 10 |
| exploration schedule | 1.0 to 0.01 over 50000 steps |
| mixer embedding size | 32 |
| agent hidden size | 64 |
| temperature | $\beta = 0.01$ |
| target $Q$-functions | 2 |

Table 3: Hyperparameter values for EMIX agents

### D.3 Selection & Tuning of $\beta$

One can manually tune $\beta$ using a fine-grained hyperparameter search. We tune $\beta$ between 0.001 and 1 in intervals of 0.01 with best performance observed at $\beta = 0.01$. However, we find two additional methods helpful for obtaining more accurate values. These are described as follows-

**Armijo's Line Search:** One can borrow from optimization theory and utilize Armijo's line search [41] by setting a termination condition. The method starts with a constant value of $\beta$ which is iteratively incremented/decremented until a termination criterion (example- $\|\nabla L(\theta)\| < \epsilon$ with $\epsilon$ a constant) is reached. While line search is proven to converge towards globally optimal values, its $\mathcal{O}(n^2)$ convergence may be computationally expensive that too in the MARL setting. Thus, we turn to the more efficient automatic tuning.

**Algorithm 2** Armijo's Line Search

1: Initialize $\beta$, $\delta \in (0, 1]$, EMIX & $\mathcal{T}V^a_{\text{surp}}$;
2: **while** EMIX($Q+\beta*\mathcal{T}V^a_{\text{surp}}$) > EMIX($Q$) + $\alpha * \beta * \nabla$EMIX$(\mathcal{Q})^{\text{T}}\mathcal{T}V^a_{\text{surp}}$ **do**
3: $\quad\beta = \delta * \beta$
4: **end while**
5: return $\beta$

**Algorithm 3** Automatic Tuning

1: Initialize $\beta$, $\delta \in (0, 1]$, EMIX & $\mathcal{T}V^a_{\text{surp}}$;
2: EMIX($Q + \beta * \mathcal{T}V^a_{\text{surp}}$)
3: beta_loss = $\beta * 0.5 * (\mathcal{T}V^a_{\text{surp}} - 0)^2$
4: beta_loss.backward()
5: return $\beta$

**Automatic Tuning:** We choose to automatically tune $\beta$ following single-agent RL literature [20, 28]. This is achieved by treating $\beta$ as a parameter and adaptively optimizing over it using Adam. We treat a surprise value of 0 as our target value. The method works well in practice and provides $\beta$ values closer to 0.01 (our manual selection).

## E Additional Results

### E.1 Statistical Significance

| Scenarios | EMIX | SMiRL-QMIX | QMIX | VDN | COMA | IQL |
|---|---|---|---|---|---|---|
| 2s_vs_1sc | 14 | 7 | - | 21 | **25** | 4 |
| 2s3z | **15** | 9 | - | 6 | 0 | 0 |
| 3m | **17** | 0 | - | 0 | 2 | 12 |
| 3s_vs_3z | **11** | 3 | - | 0 | 0 | 1 |
| 3s_vs_4z | **21** | 0 | - | 2 | 0 | 0 |
| 3s_vs_5z | 5 | 0 | - | **25** | 0 | 0 |
| 3s5z | 7 | **13** | - | 0 | 0 | 0 |
| 8m | **15** | 1 | - | 1 | 3 | 0 |
| 8m_vs_9m | 7 | **11** | - | 0 | 0 | 0 |
| 10m_vs_11m | 14 | **25** | - | 6 | 0 | 0 |
| so_many_baneling | **24** | 14 | - | 9 | 4 | 0 |
| 5m_vs_6m | **21** | 15 | - | 18 | 0 | 0 |

Table 4: Comparison of the $\mathcal{U}$ statistic on StarCraft II benchmark. $\mathcal{U}$ here denotes the statistical significance of an algorithm against QMIX (higher is better).

We follow the recommendation of [33] and evaluate the statistical significance of our results by carrying out the Mann-Whitney U test [38]. All 5 seeds of an algorithm (on each task) are compared to that of QMIX to yield the $\mathcal{U}$ statistic. $\mathcal{U}$ here denotes the statistical significance of performance with higher values being desirable.

Table 4 presents the comparison of $\mathcal{U}$ statistic on the StartCraft II benchmark. EMIX demonstrates consistently high values of $\mathcal{U}$ across a diverse set of tasks when compared to SMiRL and prior MARL agents. This highlights the consistent surprise-minimizing performance of EMIX across random seeds.

