QMIX Analysis

Yanjie Ze, May 2021

1 QMIX's disadvantages

1. Monotonic $\Rightarrow$ Suboptimal
2. Lack of `committed exploration` (a general disadvantage for the MARL algorithm like QMIX and QTRAN)

   $\Rightarrow$ In our setting, Only QBot uses ϵ-greedy.

2 Improve?

1. Delete the **absolute** constraint in **QMIX**. Will this work? (possibly hard to converge)
2. How to add the committed exploration in multi agent?

MAVEN

Yanjie Ze, May 2021

0 Related Links

MAVEN: Multi-Agent Variational Exploration , NeurIPS2019

paper link

code link

1 Motivation

To overcome the detrimental effects of QMIX's monotonicity constraint on exploration.

(Actually, the original model name of MAVEN is **Noise Q** ,shown in their codes. So I think the original idea of them is to simply add some noise to increase the exploration.)

2 Architecture

2.1 QMIX(ICML 2018)

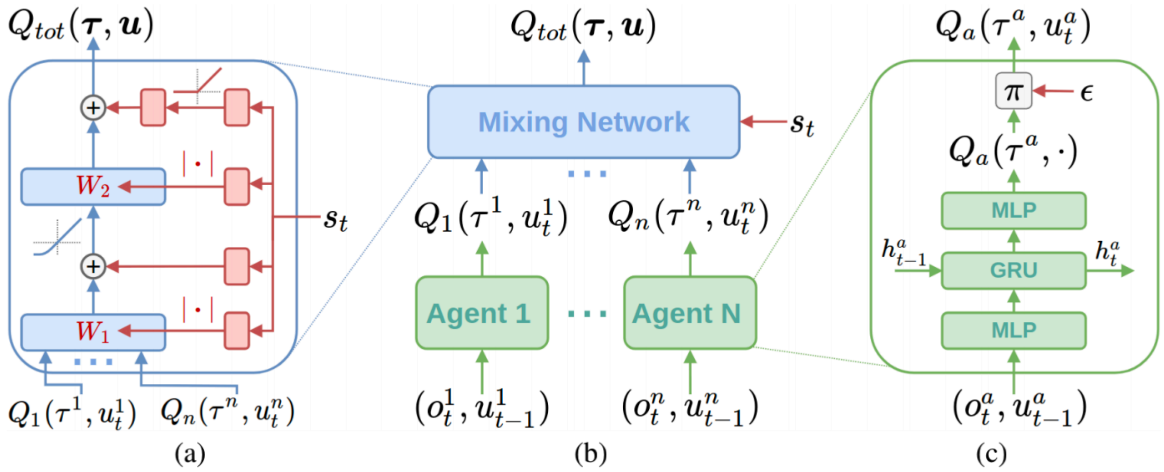


Figure 8: The overall setup of QMIX, reproduced from the original paper [40] (a) Mixing network structure. In red are the hypernetworks that produce the weights and biases for mixing network layers shown in blue. (b) The overall QMIX architecture. (c) Agent network structure. Best viewed in colour.

