Learning And Mining from DatA

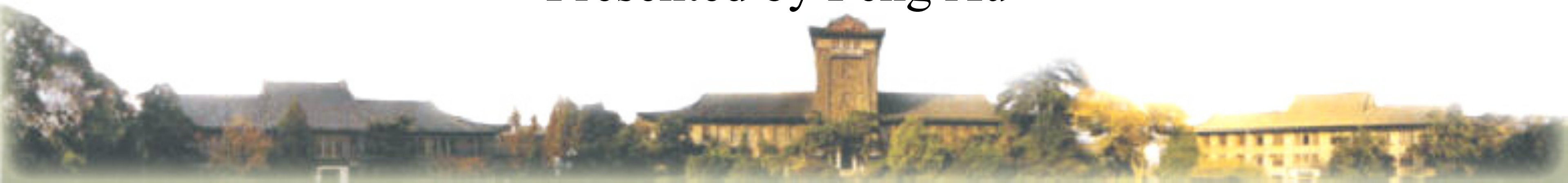
www.lamda.nju.edu.cn

Credit Assignment with Meta-Policy Gradient for Multi-Agent Reinforcement Learning

Jianzhun Shao, Hongchang Zhang, Yuhang Jiang, Shuncheng He, Xiangyang Ji

Department of Automation
Tsinghua University

Presented by Feng Xu



Background

- Centralized training with decentralized execution (CTDE):
  – Enhance agents' ability with emphasis on individually processing local observations (ROMA, RODE, Maven)
  – decompose the single reward to each agent (Qmix, Qatten, Qplex, Qtran)
- Reward decomposition

Motivation

- Inspired by Meta-DDPG
- An explicit hierarchy to the distilled information from the full state

Problem Formalization

Learning And Mining from DatA

www.lamda.nju.edu.cn

- Decentralized Partially Observable Markov Decision Process (Dec-POMDP)
  - $G = < S, A, I, P, r, Z, O, n, \gamma >$
    - S: global true state
    - $a \in A = A^n$: actions taken by individual agents form a joint action space
    - $i \in I$: agent, n: number of agents
    - $r(s, \mathrm{a}): S \times A \to R$: reward function
    - $z \in Z$: local observation
    - $O(s, i) \to Z$: partial observation of individual agents
    - γ: discount factor

