Conditional Diffusion Model for Multi-Agent Dynamic Task Decomposition

Yanda Zhu[1], **Yuanyang Zhu**[2†], **Daoyi Dong**[3], **Caihua Chen**[1†], **Chunlin Chen**[4,1]

[1] School of Management and Engineering, Nanjing University,
[2] School of Information Management, Nanjing University
[3] Australian Artificial Intelligence Institute, University of Technology Sydney,
[4] School of Robotics and Automation, Nanjing University
yandazhu@smail.nju.edu.cn, {yuanyangzhu, chchen, clchen}@nju.edu.cn, daoyidong@gmail.com

Abstract

Task decomposition has shown promise in complex cooperative multi-agent reinforcement learning (MARL) tasks, which enables efficient hierarchical learning for long-horizon tasks in dynamic and uncertain environments. However, learning dynamic task decomposition from scratch generally requires a large number of training samples, especially exploring the large joint action space under partial observability. In this paper, we present the Conditional Diffusion Model for Dynamic Task Decomposition (CD³T), a novel two-level hierarchical MARL framework designed to automatically infer subtask and coordination patterns. The high-level policy learns subtask representation to generate a subtask selection strategy based on subtask effects. To capture the effects of subtasks on the environment, CD³T predicts the next observation and reward using a conditional diffusion model. At the low level, agents collaboratively learn and share specialized skills within their assigned subtasks. Moreover, the learned subtask representation is also used as additional semantic information in a multi-head attention mixing network to enhance value decomposition and provide an efficient reasoning bridge between individual and joint value functions. Experimental results on various benchmarks demonstrate that CD³T achieves better performance than existing baselines.

Introduction

Cooperative multi-agent reinforcement learning (MARL) has achieved great improvements and holds great promise for real-world challenging problems, such as sensor networks (Zhang and Lesser 2011), coordination of robot swarms (Hüttenrauch, Šošić, and Neumann 2017), and autonomous vehicles (Pham et al. 2018). Learning effective control policies under partial observation for coordinating such systems remains challenging. The centralized training with decentralized execution (CTDE) paradigm (Oliehoek, Spaan, and Vlassis 2008; Kraemer and Banerjee 2016; Liu et al. 2025) alleviates partial observability yet struggles with the exponential growth of the joint action-observation space as agent numbers increase, which makes exploration of valuable states rare and coordination difficult.

To deal with uncertainty and adapt to the dynamics of an environment, all agents learn and share a decentralized policy network under the CTDE framework. Memory-based architectures, such as recurrent neural networks (RNNs), long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997), and gated recurrent units (GRUs), help agents to capture long-term dependencies in their action-observation history (Sunehag et al. 2018; Rashid et al. 2018; Son et al. 2019). Recent transformer-based methods have shown superior performance by modeling both long- and short-term dependencies, offering a powerful solution to partial observability (Parisotto et al. 2020; Yang et al. 2022b; Wen et al. 2022). However, parameter sharing among agents can lead to similar behavior, hindering diversity. The challenge is to balance agent specialization and dynamic sharing to promote cooperation (Christianos et al. 2021).

A natural solution to this challenge is decomposing complex tasks into subtasks (Butler 2011). This decomposition not only simplifies the task but also allows agents to focus on solving specific subtasks, which can reduce the complexity of the action-observation space and enhance overall learning efficiency. Building on this idea, recent research has explored the integration of roles and skills into MARL. In the learning of roles (Wang et al. 2020a, 2021b; Li et al. 2021), skills (Yang et al. 2022a; Liu et al. 2022), or groups (Zang et al. 2023), existing works generally use a simple network structure to extract action representations for agents and may neglect fully considering the dynamic interactions among agents and the environment. The representational capacity of such a setting poses a bottleneck when trying to learn distinct latent representations for all subtasks.

