Offline Multi-agent Continual Cooperation via Skill Partition and Reuse

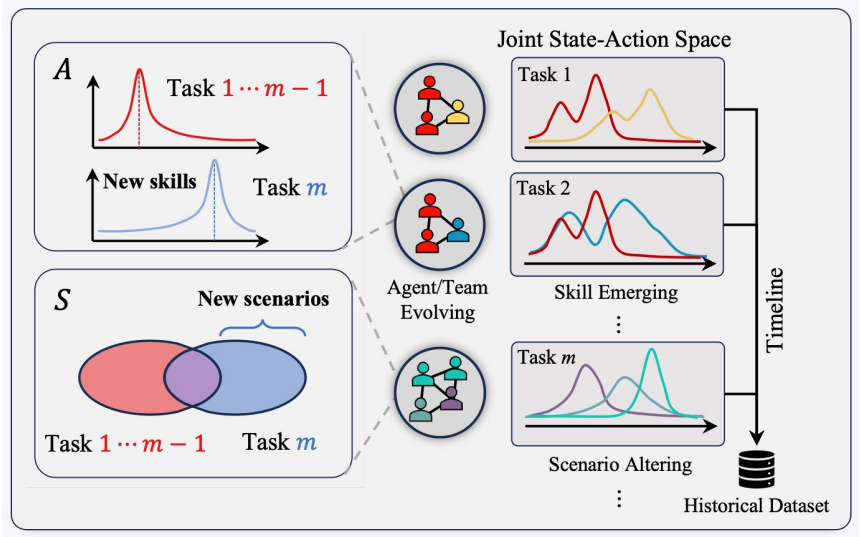
(ICML 26)

Yuchen Xiao, Lei Yuan, Ruiqi Xue, Tieyue Yin, Yang Yu

Yuchen Xiao

2026.05.01

Background



The stream of tasks

Real world scenarios where multi-agent datasets are provided in a sequential manner.

- Continual Reinforcement Learning

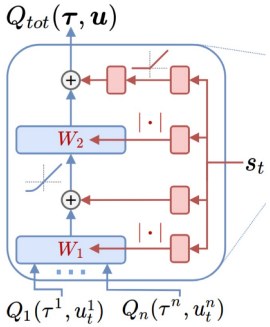
  Effectively learn from a sequential stream of tasks in open-ended environments, by preserving and reusing knowledge (e.g. skills) among **agents** and **tasks**.

- In MA: harder transfer due to agent knowledge combinatorial explosion
- In Offline: inter- and intra-task distributional shifts

Related Work

- MARL

Value Decomposition



Implicit Constraint

$$\pi_{k+1} = \arg\max_{\pi} \mathbb{E}_{a\sim\pi(\cdot|\tau)}[Q^{\pi_k}(\tau, a)],$$
$$\text{s.t.} \quad D_{\mathrm{KL}}(\pi \parallel \mu)[\tau] \leq \epsilon.$$

*Advantage weighted regression(AWR)

Implicit Constraint with closed form optimal policy (OMIGA[1])

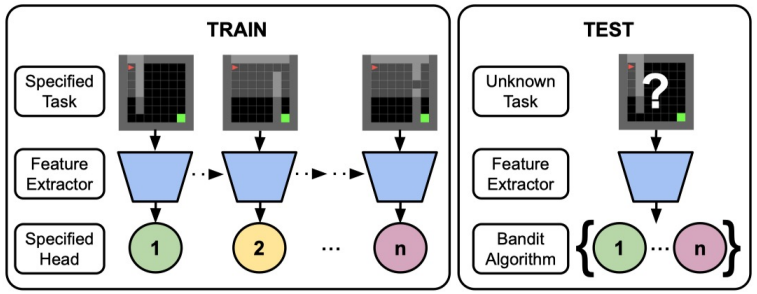
$$Q^{tot}(\boldsymbol{o}, \boldsymbol{a}) = \sum_i w^i(\boldsymbol{o}) Q^i(o^i, a^i) + b(\boldsymbol{o}),$$
$$V^{tot}(\boldsymbol{o}) = \sum_i w^i(\boldsymbol{o}) V^i(o^i) + b(\boldsymbol{o}),$$
$$w^i \geq 0, i = 1, \cdots, n.$$

$$\pi^*_{tot}(\boldsymbol{a}|\boldsymbol{o}) = \mu_{tot}(\boldsymbol{a}|\boldsymbol{o}) \cdot \exp\left(\frac{Q^*_{tot}(\boldsymbol{o}, \boldsymbol{a}) - u^*(\boldsymbol{o})}{\alpha} - 1\right)$$

[1] Wang, X. et al. Offline multiagent reinforcement learning with implicit global-to-local value regularization. NeurIPS, 2023.

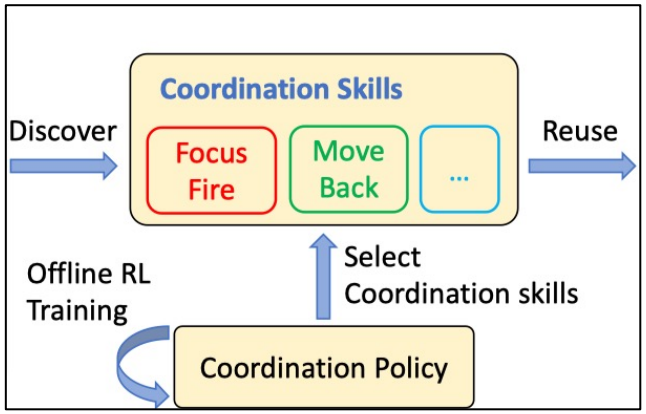
Related Work

- Continual RL



- Parameter isolation (OWL[2])
- Regularization
- Gradient Projection Based
- Hypernetwork; Task embedding
- ……(Mostly simply adapted)

- Skill Discovery: learning **action chunk** for guiding decision-making



Learning skill embedding via
- VAE, hVAE (e.g. ODIS[3])
- Mutual information
- Option framework
- Metric based
- ……

[2] Kessler, S., et al. Same state, different task: Continual reinforcement learning without interference. AAAI, 2022.
[3] Zhang, F., et al. Discovering generalizable multi-agent coordination skills from multi-task offline data. ICLR, 2023.

