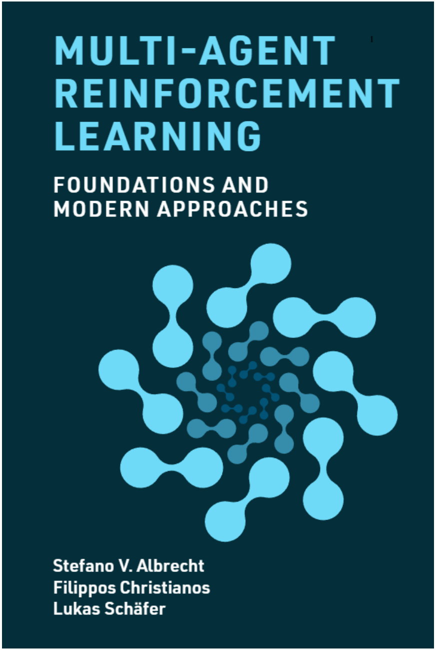
MARL's Zoo

An Introduction to Multi-Agent RL

Vojtěch Formánek
xforman@fi.muni.cz

Masaryk University

December 27, 2025


MULTI-AGENT
REINFORCEMENT
LEARNING
FOUNDATIONS AND
MODERN APPROACHES
Stefano V. Albrecht
Filippos Christianos
Lukas Schäfer

Today

- Intro
- Solving Joint Policies
- Independent -DQN, -REINFORCE, -A2C
- Centralized Critics
- Value Decomposition (VDN, QMIX)

- **Intro**
- Solving Joint Policies
- Independent -DQN, -REINFORCE, -A2C
- Centralized Critics

Definition 1 (Stochastic Game) *consists of:*

- *Finite set of agents* $I = 1, \dots, n$
- *Finite set of states* S, with a subset of terminal $\overline{S} \subset S$
- *State transition function* $\mathcal{T} : S \times A \times S \to [0, 1]$ *such that*

$$\forall s \in S, a \in A : \sum_{s' \in S} \mathcal{T}(s, a, s') = 1,$$

  where $A = A_1 \times \cdots \times A_n$
- *Initial state distribution* $S \to [0, 1]$

$$\sum_{s \in S} \mu(s) = 1 \quad \textit{and} \quad \forall s \in \overline{S} : \mu(s) = 0$$

- *For each* $i \in I$
  - *Reward function* $\mathcal{R}_i : S \times A \times S$
  - *Finite set of actions* A_i

Definition 2 (Partially observable SG) *consists of:*

- *Finite set of agents* $I = 1, \ldots, n$
- *Finite set of states* S, with a subset of terminal $\overline{S} \subset S$
- *State transition function* $\mathcal{T} : S \times A \times S \to [0, 1]$ *such that*

$$\forall s \in S, a \in A : \sum_{s' \in S} \mathcal{T}(s, a, s') = 1,$$

  where $A = A_1 \times \cdots \times A_n$
- *Initial state distribution* $S \to [0, 1]$

$$\sum_{s \in S} \mu(s) = 1 \quad \text{and} \quad \forall s \in \overline{S} : \mu(s) = 0$$

- *For each* $i \in I$
  - *Reward function* $\mathcal{R}_i : S \times A \times S$
  - *Finite set of actions* A_i *and observations* O_i
  - *Observation function* $\mathcal{O}_i : A \times S \times O_i \to [0, 1]$ such that

$$\forall a \in A, s \in S : \sum_{o_i \in O_i} \mathcal{O}_i(a, s, o_i) = 1$$

Notation

- $s^t = (s_1^t, \ldots, s_n^t), a^t = (a_1^t, \ldots, a_n^t)$, at time t
- *observations*
  - SG: $o_i^t = (s^t, a^{t-1})$
  - Other actions unobserved: $o_i^t = (s^t, a_i^{t-1})$
  - Limited view: $o_i^t = (\bar{s}^t, \bar{a}^t)$, where $\bar{s}^t \subset s^t$ and $\bar{a}^t \subset a^t$
- history
  - full: $\hat{h}^t = \{s^0, o^0, a^0, \ldots, s^t, o^t\}$
  - observations: $\sigma(\hat{h}^t) = \{o^0, \ldots, o^t\}$