Diffusion Models (Ho, Jain, and Abbeel 2020; Sohl-Dickstein et al. 2015a), a novel class of generative models known for their impressive performance in image generation tasks, offer a promising avenue to address such challenges. These models are well-suited for handling the stochasticity inherent in complex environments due to their ability to model stochastic processes through iterative denoising. Furthermore, since the spaces of histories and states in MARL are often continuous and high-dimensional, diffusion models are particularly effective because of their robust representational capacity in expansive spaces. These benefits of the diffusion model stimulate our thinking in MARL domains, i.e., *can we harness modern generative models, such as diffusion models, trained on offline data and capture useful latent representation that facilitates online MARL?*

†Corresponding author: Yuanyang Zhu and Caihua Chen.

To explore this, we propose the Conditional Diffusion Model for Dynamic Task Decomposition CD^{3}T. To reflect the potential characters of agents and subtasks, a set of action representations is used as the input conditioned on observations and actions of agents, while the reward of the environment and the next observations are used as the output. With this representation, we can derive subtasks by clustering and devise a subtask selection mechanism that assigns an agent to a subtask. Due to characters encoded in the subtask representation, this mechanism could select and share proper skills for agents based on parameter sharing. Benefit from the powerful representational capacity of the diffusion model, CD^{3}T not only owns a better ability to model stochastic processes through the iterative denoising inductive bias but also learns distinguishable subtasks to explore the environment. Inspired by recent works addressing spurious correlations between global states and joint values (Li et al. 2022; Wang et al. 2023; Liu, Zhu, and Chen 2023), we incorporate subtask representations with global state information to better estimate credit assignment.

We evaluate CD^{3}T across a range of benchmarks, including Level-based Foraging (LBF), StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al. 2019), and SMACv2 (Ellis et al. 2023). The results show that our CD^{3}T improves the performance on SMAC compared to the baselines, especially on *Hard* and *Super Hard* scenarios. Ablation studies confirm the efficacy of task decomposition and credit assignment design, and visualizations illustrate meaningful dynamic task decompositions and cooperation.

Preliminaries

Our work focuses on a fully cooperative multi-agent task with only partial observation for each agent, which typically is modeled as a decentralized partially observable Markov decision process (Dec-POMDP) (Oliehoek and Amato 2016) and described with the tuple $\mathcal{M} = \langle \mathcal{I}, \mathcal{S}, \mathcal{A}, P, R, \Omega, O, \gamma \rangle$. At each time step, each agent $i \in \mathcal{I}$ receives an observation $o_i \in \Omega$, drawn from the observation function $O(s, i)$, where $s \in \mathcal{S}$ is the global state of the environment, and selects an action $a_i \in \mathcal{A}$, producing a joint action $\boldsymbol{a} = (a_1, \ldots, a_n)$. This joint action would lead to the next state s' according to the state transition function $P(s'|s)$, and all agents would receive a shared team reward $r = R(s, \boldsymbol{a})$. We use $\boldsymbol{\tau} \in \mathcal{T} \equiv (\Omega \times \mathcal{A})^*$ to denote the joint action-observation history, where $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_n)$, and $\tau_i = (o_i^1, a_i^1, \ldots, o_i^{t-1}, a_i^{t-1}, o_i^t)$ represents the trajectory of agent i. The target is to find the optimal joint policy $\boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{\tau})$ that maximizes the discounted return, defined as $Q^{\pi}(\boldsymbol{\tau}, \boldsymbol{a}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \boldsymbol{a}_t) \mid s_0 = s, \boldsymbol{a}_0 = \boldsymbol{a}, \boldsymbol{\pi}\right]$, where $\gamma \in [0, 1)$ is the discount factor.

Unlike recent MARL works (Sunehag et al. 2018; Rashid et al. 2018; Son et al. 2019; Wang et al. 2021a), we propose to decompose a fully cooperative multi-agent task into subtasks and present the definition of subtasks in the following.