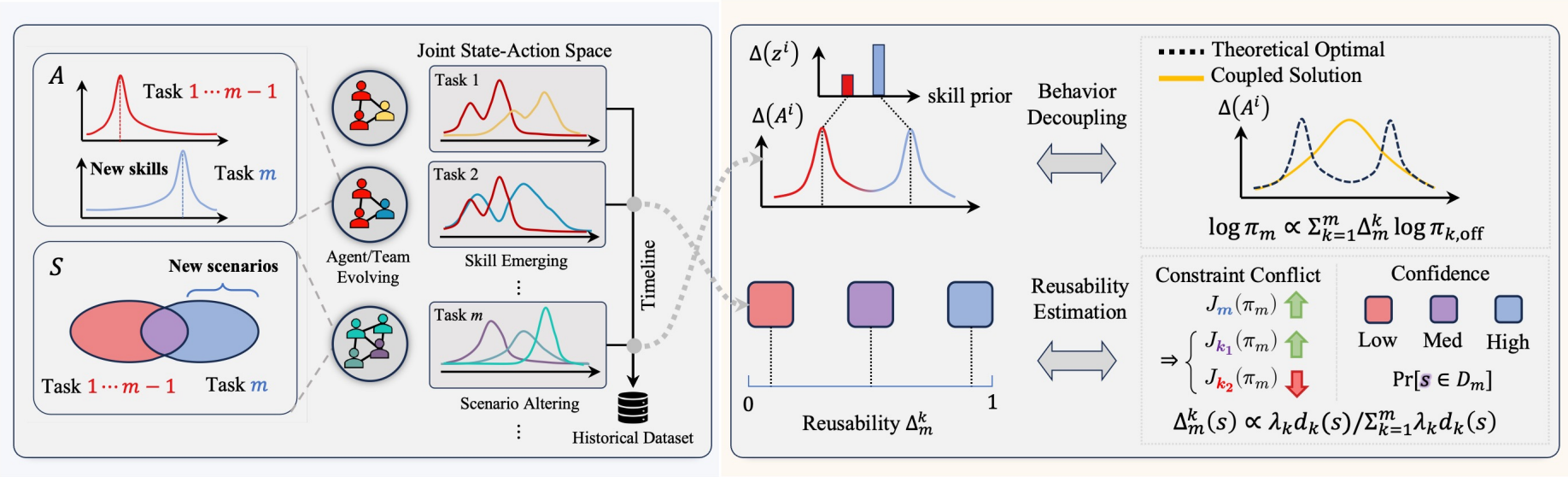
Problem Formulation

- Offline Multi-agent Continual Skill Discovery
  - Learning from MA dataset stream effectively by discovering, preserving and reusing skills
- Dec-POMDP (Single-task)

$$\mathcal{M} = \langle N, S, A, \Omega, O, R, P, \gamma \rangle,$$

- Offline Continual MARL with Skills
  - Learning from an infinite stream of tasks $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_m, \cdots$
  - Skill z_m^i from a finite set Z_m^i, for each agent i and task m
  - Task bounded setting: finite skill conditioned policy $\pi_m^i(a^i|o^i, z_m^i)$
  - **Objective**: maximizing performance $J_m(\pi_m)$ on current task while maintaining performance on old tasks $J_k(\pi_m) \geq L_k$

Method: Overview



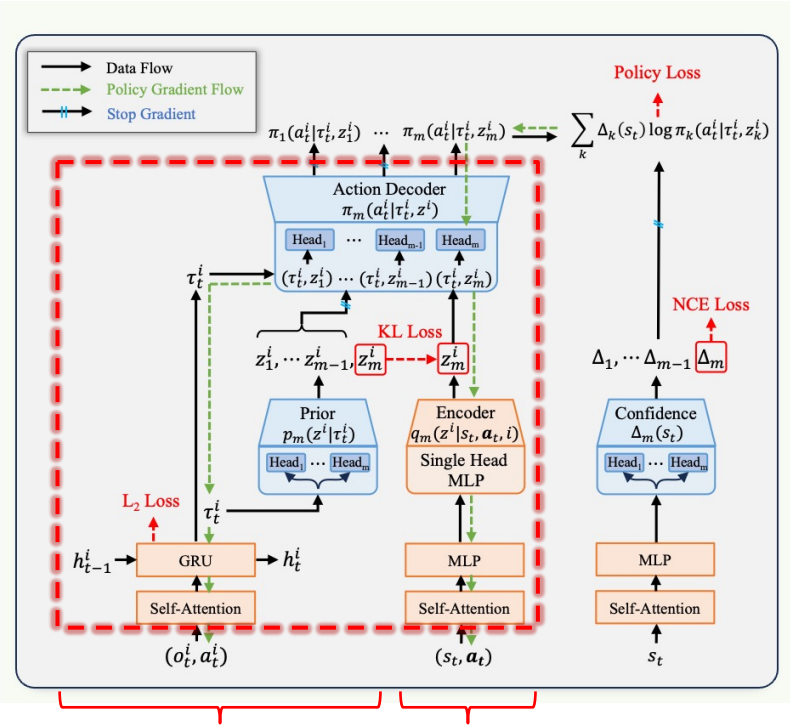
- New skills from agent team evolving & task scenario altering
  - Discover novel skills
  - Partition different skills
  - Selectively reuse skill for effective transfer
  - Theoretical analysis

Method.1 Discovering Skills

- Population-invariant networks

$$Q = \text{MLP}_q([o^{i,\text{env}}, o^{i,1}, \ldots, o^{i,n}])$$
$$K = \text{MLP}_k([o^{i,\text{env}}, o^{i,1}, \ldots, o^{i,n}])$$
$$V = \text{MLP}_v([o^{i,\text{env}}, o^{i,1}, \ldots, o^{i,n}])$$
$$[e^{i,\text{env}}, e^{i,1}, \ldots, e^{i,n}] = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V,$$

