MTRL-CG: Multi-Task Reinforcement Learning Method with Spectral Clustering-Based Task Grouping

Wenjia Meng[1], **Teng Zhang**[1*], **Haoliang Sun**[1], **Yilong Yin**[1]

[1]School of Software, Shandong University, Jinan, China
wjmeng@sdu.edu.cn, zhang_teng@mail.sdu.edu.cn, {haolsun,ylyin}@sdu.edu.cn

Abstract

Multi-task reinforcement learning (RL) aims to enhance agent performance across multiple tasks by enabling effective knowledge transfer. However, these methods adopt a fully shared policy across all tasks without explicitly distinguishing between related and conflicting ones, making them suffer from negative interference issue, where updates beneficial to one task adversely affect others and lead to degraded overall performance. In this paper, we propose a multi-task reinforcement learning method with spectral clustering-based task grouping (MTRL-CG), which leverages spectral clustering to group related tasks and separate conflicting ones, enabling group-wise policy learning to mitigate negative interference. We first quantify inter-task affinity by measuring the influence of task-specific updates on others within a shared model, and construct an affinity matrix to capture these relationships. Spectral clustering is then applied to partition tasks via spectral embedding and k-means clustering. Each task group is trained with a dedicated policy network to promote focused learning. Built upon the Soft Actor-Critic (SAC) algorithm, MTRL-CG can be readily integrated into existing SAC-based multi-task RL methods. Extensive experiments on the Meta-World benchmark demonstrate the effectiveness of the proposed MTRL-CG method.

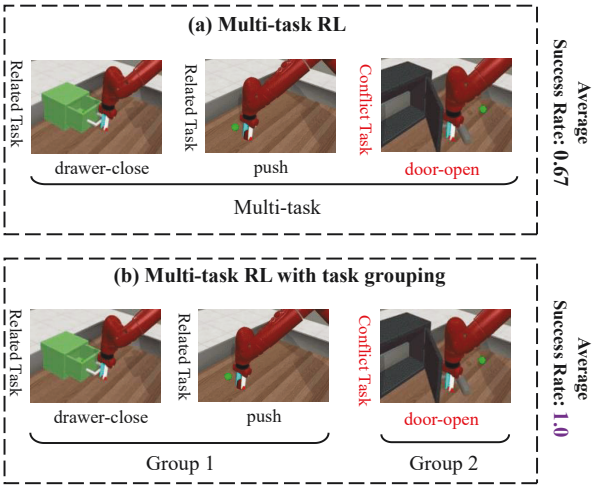

Figure 1: Multi-task RL with vs. without task grouping: (a) Shared policy suffers from conflicts (0.67); (b) Grouped policies reduce conflicts and improve performance (1.0).

Introduction

Deep Reinforcement Learning (RL) (Sutton, Barto et al. 1998) has demonstrated remarkable success across a range of single-task decision-making domains, *e.g.*, autonomous driving (Kiran et al. 2021; El Sallab et al. 2017), robotics (Kober et al. 2013; Zhao et al. 2020), finance (Hambly et al. 2023), and personalized recommendation (Afsar et al. 2022). While significant advances have been made, most RL methods remain restricted to single-task scenarios and struggle to generalize learned skills for solving complex tasks (Lan et al. 2023; He et al. 2024; Xu et al. 2020; Georgiev et al. 2025). This limitation hinders their effectiveness in training agents for real-world applications (Yu et al. 2020b; Yang et al. 2020; Sodhani et al. 2021).

Multi-task RL methods are proposed to improve agent performance across multiple tasks by enabling knowledge transfer, enhancing their applicability to real-world scenarios. These methods can be broadly categorized into four classes: knowledge transfer, structure sharing, representation learning, and gradient manipulation. *Knowledge transfer* methods (Teh et al. 2017; Xu et al. 2020; Arora et al. 2018) aim to distill the knowledge or behaviors of single-task experts into a unified policy that generalizes across tasks. *Structure sharing* methods (Devin et al. 2017; Yang et al. 2020; Sun et al. 2022; He et al. 2024) aim to construct compositional policies by reusing shared modules across tasks, thereby promoting parameter efficiency and enhancing policy generalization. *Representation learning* methods (Sodhani et al. 2021; Lan et al. 2023; D'Eramo et al. 2020) learn shared representations to improve policy and value learning across tasks. *Gradient manipulation* methods (Yu et al. 2020a; Liu et al. 2021) learn a generalizable policies by controlling gradient direction and scale during multi-task RL training. However, these methods ***employ a fully shared model across all tasks without explicitly distinguishing between related and conflicting ones, and thus still suffer from negative interference problem***, where ben-

*Corresponding author.

eficial updates to one task harm others due to task conflicts, ultimately resulting in overall performance degradation (Teh et al. 2017; Yang et al. 2020; Sodhani et al. 2021).

In this paper, we propose a **M**ulti-**T**ask **R**einforcement **L**earning method with spectral **C**lustering-based task **G**rouping (MTRL-CG), which utilizes spectral clustering to organize related tasks into groups and separate conflicting ones, enabling group-specific policy learning that mitigates negative interference (see Figure 1). Specifically, we begin by quantifying inter-task affinity, which captures the extent to which parameter updates from one task influence the optimization objectives of others under a shared model. These affinities are then aggregated into an inter-task affinity matrix that characterizes the mutual influence across all tasks. Based on this matrix, we perform spectral clustering to partition tasks into groups, consisting of two key stages: constructing spectral representations and conducting k-means clustering in the spectral space to assign tasks to groups. Based on the obtained task groups, we conduct group-wise training, where a separate policy network is optimized for each group to facilitate focused learning and reduce task interference. Additionally, we provide the pseudocode of the proposed MTRL-CG framework, which is built upon the Soft Actor-Critic (SAC) algorithm for policy and value function optimization. Owing to its SAC-based foundation, MTRL-CG can be seamlessly incorporated into existing multi-task RL methods that utilize SAC as their backbone. Furthermore, extensive experiments on the Meta-World benchmark validate the overall performance of our method, as well as the effectiveness of the task grouping and the validity of the inter-task affinity matrix. Our contributions are summarized as follows,

- To the best of our knowledge, this is the first work to introduce task grouping in multi-task RL, aiming to cluster related tasks and separate conflicting ones in order to mitigate negative interference.
- We propose the MTRL-CG method, a multi-task RL method that leverages spectral clustering to group tasks and performs group-wise training by optimizing separate policy network for each group, thereby enhancing reducing negative interference.
- We conduct extensive experiments on the Meta-World benchmark, and the results demonstrate that our method achieve superior overall performance compared to the state-of-the-art multi-task RL methods.