Today

- Intro
- **Solving Joint Policies**
- Independent -DQN, -REINFORCE, -A2C
- Centralized Critics

Policy

Joint policy $\pi = (\pi_1, \ldots, \pi_n)$ *satisfies requirements in terms of expected return* U_i^π*, where*

$$U_i^\pi = \lim_{t \to \infty} \mathbb{E}_{\hat{h}^t \sim (\mu, \mathcal{T}, \mathcal{O}, \pi)} \left[\sum_{\tau=0}^{t-1} \gamma^\tau \mathcal{R}_i \right]$$

for each agent i*.*

Recursive Expected Return

$$
\begin{aligned}
V_i^{\pi}(\hat{h}) &= \sum_{a \in A} \pi(a \mid \sigma(\hat{h}))\, Q_i^{\pi}(\hat{h}, a) \\
Q_i^{\pi}(\hat{h}, a) &= \sum_{s' \in S} \mathcal{T}\left[\mathcal{R}_i + \gamma \sum_{o' \in O} \mathcal{O}(o' | a, s')\, V_i^{\pi}(\langle \hat{h}, a, s', o' \rangle)\right] \\
U_i^{\pi} &= \mathbb{E}^{\pi}[V_i^{\pi}(\langle s^0, o^0 \rangle)]
\end{aligned}
$$

Solutions
Equilibrium

In a general-sum game π *is a* **Nash Equilibrium** *if*

$$\forall i, \pi'_i : U_i^{\langle \pi'_i, \pi_{-i} \rangle} \leq U_i^{\pi}$$

- Sub-optimality
- Non-uniqueness
- Incompleteness

Solutions
Non-Equilibrium

Pareto Optimality:
A joint policy π *is* Pareto-dominated *by* π' *if*

$$\forall i : U_i^{\pi'} \geq U_i^{\pi} \text{ and } \exists i : U_i^{\pi'} > U_i^{\pi},$$

π *is* Pareto-optimal *if it is not* Pareto-dominated

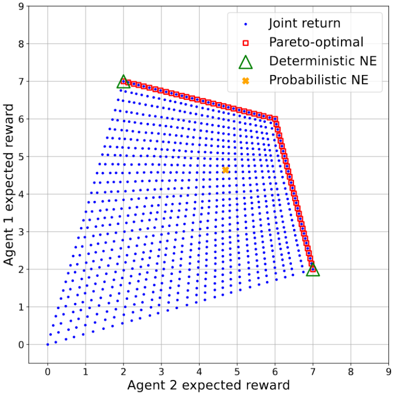


Figure: Feasible expected joint rewards and Pareto frontier in the Chicken matrix game.

Three Paradigms

- Centralized training and execution
- Decentralized training and execution
- Centralized training and decentralized execution

Algorithm 4 Central Q-learning (CQL) for stochastic games

Initialize: $Q(s,a)=0$ for all $s \in S$ and $a \in A = A_1 \times \ldots \times A_n$

Repeat for every episode:

for $t = 0, 1, 2, \ldots$ **do**

    Observe current state s^t

    With probability ϵ: choose random joint action $a^t \in A$

    Otherwise: choose joint action $a^t \in \arg\max_a Q(s^t, a)$

    Apply joint action a^t, observe rewards $r_1^t, \ldots, r_n^t$ and next state s^{t+1}

    Transform $r_1^t, \ldots, r_n^t$ into scalar reward r^t

    $Q(s^t, a^t) \leftarrow Q(s^t, a^t) + \alpha \left[r^t + \gamma \max_{a'} Q(s^{t+1}, a') - Q(s^t, a^t) \right]$

Algorithm 5 Independent Q-learning (IQL) for stochastic games

// Algorithm controls agent i

Initialize: $Q_i(s, a_i) = 0$ for all $s \in S, a_i \in A_i$

Repeat for every episode:

for $t = 0, 1, 2, \ldots$ **do**

    Observe current state s^t

    With probability ϵ: choose random action $a_i^t \in A_i$

    Otherwise: choose action $a_i^t \in \arg\max_{a_i} Q_i(s^t, a_i)$

    (meanwhile, other agents $j \neq i$ choose their actions a_j^t)

