DM²: Decentralized Multi-Agent Reinforcement Learning via Distribution Matching


Caroline Wang*[1]


Ishan Durugkar*[1]


Elad Liebman*[2]


Peter Stone[1, 3]

[1] The University of Texas at Austin

[2] SparkCognition Research, [3] Sony AI

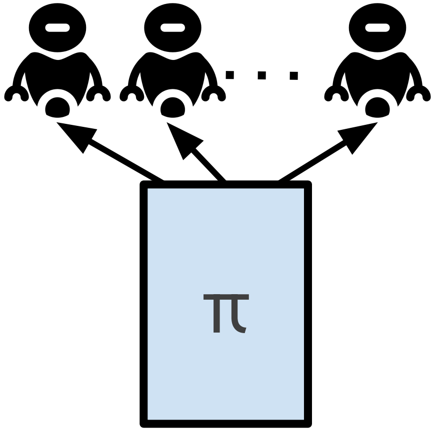
AAAI 2023

*Equal contribution

Motivation:

- Multi-agent reinforcement learning (MARL) is challenging – agents learning simultaneously makes the environment nonstationary

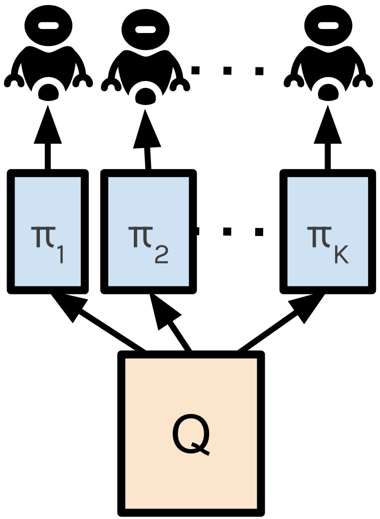
- Strategies:
  - Fully centralized learning



Motivation:

- Multi-agent reinforcement learning (MARL) is challenging – agents learning simultaneously makes the environment nonstationary

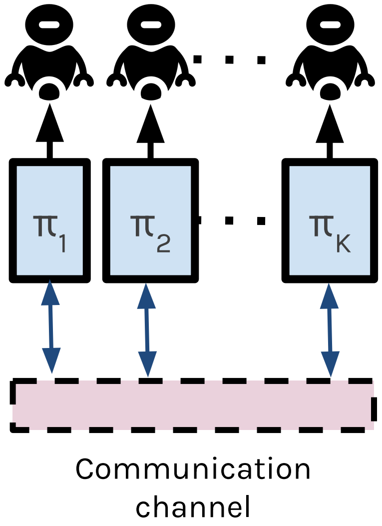
- Strategies:
  - Fully centralized learning
  - Centralized training, decentralized execution (CTDE) [1]



[1] Sunehag et al., Value Decomposition Networks for Cooperative Multiagent learning, AAMAS 2018.

Motivation:

- Multi-agent reinforcement learning (MARL) is challenging – agents learning simultaneously makes the environment nonstationary
- Strategies:
  - Fully centralized learning
  - Centralized training, decentralized execution (CTDE) [1]
  - Decentralized learning + communication[2]



[1] Sunehag et al., Value Decomposition Networks for Cooperative Multiagent learning, AAMAS 2018.
[2] Jaques et al., Social Influence as Intrinsic Motivation for Multi-Agent Deep Reinforcement Learning, ICML 2019.

Motivation:

- Multi-agent reinforcement learning (MARL) is challenging – agents learning simultaneously makes the environment nonstationary
- Strategies:
  - Fully centralized learning
  - Centralized training, decentralized execution (CTDE) [1]
  - Decentralized learning + communication[2]

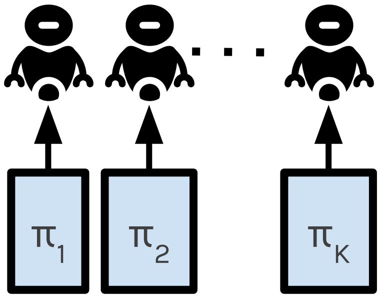
share model components or require communication

[1] Sunehag et al., Value Decomposition Networks for Cooperative Multiagent learning, AAMAS 2018.
[2] Jaques et al., Social Influence as Intrinsic Motivation for Multi-Agent Deep Reinforcement Learning, ICML 2019.

