Cooperative Exploration for Multi-Agent Deep Reinforcement Learning

ILLINOIS

Iou-Jen Liu, Unnat Jain, Raymond A. Yeh, Alexander G. Schwing
University of Illinois at Urbana-Champaign

ICML 2021

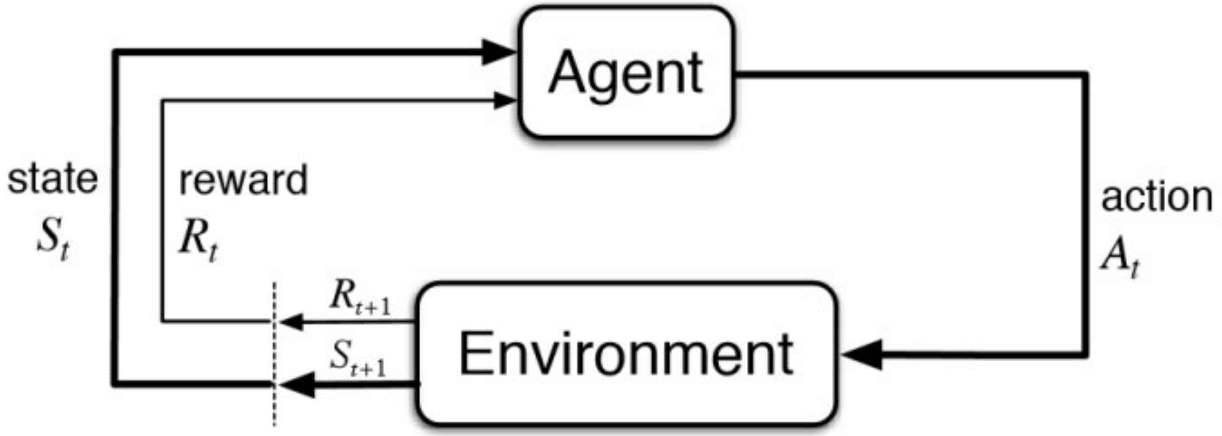
ICML
International Conference On Machine Learning

Introduction

Multi-agent systems are everywhere

Goal of RL:

- Learn a policy that will maximize the expected reward



Needs access to a reward function

Sparse Reward

Reward is provided only when a task is completed

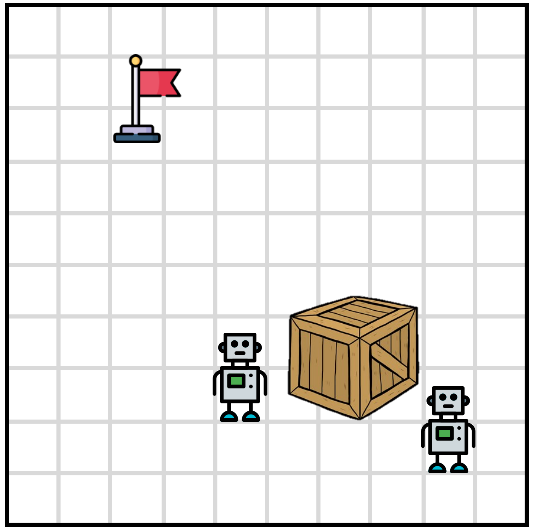
- Only define the criteria for completing a task
- Difficult policy optimization
- **Requires efficient exploration strategy**

Challenges of Multi-Agent Exploration with Sparse Reward

Challenge 1: Identify states that are worth exploring

- States grow exponentially with the number of agents
- Infeasible to explore all states

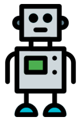
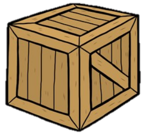
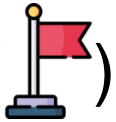
Challenges of Multi-Agent Exploration with Sparse Reward



Challenge 1: Identify states that are worth exploring

- States grow exponentially with the number of agents
- Infeasible to explore all states

Example: push-box task

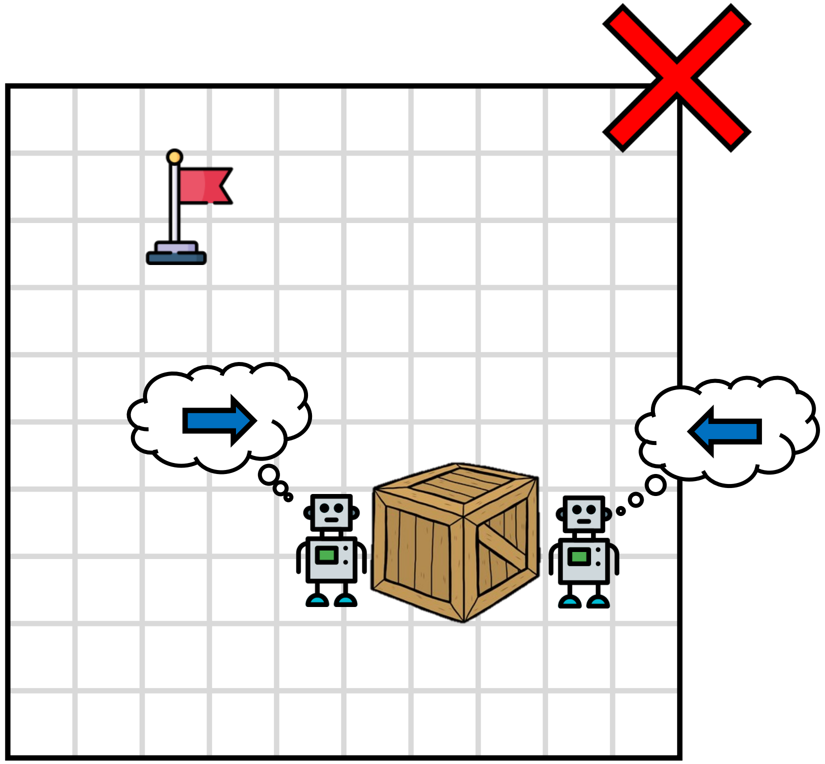
- Agents () push a heavy box () to a goal () in an $L \times L$ grid
- Only receive reward when the box is pushed to the goal
- State contains x, y location of the agents and the box
- Two agents: $(L^2)^{1+2}$ states to explore
- N agents: $(L^2)^{1+N}$ states to explore

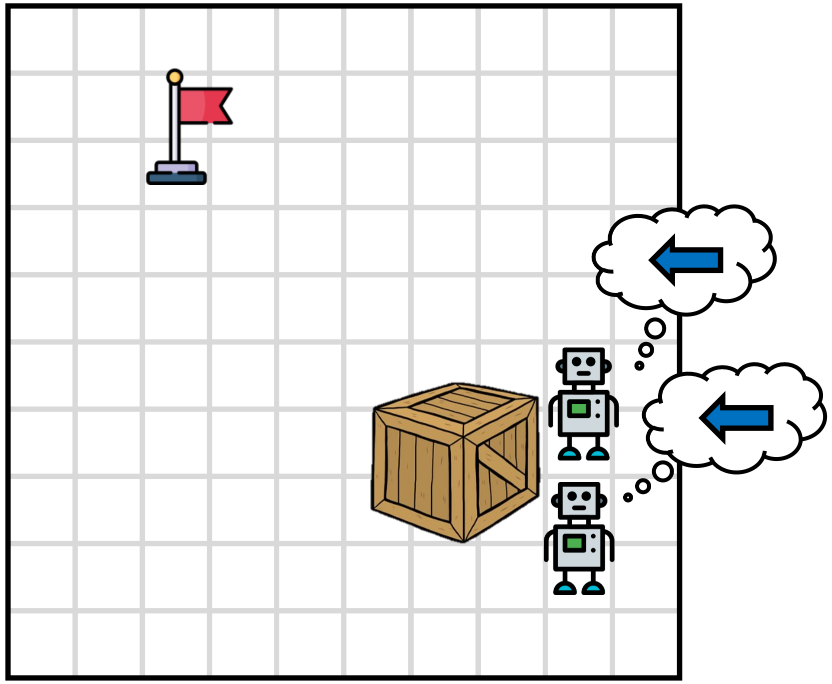
Challenges of Multi-Agent Exploration with Sparse Reward

Challenge 2: Coordinate agents' exploration efforts

- Uncoordinated exploration is inefficient

Example: push-box task



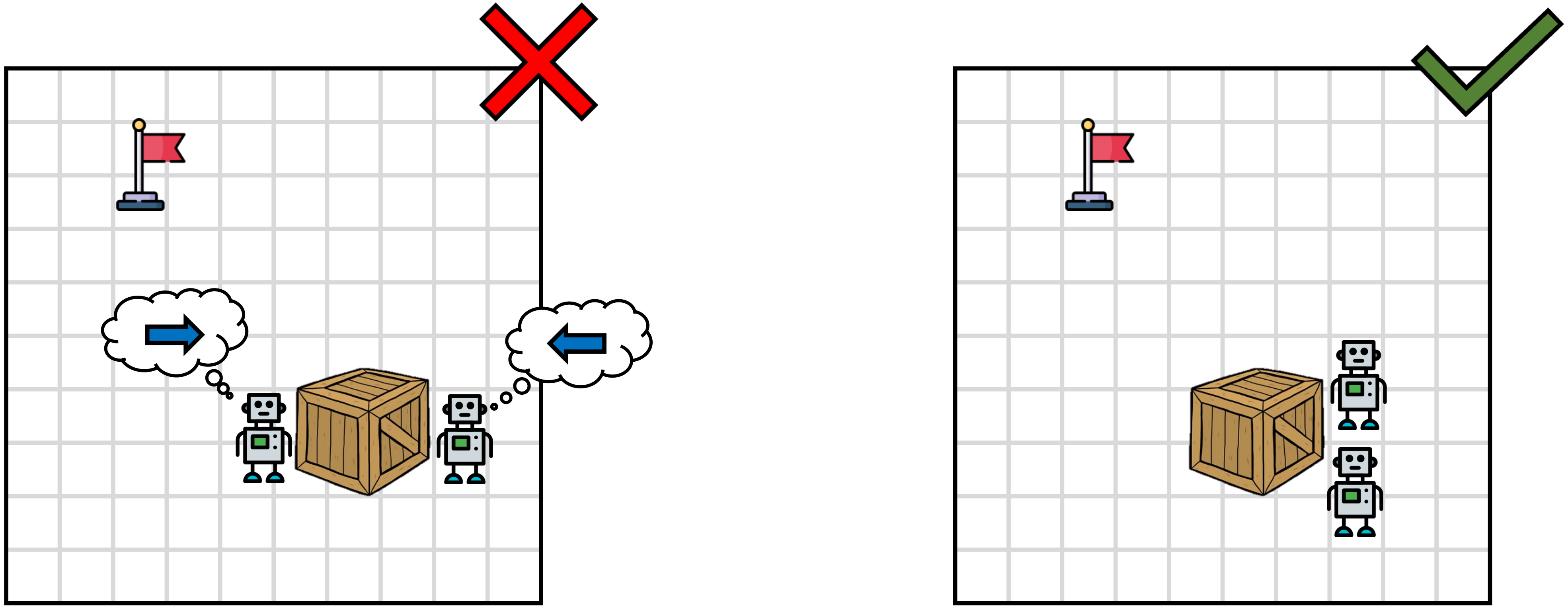


Challenges of Multi-Agent Exploration with Sparse Reward

Challenge 2: Coordinate agents' exploration efforts

- Uncoordinated exploration is inefficient

Example: push-box task



Cooperative Multi-Agent Exploration (CMAE)

Challenge 1: Identify states that are worth exploring

CMAE: Restricted space exploration

- Identify under-explored low-dimensional restricted space
- Avoid exploring the exponentially-growing full state space

Challenge 2: Coordinate agents' exploration efforts

CMAE: Shared goal exploration

- Agents share a common goal while exploring
- Enable coordinated multi-agent exploration

Cooperative Multi-Agent Exploration (CMAE)