Related Work

Multi-Task Reinforcement Learning

Multi-task reinforcement learning (RL) improves sample efficiency by jointly training a policy across tasks to promote knowledge sharing and reuse. Existing multi-task RL methods can be grouped into four major directions. Knowledge distillation approaches (Teh et al. 2017; Xu et al. 2020; Arora et al. 2018) focus on aggregating task-specific expertise into a centralized policy; for example, Distral (Teh et al. 2017) aligns task policies with a shared prior using KL regularization, while KTM-DRL (Xu et al. 2020) merges offline and online learning signals. Structure sharing approaches (Devin et al. 2017; Yang et al. 2020; Sun et al. 2022; He et al. 2024) improve scalability by assembling policies from reusable components, as seen in soft-Modularization (Yang et al. 2020) and PaCo (Sun et al. 2022). Shared representation methods (Sodhani et al. 2021; Lan et al. 2023; D'Eramo et al. 2020) emphasize learning task-invariant features to facilitate joint learning across tasks, for instance, CARE (Sodhani et al. 2021) incorporates language-guided embeddings, and CMTA (Lan et al. 2023) combines contrastive and temporal cues. Finally, gradient-level techniques (Yu et al. 2020a; Liu et al. 2021), such as PCGrad (Yu et al. 2020a), modify gradient updates to enhance stability during joint optimization. Unlike these methods, our approach explicitly separates related and conflicting tasks to alleviate negative interference.

Multi-task Learning

Multi-task learning (MTL) improves performance by jointly training multiple tasks. Existing methods are categorized into multi-task architecture (MTA) and multi-task optimization (MTO). MTA methods focus on parameter sharing: hard sharing uses a common backbone with task-specific heads (*e.g.*, Deep Relationship Networks (Long et al. 2017), UberNet (Kokkinos 2017)), while soft sharing keeps separate backbones with information exchange (*e.g.*, Neural Network Parser (Duong et al. 2015), Cross-Stitch Networks (Misra et al. 2016)). MTO methods aim to reduce gradient conflicts and balance training, such as GradNorm (Chen et al. 2018) and PCGrad (Yu et al. 2020a). Unlike these methods for MTL, our work focuses on multi-task RL, where tasks involve sequential decision making.

Grouping in Multi-task Learning

Task grouping in multi-task learning clusters related tasks to promote intra-group knowledge sharing and reduce interference from unrelated ones (Gao et al. 2024; Kang et al. 2011; Kumar and Daume III 2012). Standley et al.(Standley et al. 2020) improved performance by assigning groups to separate networks. TAG (Fifty et al. 2021) forms groups by evaluating gradient influence across tasks, while MTG-Net (Song et al. 2022) uses a meta-learning framework to predict performance gains for guiding grouping. DMTG (Gao et al. 2024) models grouping as differentiable pruning via a categorical distribution, enabling one-shot grouping and training with high-order affinity modeling. This work fills the gap between the task grouping and multi-task RL.

Preliminaries

Reinforcement Learning

Reinforcement learning (RL) (Sutton, Barto et al. 1998) is commonly formulated as a Markov decision process (MDP) (Puterman 2014), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ represent the state and action spaces, respectively; $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denotes the reward function; $\mathcal{P}(s' \mid s, a)$ characterizes the transition dynamics; and $\gamma \in [0, 1]$ is the discount factor. In RL, an agent interacts

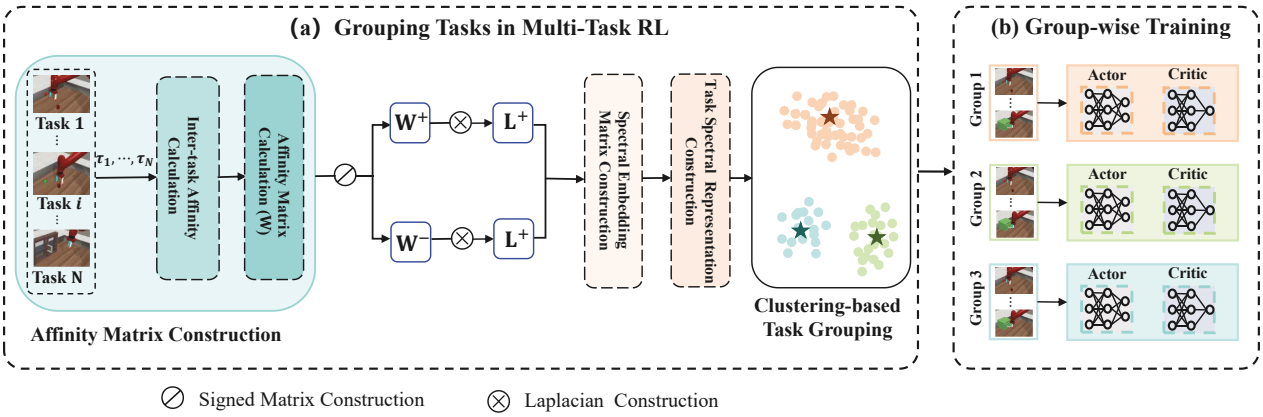

Figure 2: Framework of proposed MTRL-CG method. (a) Grouping Tasks in Multi-Task RL. (b) Grouping-wise Training.

with the environment in discrete time steps. At each timestep t, the agent observes the current state s_t, selects an action a_t according to a policy $\pi(a_t \mid s_t)$, receives a reward r_t, and transitions to the next state s_{t+1} according to the transition dynamics $\mathcal{P}$. The goal of RL is to learn a policy π that maximizes the expected cumulative discounted return:

$$J_\pi = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]. \tag{1}$$

Soft Actor-Critic (SAC) is a widely adopted and representative reinforcement learning algorithm, which we employ to train the policy in this paper. In SAC, the policy function $\pi_\phi(a|s)$ and the Q function $Q_\theta(s,a)$ are parameterized by ϕ and θ, respectively. The objective of policy function is