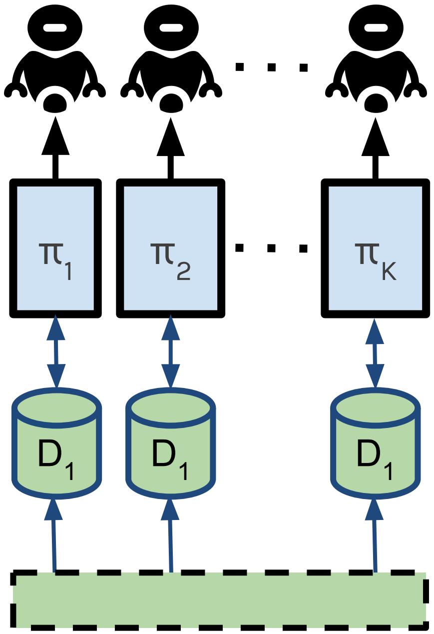
How can we foster team cooperation in the decentralized learning scenario w/o explicit communication?

Fully decentralized learning: no shared model components or communication between agents during training or execution

- Search-and-rescue robotics
- Autonomous driving
- Scalability
- Parallelism



DM^2: a MARL algorithm that enables cooperation in the decentralized setting w/o explicit communication



Expert Team Demo

Contributions

- Propose DM^2, a decentralized MARL algorithm based on independent **distribution matching to encourage coordination**
- Theoretical analysis shows
  - Conditions under which DM^2 **converges**
  - **Expert policies are a Nash equilibrium** for mixed task and distribution matching reward
- **Empirical validation** in StarCraft II tasks

Background: Stochastic Games

- Stochastic game[1] $\langle K, \mathcal{S}, \mathcal{A}, \rho_0, \mathcal{T}, R, \gamma \rangle$
  - Number of agents K
  - State space $\mathcal{S}$
  - Action space $\mathcal{A} \equiv A^K$
  - Initial state distribution $\rho_0 : \Delta(\mathcal{S})$
  - Transition function $\mathcal{T} : \mathcal{S} \times A_0 \times \cdots \times A_{K-1} \mapsto \Delta(\mathcal{S})$
  - Reward function $R_i : \mathcal{S} \times A_0 \times \cdots \times A_{K-1} \mapsto \mathbb{R}$
  - Discount factor γ
- Per-agent policy $\pi_i : \mathcal{S} \mapsto \Delta(A_i)$

[1] Littman, Markov Games as a Framework for Multi-agent Reinforcement Learning, ICML 1994.

Background: Distribution Matching

- Approach to imitation learning (IL) [1, 2]
- The **per agent** state-action visitation distribution

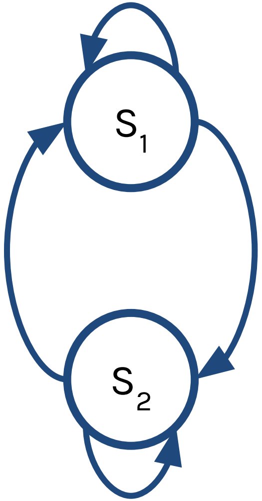
$$\rho_{\pi_i,\pi_{i-}}(s, a_i) := (1-\gamma)\pi_i(a_i|s)\sum_{t=0}^{\infty}\gamma^t P(s_t = s|\pi_i, \pi_{i-})$$