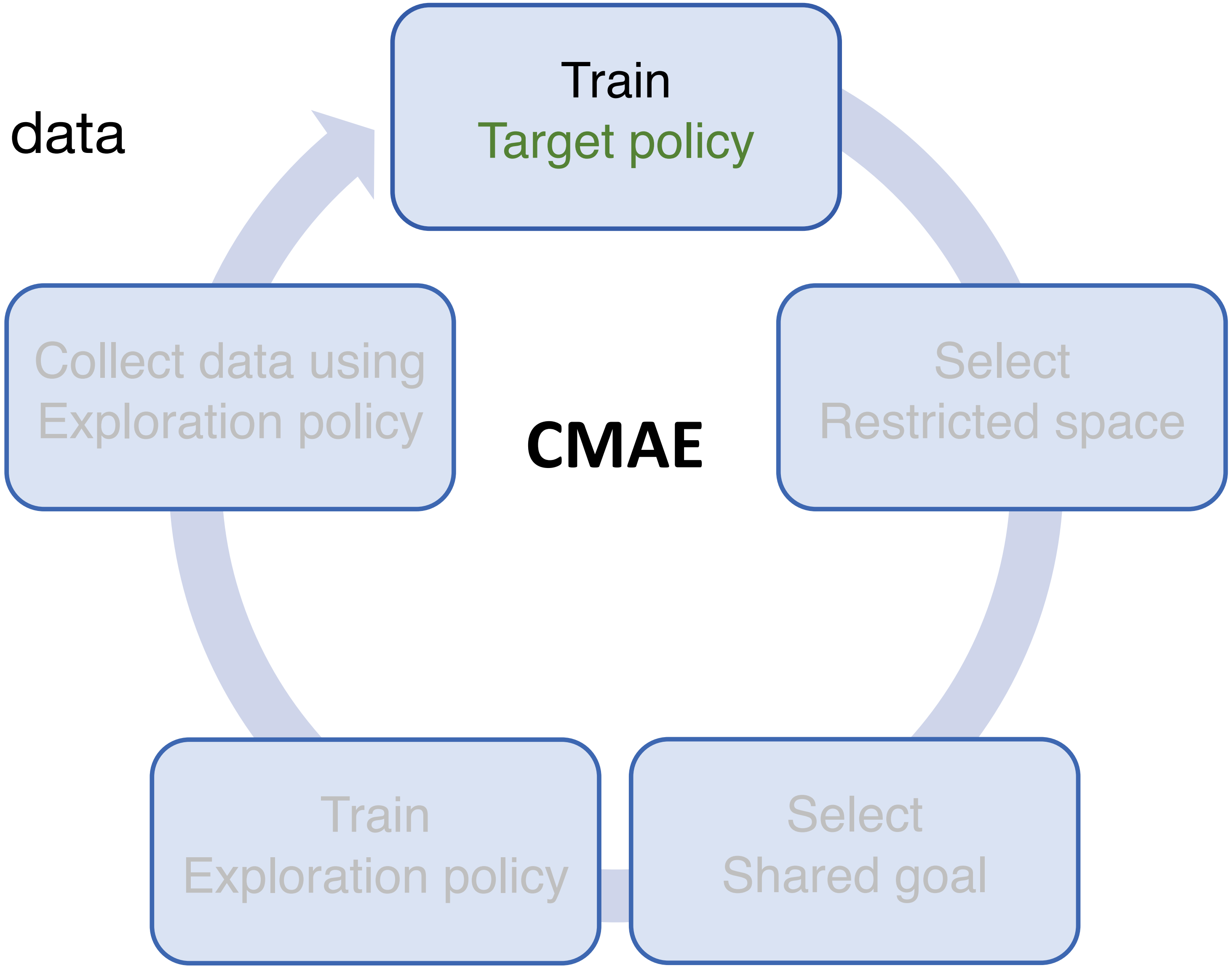
Policy Decoupling

- Exploration policy: Collect data from rarely visited states
- Target policy: Maximize external reward

Cooperative Multi-Agent Exploration (CMAE)

Policy Decoupling

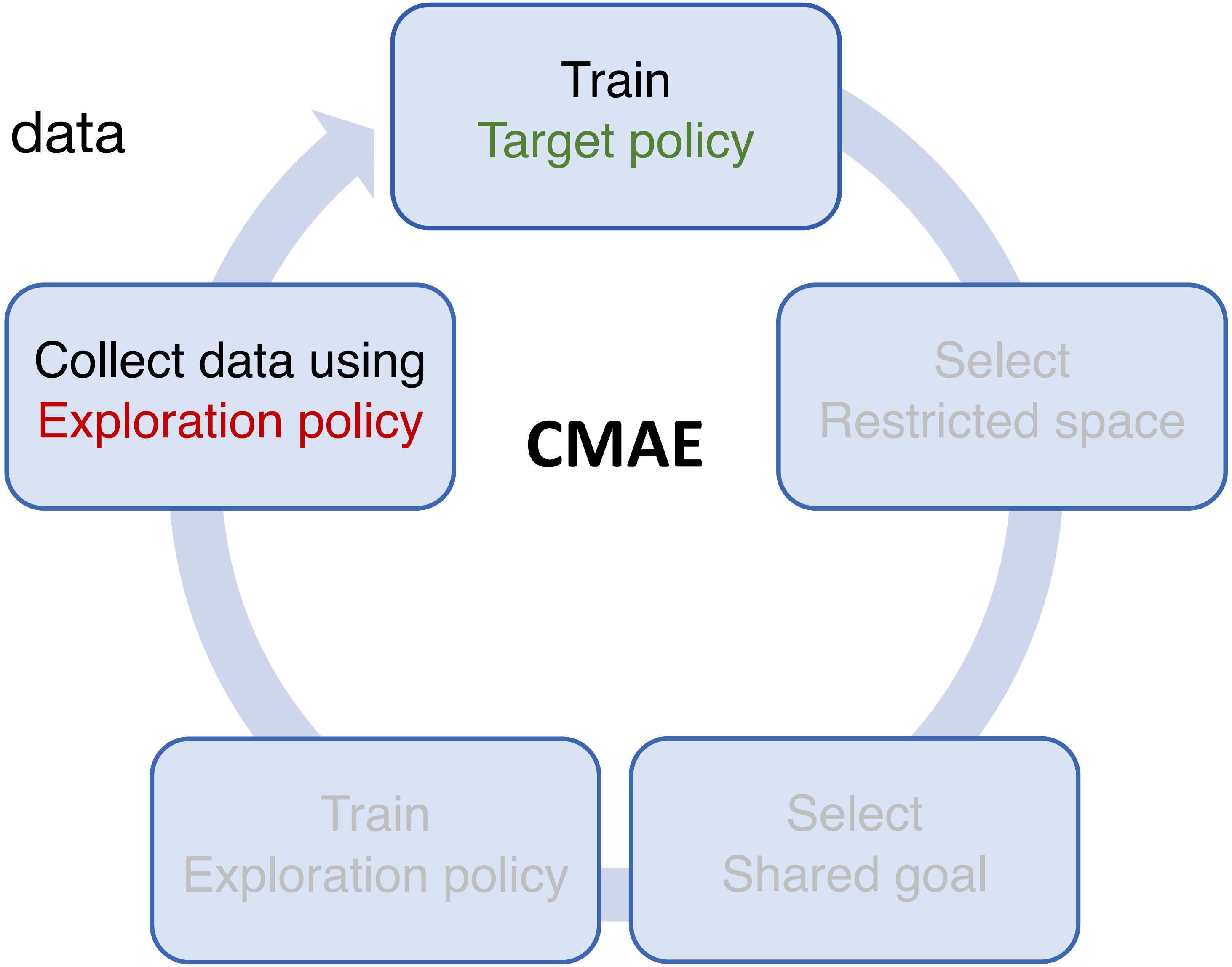
- Exploration policy: Collect data from rarely visited states
- Target policy: Maximize external reward



Cooperative Multi-Agent Exploration (CMAE)

Policy Decoupling

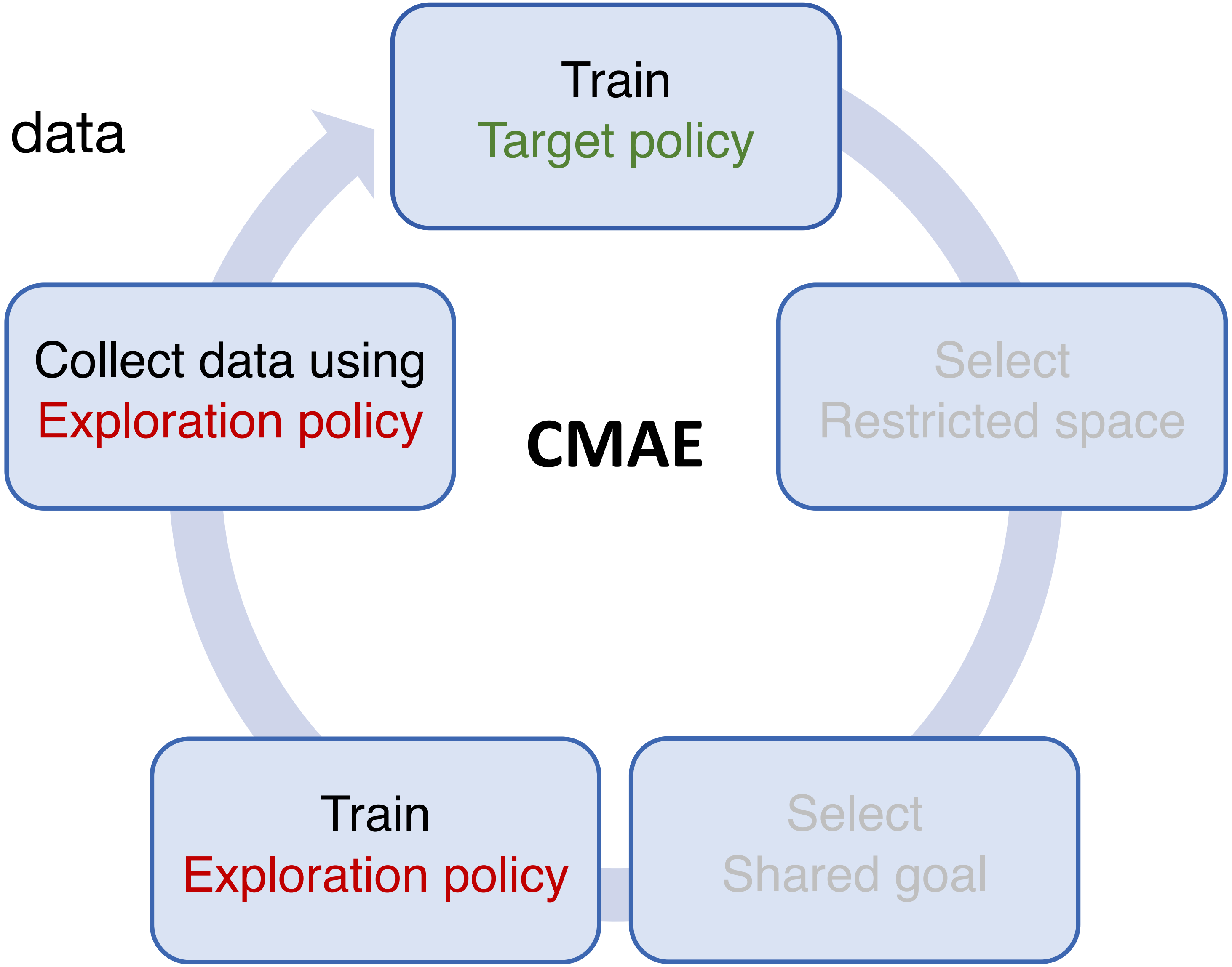
- Exploration policy: Collect data from rarely visited states
- Target policy: Maximize external reward



Cooperative Multi-Agent Exploration (CMAE)