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim \mathcal{D}} \Big[\mathbb{E}_{a_t \sim \pi_\phi} \left[\alpha \log \pi_\phi(a_t|s_t) - Q_\theta(s_t,a_t) \right] \Big]. \tag{2}$$

where α is an entropy temperature coefficient. The Q function $Q_\theta(s,a)$ is optimized by minimizing the objective:

$$J_Q(\theta) = \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}} \Big[\Big(Q_\theta(s_t,a_t) - \big(r_t + \gamma \mathbb{E}_{s_{t+1}\sim\mathcal{P}}[V_{\bar{\theta}}(s_{t+1})]\big) \Big)^2 \Big], \tag{3}$$

where γ represents the discount rate, and $V_{\bar{\theta}}(s_{t+1}) = \mathbb{E}_{a_{t+1}\sim\pi_\phi} \big[Q_{\bar{\theta}}(s_{t+1},a_{t+1}) - \alpha \log \pi_\phi(a_{t+1}|s_{t+1}) \big]$, and $Q_{\bar{\theta}}(s_{t+1},a_{t+1})$ denotes the target Q function with $\bar{\theta}$.

Multi-Task Reinforcement Learning

In multi-task reinforcement learning, we consider a set of M tasks, each sampled from a distribution $p(\mathcal{T})$, where each task $\mathcal{T}$ corresponds to a distinct Markov decision process (MDP). Given the prevalence and superior performance of SAC in multi-task reinforcement learning, we adopt SAC as the foundation of our approach and detail its optimization objectives under the multi-task RL. Specifically, SAC in multi-task setting aims to learn a policy that maximizes the average expected return across tasks sampled from the distribution $p(\mathcal{T})$, formulated as:

$$J_\pi(\phi) = \mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})} \left[J_{\pi,\mathcal{T}}(\phi) \right], \tag{4}$$

where $J_{\pi,\mathcal{T}}(\phi)$ denotes the task-specific policy objective defined in Eq.(2). Similarly, the Q function is optimized by minimizing the average Q loss over tasks:

$$J_Q(\theta) = \mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})} \left[J_{Q,\mathcal{T}}(\theta) \right], \tag{5}$$

where $J_{Q,\mathcal{T}}(\theta)$ is the Q function objective for task $\mathcal{T}$, as defined in Eq. (3).

Proposed Method

In this section, we propose a multi-task reinforcement learning method with spectral clustering-based task grouping (MTRL-CG), which clusters related tasks and isolates conflicting ones to reduce negative interference. We begin by outlining the overall MTRL-CG framework, which consists of a task grouping stage and a group-wise multi-task reinforcement learning stage. We then introduce the task grouping process, including inter-task affinity computation, spectral embedding, and clustering. Next, we describe the group-wise training procedure, where a multi-task SAC algorithm is used to train a shared policy within each group. Finally, we present the detailed pseudocode of the proposed method.

Framework of Proposed Method

This section outlines the proposed MTRL-CG framework (Figure 2), which consists of two main components: grouping tasks and group-wise training. In the grouping stage, we first quantify the mutual influence among tasks by computing an affinity score for each task pair, which is assembled into an inter-task affinity matrix. Based on this matrix, we perform spectral clustering to partition tasks into groups, which involves two key steps: constructing task spectral representations and applying k-means clustering to form the final groups. Specifically, we perform eigenvalue decomposition on the affinity matrix and use the resulting eigenvectors to obtain a low-dimensional spectral embedding for each task. These embeddings are then clustered using the k-means algorithm, allowing related tasks to be grouped together. In the group-wise training stage, each task group is assigned a dedicated SAC learner to enable focused policy learning, promoting intra-group knowledge sharing while reducing interference from unrelated tasks.

Grouping Tasks in Multi-Task RL

This section presents the task grouping procedure in multi-task RL, which aims to cluster related tasks together while separating conflicting ones. Specifically, we first describe the construction of the inter-task affinity matrix, which quantifies the mutual influence among tasks. Based on the constructed affinity matrix, we adopt a spectral clustering approach to group tasks, which involves two steps: constructing task spectral embeddings and applying k-means clustering to form task groups.

Inter-Task Affinity Matrix Construction In multi-task RL, all tasks are jointly optimized over shared policy and Q-functions (see Eq. (4) and (5)), enabling implicit knowledge transfer and mutual influence among tasks. To quantify such interdependence, we propose to measure *inter-task affinity* by assessing the extent to which the gradient updates from one task on the shared function affect the objective values of other tasks. We adopt the Q function for this analysis, as Q values provide a direct evaluation of policy's performance.

Given a set of N reinforcement learning tasks $\mathcal{S} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N\}$, a shared Q function $Q_\theta(s,a)$ is learned by jointly optimizing the objective in Eq. (5). To analyze inter-task affinity, we perform a single-task update using task $\mathcal{T}_i$, yielding updated parameters:

$$\theta_{\mathcal{T}_i} := \theta - \eta \nabla_\theta J_{Q,\mathcal{T}_i}(\theta), \tag{6}$$

where η is the learning rate and $J_{Q,\mathcal{T}_i}(\theta)$ is the Q function objective for task $\mathcal{T}_i$ (Eq. (3)). We quantify the affinity from task $\mathcal{T}_i$ to task $\mathcal{T}_j$ by measuring the relative change in the Q value of $\mathcal{T}_j$ induced by applying the gradient update from $\mathcal{T}_i$ to the shared parameters:

$$W_{i,j} = \frac{Q_{\theta_{\mathcal{T}_i}}(s,a) - Q_\theta(s,a)}{|Q_\theta(s,a)|},\ (s,a) \sim \mathcal{T}_j, \tag{7}$$

where $|\cdot|$ denotes the ℓ_1 norm, (s,a) is the state and action pairs of task $\mathcal{T}_j$. For all $i,j \in \{1,\ldots,N\}$, $W_{i,j}$ is defined by Eq. (7) for $i \neq j$, and $W_{i,i} = 0$. With the above affinity, the inter-task affinity matrix $\mathbf{W}$ is:

$$\mathbf{W} = \begin{bmatrix} W_{1,1} & \cdots & W_{1,N} \\ \vdots & \ddots & \vdots \\ W_{N,1} & \cdots & W_{N,N} \end{bmatrix}. \tag{8}$$

To enable spectral clustering for task grouping, we construct a symmetric affinity matrix by averaging the original affinity matrix $\mathbf{W}$ with its transpose $\mathbf{W}^\top$. This yields a symmetric estimate of mutual influence between tasks:

$$\mathbf{W} = \frac{\mathbf{W} + \mathbf{W}^\top}{2}, \tag{9}$$

where $\mathbf{W}$ is the final inter-task affinity matrix for clustering.