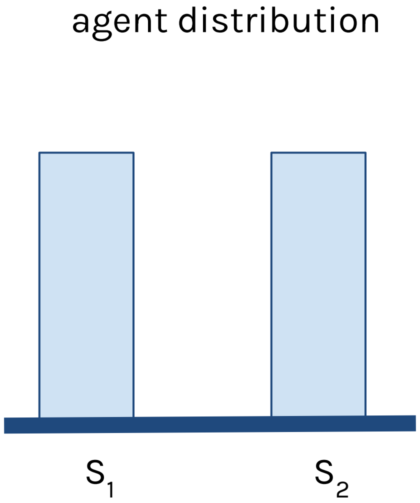
...should match the **per expert** state-action visitation distribution $\rho_{\pi_{E_i},\pi_{E_i-}}(s, a_i)$

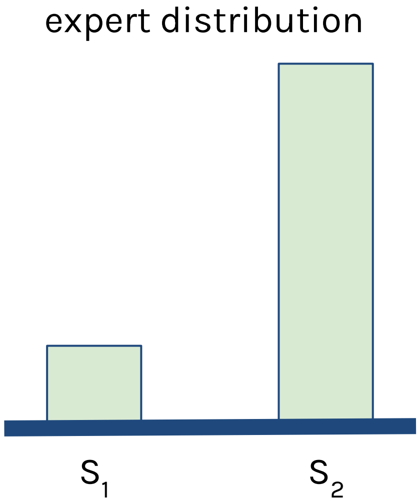
[1] Schaal, Learning from demonstration, NeurIPS 1997
[2] Ho and Ermon, Generative adversarial imitation learning, NeurIPS 2016

Background: Distribution Matching

- Approach to imitation learning (IL) [1, 2]



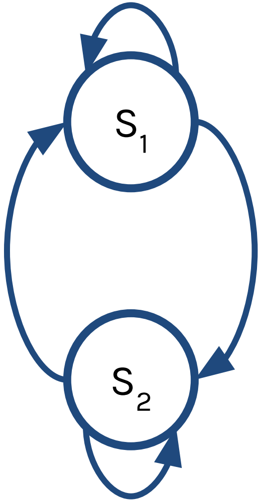


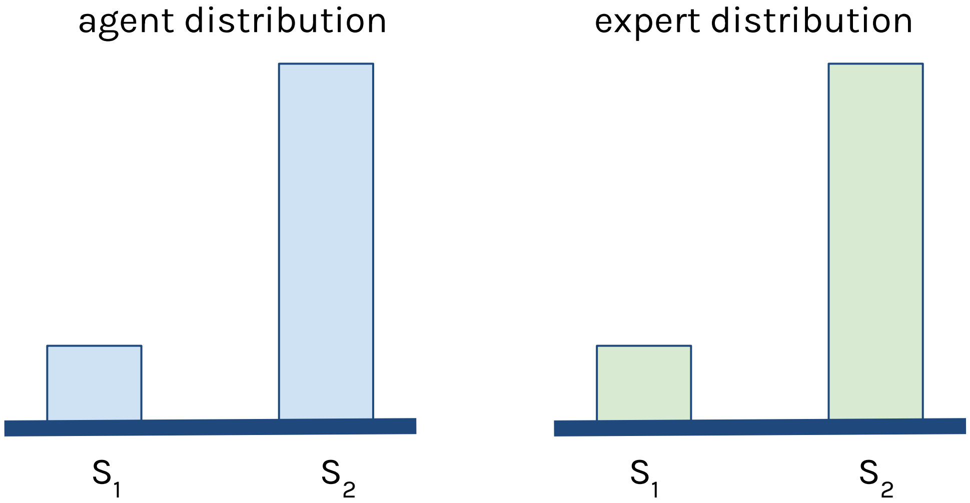


[1] Schaal, Learning from demonstration, NeurIPS 1997
[2] Ho and Ermon, Generative adversarial imitation learning, NeurIPS 2016