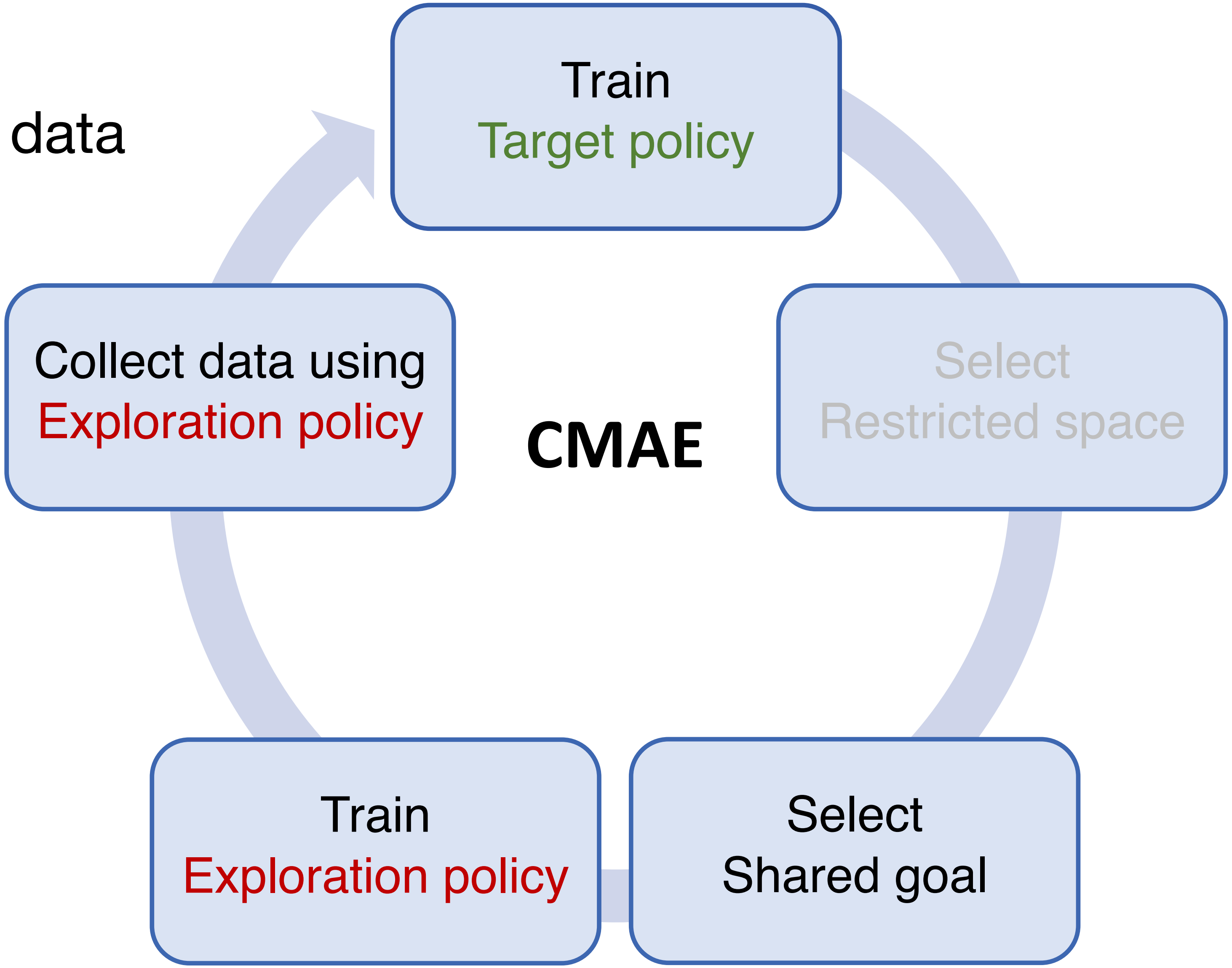
Policy Decoupling

- Exploration policy: Collect data from rarely visited states
- Target policy: Maximize external reward



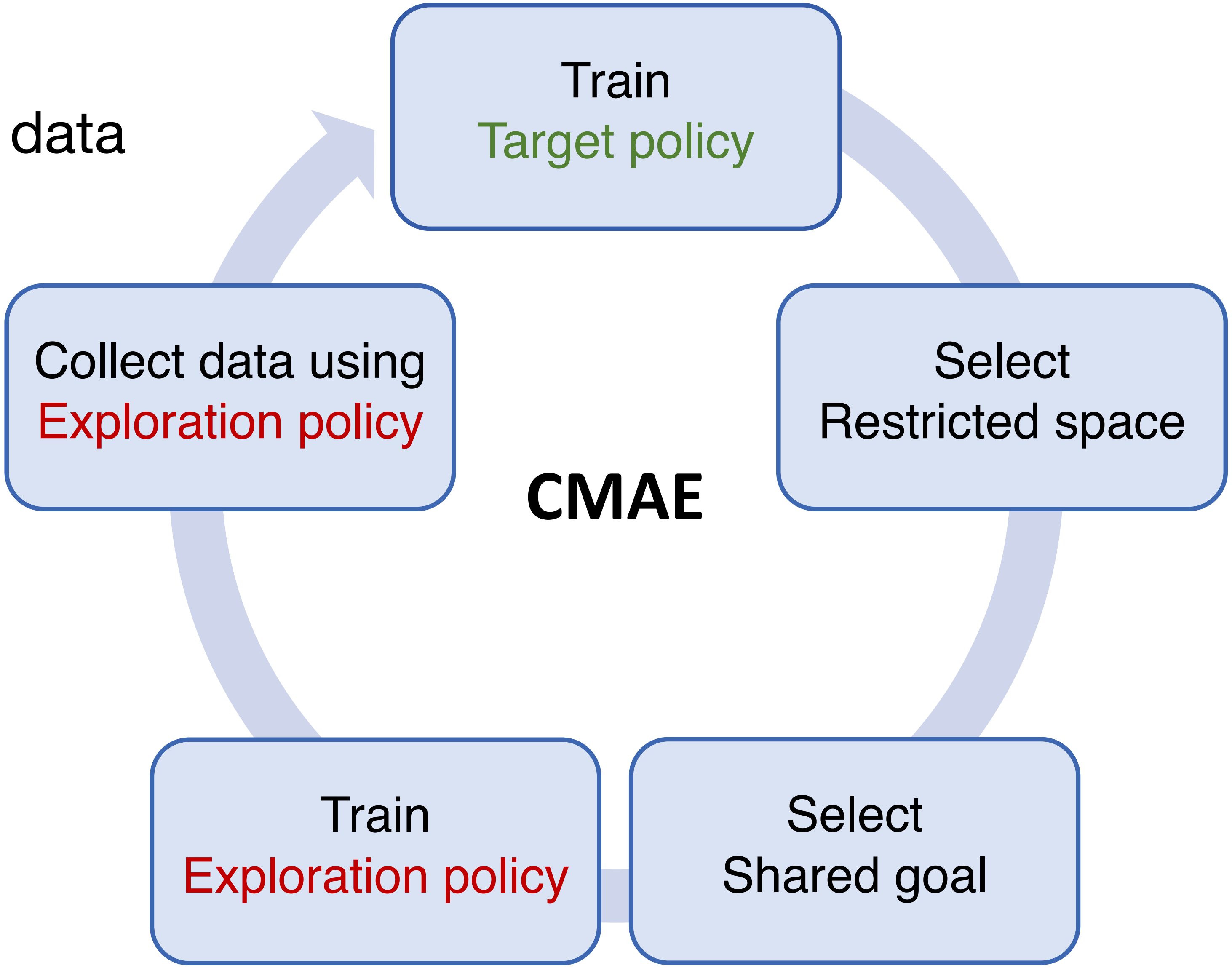
Policy Decoupling

- Exploration policy: Collect data from rarely visited states
- Target policy: Maximize external reward



Policy Decoupling

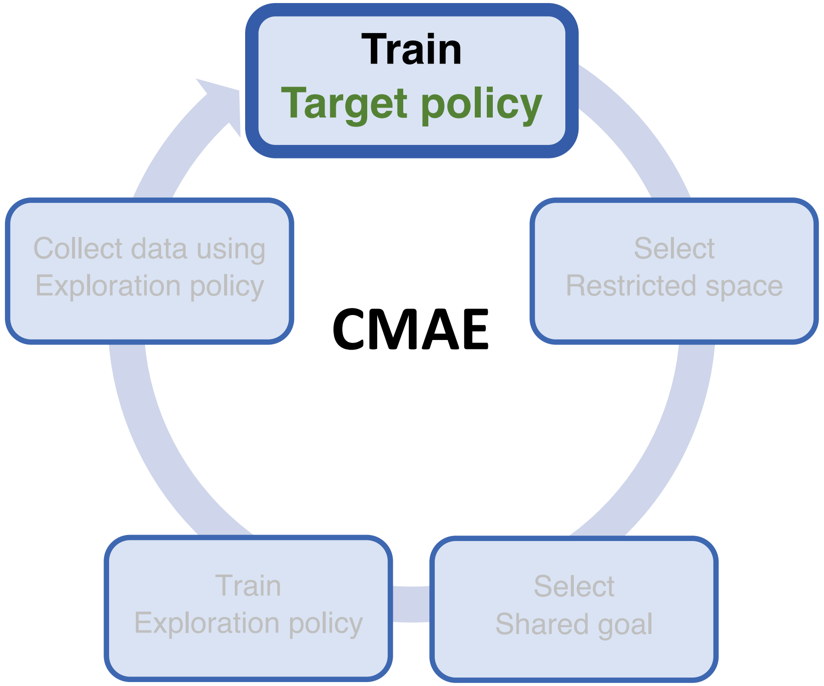
- Exploration policy: Collect data from rarely visited states
- Target policy: Maximize external reward



Train Target Policy and Data Collection

Maximize the external environment reward

- Use off-policy algorithms (e.g., MADDPG, QMIX)
- Use previously collected data



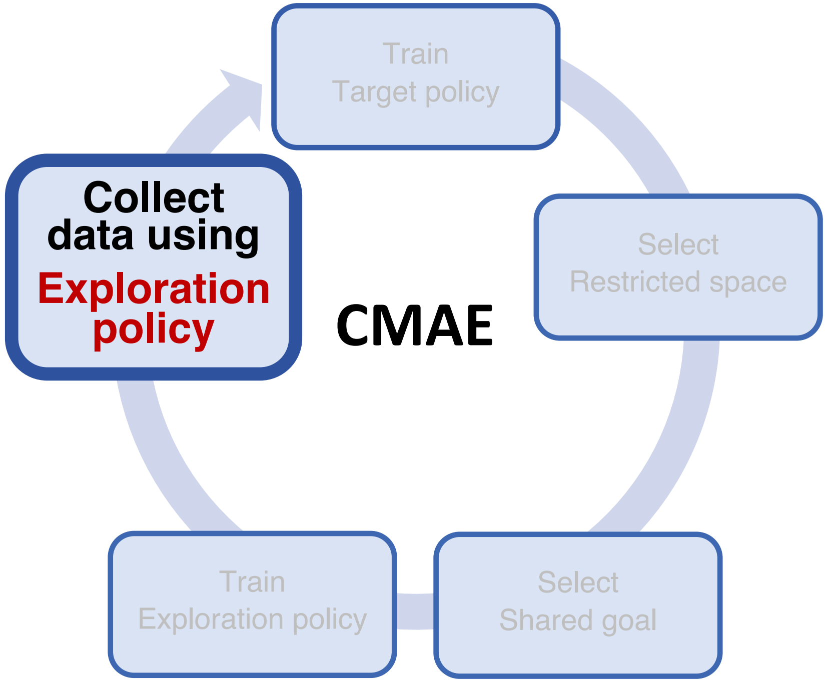
Train Target Policy and Data Collection

Maximize the external environment reward

- Use off-policy algorithms (e.g., MADDPG, QMIX)
- Use previously collected data

Exploration policy interacts with environment

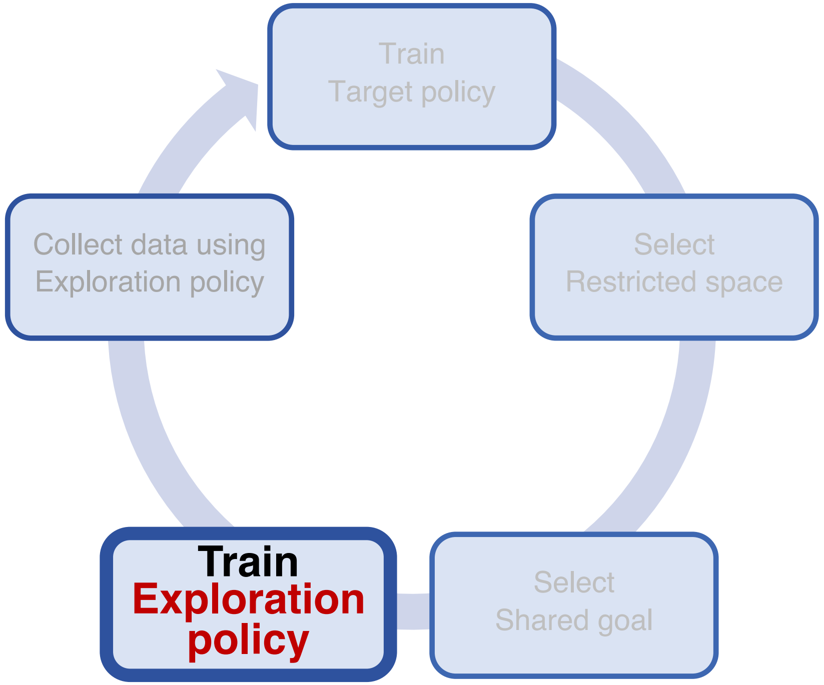
- Collected data is used to train the target policy
- The data contains under-explored states