    Observe own reward r_i^t and next state s^{t+1}

    $Q_i(s^t, a_i^t) \leftarrow Q_i(s^t, a_i^t) + \alpha \left[r_i^t + \gamma \max_{a_i'} Q_i(s^{t+1}, a_i') - Q_i(s^t, a_i^t) \right]$

A Big Step in Theory

Challenges

- Non-stationarity
- Equilibrium selection
- Credit assignment
- Scaling to many agents

Today

- Intro
- Solving Joint Policies
- **Independent -DQN, -REINFORCE, -A2C**
- Centralized Critics

Algorithm 17 Independent deep Q-networks

Initialize n value networks with random parameters $\theta_1, \dots, \theta_n$
Initialize n target networks with parameters $\bar{\theta}_1 = \theta_1, \dots, \bar{\theta}_n = \theta_n$
Initialize a replay buffer for each agent $D_1, D_2, \dots, D_n$
for time step $t = 0, 1, 2, \dots$ **do**
  Collect current observations $o_1^t, \dots, o_n^t$
  for agent $i = 1, \dots, n$ **do**
    With probability ϵ: choose random action a_i^t
    Otherwise: choose $a_i^t \in \arg\max_{a_i} Q(h_i^t, a_i; \theta_i)$
  Apply actions $(a_1^t, \dots, a_n^t)$; collect rewards $r_1^t, \dots, r_n^t$ and next observations $o_1^{t+1}, \dots, o_n^{t+1}$
  for agent $i = 1, \dots, n$ **do**
    Store transition $(h_i^t, a_i^t, r_i^t, h_i^{t+1})$ in replay buffers D_i
    Sample random mini-batch of B transitions $(h_i^k, a_i^k, r_i^k, h_i^{k+1})$ from D_i
    if s^{k+1} is terminal[2] **then**
      Targets $y_i^k \leftarrow r_i^k$
    else
      Targets $y_i^k \leftarrow r_i^k + \gamma \max_{a_i' \in A_i} Q(h_i^{k+1}, a_i'; \bar{\theta}_i)$
    Loss $\mathcal{L}(\theta_i) \leftarrow \frac{1}{B} \sum_{k=1}^{B} \left(y_i^k - Q(h_i^k, a_i^k; \theta_i) \right)^2$
    Update parameters θ_i by minimizing the loss $\mathcal{L}(\theta_i)$
    In a set interval, update target network parameters $\bar{\theta}_i$

Addressing Non-stationarity

- Buffer can store outdated experiences
- Use smaller buffers
- Importance Sampling: reweigh based on a_{-i}
- Switch to on-policy

Algorithm 18 Independent REINFORCE

Initialize n policy networks with random parameters $\phi_1, \ldots, \phi_n$
Repeat for every episode:
for time step $t = 0, 1, 2, \ldots, T-1$ **do**
  Collect current observations $o_1^t, \ldots, o_n^t$
  for agent $i = 1, \ldots, n$ **do**
    Sample actions a_i^t from $\pi(\cdot \mid h_i^t; \phi_i)$
  Apply actions $(a_1^t, \ldots, a_n^t)$; collect rewards $r_1^t, \ldots, r_n^t$ and next observations $o_1^{t+1}, \ldots, o_n^{t+1}$
for agent $i = 1, \ldots, n$ **do**
  Loss $\mathcal{L}(\phi_i) \leftarrow -\frac{1}{T} \sum_{t=0}^{T-1} \left(\sum_{\tau=t}^{T-1} \gamma^{\tau-t} r_i^{\tau} \right) \log \pi(a_i^t \mid h_i^t; \phi_i)$
  Update parameters ϕ_i by minimizing the loss $\mathcal{L}(\phi_i)$