Value Net Policy Net



- Advantage-weighted regression + CVAE

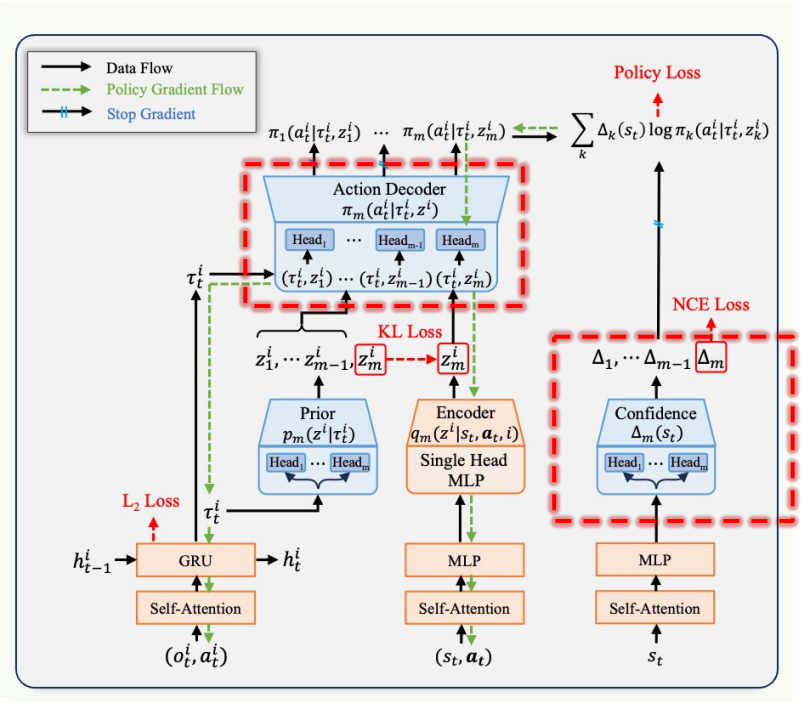
$$L_{\pi,q}^{(1)} = -\mathbb{E}_{(\tau^i,a^i)\sim D_m, z_m^i \sim q_m}$$
$$\left[\exp\left(\frac{w_m^i(\boldsymbol{o})}{\beta_m} A_m^i(o^i,a^i)\right)\cdot \log \pi_m^i(a^i|\tau^i, z_m^i)\right]$$

- Local inference

$$L_p = \mathbb{E}_{(s,\tau^i,\boldsymbol{a})\sim D_m}\left[D_{KL}(q_m(\cdot|s,\boldsymbol{a},i)\|p_m(\cdot|\tau^i))\right]$$

Method.2 Skill Partition and Reuse

- Multi-head arch for handling multi-modal action distribution



Action Decoder: $\pi_m^i(a^i|\tau^i; z_k^i) = G_k^\pi(F(\tau^i, z_k^i)),$

Skill Encoder: $p_m(z_k^i|\tau^i) = G_k^p(F(\tau^i, z_k^i)).$

Confidence: $E_k(s) = G_k^E(F^E(s))$

- Augmented Advantage Function

$$\tilde{A}_m^i(s, \boldsymbol{o}, a^i, \{\tilde{z}_k^i\}_{k=1}^m) := \underbrace{\frac{w_m^i(\boldsymbol{o})}{\beta_m} A_m^i(o^i, a^i)}_{\text{AWR term}} + \underbrace{\sum_{k=1}^m \Delta_m^k(s) \log \pi_m^i(a^i|\tau^i, \tilde{z}_k^i)}_{\text{Gated skill augmentation via action logits}},$$

$$\Delta_m^k(s) = \frac{\beta_k d_k(s)}{\sum_{l=1}^m \beta_l d_l(s)}$$

Reusability score via state visitation frequency

Method.2 Skill Partition and Reuse

- Augmented Advantage Function

$$\tilde{A}_m^i(s, \boldsymbol{o}, a^i, \{\tilde{z}_k^i\}_{k=1}^m) := \underbrace{\frac{w_m^i(\boldsymbol{o})}{\beta_m} A_m^i(o^i, a^i)}_{\text{AWR term}} + \sum_{k=1}^m \underbrace{\Delta_m^k(s) \log \pi_m^i(a^i|\tau^i, \tilde{z}_k^i)}_{\text{Gated skill augmentation via action logits}},$$

$$\Delta_m^k(s) = \frac{\beta_k d_k(s)}{\sum_{l=1}^m \beta_l d_l(s)}$$

$$E_k(s) \approx d_k(s)/\mathcal{N}(s, \sigma_s I)$$

Higher: less like a random s

Reusability score
via state visitation frequency

- Losses for Continual MA skill learning

$$L_{\pi,q}^{(2)} = -\mathbb{E}_{(s,\tau^i,a^i)\sim D_m, z_m^i \sim q_m, \{\tilde{z}_k^i\}_{k=1}^m \sim p_m} \left[\exp\left(\tilde{A}_m^i(s, \boldsymbol{o}, a^i, \{\tilde{z}_k^i\}_{k=1}^m)\right) \cdot \log \pi_m^i(a^i|\tau^i, z_m^i)\right]$$

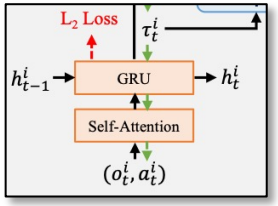
- Augmented AWR

$$L_{NCE} = -\mathbb{E}_{s\sim D_k}\left[\log \sigma(E_{\eta_k}(s))\right] - \mathbb{E}_{s^-\sim p}\left[\log \sigma(-E_{\eta_k}(s^-))\right]$$

- NCE for density estimation $\sigma(x) = \frac{1}{1+e^{-x}}$

$$L_F = \lambda_{\text{reg}} \|\theta_{F,m} - \theta_{F,1}\|_2^2,$$

- L2 loss for feature extractors



Method.3 Overall Pipeline

Algorithm 1 COMAD Training

Input: Offline Datasets $\mathcal{D} = \{D_1, \ldots, D_M\}$ (Sequentially)
Initialize: Feature extractor F, critics Q, V, mixer w, b, target networks, density estimator feature extractor F^E.
for task $m = 1$ to M **do**
  // **Head Expansion**
  if $m = 1$ **then**
    Allocate heads G_1^π, G_1^p, G_1^E.
    Set active head index $k^* \leftarrow 1$.
  else
    Evaluate expected state density $\hat{d}_k \approx \mathbb{E}_{s \sim D_m}[d_k(s)]$ for old heads $k < m$.
    if $\exists k : \hat{d}_k > d_0$ **then**
      Reuse old heads: $k^* \leftarrow \arg\max_k \mathbb{E}[d_k(s)]$.
    else
      Expand new heads G_m^π, G_m^p, G_m^E.
      Set active head index $k^* \leftarrow m$.
    end if
  end if
  // **Optimization Loop**
  for sampled batch $B \sim D_m$ **do**
    Update Q, V, w, b by minimizing critic losses in Equation (2) and (3).
    Update local encoder p_{k^*} by minimizing KL loss in Equation (6).
    Calculate NCE loss L_{NCE} via Equation (10).
    if $m = 1$ or in Stage 1 **then**
      $L_{\pi,q} \leftarrow$ vanilla policy loss in Equation (5).
    else
      $L_{\pi,q} \leftarrow$ skill-augmented policy loss in Equation (11).
      Apply regularization: $L_{\pi,q} \leftarrow L_{\pi,q} + \lambda_{reg}\|\theta_{F,m} - \theta_{F,1}\|_2^2$, $L_{NCE} \leftarrow L_{NCE} + \lambda_{reg}\|\theta_{F^E,m} - \theta_{F^E,1}\|_2^2$.
    end if
    Update state density estimator E_{k^*} by minimizing L_{NCE}.
    Update action decoder π_{k^*} and global encoder q_{k^*} by minimizing $L_{\pi,q}$.
  end for
  if $m = 1$ **then**
    Save checkpoints $\theta_{F,1}$ and $\theta_{F^E,1}$.
  end if
end for