Overview of MNMPG

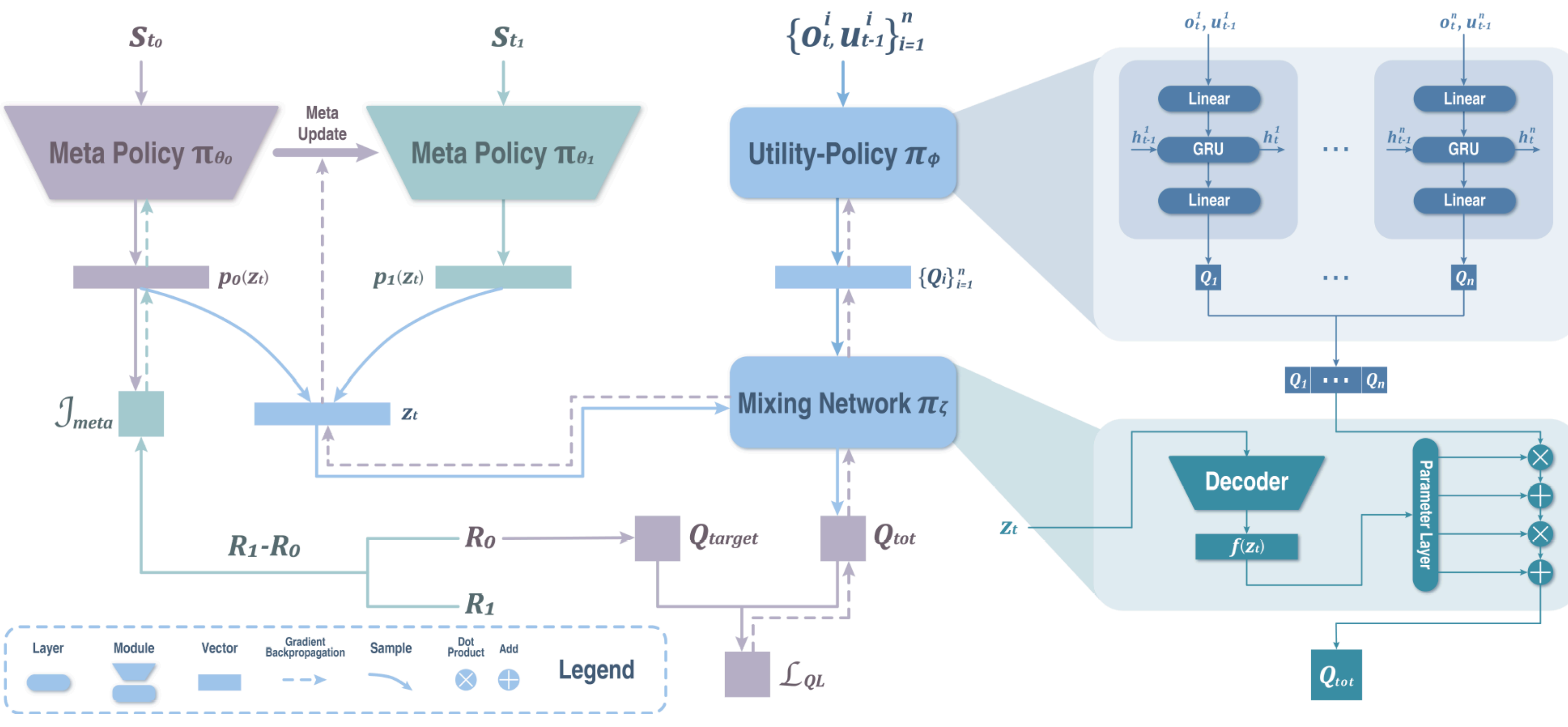


Figure 1: The MNMPG framework.

MNMPG

www.lamda.nju.edu.cn

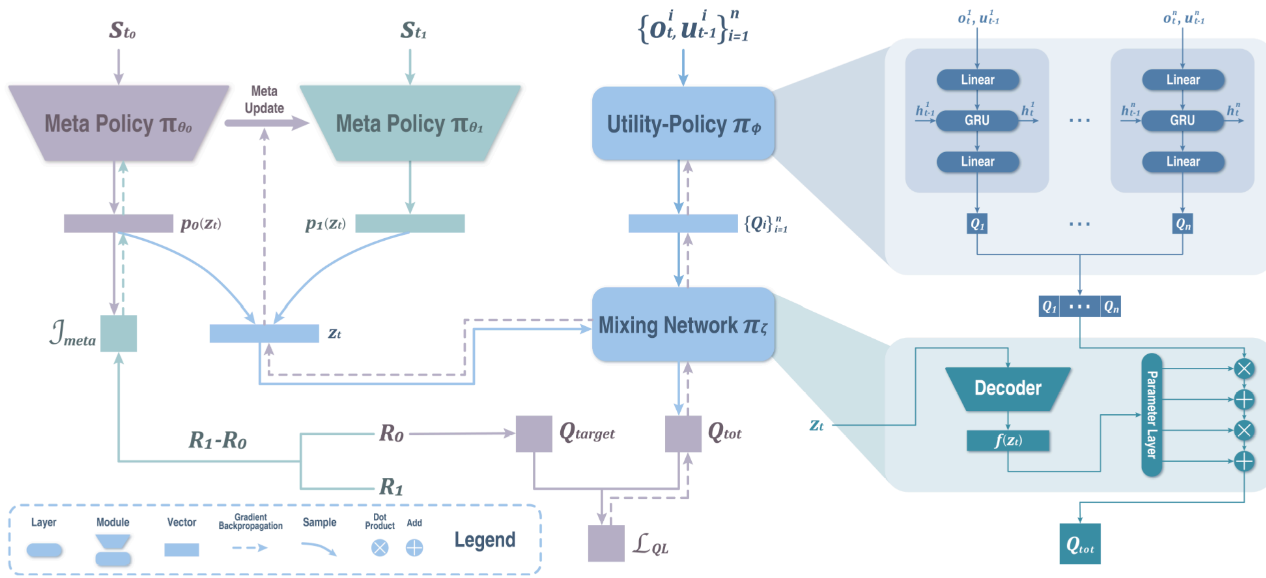
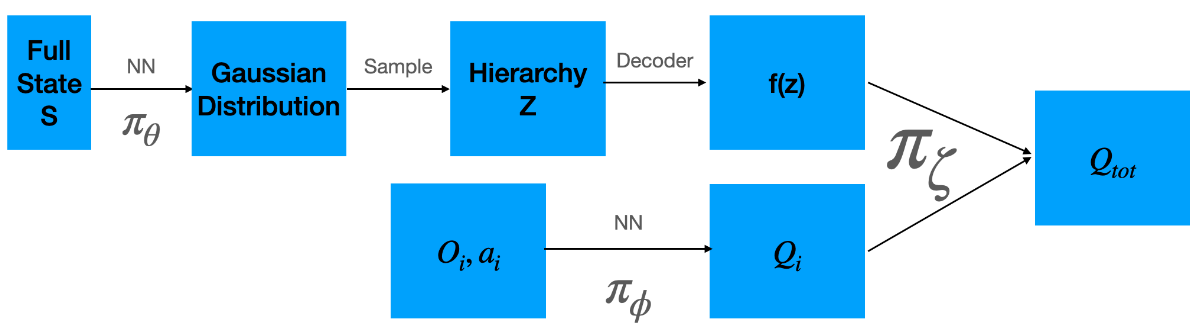
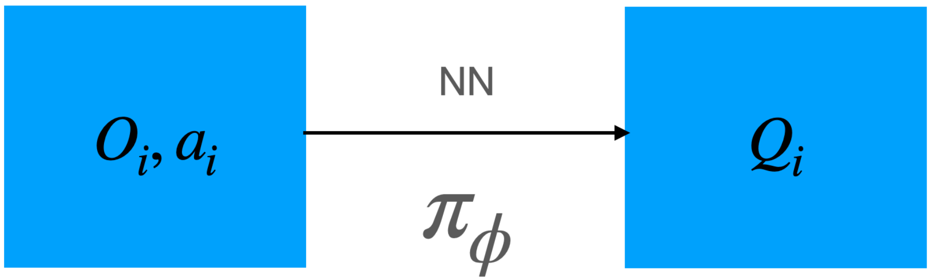