2.2 MAVEN

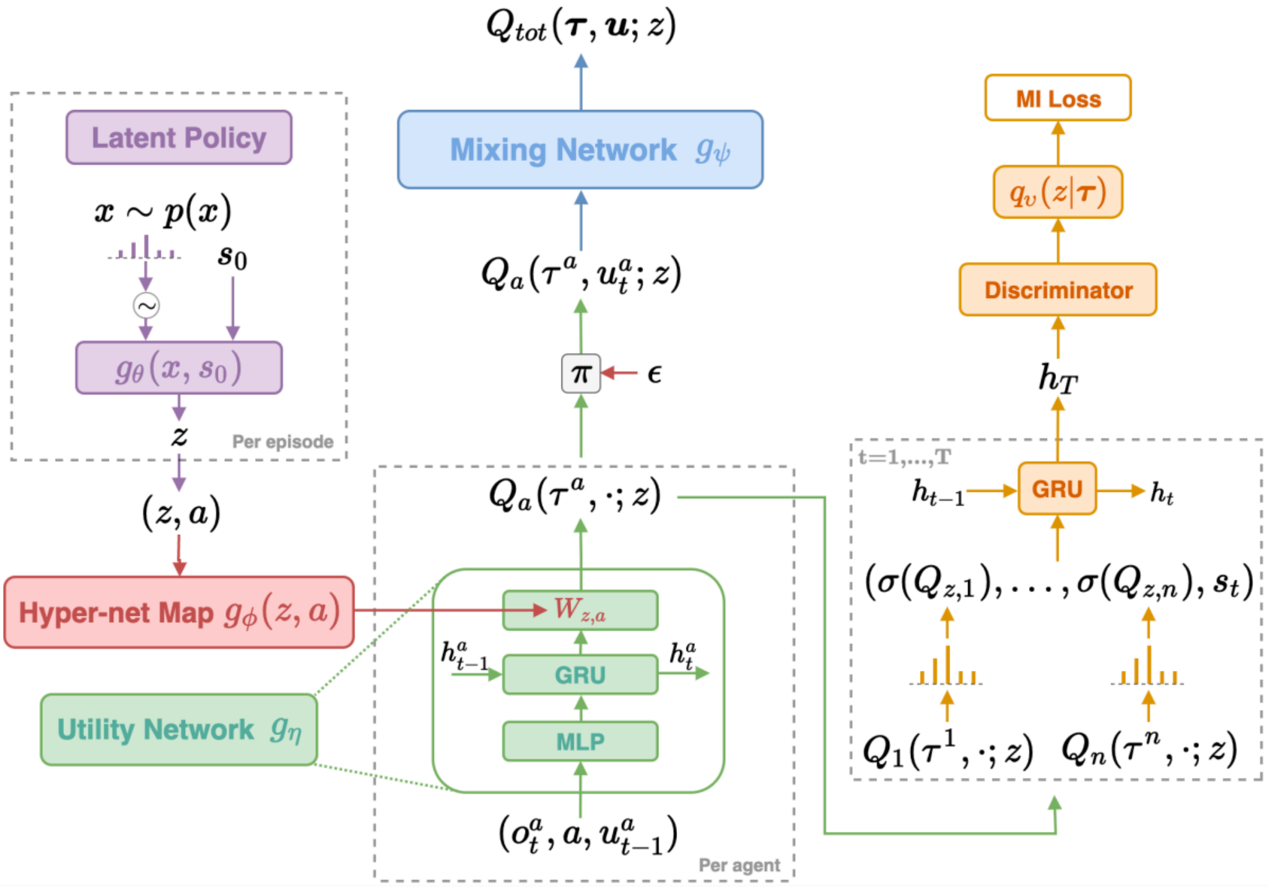


Figure 2: Architecture for MAVEN.

2.3 Latent Policy

x is a simple random variable, $x \sim p(x)$

Natural choices for *p*(*x*) are uniform for discrete *z* and uniform or normal for continuous *z*.

s_0 is the initial state.

z is the latent variable, $z \sim g_\theta(x, s_0)$

2.4 Boltzmann Policy

σ is an operator that returns a per-agent Boltzmann policy w.r.t. the utilities at each time step *t*.

$$\pi(\mathbf{a}_t|\mathbf{s}_t) \propto \exp\left(-\mathcal{E}(\mathbf{s}_t, \mathbf{a}_t)\right),$$

$$\sigma(Q_{z,t}) = exp(\frac{1}{\alpha}Q_{z,t})$$

3 Optimise the parameter

3.1 Q-learning Loss

Fix z, get the Q-learning loss:

$$\mathcal{L}_{QL}(\phi| \eta, \psi) = \mathbb{E}_{\pi_{\mathcal{A}}}[(Q(\mathbf{u}_t, s_t; z) - [r(\mathbf{u}_t, s_t) + \gamma \max_{\mathbf{u}_{t+1}} Q(\mathbf{u}_{t+1}, s_{t+1}; z)])^2],$$

3.2 Latent Policy Optimization

Next, fixing *ϕ, η, ψ*, the hierarchical policy over $\pi_z(\cdot|s_0;\theta)$ is trained on the cumulative trajectory reward

$$R(\tau, z|\phi, \eta, \psi) = \sum_t r_t$$

where $\boldsymbol{\tau}$ is the joint trajectory.