Retrieve & Allocate

Optimize

~evaluation

Directly evaluate reusability:

$$\mathbb{E}_{s \sim D_m}[d_k(s)]$$

Method.4 Theoretical Analysis

- Formulation: sequential behavior constrained Dec-POMDP
  - with skill-conditioned policies $\Pi(\mathcal{Z}_m)$

$$\begin{aligned}\max_{\pi_m \in \Pi(\mathcal{Z}_m)} \quad & J_m(\pi_m) - \beta_m D_{KL}(\pi_m \| \mu_m) \\ \text{s.t.} \quad & J_k(\pi_m) - \beta_k D_{KL}(\pi_m \| \mu_k) \\ & \geq J_k(\pi_k) - \beta_k D_{KL}(\pi_k \| \mu_k) - \delta_k, \\ & \forall k \in \{1, \cdots, m-1\},\end{aligned}$$

"maximizing performance on current task while maintaining performance on old tasks"

- Main result: $\pi_m^i(a^i|o^i, z_m^i) \propto \exp\left(\sum_{k=1}^{m} \Delta_m^{k,*}(s) \log \tilde{\pi}_k^{i,*}(a^i|o^i, z_k^i)\right),$

where $\Delta_m^{k,*}(s) = \frac{\lambda_k \beta_k d_k(s)}{\sum_{l=1}^{m} \lambda_l \beta_l d_l(s)}$, $\tilde{\pi}_k^{i,*}(a^i|o^i, z_k^i) = \exp\left(\frac{w_k^{i,*}(\boldsymbol{o})}{\beta_k} A_k^{i,*}(o^i, a^i) + \log \mu_k^i(a^i|o^i, z_k^i)\right)$ is the single task optimal policy, $\lambda_m = 1$, and $\lambda_k, k = 1, \cdots, m-1$ are Lagrange multipliers satisfying

$$\begin{aligned}&\lambda_k[(J_k(\pi_m) - \beta_k D_{KL}(\pi_m \| \mu_k)) \\ &- (J_k(\pi_k) - \beta_k D_{KL}(\pi_k \| \mu_k) - \delta_k)] = 0.\end{aligned} \tag{21}$$

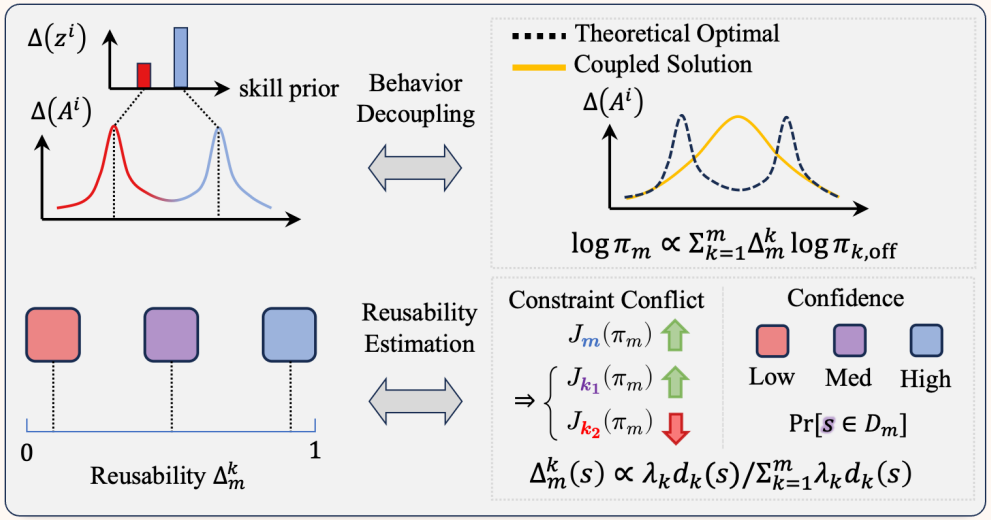
Method.4 Theoretical Analysis

- Main result:

$$\pi_m^i(a^i|o^i, z_m^i) \propto \exp\left(\sum_{k=1}^{m} \Delta_m^{k,*}(s) \log \tilde{\pi}_k^{i,*}(a^i|o^i, z_k^i)\right),$$

$$\Delta_m^{k,*}(s) = \frac{\lambda_k \beta_k d_k(s)}{\sum_{l=1}^{m} \lambda_l \beta_l d_l(s)},$$

$$\lambda_k[(J_k(\pi_m) - \beta_k D_{KL}(\pi_m \| \mu_k)) - (J_k(\pi_k) - \beta_k D_{KL}(\pi_k \| \mu_k) - \delta_k)] = 0.$$



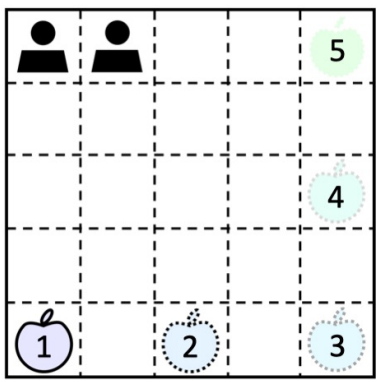
- The optima are **geometric mixtures**, with Lagrange multipliers as skill gates.
- Skill gates (reusable) + visitation frequency (confident) -> reusability
  - Select skills that are conflict-free, then mix. Maintain modes.
- Essence: KL as Bregman Divergence $D_\phi(p\|q) = \phi(p) - \phi(q) - \langle \nabla\phi(q), p - q \rangle.$

$$\min_p \sum_k w_k D_\phi(p\|q_k), \sum_k w_k = 1, \Rightarrow \nabla\phi(p^*) = \sum_k w_k \nabla\phi(q_k). \Rightarrow \log p^* = \sum w_k \log q_k,$$