Definition 1 (Subtasks)**.** *Given a cooperative task $\mathcal{M} = \langle \mathcal{I}, \mathcal{S}, \mathcal{A}, P, R, \Omega, O, \gamma \rangle$, we assume there exists a set of g subtasks, denoted as $\Phi = \{\phi^1, \phi^2, ..., \phi^g\}$, where $g \in \mathbb{N}^+$ is unknown and considered as a tunable hyperparameter. Each subtask is expressed as a tuple $\langle \mathcal{M}_{\phi^j}, \pi_{\phi^j} \rangle$, where $j \in \{1, 2, ..., g\}$ is the identity of subtask, $\mathcal{M}_{\phi^j} = \langle \mathcal{I}_{\phi^j}, \mathcal{S}, \mathcal{A}_{\phi^j}, P, R, \Omega_{\phi^j}, O, \gamma \rangle$ and $\pi_{\phi^j} : \mathcal{T} \times \mathcal{A}_{\phi^j} \to [0, 1]$. $\mathcal{I}_{\phi^j}$ is the set of agents assigned to subtask ϕ^j and each agent can only select one subtask to solve at each timestep, i.e., $\mathcal{I}_{\phi^j} \subset \mathcal{I}$, $\cup_{\phi^j} \mathcal{I}_{\phi^j} = \mathcal{I}$ and $\mathcal{I}_{\phi^j} \cap \mathcal{I}_{\phi^k} = \varnothing$ if $j \neq k$. $\mathcal{A}_{\phi^j}$ is the action space of the subtask, $\mathcal{A}_{\phi^j} \subset \mathcal{A}$, $\cup_{\phi^j} \mathcal{A}_{\phi^j} = \mathcal{A}$, but we allow action spaces of different subtasks to overlap: $|\mathcal{A}_{\phi^j} \cap \mathcal{A}_{\phi^k}| \geq 0$ if $j \neq k$. Each agent $i \in \mathcal{I}_{\phi^j}$ shares the policy parameters of π_{ϕ^j}.*

With the set of subtasks Φ defined, each agent $i_{\phi^j} \in \mathcal{I}_{\phi^j}$ is assigned subtask ϕ^j through a shared subtask selector. This enables the learning of subtask-specific policies $\pi_{\phi^j} : \boldsymbol{\tau} \times \mathcal{A}_{\phi^j}$ for each subtask. Our objective is to learn the optimal set of subtasks Φ^* that maximizes the expected global return $Q_{tot}^{\Phi}(\boldsymbol{\tau}, \boldsymbol{a}) = \mathbb{E}_{s_{1:\infty}, \boldsymbol{a}_{1:\infty}}[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, \boldsymbol{a}_0 = \boldsymbol{a}, \Phi]$.

Method

Our solution for multi-agent dynamic task decomposition is illustrated in *Fig. 1*. We begin by describing how to construct action semantic representations that enable the decomposition of multi-agent tasks. Next, we explain how the representations are leveraged to generate subtasks. Based on the generated subtasks and their corresponding latent representations, we introduce a hierarchical architecture consisting of a subtask selector and a set of subtask policies. Finally, we detail the training objective and inference strategy for both the subtask selector and the subtask policies.

Action Representation Learning via Diffusion

The latent action representations are designed to induce diverse subtasks with distinct responsibilities, capturing characteristic agent behaviors for more appropriate subtask selection. While this design allows CD^{3}T to adapt to dynamic environments, it may lead to rapid subtask shifts and instability during learning. Moreover, if the induced subtasks are overly similar, decomposition becomes ineffective. Therefore, two key challenges arise: 1) ensuring temporal stability to maintain adaptability, and 2) enhancing subtask diversity through efficient modeling.

To this end, we first construct the action encoder component to map the one-hot action a_i of agent i to the d-dimensional representation z_{a_i}, which serves as unmodified examples z_0. Then the UNet backbone with cross-attention is employed as a flexible feature extractor in the denoising network $\epsilon_{\theta_d}(z_k, k, o_i, a_{-i})$ to recover z_{a_i} from Gassian noise $\epsilon \sim \mathcal{N}(0, I)$ conditioned on corresponding o_i and a_{-i}. Following the simplified objective (Ho, Jain, and Abbeel 2020), we formulate a learning objective for action representations via the conditional diffusion model parameterized by θ_d, which is trained by minimizing the loss function:

$$\mathcal{L}_d(\theta_d) = \mathbb{E}_{\substack{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \\ (\boldsymbol{o}, \boldsymbol{a}) \sim \mathcal{D}}} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta_d}(\boldsymbol{z}_k, k, \boldsymbol{o}_i, \boldsymbol{a}_{-i}) \right\|^2 \right], \quad (1)$$

where $\mathcal{D}$ denotes the replay buffer, k is the diffusion iteration uniformly sampled from $\{1, 2, \ldots, K\}$ and z_k is the noisy version of z_0. Here, o_i denotes the observation of agent i,