Task Spectral Representation Construction Given the inter-task affinity matrix $\mathbf{W}$ in Eq. (9), we employ a signed spectral embedding method (Cucuringu et al. 2019) to extract task representations for clustering-based grouping. Specifically, we first build a signed matrix and compute its signed Laplacian, followed by spectral embedding matrix construction to derive task representations in the spectral space. These processes are as follows,

- *Signed Matrix Construction.* We construct the signed affinity matrices ($\mathbf{W}^+$ and $\mathbf{W}^-$) by separating the positive and negative components of $\mathbf{W}$ as follows:
$$W^+_{i,j} = \max(W_{i,j}, 0), \quad W^-_{i,j} = \max(-W_{i,j}, 0). \tag{10}$$
Here, $\mathbf{W}^+$ captures cooperative relationships, while $\mathbf{W}^-$ models conflicting ones.
- *Signed Laplacian Construction.* We then construct the corresponding signed Laplacians:
$$\mathbf{L}^+ = \mathbf{D}^+ - \mathbf{W}^+, \quad \mathbf{L}^- = \mathbf{D}^- - \mathbf{W}^-, \tag{11}$$
where $\mathbf{D}^+$ and $\mathbf{D}^-$ are diagonal matrices with $D^+_{ii} = \sum_j W^+_{ij}$ and $D^+_{ij} = 0$ for $i \neq j$; similarly for $\mathbf{D}^-$.
- *Spectral Embedding Matrix Construction.* We compute the spectral embedding by solving the generalized eigenvalue problem:
$$(\mathbf{L}^+ + \tau^- \mathbf{D}^-)\mathbf{v} = \lambda(\mathbf{L}^- + \tau^+ \mathbf{D}^+)\mathbf{v}, \tag{12}$$
where τ^+ and τ^- are hyperparameters, and λ, $\mathbf{v}$ denote the eigenvalue and eigenvector. The first K smallest non-zero eigenvectors form the spectral embedding matrix $\mathbf{V} \in \mathbb{R}^{N \times K}$:
$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_K], \tag{13}$$
where each row represents a task in the spectral space.
- *Task Representation Construction.* Each row of the embedding matrix $\mathbf{V}$ (Eq. (13)) serves as a low-dimensional spectral representation of a task:
$$\mathbf{u}_i = (\mathbf{V}_{i,1}, \cdots, \mathbf{V}_{i,j}, \cdots, \mathbf{V}_{i,K}), \tag{14}$$
where $\mathbf{u}_i$ denotes the representation of task $\mathcal{T}_i$, and $\mathbf{V}_{i,j}$ is the element at row i and column j, with $i \in \{1, \cdots, N\}$.

k-means Clustering-based Task Grouping Given the task representations $(\mathbf{u}_1, \cdots, \mathbf{u}_N)$ (Eq. (14)), we employ the k-means clustering algorithm to partition the N tasks into K groups as follows.

- *Initialization.* For the set of low-dimensional task embeddings $(\mathbf{u}_1, \cdots, \mathbf{u}_N)$, we initialize K cluster centers $\mathcal{C} = \{c_1, \cdots, c_K\}$ using the k-means++ algorithm (Arthur and Vassilvitskii 2007).
- *Group Assignment.* For each $i \in \{1, \cdots, K\}$, task $\mathcal{T}_\mathbf{u}$ is assigned to task group G_i if its embedding $\mathbf{u}$ is closer to the center c_i than to any other center c_j with $j \neq i$; *i.e.*,
$$G_i = \{\mathcal{T}_\mathbf{u} \mid \|\mathbf{u} - c_i\| < \|\mathbf{u} - c_j\|,\ \forall j \neq i\}, \tag{15}$$
where $\|\cdot\|$ denotes the Euclidean distance.
- *Cluster Center Update.* For each $i \in \{1, \cdots, K\}$, we set cluster center c_i be the center of all related embeddings:
$$c_i = \frac{1}{|G_i|} \sum_{\mathcal{T}_\mathbf{u} \in G_i} \mathbf{u}, \tag{16}$$
where $\mathcal{T}_\mathbf{u}$ denotes the task represented by embedding $\mathbf{u}$.
- *Repeat.* We repeat the assignment and update steps until the task groups become stable:
$$\mathcal{G} = \{G_1, G_2, \ldots, G_K\}. \tag{17}$$

As a result, the N tasks are clustering into K groups, yielding the final task grouping result $\mathcal{G}$.

Group-wise Training in Multi-Task RL

Using the above task grouping strategy, the task set $\mathcal{S} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N\}$ is clustered into K groups $\mathcal{G} = \{G_1, G_2, \ldots, G_K\}$. In the following, we describe how to leverage these grouping results to perform group-wise reinforcement learning training, aiming to mitigate negative interference among tasks.

Given the task groups $\mathcal{G} = \{G_1, G_2, \ldots, G_K\}$, we assign a dedicated policy function π_{ϕ_k} and Q-function Q_{θ_k} to each group G_k $(1 \leq k \leq K)$, enabling joint optimization of the tasks within each group. For each group G_k, we adopt the multi-task SAC algorithm (Haarnoja et al. 2018) to learn the corresponding policy π_{ϕ_k} and Q-function Q_{θ_k}. The policy is optimized by maximizing the expected task-specific objective over tasks in the group:

$$J_\pi(\phi_k) = \mathbb{E}_{\mathcal{T}\sim G_k}\left[J_{\pi,\mathcal{T}}(\phi_k)\right], \tag{18}$$

where $J_{\pi,\mathcal{T}}(\phi_k)$ denotes the policy objective for task $\mathcal{T}$, as defined in Eq. (2). Similarly, the Q-function is optimized by minimizing the expected loss over tasks in G_k:

$$J_Q(\theta_k) = \mathbb{E}_{\mathcal{T}\sim G_k}\left[J_{Q,\mathcal{T}}(\theta_k)\right], \tag{19}$$

where $J_{Q,\mathcal{T}}(\theta_k)$ is the Q-function objective for task $\mathcal{T}$, as defined in Eq. (3). By training a dedicated agent for each group, the proposed approach facilitates intra-group knowledge sharing while mitigating inter-group interference, thereby enabling more efficient multi-task learning.