### E.2 StarCraft II Benchmark

| Scenarios | EMIX | SMiRL-QMIX | QMIX | VDN | COMA | IQL |
|---|---|---|---|---|---|---|
| 2s_vs_1sc | 90.33 ± 0.72 | 88.41 ± 1.31 | 89.19 ± 3.23 | 91.42 ± 1.23 | **96.90 ± 0.54** | 86.07 ± 0.98 |
| 2s3z | **95.40±0.45** | 94.93±0.32 | **95.30±1.28** | 92.03±2.08 | 43.33±2.70 | 55.74±6.84 |
| 3s_vs_3z | **99.58±0.07** | 97.63±1.08 | **99.43±0.20** | 97.90±0.58 | 0.21±0.54 | 92.32±2.83 |
| 3s_vs_5z | 52.91±11.80 | 0.00±0.00 | 43.44±7.09 | **68.51±5.60** | 0.00±0.00 | 18.14±2.34 |
| 3s5z | **88.88±1.07** | 88.53±1.03 | **88.49±2.32** | 63.58±3.99 | 0.25±0.11 | 7.05±3.52 |
| 8m | **94.47±1.38** | 89.96±1.42 | **94.30±2.90** | 90.26±1.12 | 92.82±0.53 | 83.53±1.62 |

Table 5: Comparison of success rate percentages between EMIX and prior MARL methods on StarCraft II micromanagement scenarios. EMIX is comparable to or improves over QMIX agent. In comparison to SMiRL-QMIX, EMIX demonstrates improved minimization of surprise. Results are averaged over 5 random seeds.

### E.3 Predator-Prey Benchmark

We consider a simple toy task from the Predator-Prey benchmark to demonstrate the importance of surprise minimization. We select *predator_prey_easy* due to its simplicity and convenient dynamics. The task consists of 3 agents and 3 opponents. We increase the number of opponents while keeping the task fixed. This way the dynamics of the MDP remain unchanged and the only changing factor is opponent behaviors.

Fig. 8 presents the variation of average returns for EMIX and QMIX over 5 random seeds. While QMIX agents undergo a steady decrease in performance, EMIX agents are found robust to this fast degradation. Even after the addition of 20 opponents (against only 3 agents), EMIX is able to retain positive returns. The algorithm acquires a surprise robust-policy early on during training to tackle fast-paced changes introduced by the large number of agents.

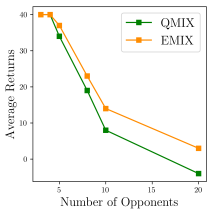


Figure 8: Variation in performance with increasing number of agents.

### E.4 Note on Minimum Entropy Conjugate Objective

The minimum conjugate entropy objective denotes the dual problem to surprise minimization. If we compute the Legendre Transform of our energy-based operator $\mathcal{T}V^a_{\text{surp}}(s,u,\sigma) = \log \sum_{a=1}^{N} \exp(V^a_{\text{surp}}(s,u,\sigma))$ we obtain the entropy function $\mathcal{H}(x)$ where x is the gradient of the operator, $x = \mathcal{T}V^a_{\text{surp}}(s,u,\sigma)$. This insight indicates that minimizing the energy operator $\mathcal{T}V^a_{\text{surp}}(s,u,\sigma)$ is same as minimizing entropy in the space of gradients. Intuitively, our objective aims to minimize uncertainty in the learning signal.

## F Additional Related Work on Multi-Agent Value Factorization

We discuss recent MARL methods within the Centralised Training and Decentralised Control paradigm [26] which improve value factorization. The original work of QTRAN [51] improves representational capacity of factorization schemes by generalizing methods such as QMIX [44] and VDN [53]. More recent advances combine techniques from dueling networks and temporal abstraction to learn MARL agent factorizations with sufficient representations [59]. A notable work is that of [60] which employs pretrained action representations to learn agent-specific roles. Decomposing policy optimization into role selection and role execution stages allows larger number of MARL agents to collaborate well even in unseen scenarios. Alternate works consider information theoretic objectives to introduce diversity in optimization and representation of shared multi-agent parameters [10]. Lastly, [61] extend these ideas by decomposing value learning in multi-agent actor-critic methods. Within the off-policy setting, these agents highlight sufficient representational capacity in both discrete and continuous action spaces.