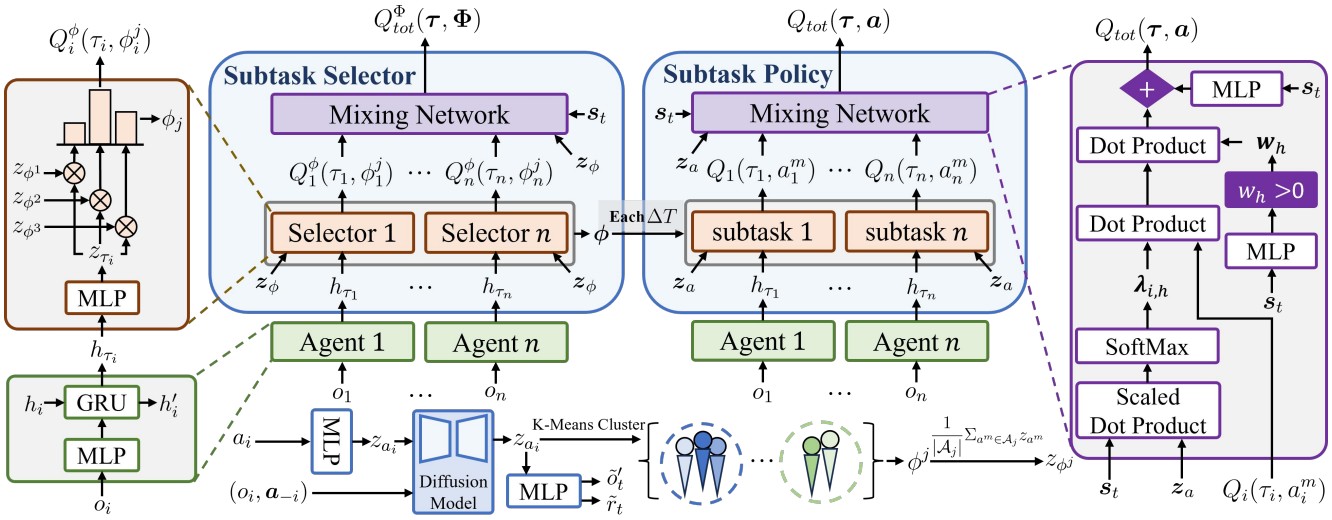


Figure 1: The overall framework of CD^{3}T. We first derive a latent action representation z_{a_i} for each agent from its action space, conditioned on its local observation o_i and other agents' one-hot actions $\boldsymbol{a}_{-i}$, to pretrain a diffusion model. Latent representations are then clustered to define subtask-specific action spaces. The subtask selector and subtask policy share the same architecture with different parameters. At every ΔT steps, the selector assigns a subtask to each agent and estimates the joint Q-value Q_{tot}^{Φ} using the global state s_t and subtask representation $\boldsymbol{z}_\phi$, while the subtask policy computes Q_{tot} with s_t and the action representation $\boldsymbol{z}_a$.

while $\boldsymbol{a}_{-i}$ denotes the joint actions of all other agents. The detailed derivation can be found in Appendix A.