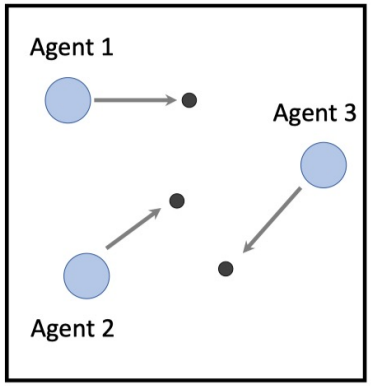
KL: $\phi(p) = \sum p \log p$

Experiment Setup

- Environments

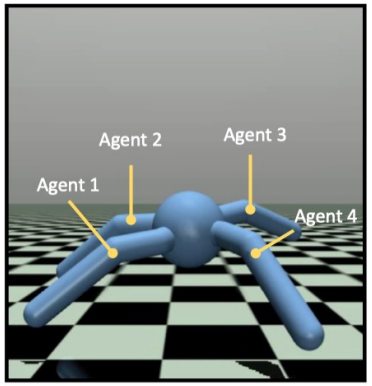


(a) Level-based Foraging



(b) Cooperative Navigation

(c) SMAC & SMACv2



(d) MAMuJoCo (Ant 4x2)

- Baselines: skill discovery(ODIS, HiSSD);

  Continual RL(EWC, OWL); Vanilla (multi-task, fine-tune, from-scratch, Rehearsal)

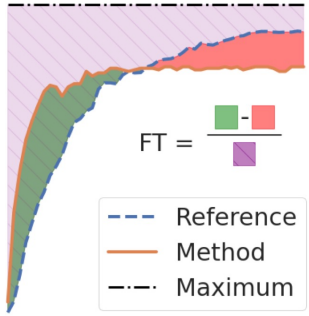
- Metrics:

  Average performance $\mathrm{P}(t) := \frac{1}{M}\sum_{k=1}^{M} p_k(t)$

  Forward transfer $\mathrm{FwT}_k = \frac{1}{\Delta}\sum_{t=(k-1)\Delta}^{k\Delta}(p_k(t) - p_k^b(t))$

  Backward transfer $\mathrm{BwT}_k = p_k(T) - p_k(k\Delta)$

  Reported: $\mathrm{P} = \mathrm{P}(T)$ $\mathrm{FwT} = \frac{1}{M}\sum_{k=1}^{M}\mathrm{FwT}_k$ $\mathrm{BwT} = \frac{1}{M}\sum_{k=1}^{M}\mathrm{BwT}_k$



Experiment Setup

- Environments
  - Task stream:
    - LBF: BottomLeft, Bottom, BottomRight, Right, TopRight
    - CN: CN-2, CN-3, CN-4, CN-5
    - SMAC: Marines sequence, SZ sequence
    - SMACv2: Protoss, Zerg, Terran
    - MaMuJoCo: Reward; Dynamics
  - Data quality: Expert, Medium

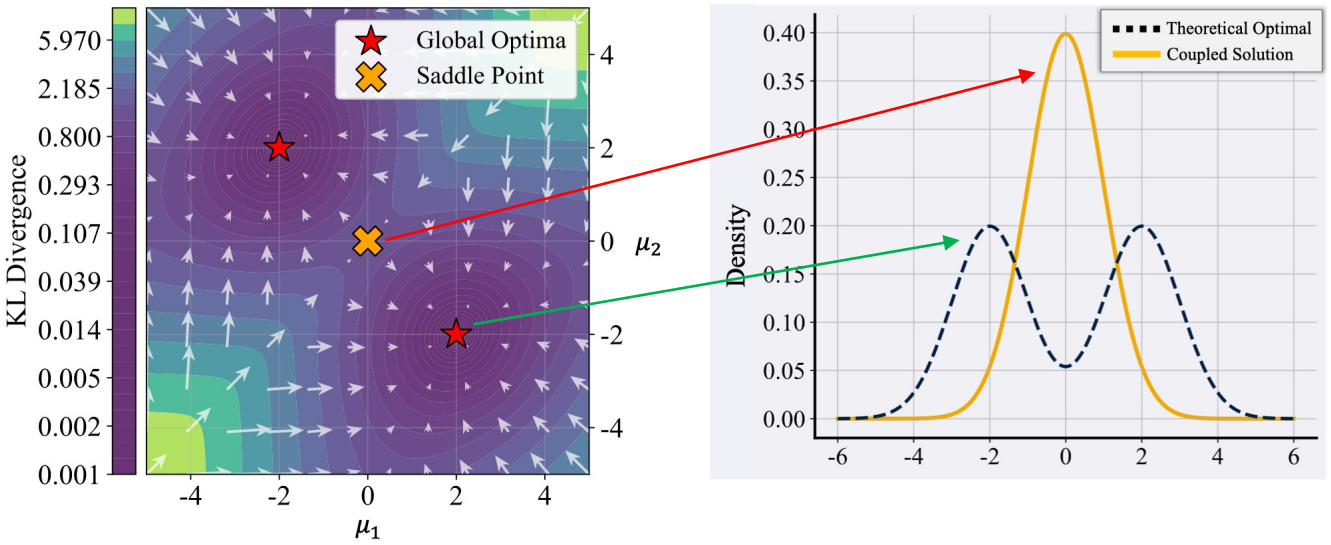
An Illustrative Example

- The necessity of multi-head arch for handling multi-modal action data

$$p_{\text{data}} \propto \mathcal{N}(-2,1) + \mathcal{N}(2,1)$$
$$p_{\text{model}} \propto \mathcal{N}(\mu_1,1) + \mathcal{N}(\mu_2,1)$$
$$d(\mu_1,\mu_2) = D_{KL}(p_{\text{data}} \| p_{\text{model}}),$$