The hierarchical policy objective for *z*, freezing the parameters *ψ, η, ϕ* is given by:

$$\mathcal{J}_{RL}(\theta) = \int \mathcal{R}(\tau_{\mathcal{A}}|z) p_\theta(z|s_0)\rho(s_0) dz ds_0.$$

3.3 Mutual Information Loss

However, the formulation so far does not encourage diverse behaviour corresponding to different values of *z* and all the values of *z* could collapse to the same joint behaviour. To prevent this, we introduce a *mutual information* (MI) objective between the observed trajectories $\tau = (u_t, s_t)$,

Use an **RNN** to encode the entire trajectory.

Intuitively:

the MI objective encourages visitation of diverse trajectories **τ** while at the same time making them identififiable given *z*, thus elegantly **separating the *z* space into different exploration modes**.

$$\mathcal{J}_{MI} = \mathcal{H}(\sigma(\boldsymbol{\tau})) - \mathcal{H}(\sigma(\boldsymbol{\tau})|z) = \mathcal{H}(z) - \mathcal{H}(z|\sigma(\boldsymbol{\tau})),$$

Where H is the entropy. The model is shown in the right side of the architecture.

However, **neither the entropy of *σ*(τ) nor the conditional of *z* given the former is tractable for nontrivial mappings**, which makes directly using MI **infeasible**.

Therefore, we introduce a **variational distribution** parameterised by *υ* as a proxy for the posterior over *z*, which provides a lower bound on J_{MI}:

$$\mathcal{J}_{MI} \geq \mathcal{H}(z) + \mathbb{E}_{\sigma(\boldsymbol{\tau}),z}[\log(q_\upsilon(z|\sigma(\boldsymbol{\tau})))].$$

We refer to the righthand side of the above inequality as the variational MI objective $J_V(\upsilon, \varphi, \eta, \psi)$.

3.4 Overall

$$\max_{\upsilon,\phi,\eta,\psi,\theta} \mathcal{J}_{RL}(\theta) + \lambda_{MI}\mathcal{J}_V(\upsilon,\phi,\eta,\psi) - \lambda_{QL}\mathcal{L}_{QL}(\phi,\eta,\psi),$$

4 More About Mutual Information

Definition:

$$\boxed{I(X;Y) = D_{\mathrm{KL}}(P_{(X,Y)} \| P_X \otimes P_Y)}$$

$$D_{KL}(p||q) = \sum_{i=1}^{n} p(x_i) log(\frac{p(x_i)}{q(x_i)})$$

Discrete:

$$\boxed{\mathrm{I}(X;Y) = \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} p_{(X,Y)}(x,y)\log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)\,p_Y(y)}\right), \qquad \text{(Eq.1)}}$$

Continuous:

$$\boxed{\mathrm{I}(X;Y) = \int_{\mathcal{Y}}\int_{\mathcal{X}} p_{(X,Y)}(x,y)\log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)\,p_Y(y)}\right) dx\,dy, \qquad \text{(Eq.2)}}$$

Other forms:

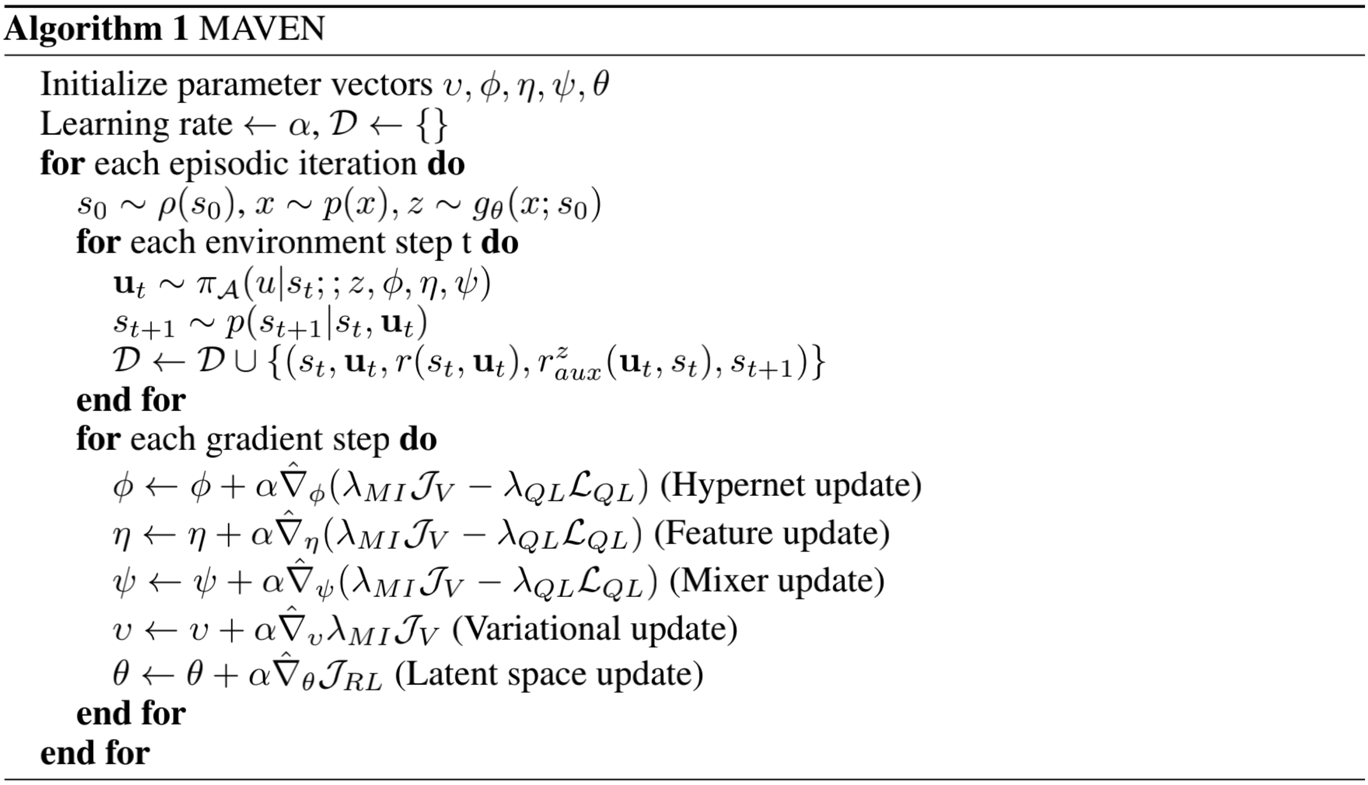
$$\begin{aligned} \mathrm{I}(X;Y) &\equiv \mathrm{H}(X) - \mathrm{H}(X \mid Y) \\ &\equiv \mathrm{H}(Y) - \mathrm{H}(Y \mid X) \\ &\equiv \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X,Y) \\ &\equiv \mathrm{H}(X,Y) - \mathrm{H}(X \mid Y) - \mathrm{H}(Y \mid X) \end{aligned}$$

Intuitively, **mutual information measures the information that X and Y share:**

It measures how much knowing one of these variables reduces uncertainty about the other. For example, if X and Y are independent, then knowing X does not give any information about Y and vice versa, so their mutual information is zero.