Figure 1: The MNMPG framework.



Local Utility Network



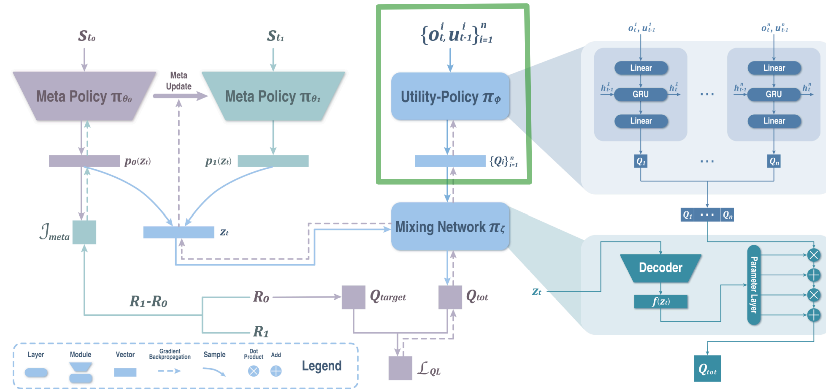


Figure 1: The MNMPG framework.

Compute the value function of each individual agent

Hierarchy Network

$$z_t \sim N(\mu_\theta(s_t), \sigma_\theta(s_t))$$

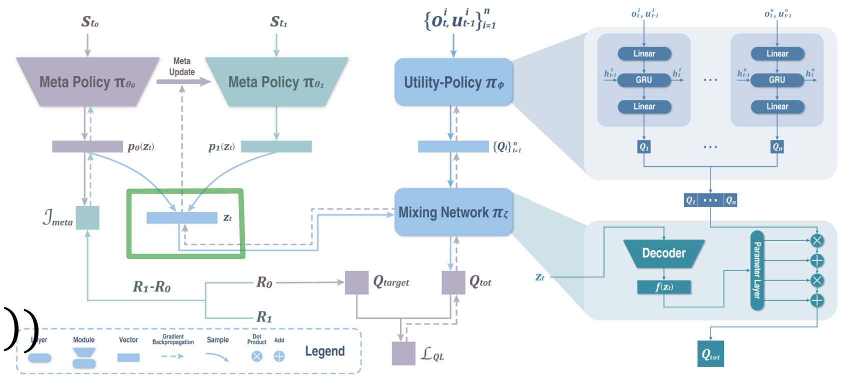
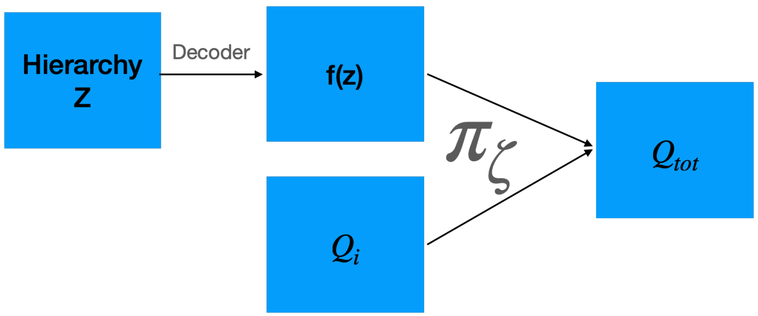


Figure 1: The MNMPG framework.

Compute a global hierarchy for mixing the Q values of each individual agent

- reflect high-level goals

Mixing Network



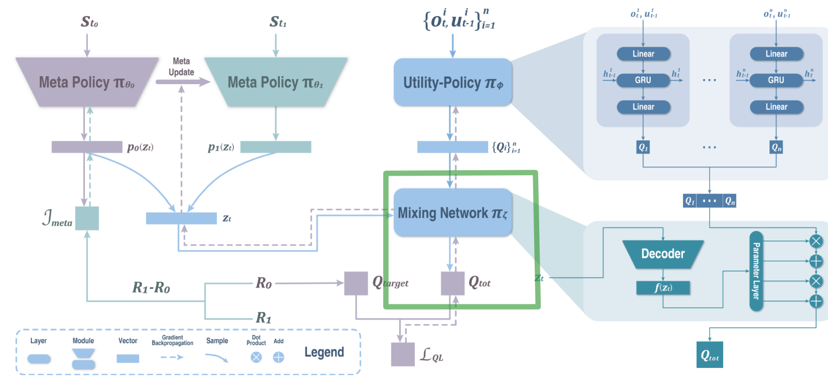


Figure 1: The MNMPG framework.