Train Exploration Policy

Exploration policy is trained to reach a selected goal

- Reshape reward in the replay buffer
- Positive reward when reaching a shared goal

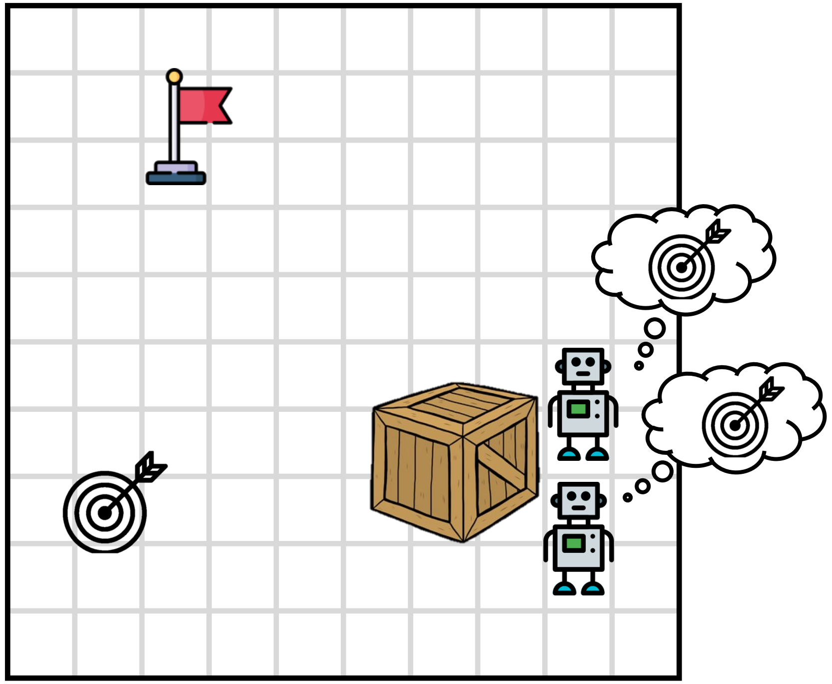


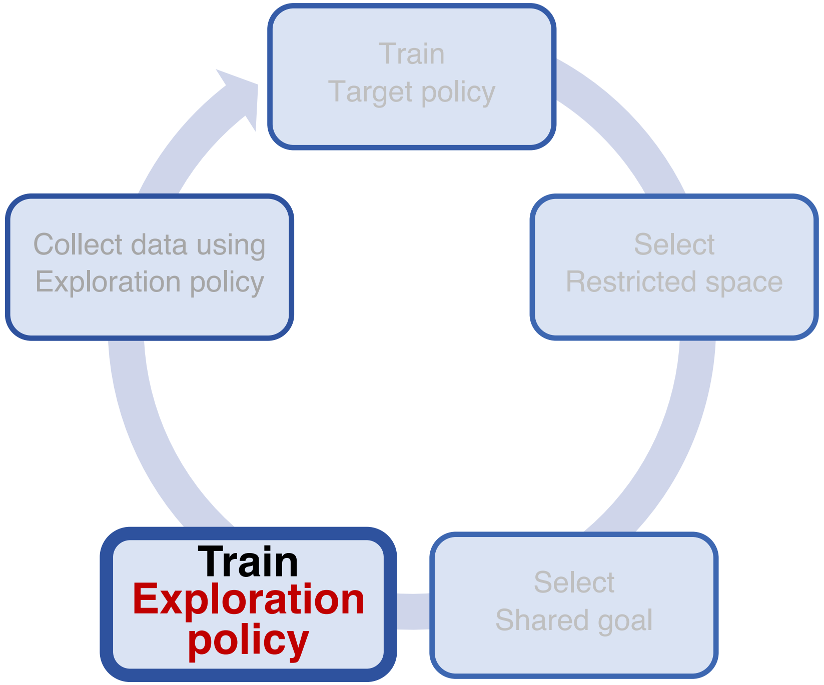
Train Exploration Policy

Exploration policy is trained to reach a selected goal (🎯)

- Reshape reward in the replay buffer
- Positive reward when reaching a shared goal

Example: push-box task



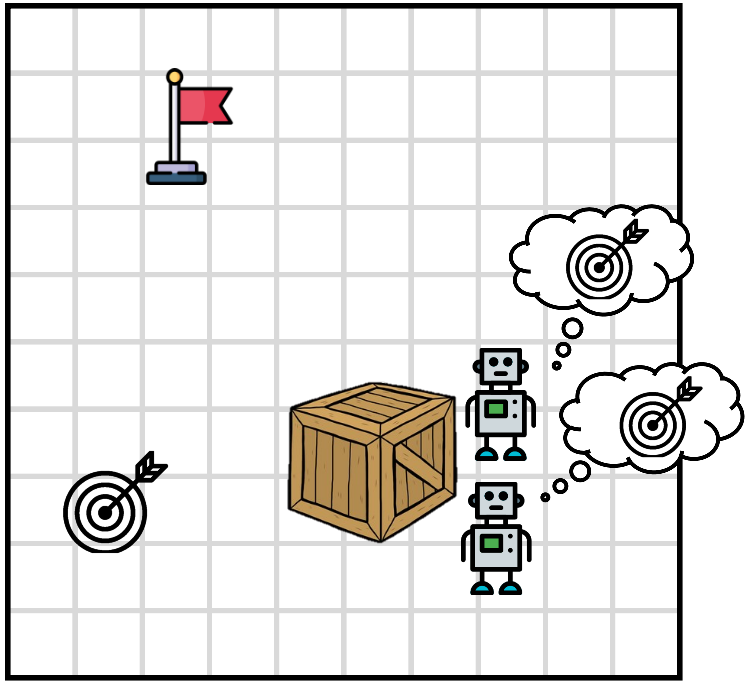


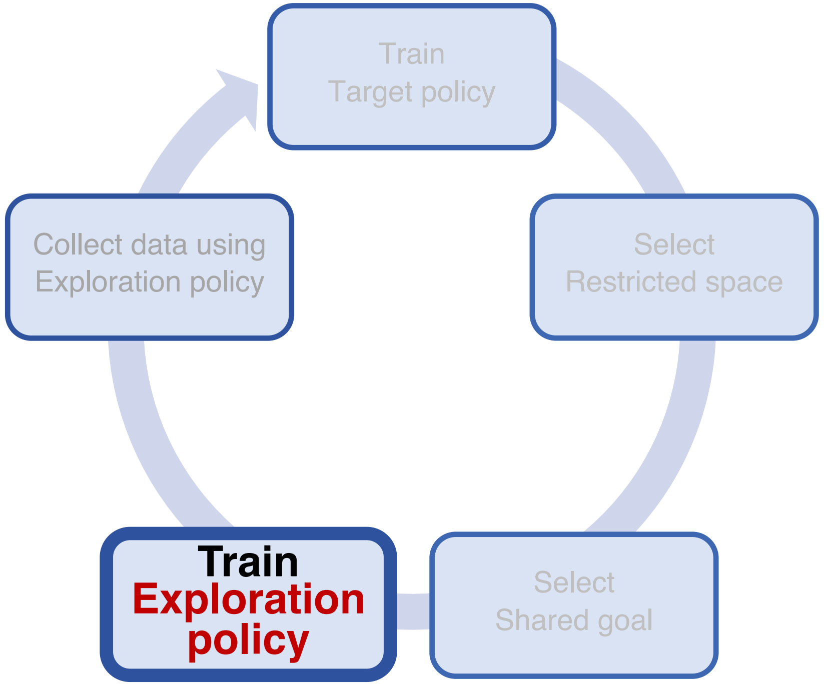
Train Exploration Policy

Exploration policy is trained to reach a selected goal (🎯)

- Reshape reward in the replay buffer
- Positive reward when reaching a shared goal

Example: push-box task



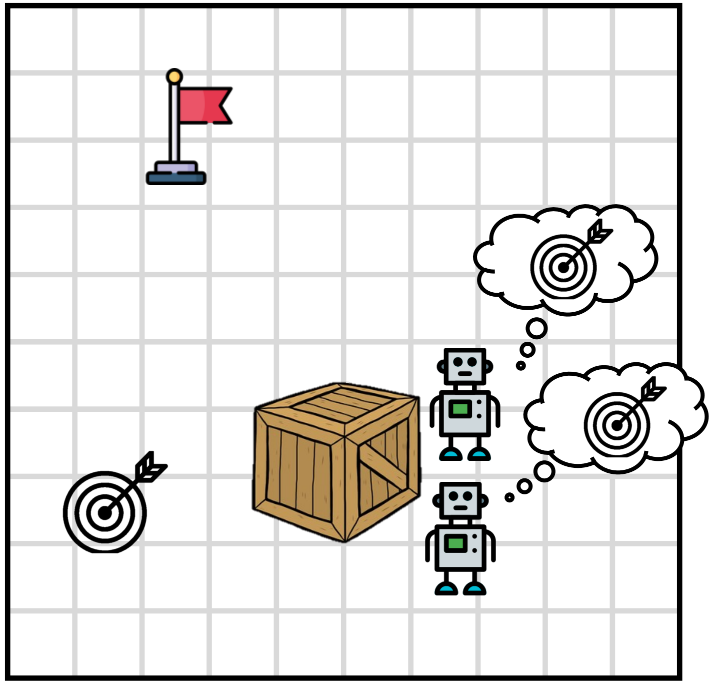


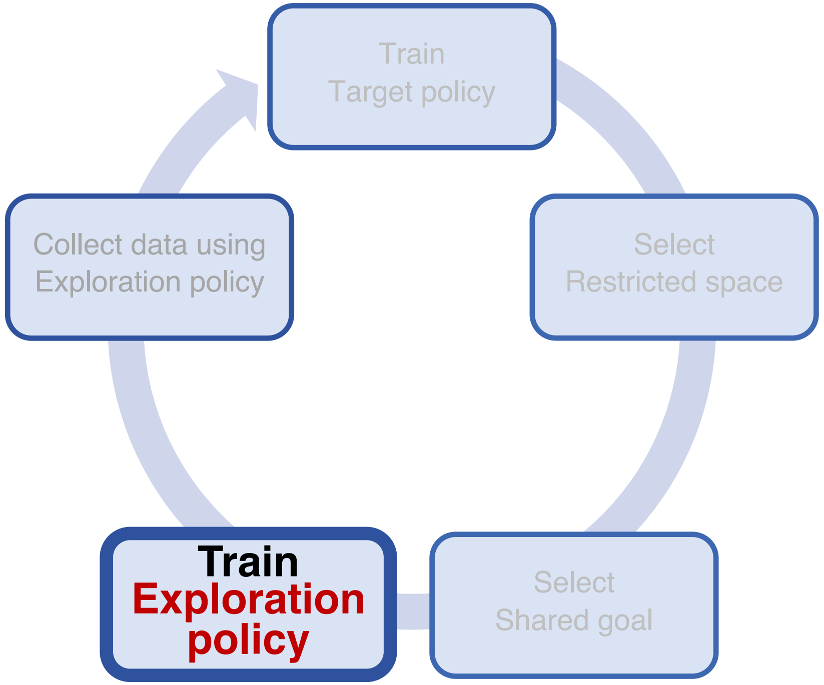
Train Exploration Policy

Exploration policy is trained to reach a selected goal ( )