Pseudocode of the Proposed Method

The overall procedure of the proposed MTRL-CG method is summarized in Algorithm 1, which consists of grouping task process and group-wise training process.

Algorithm 1: Proposed MTRL-CG method

Require : Task set $\{\mathcal{T}_i\}_{i=1}^N$, estimation interval C, timesteps T_g, T_m, learning rate η;
Initialize: Buffers $\mathcal{D}_1, \ldots, \mathcal{D}_N$, inter-task affinity matrix $\mathbf{W}$, policy π_ϕ and Q network Q_θ;

// Grouping Tasks in Multi-Task RL
for $t = 1, 2, \ldots, T_g$ **do**
  Data Collection: Collect trajectory data $\tau_1, \ldots, \tau_N$ and store them in $\mathcal{D}_1, \ldots, \mathcal{D}_N$;
  Update: Update Q_θ, π_ϕ by multi-task SAC;
  if $t \bmod C == 0$ **then**
    Compute inter-task affinity matrix $\mathbf{W}_C$ (9);
    $\mathbf{W} = \mathbf{W} + \mathbf{W}_C$;
  end
end
Obtain groups $\mathcal{G} = \{G_1, \ldots, G_K\}$ by Eq. (17);
// Group-wise Training
Initialize π_{ϕ_k} and Q_{θ_k} $(1 \leq k \leq K)$;
for $G_k \in \{G_1, \ldots, G_K\}$ **do**
  for $t = 1, 2, \ldots, T_m$ **do**
    Update $\pi_{\phi_k}, Q_{\theta_k}$ by Eq. (18) and (19):
      $\theta_k \leftarrow \theta_k - \eta\hat{\nabla}_{\theta_k} J_Q(\theta_k)$,
      $\phi_k \leftarrow \phi_k - \eta\hat{\nabla}_{\phi_k} J_\pi(\phi_k)$;
  end
end

Experimental Results

In this section, we conduct experiments to evaluate our proposed method. In the following, we first describe the experimental setup including the environments and hyperparameter settings. To evaluate the effectiveness of our proposed grouping strategy, we compare the state-of-the-art methods with and without our grouping strategy. Then we conduct the ablation study about the different group strategies and group number. Lastly, we analyze the affinity matrices and t-SNE visualizations to qualitatively validate the effectiveness of the task grouping achieved by our method.

Experimental Setup

Environments. Meta-World (Yu et al. 2020b) is an open-source simulated environment for benchmarking multi-task reinforcement learning, featuring a diverse set of manipulation tasks with varying levels of complexity based on the Sawyer robotic arm. We conduct experiments on two standard environments: MT10 and MT50, comprising 10 and 50 distinct manipulation tasks, respectively. MT10 and MT50 include two settings: *fixed* and *mixed*. In the fixed setting, each task is associated with a fixed initial and goal position. In the mixed setting, the initial and goal positions are randomly sampled from a pool of 50 predefined configurations.

Hyperparameter Settings. For the MT10 and MT50 tasks, we follow the well-defined *success rate* in Meta-World as the evaluation protocol (Yu et al. 2020b). The settings for the compared methods (*i.e.*, Multi-task SAC (MT-SAC) (Yu et al. 2020b), CARE (Sodhani et al. 2021), CMTA (Lan et al. 2023), and PaCo (Sun et al. 2022)) follow those reported in their original publications to ensure fair comparison. Results are averaged across 3 random runs. Our code is available at: https://github.com/zhangt603/MTRL-CG.git.

Comparison with State-of-the-art Methods

This section evaluates the effectiveness of MTRL-CG by comparing state-of-the-art MTRL methods (i.e., MT-SAC, CARE, CMTA, and PaCo) with and without its integration.

The experimental results are presented in Figure 3 and Table 1. As illustrated in Figure 3, the method incorporating the grouping strategy attains the same success rate with significantly fewer timesteps compared to the method without grouping. For example, under the MT10-Fixed setting, MT-SAC converges at approximately 0.30×10^7 timesteps, while MT-SAC with MTRL-CG converges at around 0.13×10^7 timesteps. Moreover, the proposed task grouping strategy consistently improves the final success rates of existing methods. For instance, CARE with the grouping strategy achieves an average success rate improvement of approximately 0.15 on the MT10-Fixed environment, and around 0.10 on the MT10-Mixed environment compared to the original CARE. Table 1 presents mean $\pm$ standard deviation over three seeds, with "Max" and "Final" indicating peak and final success rates, respectively. In most cases, it enhances ex-

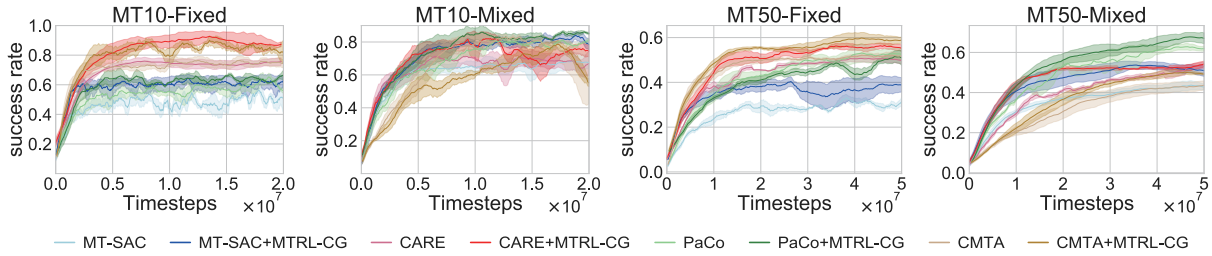

Figure 3: Success rate curves of different methods with and without our MTRL-CG strategy on MT10 and MT50. Shaded areas show half the standard deviation across three random runs. The X- and Y-axes denote timesteps and success rate, respectively.