Compute a global Q value from individual Q values

MNMPG

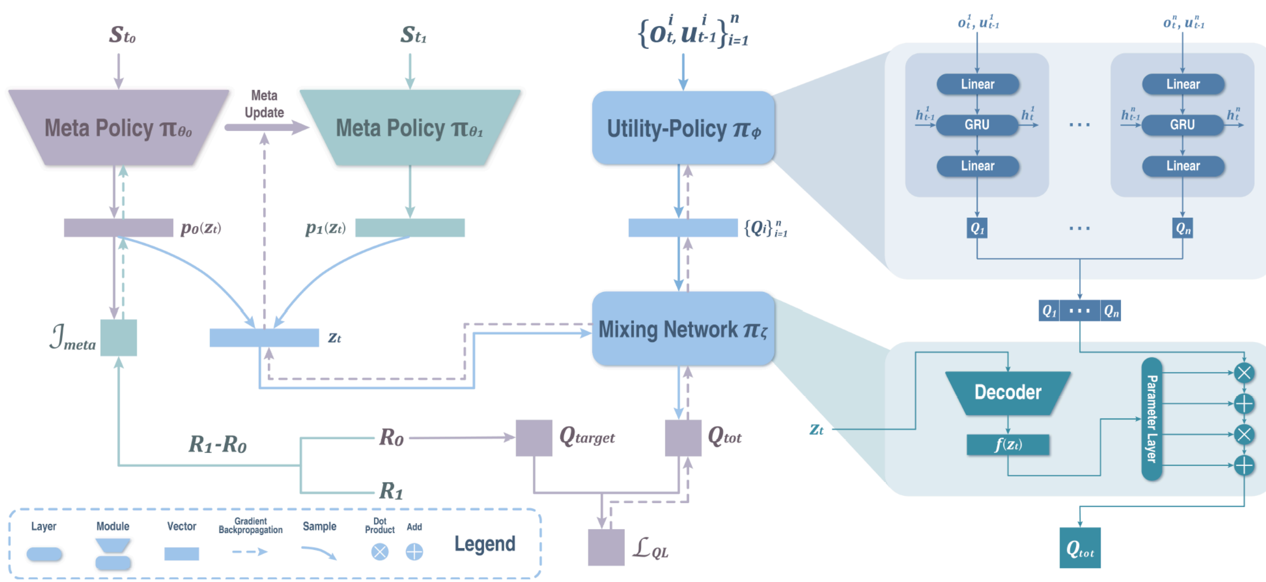
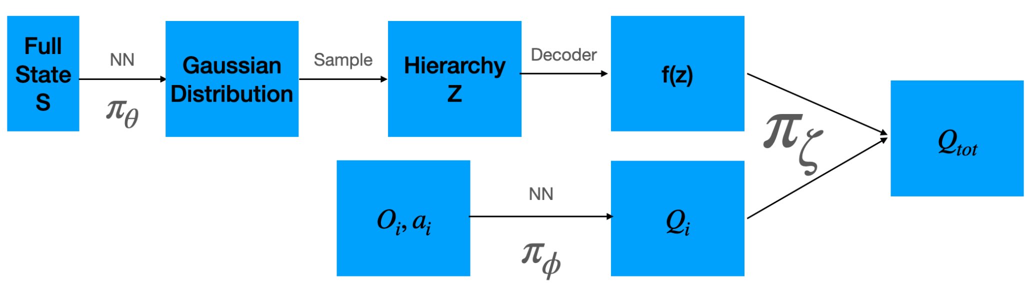


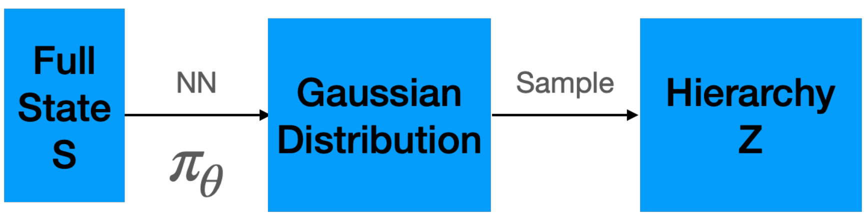
Figure 1: The MNMPG framework.