- Reshape reward in the replay buffer
- Positive reward when reaching a shared goal

Example: push-box task



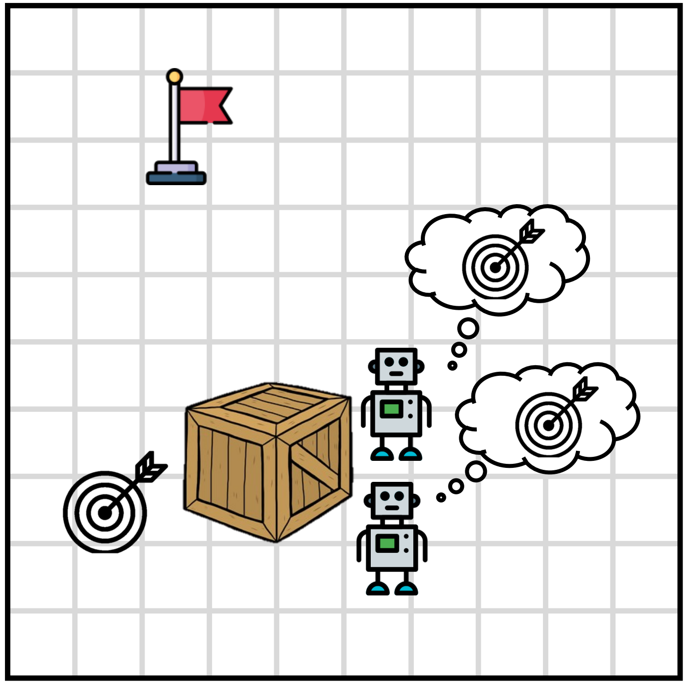


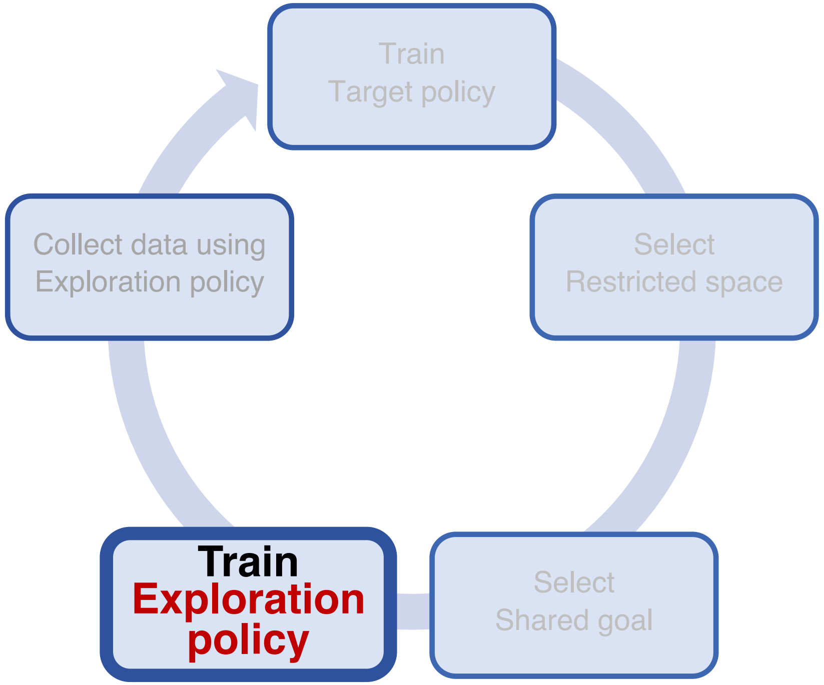
Train Exploration Policy

Exploration policy is trained to reach a selected goal (🎯)

- Reshape reward in the replay buffer
- Positive reward when reaching a shared goal

Example: push-box task



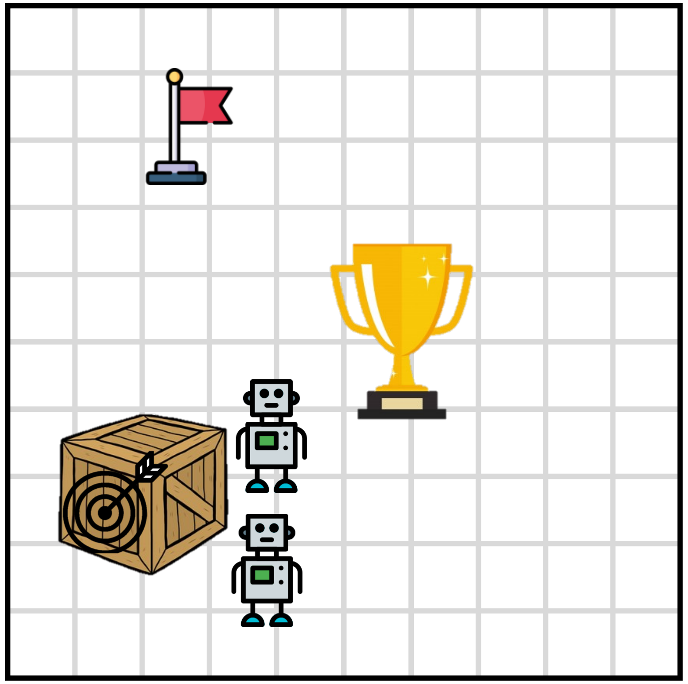


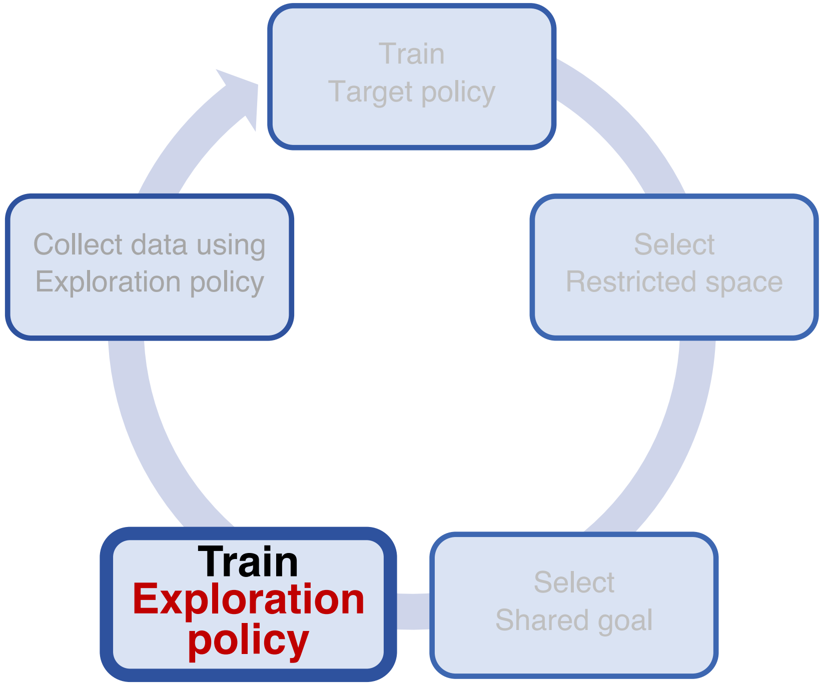
Train Exploration Policy

Exploration policy is trained to reach a selected goal ()

- Reshape reward in the replay buffer
- Positive reward when reaching a shared goal

Example: push-box task

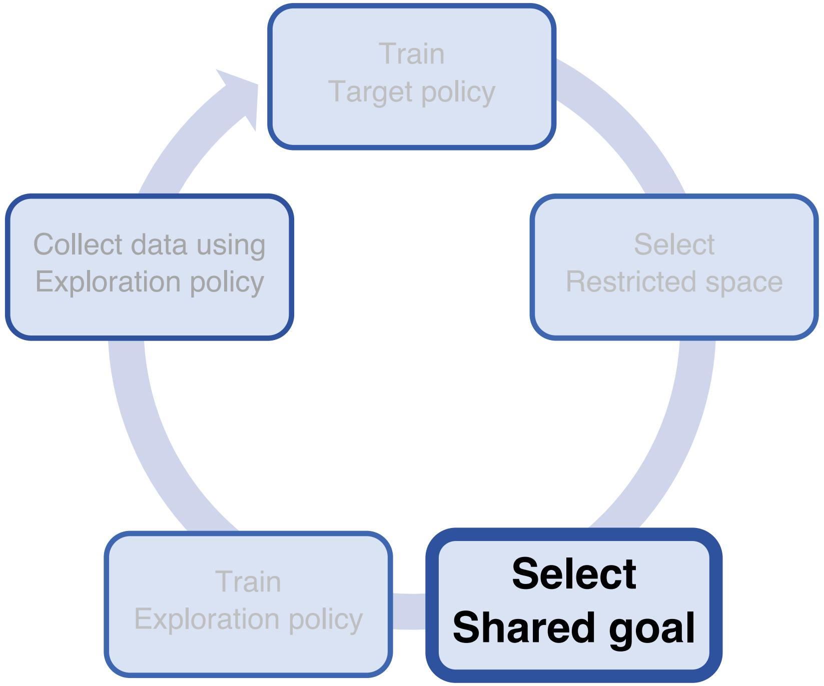




Select Shared Goal

How to select a shared goal?

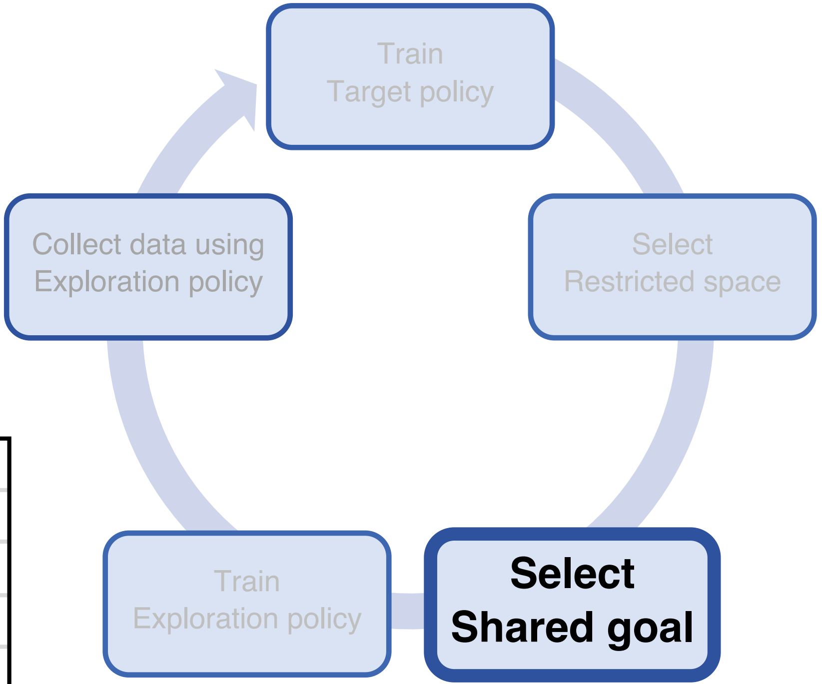
- Select a rarely visited state as shared goal
- Count in low-dimensional restricted space
- Avoid selecting goal from full state space, whose size grows exponentially



Select Shared Goal

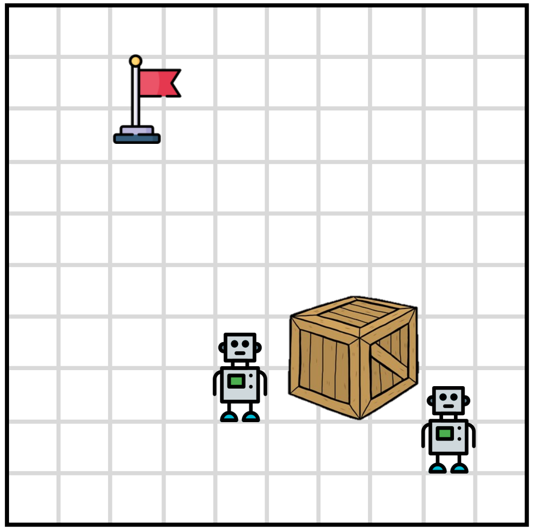
How to select a shared goal?