At the other extreme, if X is a deterministic function of Y and Y is a deterministic function of X then all information conveyed by X is shared with Y: knowing X determines the value of Y and vice versa. As a result, in this case the mutual information is the same as the uncertainty contained in Y (or X) alone, namely the entropy of Y (or X). Moreover, this mutual information is the same as the entropy of X and as the entropy of Y. (A very special case of this is when X and Y are the same random variable.)

5 Training



Algorithm 1 MAVEN

```
Initialize parameter vectors $\upsilon, \phi, \eta, \psi, \theta$
Learning rate $\leftarrow \alpha$, $\mathcal{D} \leftarrow \{\}$
for each episodic iteration do
    $s_0 \sim \rho(s_0), x \sim p(x), z \sim g_\theta(x; s_0)$
    for each environment step t do
        $\mathbf{u}_t \sim \pi_{\mathcal{A}}(u|s_t;;z, \phi, \eta, \psi)$
        $s_{t+1} \sim p(s_{t+1}|s_t, \mathbf{u}_t)$
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, \mathbf{u}_t, r(s_t, \mathbf{u}_t), r^z_{aux}(\mathbf{u}_t, s_t), s_{t+1})\}$
    end for
    for each gradient step do
        $\phi \leftarrow \phi + \alpha\hat{\nabla}_\phi(\lambda_{MI}\mathcal{J}_V - \lambda_{QL}\mathcal{L}_{QL})$ (Hypernet update)
        $\eta \leftarrow \eta + \alpha\hat{\nabla}_\eta(\lambda_{MI}\mathcal{J}_V - \lambda_{QL}\mathcal{L}_{QL})$ (Feature update)
        $\psi \leftarrow \psi + \alpha\hat{\nabla}_\psi(\lambda_{MI}\mathcal{J}_V - \lambda_{QL}\mathcal{L}_{QL})$ (Mixer update)
        $\upsilon \leftarrow \upsilon + \alpha\hat{\nabla}_\upsilon\lambda_{MI}\mathcal{J}_V$ (Variational update)
        $\theta \leftarrow \theta + \alpha\hat{\nabla}_\theta\mathcal{J}_{RL}$ (Latent space update)
    end for
end for
```

6 Test

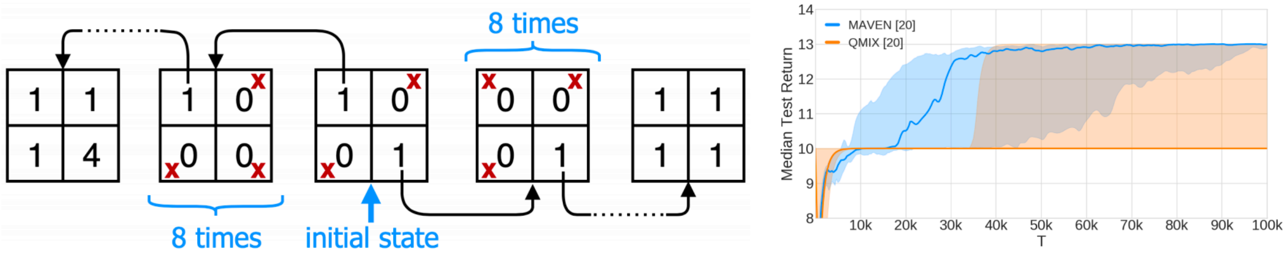
At test time, we sample *z* **at the start of an episode** and then perform a decentralised argmax on the corresponding Q-function to select actions.

7 Experiment Results

m-step matrix game

As *m* increases, it becomes increasingly difficult to discover the optimal policy using ϵ-dithering and a *committed* approach becomes necessary.

QMIX gets stuck in a suboptimal policy with payoff 10, while MAVEN successfully learns the true optimal policy with payoff 13. This example shows how representational constraints can hurt performance if they are left unmoderated.



(a) (b)

Figure 3: (a) m-step matrix game for $m = 10$ case (b) median return of MAVEN and QMIX method on 10-step matrix game for 100k training steps, averaged over 20 random initializations (2nd and 3rd quartile is shaded).