Details of MNMPG

- Training Objective
- Design Concept

Hierarchy Network



$$z_t \sim N(\mu_\theta(s_t), \sigma_\theta(s_t))$$

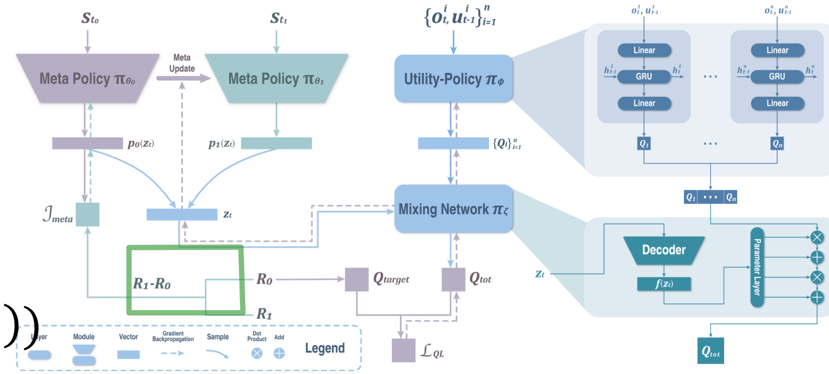


Figure 1: The MNMPG framework.

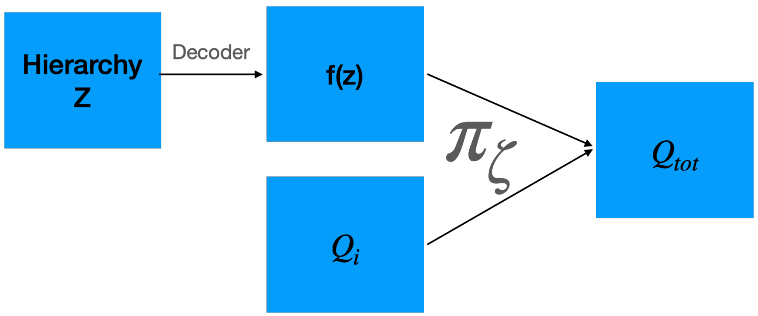
- Changes according to time: represents the current goal of the team
  - Maven computes a episodic exploration mode at the initial state
- Contains no prior information and can be improved spontaneously
  - Number of roles in RODE needs manual finetuning

$$J_{meta}(\pi_\theta) = E_{D_0 \sim \pi_{\theta,\phi,\zeta}}[R(\pi, D_0)] = E_{D_0 \sim \pi_{\theta,\phi,\zeta}}[R_{\pi'} - R_\pi]$$

$\pi' = \mathrm{QL}(\pi, D_0)$ is updated from π using D_0 sampled by π

$$\nabla_\theta J_{meta} = E_{D_0 \sim \pi_{\theta,\phi,\zeta}}[R(\pi, D_0) \nabla_\theta \log P(D_0 | \pi_{\theta,\phi,\zeta})]$$

Mixing Network



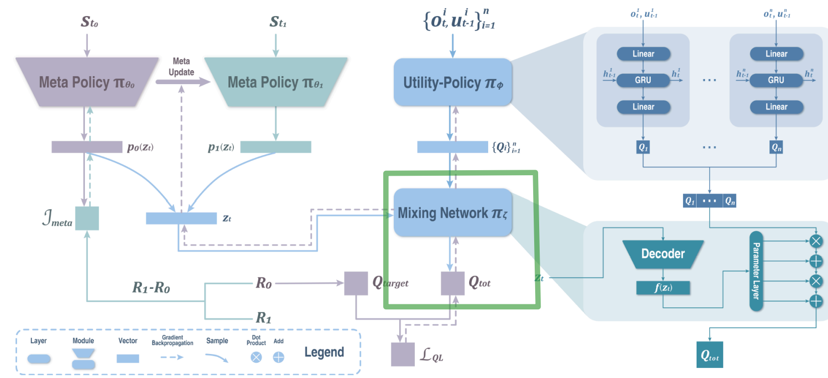


Figure 1: The MNMPG framework.

MNMPG: $Q_{tot}(\tau, a) = f_\zeta(Q_i(\tau^i, a_t^i)|z_t)$

QMix: $Q_{tot}(\tau, a) = f_\zeta\ (Q_i(\tau^i, a_t^i)|s_t)$

Maven/ROMA: $Q_{tot}(\tau, a) = f_\zeta\ (Q_i(\tau^i, a_t^i|z_i)|s_t)$

Advantage:

- Better reflect high-level goal than simply using full state
- Non-monotonic global hierarchy
- Global hierarchy is more efficient than local hierarchy