- Select a rarely visited state as shared goal
- Count in low-dimensional restricted space
- Avoid selecting goal from full state space, whose size grows exponentially



Example: 2-agent push-box

- $S_{\{\mathrm{box}_x, \mathrm{box}_y\}}$ contains box x, y
- Shared goal is a state with box in a rarely seen location

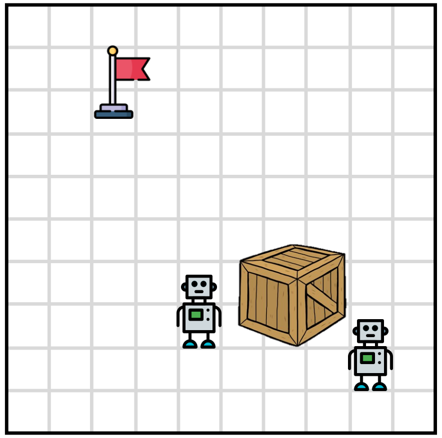


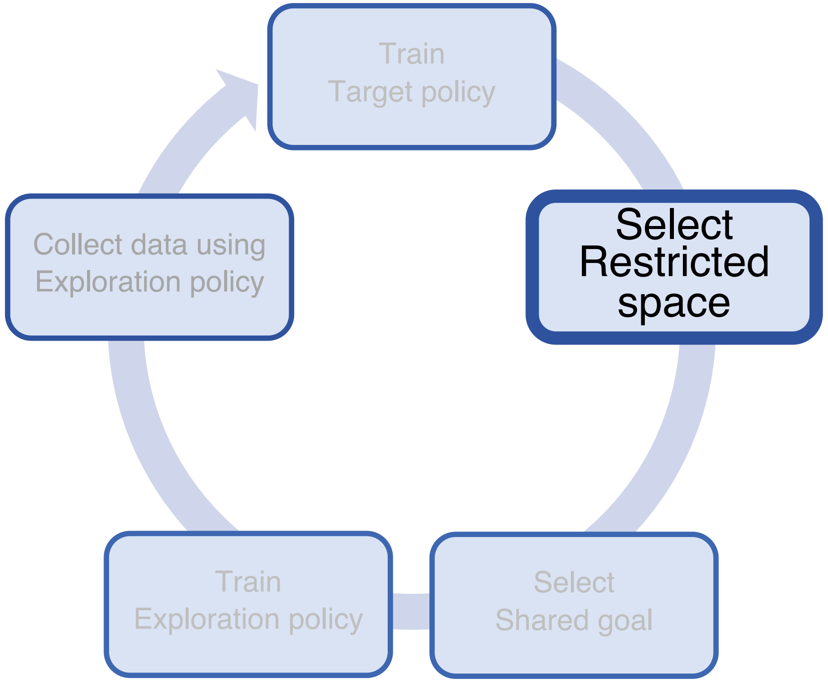
Restricted Space

- Reward function typically depends on a low-dimensional subspace of the state space

Example: N-agent push-box task in $L \times L$ grid

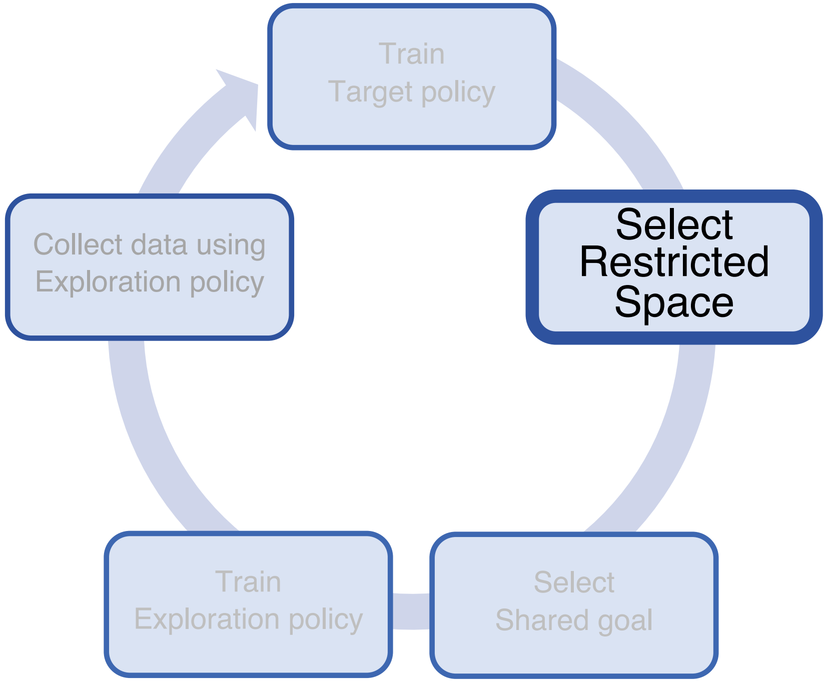
- Size of state space: $(L^2)^{1+N}$
- Reward function depends only on the box location, whose state space size is L^2





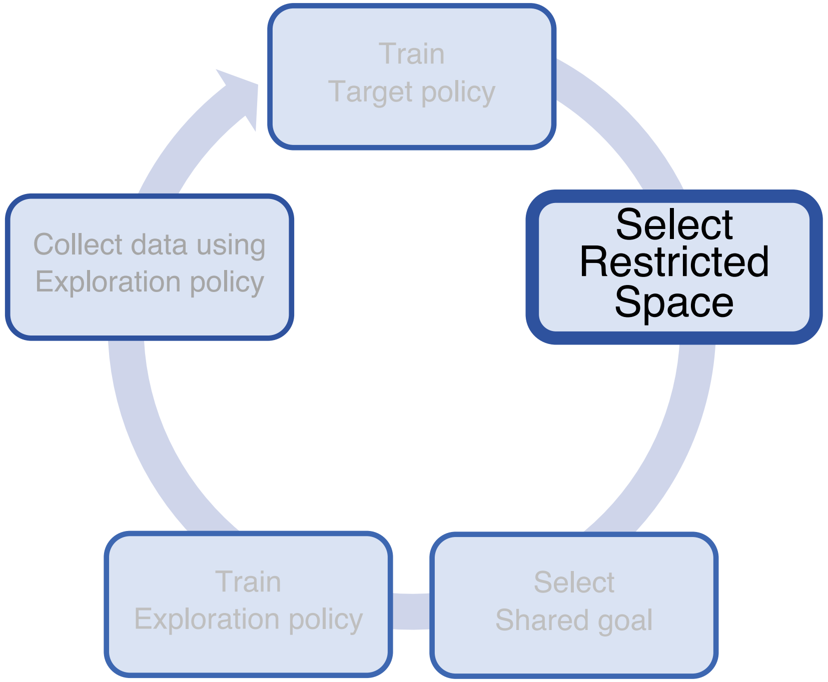
How to find an under-explored restricted space?

- Each restricted space S_k has a counter $\boldsymbol{c}_k$
- $\boldsymbol{c}_k$ tracks the number of times a state was visited
- Use $\boldsymbol{c}_k$ to compute distribution of state visitation
- Under-explored restricted space has smaller entropy



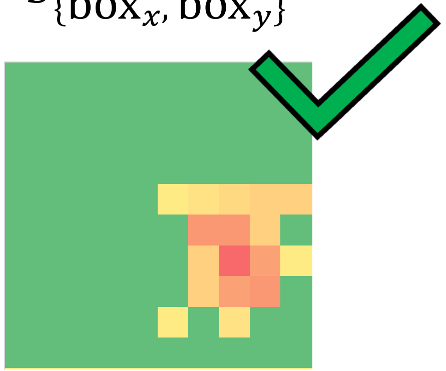
How to find an under-explored restricted space?

- Each restricted space S_k has a counter $\boldsymbol{c}_k$
- $\boldsymbol{c}_k$ tracks the number of times a state was visited
- Use $\boldsymbol{c}_k$ to compute distribution of state visitation
- Under-explored restricted space has smaller entropy



Example: 2-agent push-box

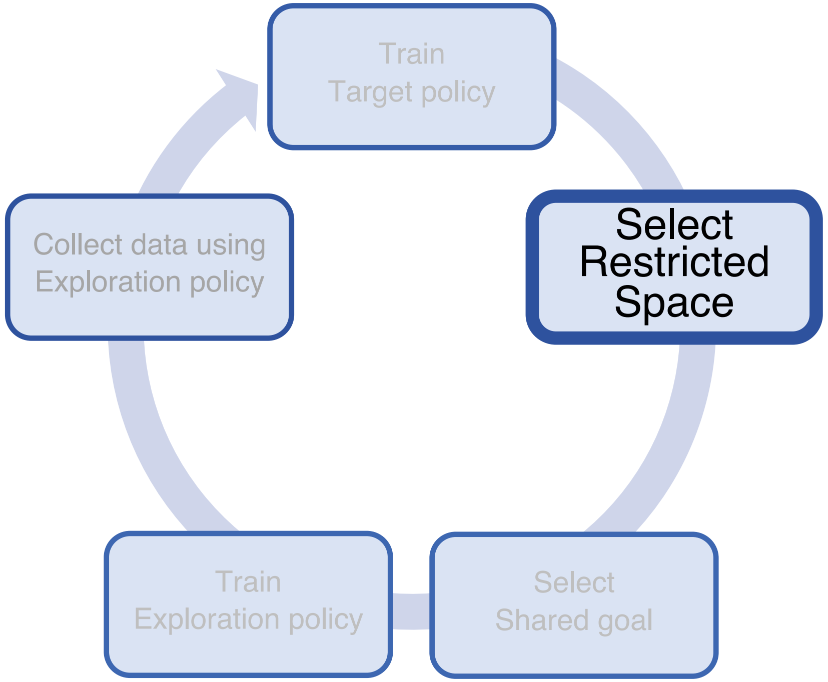
$S_{\{\mathrm{box}_x, \mathrm{box}_y\}}$



$S_{\{\mathrm{agent1}_x, \mathrm{agent1}_y\}}$

Space Tree

- Each node represents a restricted space
- Space tree is initialized with 1-dimensional restricted spaces



Select Restricted Space

Space Tree

- Each node represents a restricted space
- Space tree is initialized with 1-dimensional restricted spaces

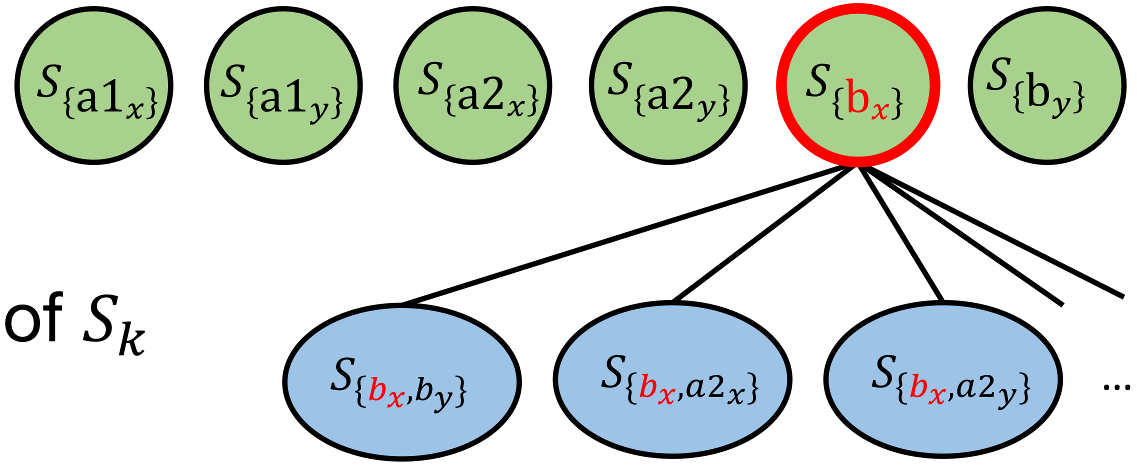
Space Tree Expansion

- Utility μ_k: negative normalized entropy of S_k
- Select restricted space S_k with high μ_k
- Add all restricted spaces of $(|k| + 1)$-dimension which contain S_k as a subset