Method	MT10-Fixed		MT10-Mixed		MT50-Fixed		MT50-Mixed	
	Max	Final	Max	Final	Max	Final	Max	Final
MT-SAC	0.63 ± 0.02	0.50 ± 0.11	0.70 ± 0.04	0.68 ± 0.03	0.35 ± 0.01	0.34 ± 0.02	0.45 ± 0.02	0.42 ± 0.01
MT-SAC+MTRL-CG	$\mathbf{0.67 \pm 0.02}$	$\mathbf{0.64 \pm 0.02}$	$\mathbf{0.87 \pm 0.01}$	$\mathbf{0.77 \pm 0.04}$	$\mathbf{0.42 \pm 0.01}$	$\mathbf{0.38 \pm 0.02}$	$\mathbf{0.55 \pm 0.02}$	$\mathbf{0.50 \pm 0.03}$
CARE	0.77 ± 0.02	0.77 ± 0.02	0.74 ± 0.03	0.69 ± 0.06	0.51 ± 0.01	0.49 ± 0.01	0.54 ± 0.01	0.54 ± 0.02
CARE+MTRL-CG	$\mathbf{0.97 \pm 0.02}$	$\mathbf{0.87 \pm 0.02}$	$\mathbf{0.87 \pm 0.02}$	$\mathbf{0.73 \pm 0.02}$	$\mathbf{0.58 \pm 0.01}$	$\mathbf{0.55 \pm 0.01}$	$\mathbf{0.55 \pm 0.01}$	$\mathbf{0.55 \pm 0.01}$
CMTA	0.77 ± 0.02	0.73 ± 0.02	$\mathbf{0.88 \pm 0.02}$	$\mathbf{0.58 \pm 0.12}$	0.54 ± 0.01	0.52 ± 0.01	0.45 ± 0.02	0.44 ± 0.02
CMTA+MTRL-CG	$\mathbf{0.93 \pm 0.02}$	$\mathbf{0.83 \pm 0.09}$	0.83 ± 0.01	0.53 ± 0.01	$\mathbf{0.63 \pm 0.01}$	$\mathbf{0.59 \pm 0.01}$	$\mathbf{0.52 \pm 0.00}$	$\mathbf{0.49 \pm 0.00}$
PaCo	0.73 ± 0.05	0.57 ± 0.02	0.85 ± 0.02	0.81 ± 0.01	$\mathbf{0.57 \pm 0.03}$	$\mathbf{0.52 \pm 0.03}$	0.66 ± 0.02	0.62 ± 0.01
PaCo+MTRL-CG	$\mathbf{0.73 \pm 0.06}$	$\mathbf{0.67 \pm 0.05}$	$\mathbf{0.89 \pm 0.01}$	$\mathbf{0.84 \pm 0.01}$	0.56 ± 0.00	0.51 ± 0.03	$\mathbf{0.69 \pm 0.02}$	$\mathbf{0.66 \pm 0.02}$

Table 1: Success rates of different methods with and without our MTRL-CG strategy on MT10 and MT50.

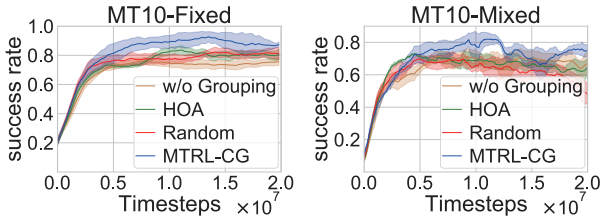

Figure 4: Success rate curves of CARE with various grouping strategies on MT10-Fixed and MT10-Mixed.

Grouping Strategy	MT10-Fixed		MT10-Mixed	
	Max	Final	Max	Final
w/o Grouping	0.77 ± 0.02	0.77 ± 0.02	0.74 ± 0.03	0.69 ± 0.06
HOA	0.87 ± 0.02	0.80 ± 0.04	0.76 ± 0.01	0.66 ± 0.04
Random	0.83 ± 0.02	0.80 ± 0.04	0.72 ± 0.02	0.47 ± 0.04
MTRL-CG	$\mathbf{0.97 \pm 0.02}$	$\mathbf{0.87 \pm 0.02}$	$\mathbf{0.87 \pm 0.02}$	$\mathbf{0.73 \pm 0.02}$

Table 2: Success rates of CARE with various grouping strategies on MT10-Fixed and MT10-Mixed.

isting multi-task RL methods (MT-SAC, CARE, CMTA, and PaCo) by improving both maximum and final success rates. The relatively poor performance of CMTA with MTRL-CG in MT10-Mixed may be due to its design, which amplifies Q value sensitivity to randomness and results in unstable task grouping. Similarly, the lower success rate of PaCo with MTRL-CG on MT50-Fixed may stem from its sensitivity to hyperparameter settings. Overall, these results demonstrate the effectiveness of the proposed MTRL-CG method.

Ablation Study

We perform an ablation study on the grouping strategy and group number selection using CARE, a representative and widely used multi-task RL method, as the backbone.

Effectiveness of the grouping strategy. To evaluate the effectiveness of our grouping strategy, we compare the success rates of CARE under different task grouping strategies (including w/o Grouping, Random, Higher Order Approximation (HOA) (Standley et al. 2020), and our MTRL-CG), as shown in Figure 4 and Table 2. Figure 4 shows that MTRL-CG achieves a higher final success rate compared to other strategies. In Table 2, MTRL-CG achieves the highest max and final success rates on MT10-Fixed and MT10-Mixed. These results confirm the effectiveness of MTRL-CG.

Comparison of different group number. We evaluate the impact of group number on MTRL-CG by testing configurations of 1, 3, 5, 7, and 10 groups. As shown in Figure 7 and Table 3, MTRL-CG with 3 groups achieves the best overall performance, requiring fewer timesteps to reach higher success rates. These results suggest that group number 3 is a reasonable choice for grouping.