$\mu_1 = \mu_2$: Fail to identify modes, collapse

free μ_1, μ_2: Gracefully fitted



(a) The KL divergence objective and its gradients

(b) Two solutions

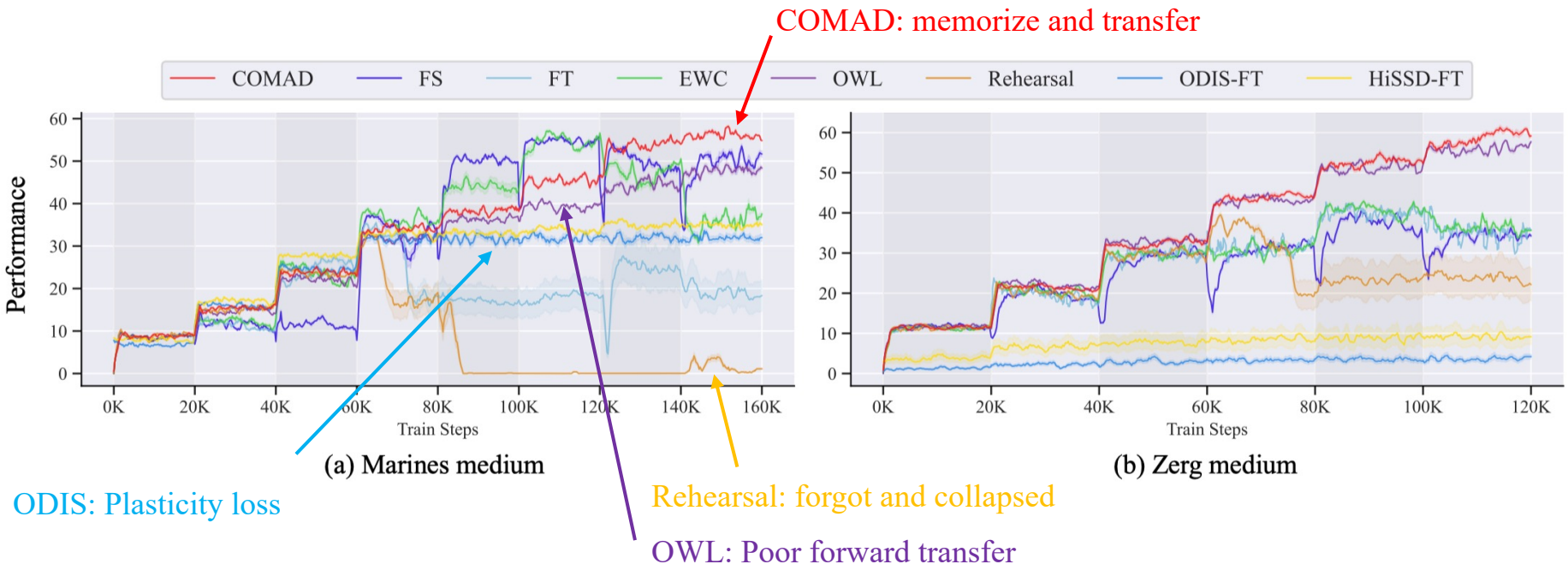
Main Results

Table 1. Average performance ± std of different algorithms on task streams from LBF, CN, SMAC, SMACv2 and MAMuJoCo environments. The best performance is marked in bold for each task stream. All results are based on 5 distinct seeds and 32 episodes per seed on each evaluation step. "Overall" means the averaged performances of one algorithms over all task streams.

Task Stream	Dataset	COMAD(ours)	MT	FS	FT	EWC	OWL	Rehearsal	ODIS-FT	HiSSD-FT
LBF	Expert	**99.26 ± 0.08**	79.26 ± 0.14	20.00 ± 0.00	7.83 ± 6.72	13.30 ± 0.53	98.65 ± 0.35	97.07 ± 0.66	16.48 ± 2.86	19.17 ± 1.70
	Medium	65.08 ± 1.34	48.49 ± 1.72	16.14 ± 0.71	7.86 ± 4.00	7.45 ± 3.26	37.83 ± 13.78	**66.79 ± 1.15**	7.06 ± 0.71	7.37 ± 1.26
CN	Expert	**78.49 ± 0.41**	69.60 ± 2.06	69.35 ± 0.93	69.10 ± 1.84	67.18 ± 3.71	64.59 ± 1.74	72.60 ± 1.01	67.26 ± 0.75	63.23 ± 2.16
	Medium	47.05 ± 2.03	32.33 ± 3.38	59.51 ± 0.60	**60.67 ± 1.04**	55.18 ± 1.43	33.33 ± 2.21	39.81 ± 1.35	10.48 ± 1.25	12.06 ± 2.01
Marines	Expert	**94.05 ± 0.83**	48.63 ± 0.49	53.84 ± 4.81	53.66 ± 4.91	58.84 ± 5.08	92.52 ± 0.97	42.45 ± 28.24	17.83 ± 0.70	14.51 ± 0.20
	Medium	**55.54 ± 0.66**	34.63 ± 1.46	51.50 ± 3.37	18.15 ± 17.84	36.91 ± 3.48	48.19 ± 2.41	1.10 ± 0.69	31.79 ± 4.23	35.06 ± 1.61
Stalker-Zealot	Expert	72.20 ± 0.77	42.66 ± 0.99	9.90 ± 0.46	10.38 ± 3.72	14.84 ± 1.42	**77.49 ± 0.29**	71.40 ± 0.99	17.83 ± 1.07	27.61 ± 0.82
	Medium	45.20 ± 1.89	**52.96 ± 1.08**	10.74 ± 1.37	9.75 ± 2.00	16.65 ± 7.67	52.54 ± 3.25	23.45 ± 22.37	35.13 ± 0.97	31.07 ± 3.45
Protoss	Medium	**59.77 ± 1.76**	47.40 ± 0.73	24.25 ± 0.82	21.00 ± 1.14	23.56 ± 1.17	57.05 ± 0.28	44.90 ± 1.27	8.39 ± 0.99	18.04 ± 2.71
Zerg	Medium	**59.98 ± 2.31**	11.97 ± 2.20	34.20 ± 1.49	35.92 ± 0.22	35.70 ± 1.32	56.89 ± 1.45	22.10 ± 21.18	4.22 ± 2.99	9.77 ± 9.10
Terran	Medium	**57.35 ± 1.60**	44.74 ± 1.03	37.54 ± 1.20	35.37 ± 2.13	38.82 ± 1.24	55.09 ± 1.34	24.21 ± 7.78	10.37 ± 8.11	4.66 ± 1.74
Reward	Expert	**77.75 ± 4.64**	63.73 ± 5.93	10.18 ± 1.30	5.09 ± 1.06	15.45 ± 1.56	75.46 ± 5.08	5.33 ± 3.73	8.20 ± 1.76	2.55 ± 0.47
	Medium	**62.16 ± 3.22**	30.24 ± 5.35	11.55 ± 2.06	9.01 ± 0.22	10.58 ± 1.00	58.15 ± 3.10	11.53 ± 2.13	11.23 ± 0.64	11.95 ± 1.17
Dynamics	Expert	46.66 ± 0.96	**55.28 ± 4.27**	23.24 ± 1.53	19.48 ± 0.53	15.65 ± 1.23	46.94 ± 4.31	14.77 ± 1.37	14.56 ± 1.52	16.86 ± 1.30
	Medium	44.54 ± 3.73	**53.14 ± 5.53**	19.52 ± 1.05	15.72 ± 0.23	16.54 ± 0.63	42.87 ± 2.45	16.39 ± 3.78	16.03 ± 1.81	16.70 ± 1.27
Overall		**64.34**	47.67	30.10	25.26	28.44	59.84	36.93	18.46	19.37