Background: Distribution Matching

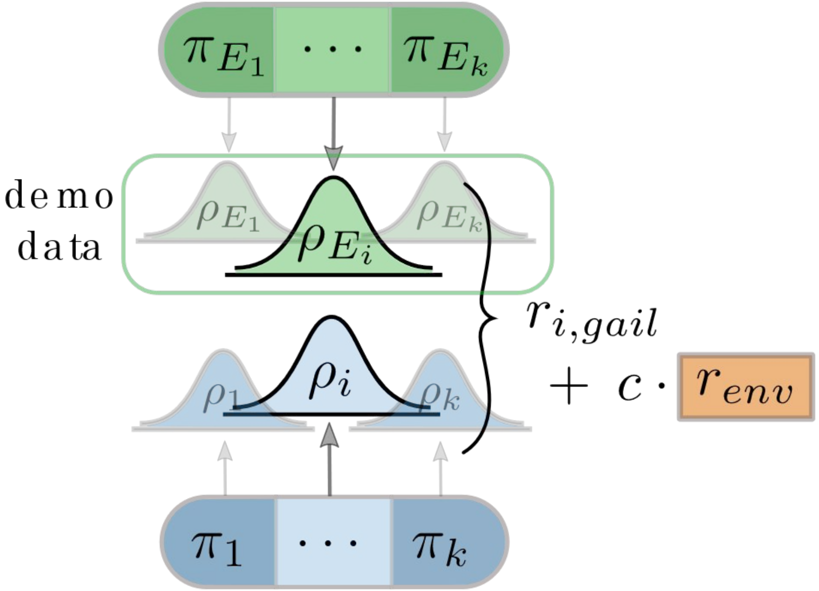
- Approach to imitation learning (IL) [1, 2]





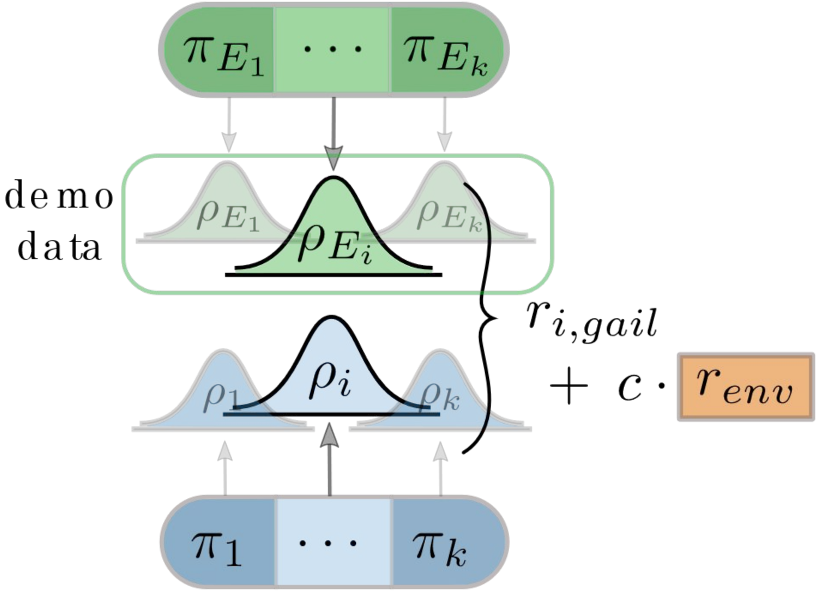
[1] Schaal, Learning from demonstration, NeurIPS 1997
[2] Ho and Ermon, Generative adversarial imitation learning, NeurIPS 2016

Theoretical Analysis



1. Individual distribution matching leads to agent policies converging to compatible expert policies
2. Expert policies also constitute a Nash equilibrium under a mixed task and distribution matching reward

DM^2: Decentralized MARL via Distribution Matching



Algorithm 1: DM^2 (**D**ecentralized **MARL** via distribution **m**atching)