Further Analysis

To further validate the effectiveness of our MTRL-CG strategy, we provide a t-SNE visualization (Figure 5) and task affinity matrix (Figure 6) in the MT10-Fixed environment.

t-SNE visualization. To validate the effectiveness of our grouping results, we present t-SNE visualization of the MTRL-CG groupings based on different multi-task RL backbones, as shown in Figure 5. Each point represents a task's transition feature from the MT10-Fixed environment, with colors indicating task groups. As illustrated in Figure 5, tasks within the same group form compact clusters,

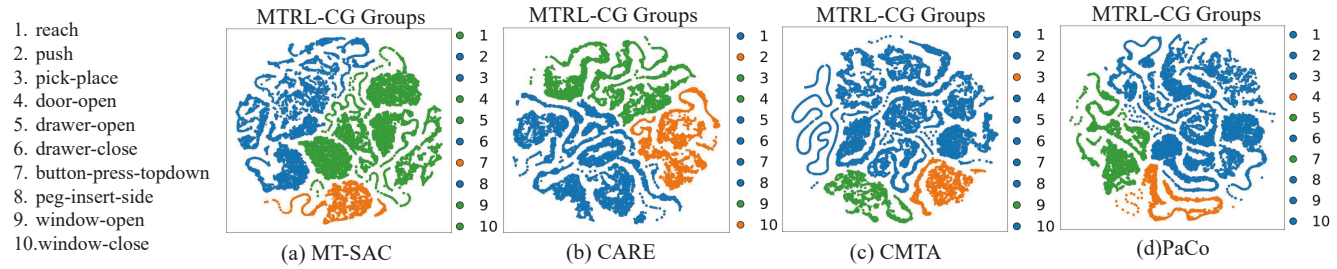

Figure 5: t-SNE visualization of our MTRL-CG grouping results based on MT-SAC, CARE, CMTA, and PaCo. Each point is a task's transition feature from the MT10-Fixed environment, with tasks in the same group indicated by the same color.

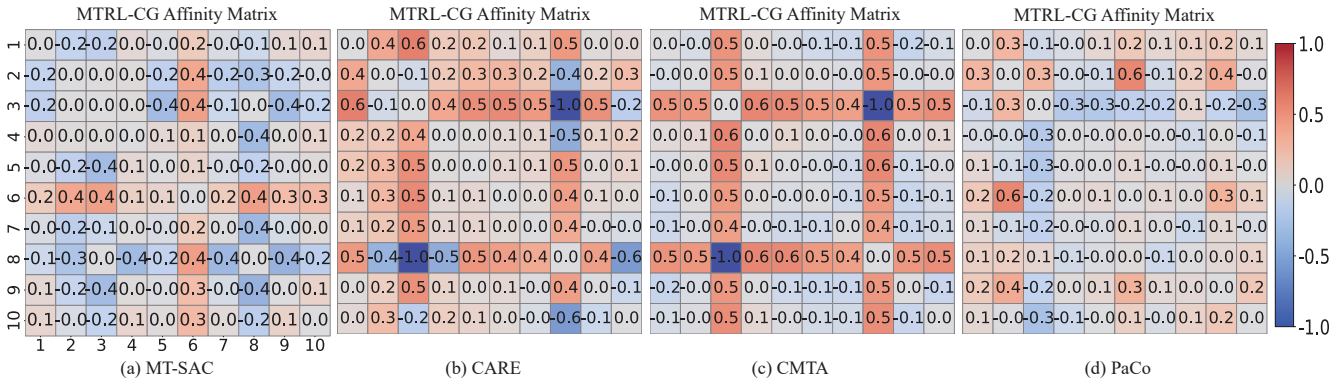

Figure 6: Visualization of inter-task affinity matrices (Eq. (9)) in our MTRL-CG grouping strategy across various multi-task RL backbones (MT-SAC, CARE, CMTA, and PaCo). These tasks are from the MT10-Fixed environment.

while different groups are clearly separated across backbones, demonstrating the effectiveness of MTRL-CG.

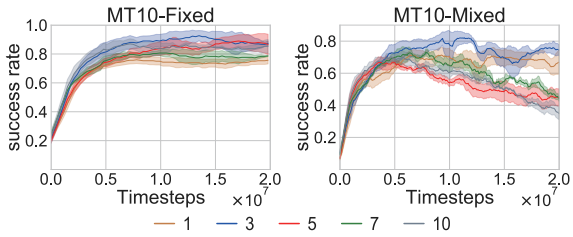

Figure 7: Success rate curves of CARE with MTRL-CG under various group numbers on MT10-Fixed and -Mixed.

Task affinity matrix. To evaluate the effectiveness of the affinity matrices (Eq. (9)), we visualize these matrices in our MTRL-CG grouping strategy across different multi-task RL backbones, with the results shown in Figure 6. The horizontal and vertical axes denote task indices, and each matrix element quantifies the affinity between the corresponding pair of tasks, where warmer (cooler) colors indicate higher (lower) affinity. As illustrated in Figure 6, the affinity matrix produced by our method aligns with intuitive human understanding of task relationships, supporting the reliability of the resulting task groupings.

Group Number	MT10-Fixed		MT10-Mixed	
	Max	Final	Max	Final
1	0.77 ± 0.02	0.77 ± 0.02	0.74 ± 0.03	0.69 ± 0.06
3	$\mathbf{0.97 \pm 0.02}$	$\mathbf{0.87 \pm 0.02}$	$\mathbf{0.87 \pm 0.02}$	$\mathbf{0.73 \pm 0.02}$
5	0.93 ± 0.05	0.87 ± 0.06	0.72 ± 0.06	0.46 ± 0.04
7	0.83 ± 0.02	0.80 ± 0.00	0.79 ± 0.03	0.42 ± 0.02
10	0.93 ± 0.02	0.87 ± 0.05	0.76 ± 0.02	0.34 ± 0.03

Table 3: Success rates of CARE with MTRL-CG under varying group numbers on the MT10-Fixed and MT10-Mixed.

Conclusion

This paper presents MTRL-CG, a novel multi-task RL method that incorporates spectral clustering-based task grouping to address the challenge of negative interference among tasks. By quantifying inter-task affinity through the impact of task-specific updates within a shared model, we construct an affinity matrix that reflects mutual influences among tasks. Spectral embedding followed by k-means clustering is then employed to partition related tasks into the same groups. To facilitate focused and interference-aware learning, each task group is assigned a dedicated policy network. Experimental results on the Meta-World benchmark demonstrate that MTRL-CG consistently improves performance across diverse tasks.