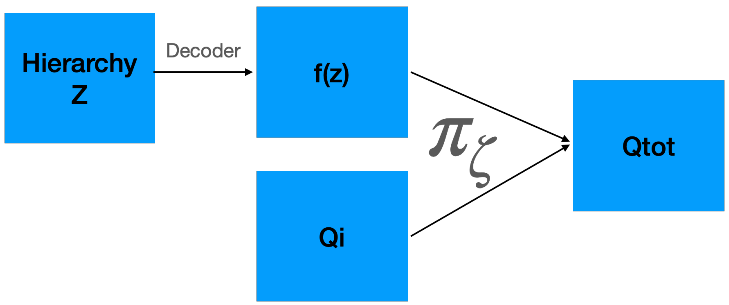
Mixing Network

Qmix enforces a **monotonic** value decomposing network that satisfies: $\frac{\partial Q_{tot}}{\partial Q_a} \geq 0, \forall$ a

- May converge to suboptimal policy: monotonicity implies that the optimal action of agent I does not depend on the actions of the other agents

Maven introduces a **diverse ensemble of monotonic approximations** with the help of a latent space. Agents act based on variable that implies the **mode of exploration**

Mixing Network



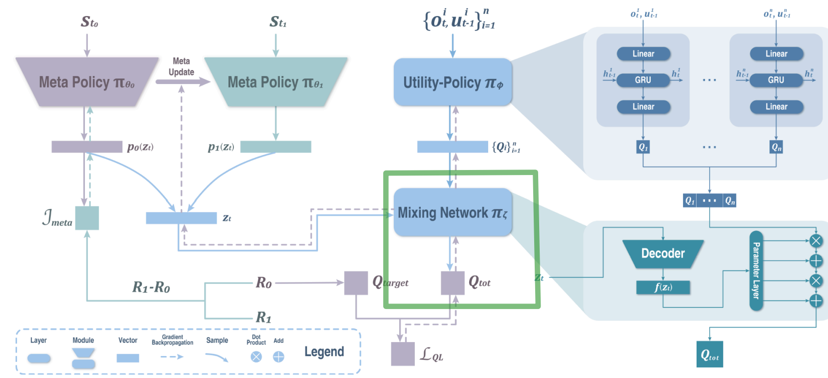


Figure 1: The MNMPG framework.

Q_{tot} is updated by Q learning loss:

$$\mathrm{L_{QL}}(\phi,\zeta) = E_{D\sim\pi_{\theta,\phi,\zeta}}[(Q_{tot}(a_t,s_t;z_t,\phi,\zeta)$$
$$-\left[r(a_t,s_t)+\gamma \max\nolimits_{a_{t+1}} Q_{tot}(a_{t+1},s_{t+1};z_{t+1},\phi,\zeta)\right]^2)]$$

Pseudocode

Algorithm 1 Mixing Network with Meta Policy Gradient

Initialize $\pi_\theta, \pi_\phi, \pi_\zeta$
Set learning rate $\leftarrow \eta$,meta learning rate$\leftarrow \lambda$,$D_{all} \leftarrow \{\}$
for each episode iteration **do**
  $D_0 \leftarrow \{\}, D_1 \leftarrow \{\}$
  Generate tuple $\{s_t, z_t, a_t, r_t, s_{t+1}, z_{t+1}\}_{t=1}^T$ by executing $\pi_{\theta,\phi,\zeta}$
  $D_0 \leftarrow D_0 \cup \{s_t, z_t, a_t, r_t, s_{t+1}, z_{t+1}\}_{t=1}^T$
  $R_0 \leftarrow \sum_{t=1}^T r_t$
  for exercise move iteration **do**
    $\phi \leftarrow \phi + \eta\hat{\nabla}_\phi \mathcal{L}_{QL}(D_0)$
    $\zeta \leftarrow \zeta + \eta\hat{\nabla}_\zeta \mathcal{L}_{QL}(D_0)$
  end for
  Generate tuple $\{s_t, z_t, a_t, r_t, s_{t+1}, z_{t+1}\}_{t=1}^T$ by executing $\pi_{\theta,\phi,\zeta}$
  $D_1 \leftarrow D_1 \cup \{s_t, z_t, a_t, r_t, s_{t+1}, z_{t+1}\}_{t=1}^T$
  $R_1 \leftarrow \sum_{t=1}^T r_t$
  $\mathcal{R} \leftarrow R_1 - R_0$
  $\theta \leftarrow \theta + \lambda\hat{\nabla}_\theta \mathcal{J}_{meta}(D_0, \mathcal{R})$
  $D_{all} \leftarrow D_{all} \cup D_0 \cup D_1$
  for Q-learning iteration **do**
    $\phi \leftarrow \phi + \eta\hat{\nabla}_\phi \mathcal{L}_{QL}(D_{all})$
    $\zeta \leftarrow \zeta + \eta\hat{\nabla}_\zeta \mathcal{L}_{QL}(D_{all})$
  end for
end for

Learning

Update utility network and mixing network (A new task for meta learning)