Algorithm 19 Independent A2C with synchronous environments

```
Initialize n actor networks with random parameters φ_1, ..., φ_n
Initialize n critic networks with random parameters θ_1, ..., θ_n
Initialize K parallel environments
for time step t = 0 ... do
    Batch of observations for each agent and environment: [o_1^{t,1} ... o_1^{t,K}; ⋱ ; o_n^{t,1} ... o_n^{t,K}]
    Sample actions [a_1^{t,1} ... a_1^{t,K}; ⋱ ; a_n^{t,1} ... a_n^{t,K}] ~ π(·|h_1^t; φ_1), ..., π(·|h_n^t; φ_n)
    Apply actions; collect rewards [r_1^{t,1} ... r_1^{t,K}; ⋱ ; r_n^{t,1} ... r_n^{t,K}] and observations [o_1^{t+1,1} ... o_1^{t+1,K}; ⋱ ; o_n^{t+1,1} ... o_n^{t+1,K}]
    for agent i = 1, ..., n do
        if s^{t+1,k} is terminal then
            Advantage Adv(h_i^{t,k}, a_i^{t,k}) ← r_i^{t,k} − V(h_i^{t,k}; θ_i)
            Critic target y_i^{t,k} ← r_i^{t,k}
        else
            Advantage Adv(h_i^{t,k}, a_i^{t,k}) ← r_i^{t,k} + γV(h_i^{t+1,k}; θ_i) − V(h_i^{t,k}; θ_i)
            Critic target y_i^{t,k} ← r_i^{t,k} + γV(h_i^{t+1,k}; θ_i)
        Actor loss L(φ_i) ← (1/K) Σ_{k=1}^{K} Adv(h_i^{t,k}, a_i^{t,k}) log π(a_i^{t,k} | h_i^{t,k}; φ_i)
        Critic loss L(θ_i) ← (1/K) Σ_{k=1}^{K} (y_i^{t,k} − V(h_i^{t,k}; θ_i))^2
        Update parameters φ_i by minimizing the actor loss L(φ_i)
        Update parameters θ_i by minimizing the critic loss L(θ_i)
```

MA Policy Gradient Theorem

Partially Observable Case

$$\nabla_{\phi_i} J(\phi_i) \propto \mathbb{E}_{\hat{h} \sim \Pr(\hat{h}|\pi), a_i \sim \pi_i, a_{-i} \sim \pi_{-i}} \left[Q_i^{\pi}(\hat{h}, \langle a_i, a_{-i} \rangle) \nabla_{\phi_i} \log \pi_i(a_i \mid h_i = \sigma_i(\hat{h}); \phi_i) \right]$$

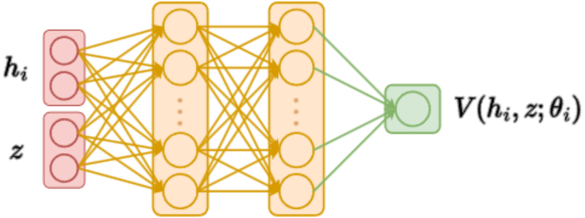
Today

- Intro
- Solving Joint Policies
- Independent -DQN, -REINFORCE, -A2C
- **Centralized Critics**

The Idea

- No constraints on critic during training
- Give critic info z about other agents
- critic as $V(h_1^t, \ldots, h_n^t; \theta_i)$
- or ...

Centralized Critic



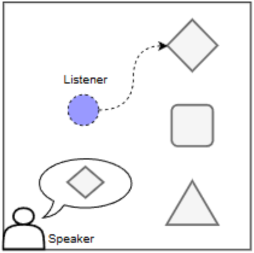
Value Loss:

$$\mathcal{L}(\theta_i) = \left(y_i - V(h_i^t, z^t; \theta_i)\right)^2 \quad \text{with} \quad y_i = r_i^t + \gamma V(h_i^t, z^t; \theta_i)$$

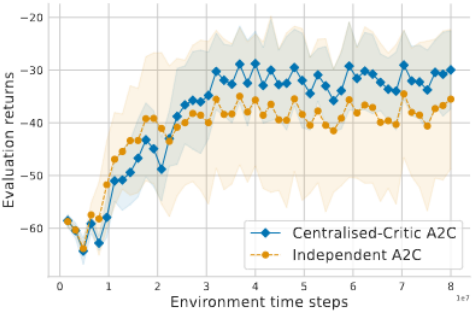
Algorithm 20 Centralized A2C with synchronous environments