Acknowledgments

This research is supported by the National Natural Science Foundation of China [No. 62206158], the Natural Science Foundation of Shandong Province [No. ZR2022QF097], Shandong University Young Scholar Future Plan,Young Expert of Taishan Scholars [No. tsqn202312026], Shandong Sci-tech SMEs Innovation Project [No. 2024TSGC0740].

References

Afsar, M. M.; Crump, T.; Far, B.; et al. 2022. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, 55(7): 1–38.

Arora, H.; Kumar, R.; Krone, J.; and Li, C. 2018. Multi-task learning for continuous control. *arXiv preprint arXiv:1802.01034*.

Arthur, D.; and Vassilvitskii, S. 2007. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1027–1035.

Chen, Z.; Badrinarayanan, V.; Lee, C.-Y.; and Rabinovich, A. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, 794–803.

Cucuringu, M.; Davies, P.; Glielmo, A.; and Tyagi, H. 2019. SPONGE: A generalized eigenproblem for clustering signed networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 1088–1098.

D'Eramo, C.; Tateo, D.; Bonarini, A.; Restelli, M.; and Peters, J. 2020. Sharing knowledge in multi-task deep reinforcement learning. In *International Conference on Learning Representations*.

Devin, C.; Gupta, A.; Darrell, T.; Abbeel, P.; and Levine, S. 2017. Learning modular neural network policies for multi-task and multi-robot transfer. In *2017 IEEE international conference on robotics and automation (ICRA)*, 2169–2176.

Duong, L.; Cohn, T.; Bird, S.; and Cook, P. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing*, 845–850.

El Sallab, A.; Abdou, M.; Perot, E.; and Yogamani, S. K. 2017. Deep Reinforcement Learning framework for Autonomous Driving. In *Autonomous Vehicles and Machines*.

Fifty, C.; Amid, E.; Zhao, Z.; Yu, T.; Anil, R.; and Finn, C. 2021. Efficiently identifying task groupings for multi-task learning. In *Advances in Neural Information Processing Systems*, 27503–27516.

Gao, Y.; Jiang, S.; Li, M.; Yu, J.-G.; and Xia, G.-S. 2024. Dmtg: One-shot differentiable multi-task grouping. In *International Conference on Machine Learning*.

Georgiev, I.; Giridhar, V.; Hansen, N.; and Garg, A. 2025. PWM: Policy Learning with Multi-Task World Models. In *International Conference on Learning Representations*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 1861–1870.

Hambly, B.; Xu, R.; Yang, H.; et al. 2023. Recent advances in reinforcement learning in finance. *Mathematical Finance*, 33(3): 437–503.

He, J.; Li, K.; Zang, Y.; Fu, H.; Fu, Q.; Xing, J.; and Cheng, J. 2024. Not all tasks are equally difficult: Multi-task deep reinforcement learning with dynamic depth routing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 12376–12384.

Kang, Z.; Grauman, K.; Sha, F.; et al. 2011. Learning with whom to share in multi-task feature learning. In *International Conference on Machine Learning*, 521–528.

Kiran, B. R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A. A.; Yogamani, S.; and Pérez, P. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems*, 23(6): 4909–4926.

Kober, J.; Bagnell, J. A.; Peters, J.; et al. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11): 1238–1274.

Kokkinos, I. 2017. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6129–6138.

Kumar, A.; and Daume III, H. 2012. Learning task grouping and overlap in multi-task learning. In *International Conference on Machine Learning*.

Lan, S.; Zhang, R.; Yi, Q.; Guo, J.; Peng, S.; Gao, Y.; Wu, F.; Chen, R.; Du, Z.; Hu, X.; et al. 2023. Contrastive modules with temporal attention for multi-task reinforcement learning. In *Advances in Neural Information Processing Systems*, 36507–36523.

Liu, B.; Liu, X.; Jin, X.; Stone, P.; and Liu, Q. 2021. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, 18878–18890.

Long, M.; CAO, Z.; Wang, J.; and Yu, P. S. 2017. Learning Multiple Tasks with Multilinear Relationship Networks. In *Advances in Neural Information Processing Systems*, 1594–1603.

Misra, I.; Shrivastava, A.; Gupta, A.; and Hebert, M. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3994–4003.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Sodhani, S.; Zhang, A.; Pineau, J.; et al. 2021. Multi-task reinforcement learning with context-based representations. In *International Conference on Machine Learning*, 9767–9779.

Song, X.; Zheng, S.; Cao, W.; Yu, J.; and Bian, J. 2022. Efficient and effective multi-task grouping via meta learning

on task combinations. In *Advances in Neural Information Processing Systems*, 37647–37659.

Standley, T.; Zamir, A.; Chen, D.; Guibas, L.; Malik, J.; and Savarese, S. 2020. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, 9120–9132.

Sun, L.; Zhang, H.; Xu, W.; and Tomizuka, M. 2022. Paco: Parameter-compositional multi-task reinforcement learning. In *Advances in Neural Information Processing Systems*, 21495–21507.

Sutton, R. S.; Barto, A. G.; et al. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Teh, Y.; Bapst, V.; Czarnecki, W. M.; Quan, J.; Kirkpatrick, J.; Hadsell, R.; Heess, N.; and Pascanu, R. 2017. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, 4496–4506.

Xu, Z.; Wu, K.; Che, Z.; Tang, J.; and Ye, J. 2020. Knowledge transfer in multi-task deep reinforcement learning for continuous control. In *Advances in Neural Information Processing Systems*, 15146–15155.

Yang, R.; Xu, H.; Wu, Y.; and Wang, X. 2020. Multi-task reinforcement learning with soft modularization. In *Advances in Neural Information Processing Systems*, 4767–4777.

Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020a. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, 5824–5836.

Yu, T.; Quillen, D.; He, Z.; Julian, R.; Hausman, K.; Finn, C.; and Levine, S. 2020b. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, 1094–1100.

Zhao, W.; Queralta, J. P.; Westerlund, T.; et al. 2020. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, 737–744.