Input: Number of agents K, expert demonstrations $\mathcal{D}_0, \ldots, \mathcal{D}_K$, environment env, number of epochs N, number of time-steps per epoch M, reward mixture coefficient c

```
for k = 0, ..., K − 1 do
    Initialize discriminator parameters φ_k;
    Initialize policy parameters θ_k;
end
for n = 0, 1, ..., N − 1 do
    Gather m = 1, ..., M steps of data
      (s^m, a^m, r^m_env) from env;
    for k = 0, ..., K − 1 do
        Sample M states from demonstration D_k;
        Update discriminator D^k_φ;
        Get GAIL reward r^m_{k,GAIL} = − log D_{k,φ}(s^m)
          for m = 1, ..., M;
        Set agent reward r^m_{k,mix} = r^m_env + r^m_{k,GAIL} * c;
        Update agent policy π^k_θ with data
          (s_m, a_m, r^m_{k,mix}) for m = 1, ..., M;
    end
end
```

Output: K agent policies $\boldsymbol{\pi}_\theta$

Experimental Setting

- StarCraft II Multi-Agent Challenge[1] tasks
  - 5m vs 6m (5v6)
  - 3s vs 4z (3sv4z)
- Baselines w/environment reward alone
  - IPPO (decentralized)
  - QMIX[2] (CTDE)
  - R-MAPPO[3] (CTDE)
- Distribution Matching Baseline: DM^2 w/SIL [4]

[1] Samvelyan et al., The StarCraft Multi-Agent Challenge, AAMAS 2019.
[2] Rashid et al., Qmix: Monotonic Value Function Factorisation for Deep Multi-agent Reinforcement Learning, ICML 2018.
[3] Yu et al., The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games, ArXiv 2021.
[4] Oh et al., Self-Imitation Learning, ICML 2018.