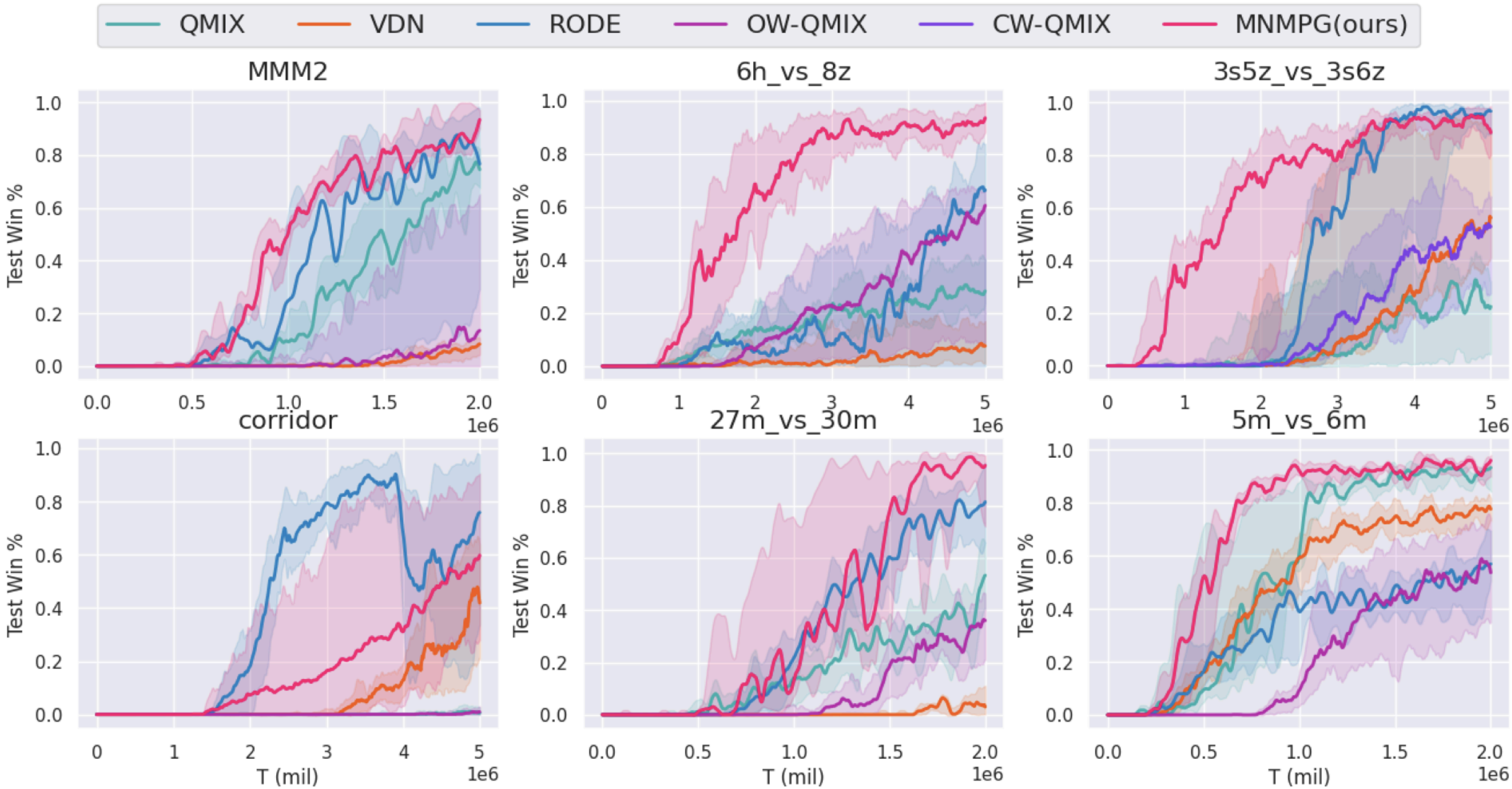
To learn

Improve sample efficiency

Experiments: Performance

Learning And Mining from DatA

www.lamda.nju.edu.cn



Experiments: Exploration

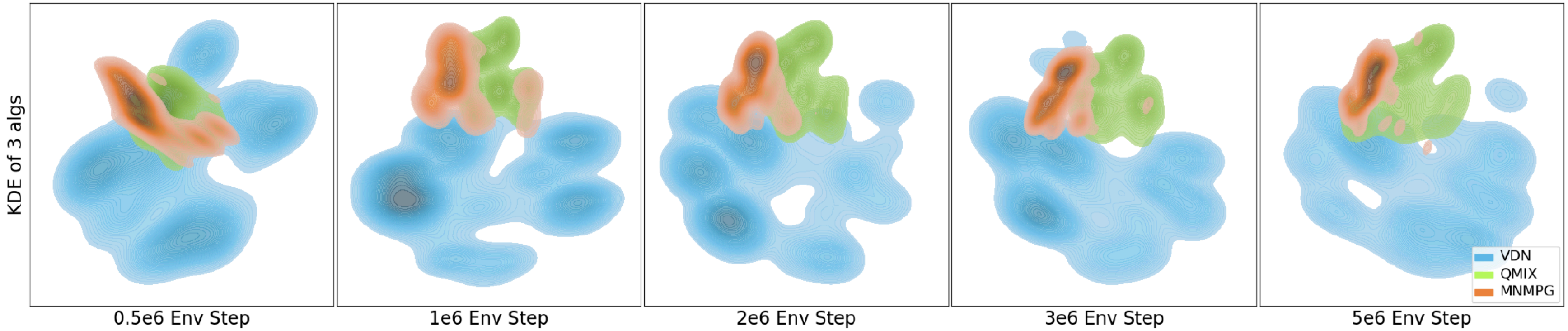


Figure 3: The kernel density estimate(KDE) map of 6h_vs_8z's state visitation after t-SNE for 3 algorithms.

VDN: Q_tot = sum(Q_i)
QMIX: Q_tot=NN(Qs)
MNMPG: Q_tot=NN(Qs,z,s)