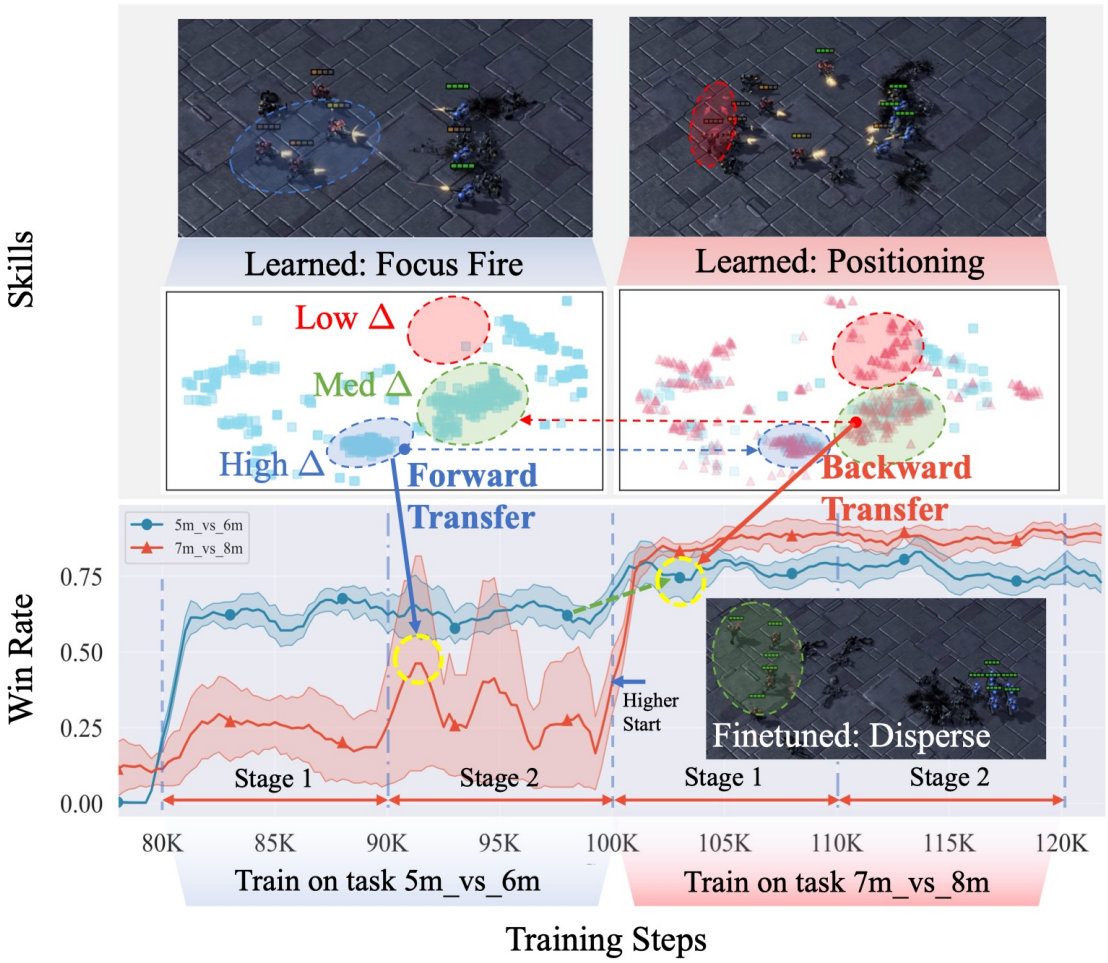
- COMAD(Our) performs the best;
- Skill discovery methods with static skill libraries (ODIS, HiSSD): plasticity loss;
- Monolithic (MT, FS, FT, EWC, Re): catastrophic forgetting due to gradient conflict;
- Parameter isolation (OWL): poor knowledge reuse and transfer

Main Results



- COMAD(Our) performs the best;
- Skill discovery methods with static skill libraries (ODIS, HiSSD): plasticity loss;
- Monolithic (MT, FS, FT, EWC): catastrophic forgetting due to gradient conflict;
- Parameter isolation (OWL): poor knowledge reuse and transfer

Skill Transfer Analysis



- Forward transfer via common skill: Focus Fire
- Backward transfer via finetuned skill: Disperse
- Learn new skill: Positioning

Ablation Study

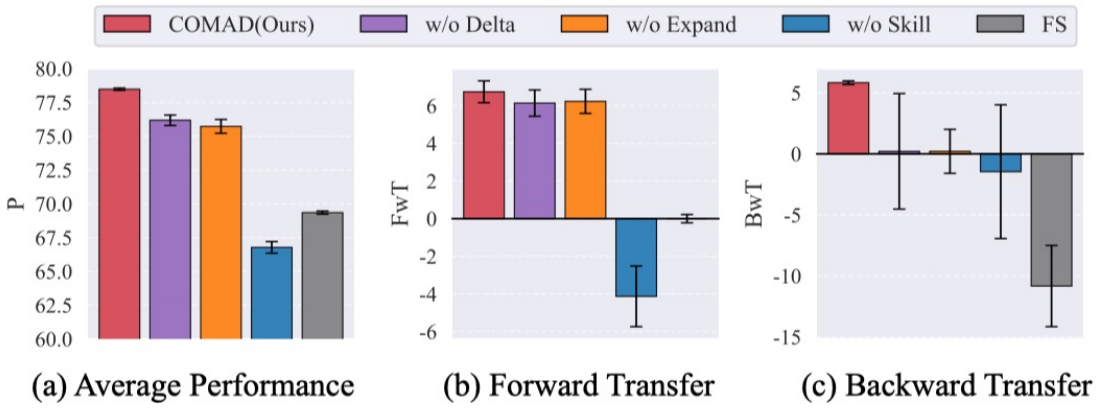


Figure 5. Metrics of ablation studies on CN-expert task.