Initialize n actor networks with random parameters $\phi_1, \ldots, \phi_n$
Initialize n critic networks with random parameters $\theta_1, \ldots, \theta_n$
Initialize K parallel environments
for time step $t = 0 \ldots$ **do**
    Batch of observations for each agent and environment: $\begin{bmatrix} o_1^{t,1} \ldots o_1^{t,K} \\ \ddots \\ o_n^{t,1} \ldots o_n^{t,K} \end{bmatrix}$
    Batch of centralized information for each environment: $\begin{bmatrix} z^{t,1} \ldots z^{t,K} \end{bmatrix}$
    Sample actions $\begin{bmatrix} a_1^{t,1} \ldots a_1^{t,K} \\ \ddots \\ a_n^{t,1} \ldots a_n^{t,K} \end{bmatrix} \sim \pi(\cdot \mid h_1^t; \phi_1), \ldots, \pi(\cdot \mid h_n^t; \phi_n)$
    Apply actions; collect rewards $\begin{bmatrix} r_1^{t,1} \ldots r_1^{t,K} \\ \ddots \\ r_n^{t,1} \ldots r_n^{t,K} \end{bmatrix}$, observations $\begin{bmatrix} o_1^{t+1,1} \ldots o_1^{t+1,K} \\ \ddots \\ o_n^{t+1,1} \ldots o_n^{t+1,K} \end{bmatrix}$, and centralized information $\begin{bmatrix} z^{t+1,1} \ldots z^{t+1,K} \end{bmatrix}$
    for agent $i = 1, \ldots, n$ **do**
        if $s^{t+1,k}$ is terminal **then**
            $Adv(h_i^{t,k}, z^{t,k}, a_i^{t,k}) \leftarrow r_i^{t,k} - V(h_i^{t,k}, z^{t,k}; \theta_i)$
            Critic target $y_i^{t,k} \leftarrow r_i^{t,k}$
        else
            $Adv(h_i^{t,k}, z^{t,k}, a_i^{t,k}) \leftarrow r_i^{t,k} + \gamma V(h_i^{t+1,k}, z^{t+1,k}; \theta_i) - V(h_i^{t,k}, z^{t,k}; \theta_i)$
            Critic target $y_i^{t,k} \leftarrow r_i^{t,k} + \gamma V(h_i^{t+1,k}, z^{t+1,k}; \theta_i)$
        Actor loss $\mathcal{L}(\phi_i) \leftarrow \frac{1}{K} \sum_{k=1}^{K} Adv(h_i^{t,k}, z^{t,k}, a_i^{t,k}) \log \pi(a_i^{t,k} \mid h_i^{t,k}; \phi_i)$
        Critic loss $\mathcal{L}(\theta_i) \leftarrow \frac{1}{K} \sum_{k=1}^{K} \left(y_i^{t,k} - V(h_i^{t,k}, z^{t,k}; \theta_i) \right)^2$
        Update parameters ϕ_i by minimizing the actor loss $\mathcal{L}(\phi_i)$
        Update parameters θ_i by minimizing the critic loss $\mathcal{L}(\theta_i)$

Empirical Results

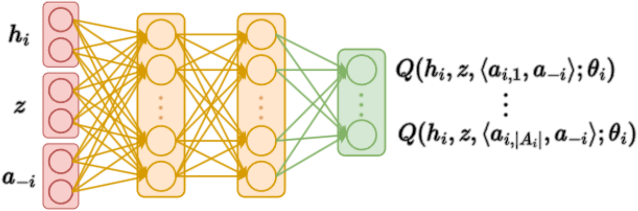


(a) Speaker-listener game



(b) Training curves

Centralized Action-Value Critics



Action-Value Loss:

$$\mathcal{L}(\theta_i) = \left(y_i - Q(h_i^t, z^t, a^t; \theta_i)\right)^2 \quad \text{with} \quad y_i = r_i^t + \gamma Q(h_i^{t+1}, z^{t+1}, a^{t+1}; \theta_i)$$

Today

- Intro
- Solving Joint Policies
- Independent -DQN, -REINFORCE, -A2C
- Centralized Critics
- **Value Decomposition (VDN, QMIX)**

Value Decomposition
Motivation

- Centralized action-value are difficult to learn
- Agents cannot select actions decentralized
- Greedy actions are costly
- Don’t rely on additional policy nets