Select Restricted Space

Space Tree

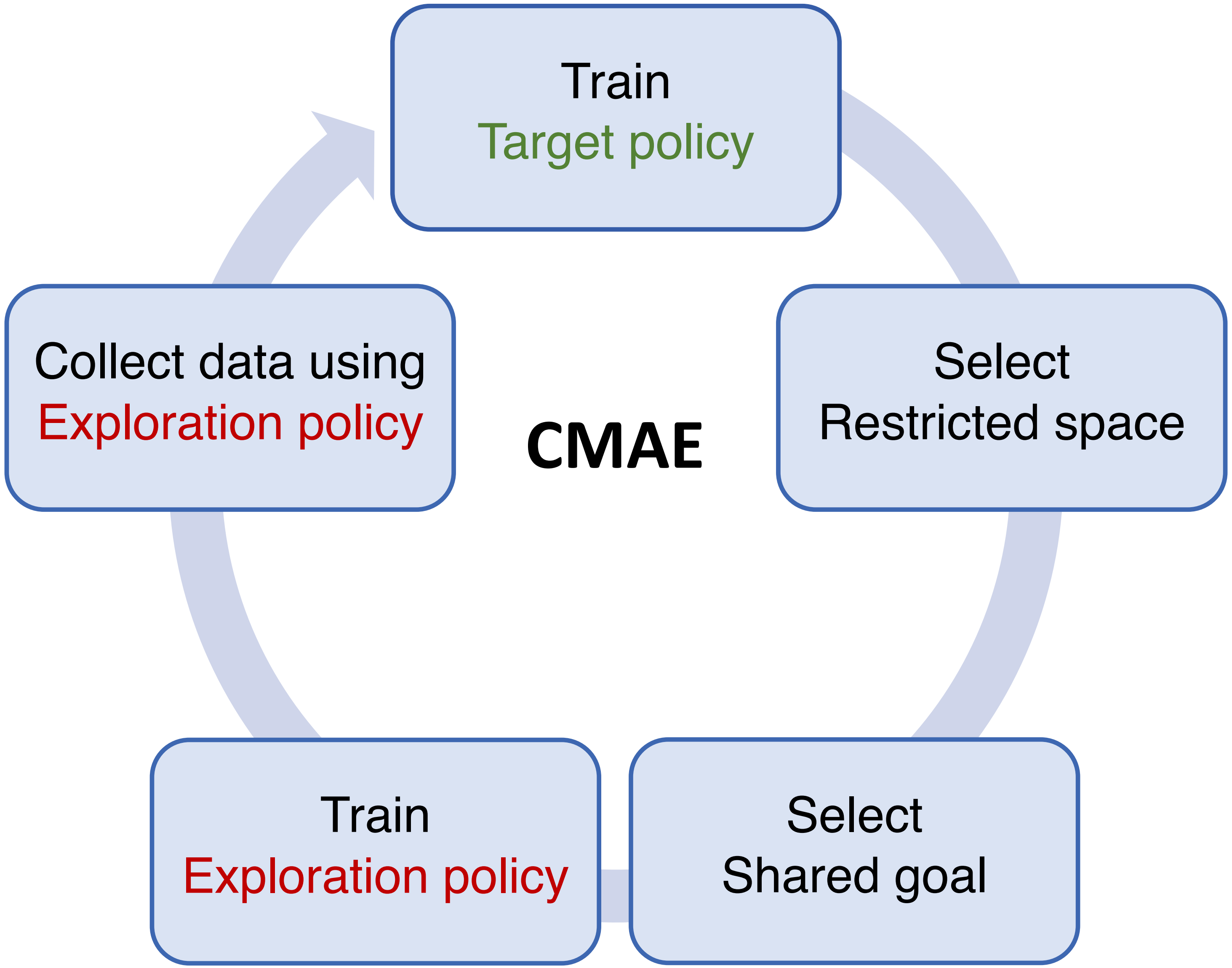
- Each node represents a restricted space
- Space tree is initialized with 1-dimensional restricted spaces



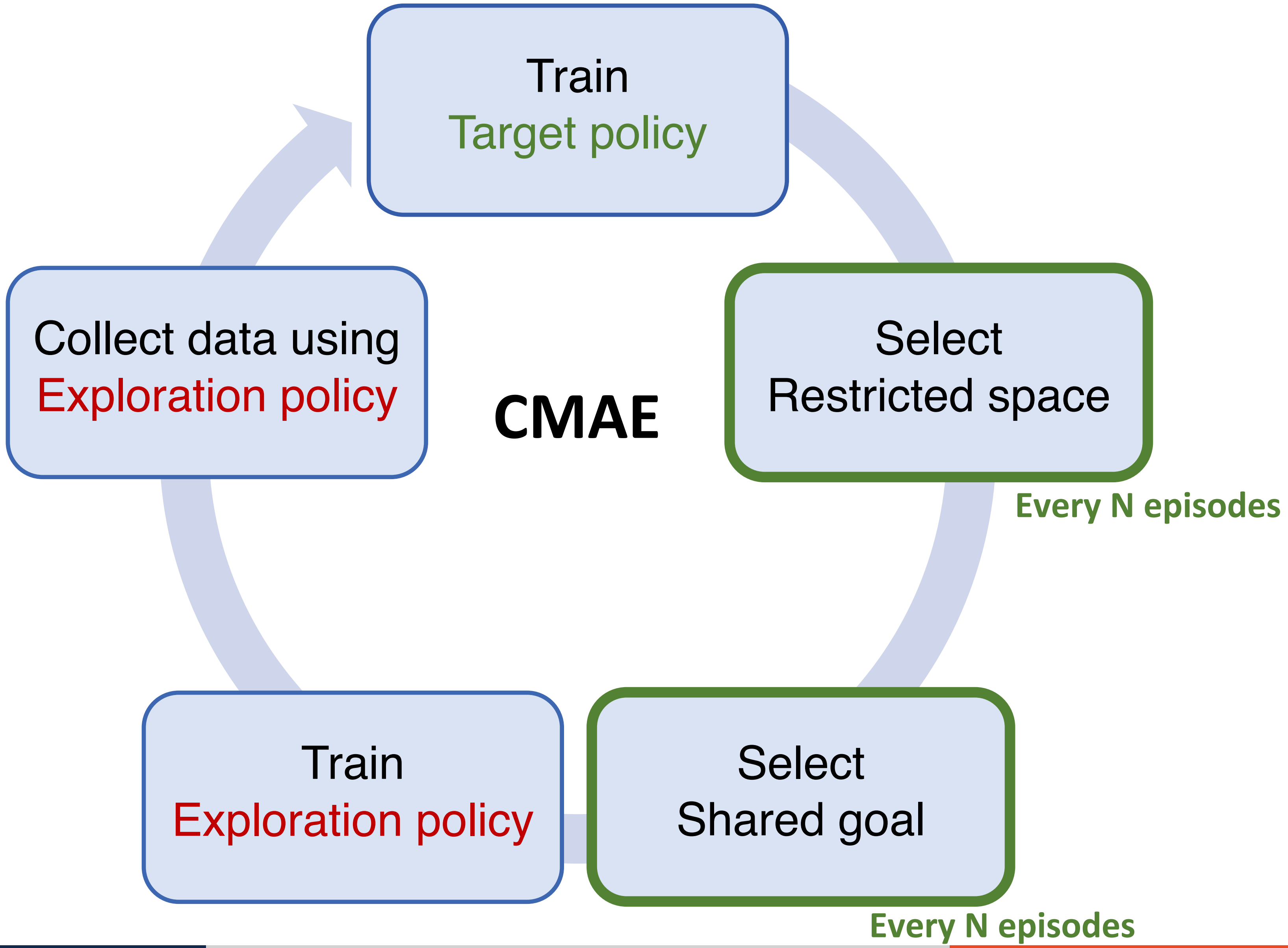
Space Tree Expansion

- Utility μ_k: negative normalized entropy of S_k
- Select restricted space S_k with high μ_k
- Add all restricted spaces of $(|k| + 1)$-dimension which contain S_k as a subset

Summary


Train
Target policy
Select
Restricted space
CMAE
Collect data using
Exploration policy
Train
Exploration policy
Select
Shared goal

Summary


Train
Target policy
Select
Restricted space
Every N episodes
Select
Shared goal
Every N episodes
Train
Exploration policy
Collect data using
Exploration policy
CMAE

Experimental Results

Multi-agent grid world tasks

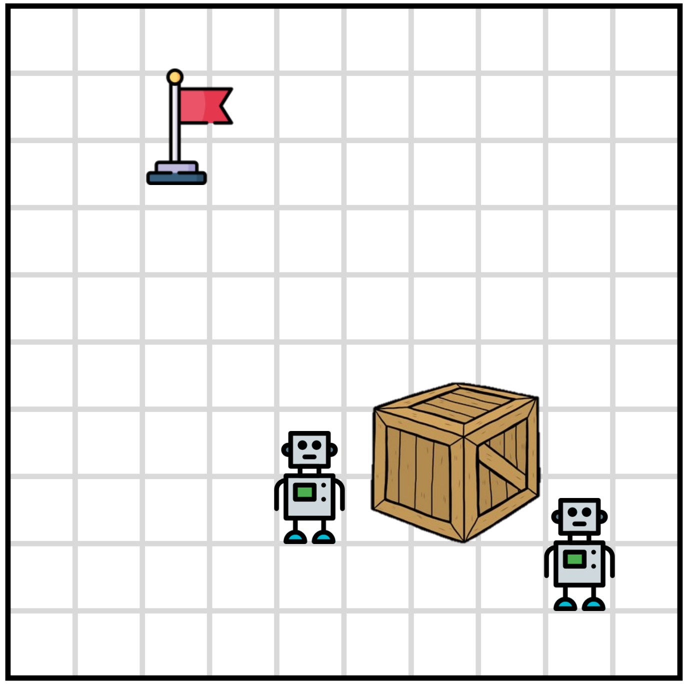
- Push-Box
- Pass
- Secret-Room

Sparse-reward StarCraft II multi-agent challenge (SMAC)