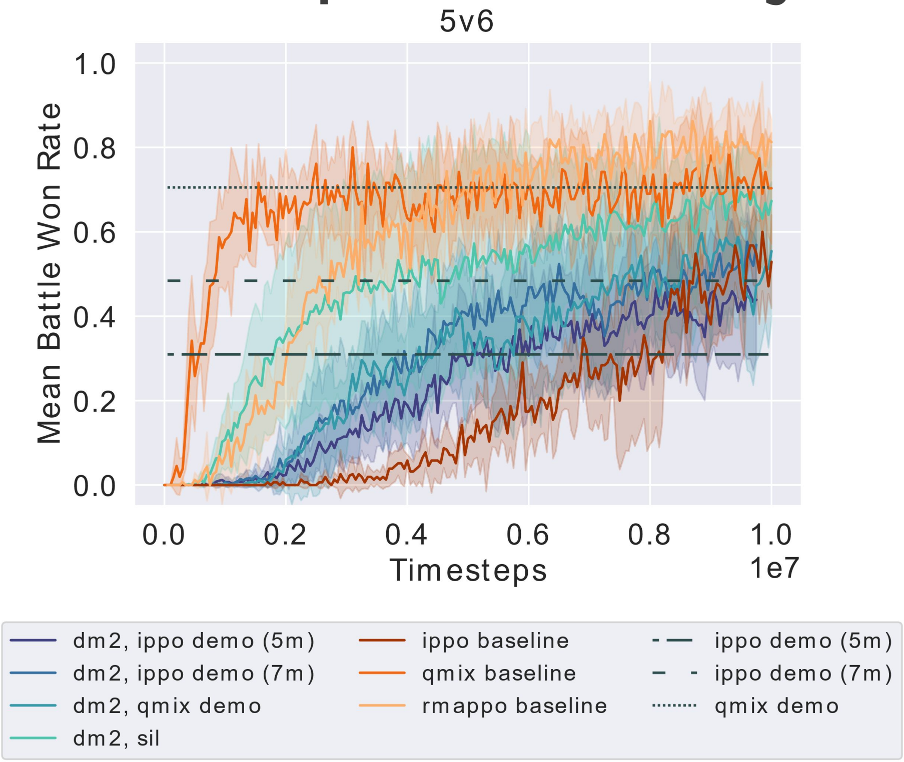
Experimental Setting

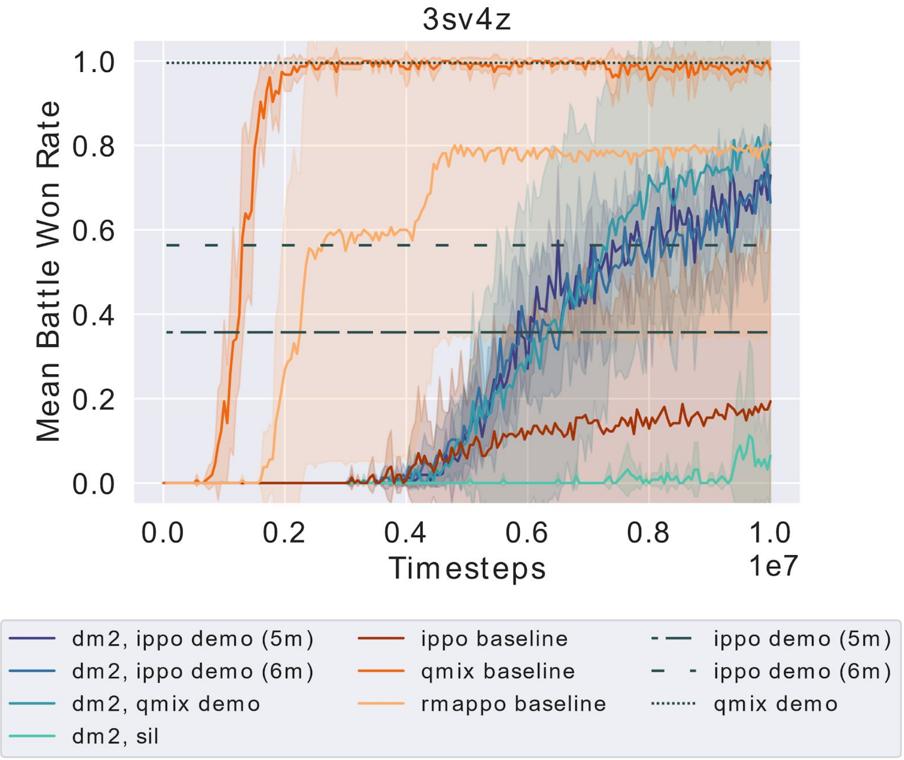
- MARL algorithm: Independent PPO (IPPO)[1]
- Demonstrations from K experts
  - State-only demonstrations sampled from saved IPPO **and** QMIX checkpoints
- Per-agent reward function:

$$r_{i,mix} = r_{env} + r_{i,GAIL} * c$$

[1] Yu et al., The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games, ArXiv 2021.