Value Decomposition

Insights

- $Q_i(s, a_i) \approx Q(s, a)$ intractable
- Agent interaction as a sparse coordination graph
- Decompose the centralized action-value function (if $\mathcal{R}_i = \mathcal{R}_j,\ i, j \in I$)

$$Q(h^t, z^t, a^t; \theta) = \mathbb{E}\left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} r^{\tau} \mid h^t, z^t, a^t\right],$$

into simpler functions

Individual-Global-Max (IGM) Property

Idea:
Greedy actions w.r.t to the centralized function $\iff$
joint actions composed of individual greedy actions

Individual-Global-Max (IGM) Property

First, formally define sets of greedy actions:

$$A^*(h, z; \theta) = \arg\max_{a \in A} Q(h, z, a; \theta)$$

$$A_i^*(h_i; \theta_i) = \arg\max_{a_i \in A_i} Q(h_i, a_i; \theta_i)$$

Definition 3. *IGM is satisfied if the following holds for all full histories $\hat{h}$, joint-observation histories $h = \sigma(\hat{h})$, individual observation histories $h_i = \sigma_i(\hat{h})$ and centralized information z.*

$$\forall a = (a_1, \ldots, a_n) \in A : a \in A^*(h, z; \theta) \iff \forall i \in I : a_i \in A_i^*(h_i; \theta_i)$$

Linear Value Decomposition

Assumption. *Decompose $r^t = \bar{r}^t_1, \ldots, \bar{r}^t_n$, where $\bar{r}^t_i$ is utility of agent i at t. $\bar{r}^t_i$ is obtained by decomposition*

Definition 4. (Linear Value Decomposition)

$$\begin{aligned}
Q(h^t, z^t, a^t; \theta) &= \mathbb{E}_{\hat{h}^t \sim \Pr(\cdot|\pi)}\left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} r^\tau \;\middle|\; h^t = \sigma(\hat{h}^t), z^t, a^t\right] \\
&= \mathbb{E}_{\hat{h}^t \sim \Pr(\cdot|\pi)}\left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} \left(\sum_{i \in I} \bar{r}^\tau_i\right) \;\middle|\; h^t, z^t, a^t\right] \\
&= \sum_{i \in I} \mathbb{E}_{\hat{h}^t \sim \Pr(\cdot|\pi)}\left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} \bar{r}^\tau_i \;\middle|\; h^t, z^t, a^t\right] \\
&= \sum_{i \in I} Q(h^t_i, a^t_i; \theta_i)
\end{aligned}$$

Proof

Value Decomposition Networks (VDN)

Given a buffer $\mathcal{D}$, the loss for a VDN is computed over a batch $\mathcal{B}$:

$$\mathcal{L}(\theta) = \frac{1}{B} \sum_{(h^t, a^t, r^t, h^{t+1}) \in \mathcal{B}} \left(r^t + \gamma \max_{a \in A} Q(h^{t+1}, a; \bar{\theta}) - Q(h^t, a^t; \theta) \right)^2$$

with

$$Q(h^t, a^t; \theta) = \sum_{i \in I} Q(h_i^t, a_i^t; \theta_i) \text{ and}$$

$$\max_{a \in A} Q(h^{t+1}, a; \bar{\theta}) = \sum_{i \in I} \max_{a_i \in A_i} Q(h_i^{t+1}, a_i; \bar{\theta}_i).$$

Algorithm 21 Value decomposition networks (VDN)