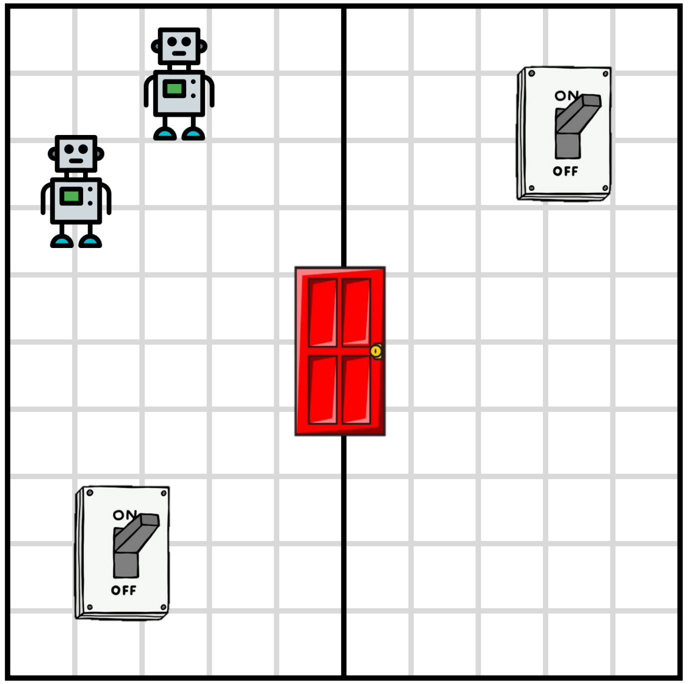
- 3m
- 2m vs. 1z
- 3m vs. 5z

Multi-Agent Grid World Tasks

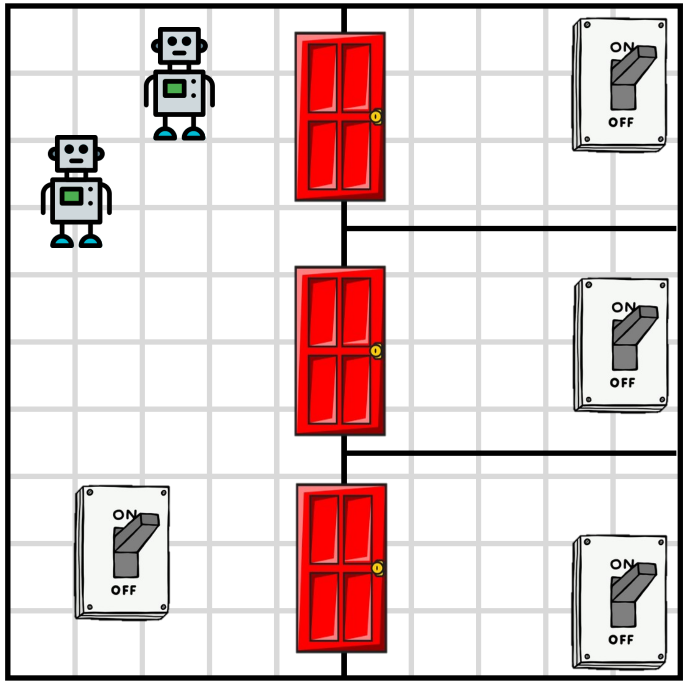
Push-Box



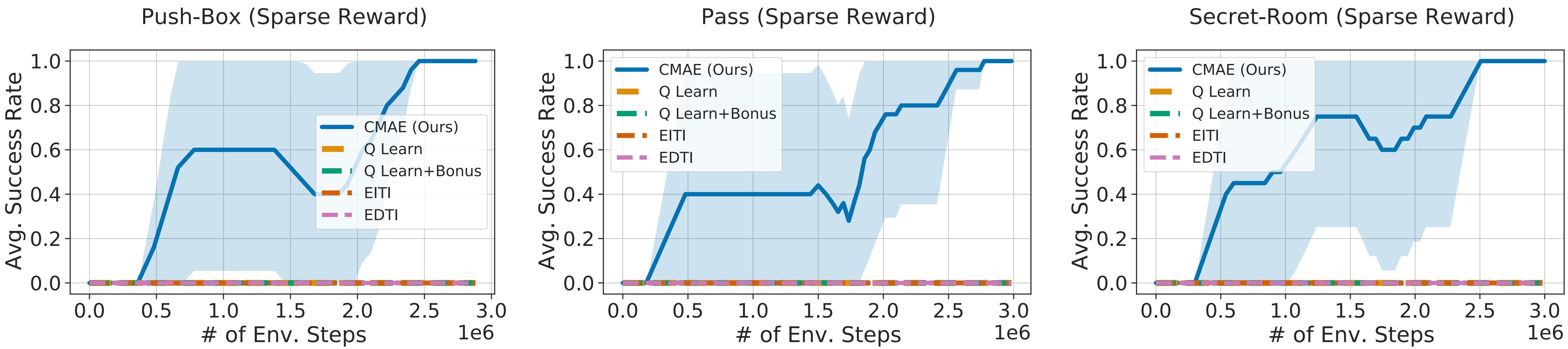
Pass



Secret-Room

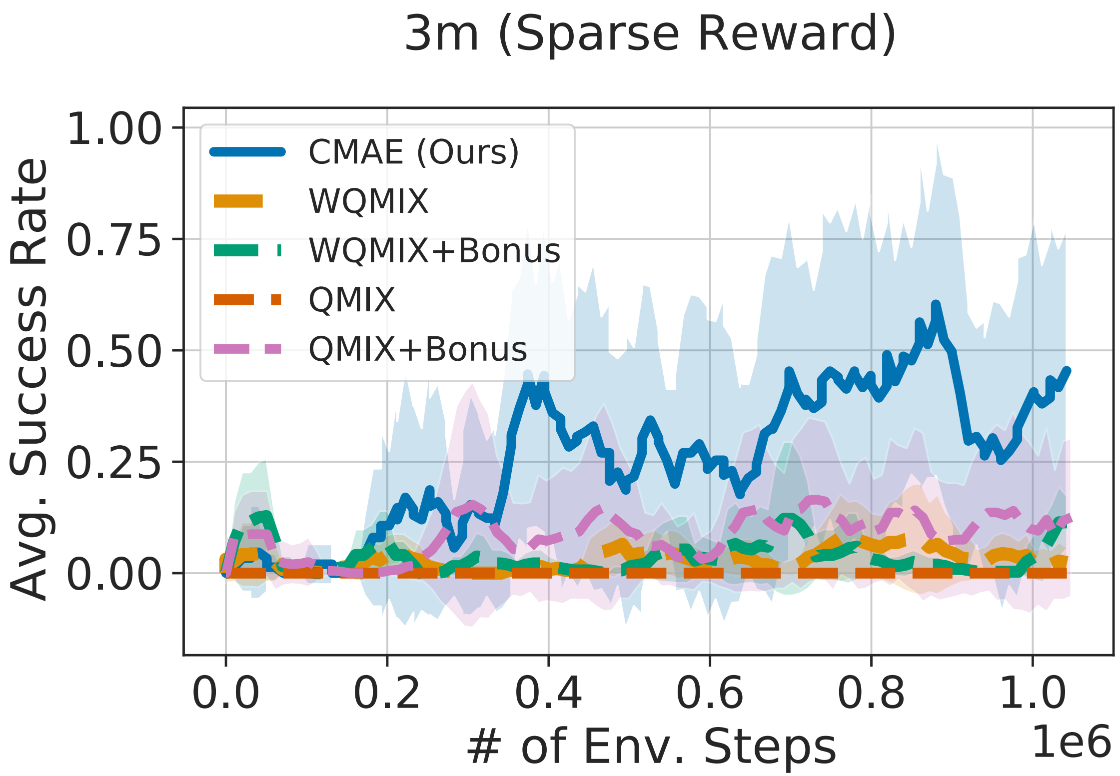


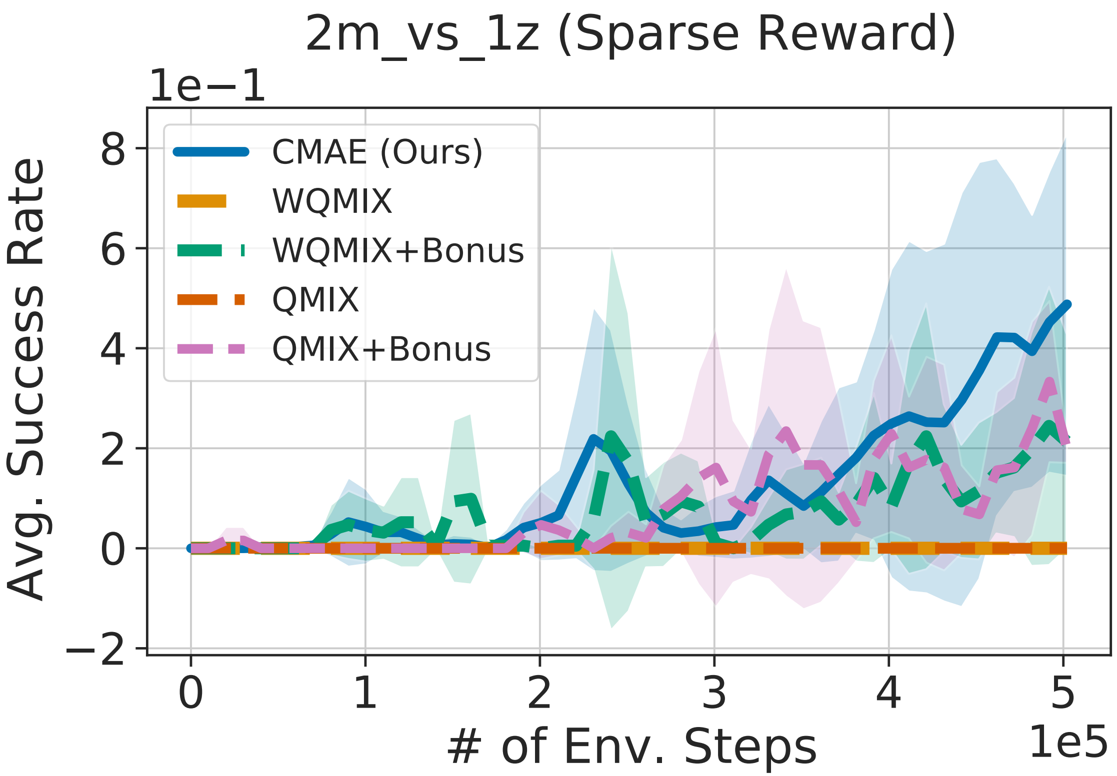
Results



Push-Box (Sparse Reward)
Pass (Sparse Reward)
Secret-Room (Sparse Reward)
Avg. Success Rate
# of Env. Steps
1e6
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
CMAE (Ours)
Q Learn
Q Learn+Bonus
EITI
EDTI

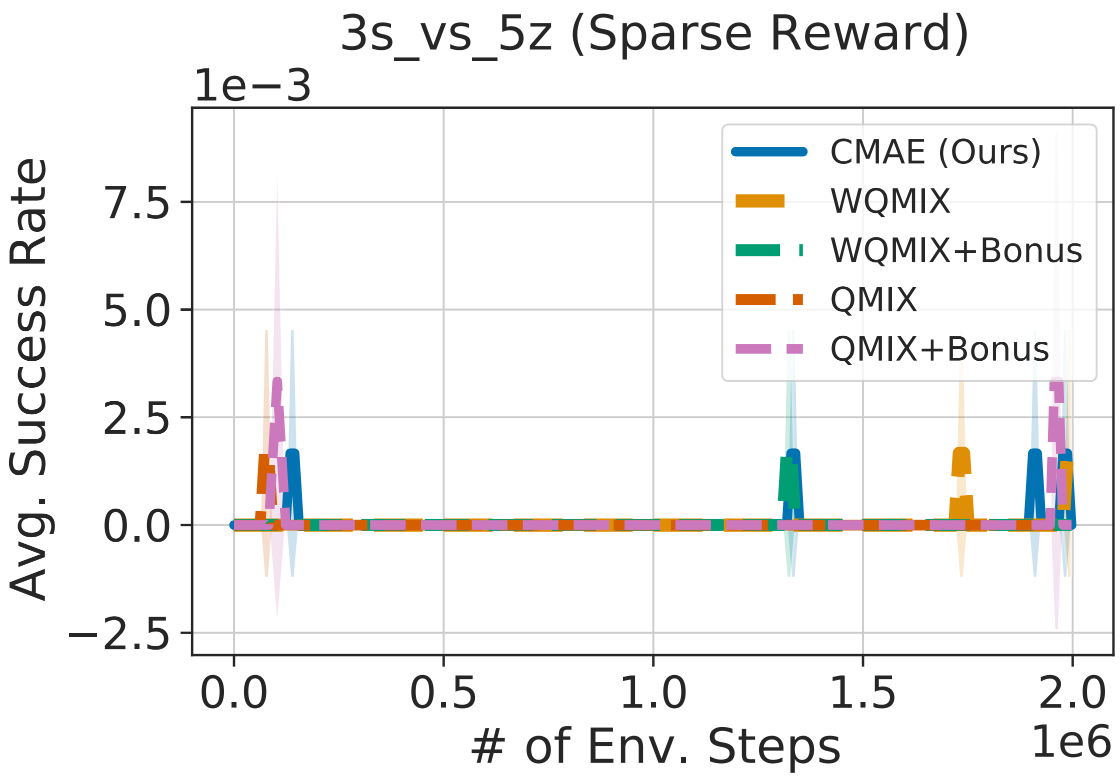
SMAC Results





Sparse 3s_vs_5z

- Winning strategy: force the enemies to scatter around the environment and attend to them one by one
- Extremely difficult without hand-crafted dense reward



Takeaways

Cooperative Multi-Agent Exploration (CMAE)

- Learns coordinated exploration policies via shared goals
- First explores low-dimensional restricted spaces
- Outperforms baselines on sparse-reward tasks

Please see us at the poster session for more details!

https://ioujenliu.github.io/CMAE