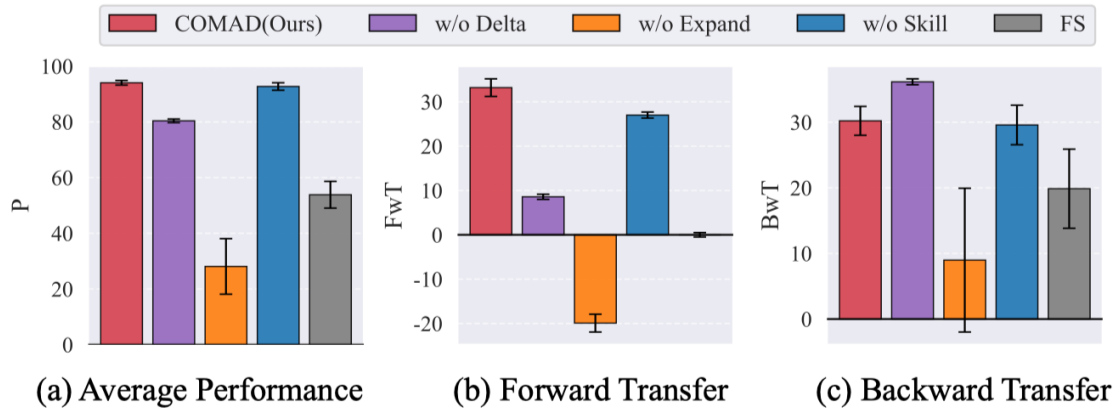


Figure 7. Metrics of ablation studies on Marines-expert task.

CN (simple)

- Skill partitioning (w/o Skill) affects both learning and bidirectional transfer the most
- Skill library expansion (w/o Expand) and reusability estimation (w/o Delta) affect mainly backward transfer

SMAC-marines (complex)

- Skill library expansion (w/o Expand) is the most important
- Reusability estimation (w/o Delta) affect mainly forward transfer on complex tasks

Sensitivity Analysis

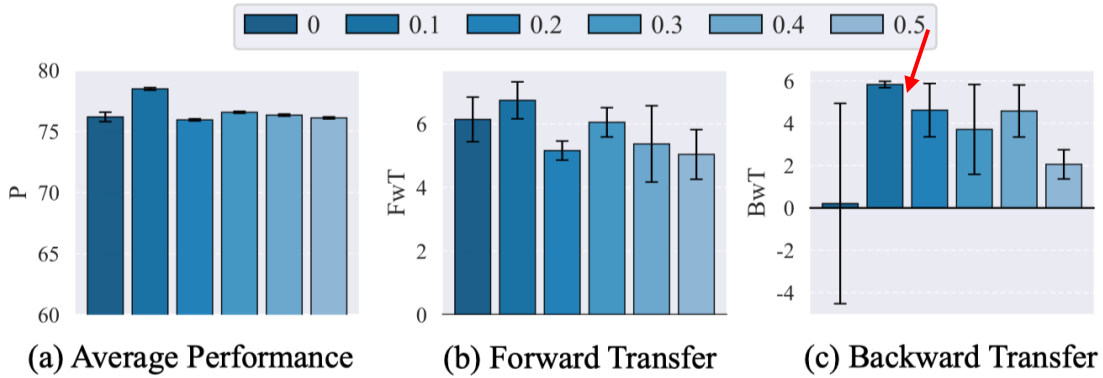


Figure 8. Sensitivity analysis of the noise scale σ_s on CN-expert task.

$$E_k(s) \approx d_k(s)/\mathcal{N}(s, \sigma_s I)$$

10: **if** $\exists k : \hat{d}_k > d_0$ **then**
11: Reuse old heads: $k^* \leftarrow \arg\max_k \mathbb{E}[d_k(s)]$.

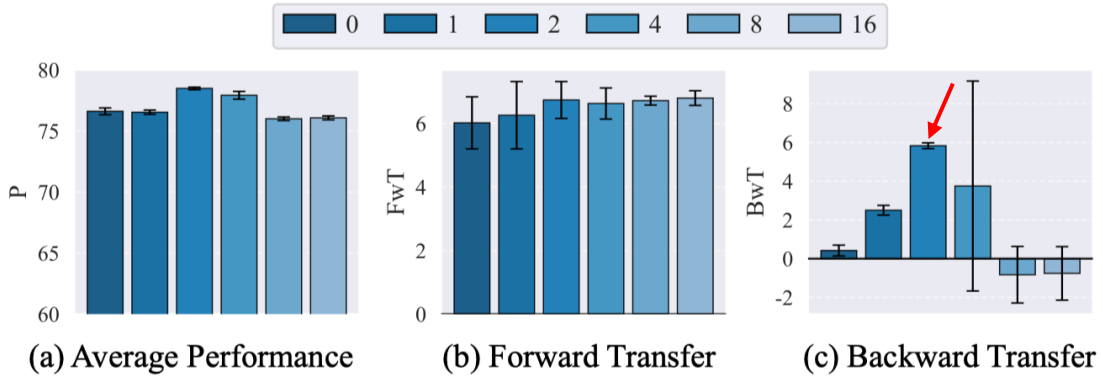


Figure 9. Sensitivity analysis of the confidence threshold d_0 on CN-expert task.

- Non-zero σ_s for identifying tasks
- Low threshold -> unselective
- High threshold -> strict

Final Remarks

- The space of coordination skills in MA task stream grows exponentially

- A principled framework for Continual Offline Multi-agent Skill Discovery via Skill Partition and Reuse
  - VAE-based skill encoder and library
  - Skill-augmented objective & reusability estimation
  - Theoretically grounded

- Future:
  - LLM based semantic reusability + visitation frequency
  - Embodied AI

Thanks!