1 Implementation Details

1.1 StarCraft II Setup

We select StarCraft II scenarios particularly for two reasons. Firstly, micromanagement scenarios consist of a larger number of agents with different action spaces. This requires a greater deal of coordination. Lastly, micromanagement scenarios in StarCraft II consist of multiple opponents which introduce a greater degree of surprise within consecutive states. Irrespective of the time evolution of an episode, environment dynamics of each scenario change rapidly as the agents need to respond to enemy's behavior. Agents were trained for a total of 5 random seeds consisting of 2 million steps in each environment. All baselines implementation consist of a Recurrent Neural Network (RNN) agent having memory consisting of past states and actions.

1.2 Model Specifications

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 $V^{a}_{surp}(s, u, \sigma)$ 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.

1.3 Hyperparameters

Table 1 presents hyperparameter values for EMIX. Value of β was tuned between 0.001 and 1 in intervals of 0.01 with best performance observed at $\beta = 0.01$. 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 1: Hyperparameter values for EMIX agents