1. Sample efficiency of DM^2 vs baselines



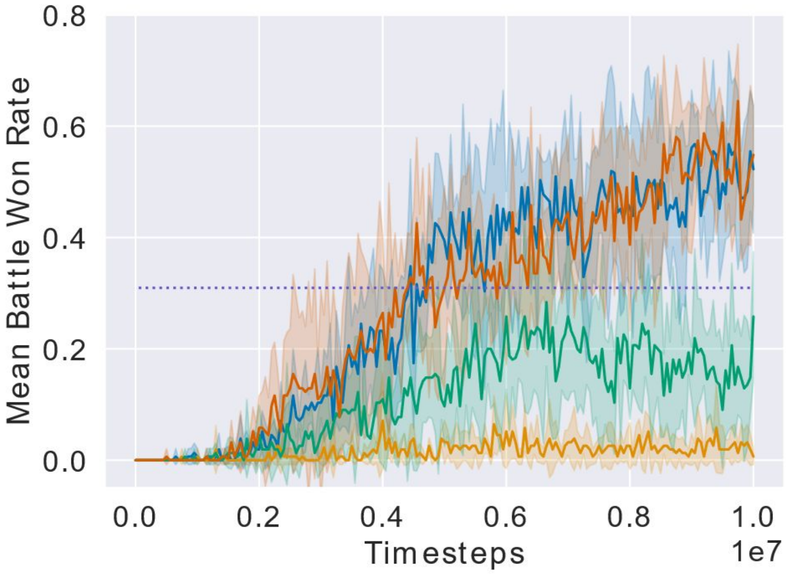


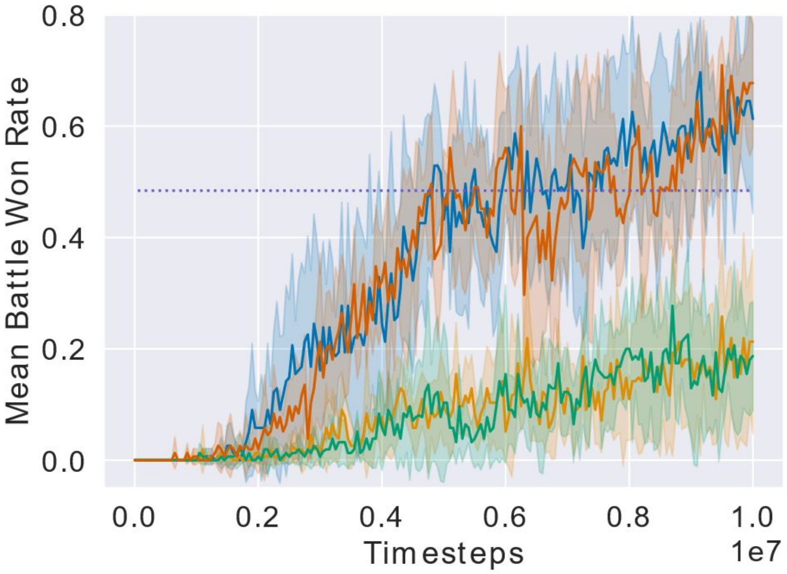
2. Coordination of expert demonstrations

Demonstrations could be ***concurrently*** sampled from ***jointly trained*** expert policies

	concurrent	nonconcurrent
joint	DM^2	ablation
not joint	ablation	ablation

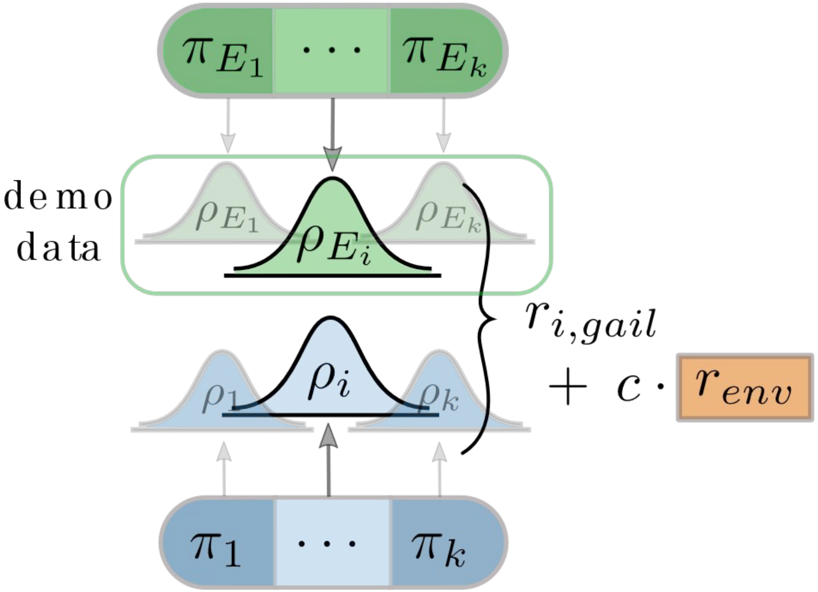
2. Coordination of expert demonstrations





legend	concurrent	nonconcurrent
joint	DM^2	ablation (1)
not joint	ablation(2)	ablation (3)

DM²: Decentralized Multi-Agent Reinforcement Learning via Distribution Matching



Caroline Wang
caroline.l.wang@utexas.edu

Ishan Durugkar
ishand@cs.utexas.edu

Elad Liebman
eliebman@sparkcognition.com

Peter Stone
pstone@cs.utexas.edu