Initialize n utility networks with random parameters $\theta_1, \ldots, \theta_n$
Initialize n target networks with parameters $\overline{\theta}_1 = \theta_1, \ldots, \overline{\theta}_n = \theta_n$
Initialize a shared replay buffer D
for time step $t = 0, 1, 2, \ldots$ **do**
    Collect current observations $o_1^t, \ldots, o_n^t$
    for agent $i = 1, \ldots, n$ **do**
        With probability ϵ: choose random action a_i^t
        Otherwise: choose $a_i^t \in \arg\max_{a_i} Q(h_i^t, a_i; \theta_i)$
    Apply actions; collect shared reward r^t and next observations $o_1^{t+1}, \ldots, o_n^{t+1}$
    Store transition (h^t, a^t, r^t, h^{t+1}) in shared replay buffer D
    Sample mini-batch of B transitions (h^k, a^k, r^k, h^{k+1}) from D
    if s^{k+1} is terminal **then**
        Targets $y^k \leftarrow r^k$
    else
        Targets $y^k \leftarrow r^k + \gamma \sum_{i \in I} \max_{a_i' \in A_i} Q(h_i^{k+1}, a_i'; \overline{\theta}_i)$
    Loss $\mathcal{L}(\theta) \leftarrow \frac{1}{B} \sum_{k=1}^{B} \left(y^k - \sum_{i \in I} Q(h_i^k, a_i^k; \theta_i) \right)^2$
    Update parameters θ by minimizing the loss $\mathcal{L}(\theta)$
    In a set interval, update target network parameters $\overline{\theta}_i$ for each agent i

QMIX

- Linear decomposition has drawbacks
- Use a monotonic decomposition
- QMIX uses a mixing network

$$Q(h, z, a; \theta) = f_{mix}(Q(h_1, a_1; \theta_1), \ldots, Q(h_n, a_n; \theta_n); \theta_{mix})$$

- Loss:

$$\mathcal{L}(\theta) = \frac{1}{B} \sum_{(h^t, z^t, a^t, r^t, h^{t+1}, z^{t+1}) \in B} \left(r^t + \gamma \max_{a \in A} Q(h^{t+1}, z^{t+1}, a; \bar{\theta}) - Q(h^t, z^t, a^t; \theta) \right)^2$$

Proof?

Algorithm 22 QMIX

Initialize n utility networks with random parameters $\theta_1, \ldots, \theta_n$
Initialize n target networks with parameters $\bar{\theta}_1 = \theta_1, \ldots, \bar{\theta}_n = \theta_n$
Initialize hypernetwork with random parameters θ_{hyper}
Initialize a shared replay buffer D
for time step $t = 0, 1, 2, \ldots$ **do**
  Collect current centralized information z^t and observations $o_1^t, \ldots, o_n^t$
  for agent $i = 1, \ldots, n$ **do**
    With probability ϵ: choose random action a_i^t
    Otherwise: choose $a_i^t \in \arg\max_{a_i} Q(h_i^t, a_i; \theta_i)$
  Apply actions; collect shared reward r^t, next centralized information z^{t+1} and observations $o_1^{t+1}, \ldots, o_n^{t+1}$
  Store transition $(h^t, z^t, a^t, r^t, h^{t+1}, z^{t+1})$ in shared replay buffer D
  Sample mini-batch of B transitions $(h^k, z^k, a^k, r^k, h^{k+1}, z^{k+1})$ from D
  if s^{k+1} is terminal **then**
    Targets $y^k \leftarrow r^k$
  else
    Mixing parameters $\theta_{\text{mix}}^{k+1} \leftarrow f_{\text{hyper}}(z^{k+1}; \theta_{\text{hyper}})$
    Targets $y^k \leftarrow r^k + \gamma f_{\text{mix}} \begin{pmatrix} \max_{a_1'} Q(h_1^{k+1}, a_1'; \bar{\theta}_1), & \\ \ddots & ; \theta_{\text{mix}}^{k+1} \\ \max_{a_n'} Q(h_n^{k+1}, a_n'; \bar{\theta}_n) & \end{pmatrix}$
  Mixing parameters $\theta_{\text{mix}}^{k} \leftarrow f_{\text{hyper}}(z^{k}; \theta_{\text{hyper}})$
  Value estimates $Q(h^k, z^k, a^k; \theta) \leftarrow f_{\text{mix}} \left(Q(h_1^k, a_1^k; \theta_1), \ldots, Q(h_n^k, a_n^k; \theta_n); \theta_{\text{mix}}^k \right)$
  Loss $\mathcal{L}(\theta) \leftarrow \frac{1}{B} \sum_{k=1}^{B} \left(y^k - Q(h^k, z^k, a^k; \theta) \right)^2$
  Update parameters θ by minimizing the loss $\mathcal{L}(\theta)$
  In a set interval, update target network parameters $\bar{\theta}_i$ for each agent i