Experiments: Adaptation to other methods

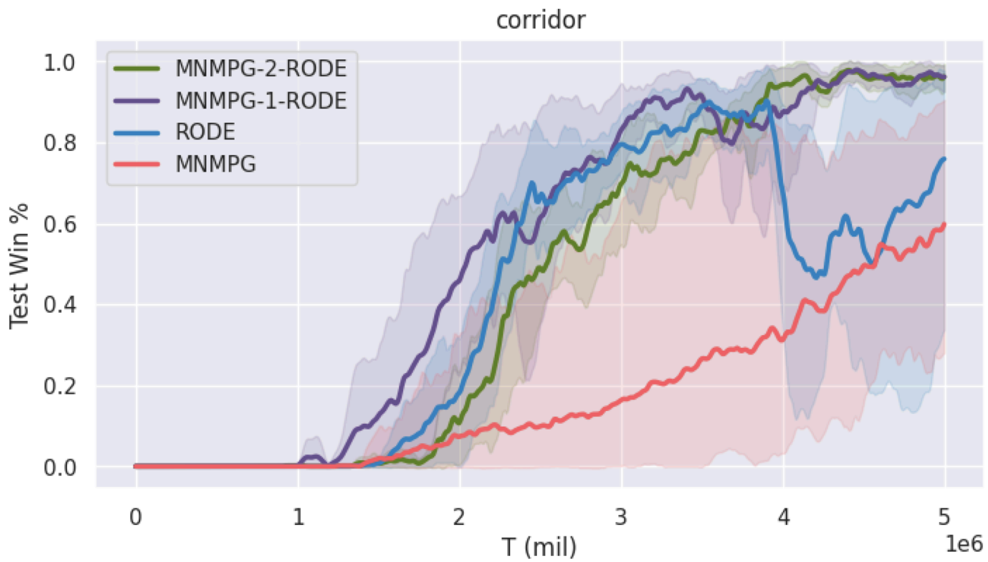
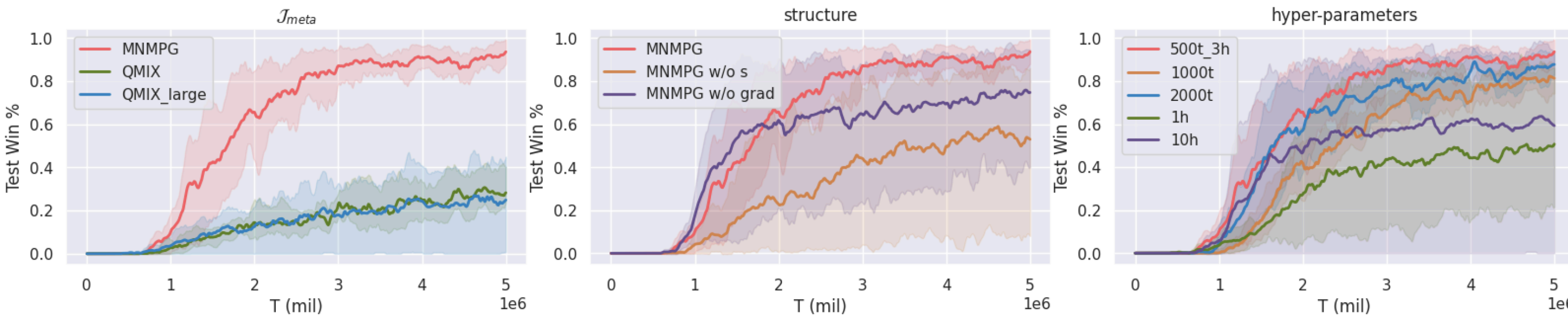


Figure 4: Adaptation to RODE.

MNMPG-1-RODE: Modification on role policy only
MNMPG-2-RODE: modification on both levels

Experiments: Ablation



(a) (b)

Figure 5: Ablation study of MNMPG.

W/o grad: no L_{QL} loss on hierarchy
W/o s: no full state input of mixing network

- Thanks