Action semantics, where different actions have varying effects on other agents, should influence the environment or their private properties. To further extract the influence of an action through the induced reward and the change in local observations, we leverage the action representations generated by the diffusion model to predict the next observations o_i' and global reward r based on o_i and $\boldsymbol{a}_{-i}$. The prediction objective can be further rewritten as a loss function:

$$\mathcal{L}_p(\theta) = \mathbb{E}_{(\boldsymbol{o},\boldsymbol{a},r,\boldsymbol{o}')\sim\mathcal{D}}\Big[\sum_i \big(\|f_{do}(z_{a_i}, \boldsymbol{o}_i, \boldsymbol{a}_{-i}) - \boldsymbol{o}_i'\|_2^2 + \lambda_{dr}\big(f_{dr}(z_{a_i}, \boldsymbol{o}_i, \boldsymbol{a}_{-i}) - r\big)^2\big)\Big], \tag{2}$$

where f_{do} and f_{dr} are predictors of observations and global rewards, respectively, and λ_{dr} is a scaling factor. Here, the summation covers all agents, and the whole action representation learning is parameterized by θ. Thus, the objective for action representation learning combines the prediction loss and the diffusion model loss with a scaling factor η_d:

$$\mathcal{L}(\theta) = \mathcal{L}_p(\theta) + \eta_d \mathcal{L}_d(\theta_d). \tag{3}$$

The current formulation does not explicitly enforce subtask specialization, which is essential for ensuring behavioral diversity across different subtasks. Prior approaches typically encourage specialization via explicit regularization techniques (Christianos et al. 2021; Wang et al. 2020a). In contrast, we utilize the diffusion model as a flexible feature extractor and enhance its UNet backbone with cross-attention to generate action representations that capture multimodal distributions. This property naturally induces subtask diversity without additional regularization.

Subtask Dynamic Decomposition

The action representation plays a critical role in assigning agents to the most suitable subtasks, where agents dealing with the same subtask share their learning of specific abilities. Considering that an agent's subtask history reflects not only its behavioral history but is also task-independent, we perform k-means clustering over these action representations to decompose the task into a finite set of subtasks after sampling and learning for the initial $50K$ timesteps of the overall training process. Given the action representations of subtask ϕ^j, the subtask representation z_{ϕ^j} can be derived as $z_{\phi^j} = \frac{1}{|\mathcal{A}_j|}\sum_{a^m\in\mathcal{A}_j} z_{a^m}$, where $\mathcal{A}_j$ represents the decomposed action space of the subtask ϕ^j and a^m denotes actions in this restricted space. With a hierarchical network consisting of the subtask selector and the subtask policies, subtask representations can be used to assign the most suitable subtask to an agent over a specific time interval ΔT. Similarly, an MLP layer and a shared GRU layer network are shared by all agents to obtain the historical information of the agent's local observations and actions, which is parameterized by θ_{τ_ϕ} and compiles it into a vector h for our subtask selection.

When selecting subtasks every ΔT timesteps, the subtask selector encodes the historical information h_τ of each agent into the hidden layer variable $\boldsymbol{z}_\tau \in \mathbb{R}^d$ with a ξ_ϕ-parameterized encoder $f_\phi(:;\xi_\phi)$. Then it is used to estimate the expected return of agent i in subtask ϕ^j with the hidden layer representation of each subtask space $Q_i^\phi(\tau_i, \phi^j) = \boldsymbol{z}_{\tau_i}^T \boldsymbol{z}_{\phi^j}$. The subtask space corresponding to the maximum

Q-value is assigned to each agent. In the next ΔT time step, each agent learns its policy in the restricted subtask space.

Learning Decomposition with Credit Assignment

The global state $\boldsymbol{s}$ in multi-agent systems contains rich information, yet only a subset is relevant to individual decision-making. To extract meaningful abstractions without relying on expert knowledge, we follow (Li et al. 2022; Liu, Zhu, and Chen 2023; Xu et al. 2025) and mitigate spurious correlations between $\boldsymbol{s}$ and Q_{tot} by leveraging agents' local histories τ_i in partially observable settings. To improve decomposition accuracy and credit assignment, we introduce an intervention-based adjustment function during training, which adheres to the Individual-Global-Maximum (IGM) principle in Appendix B.

Specifically, the credit $\lambda^{\phi}_{h,i}$ for subtask selector is computed with the subtask representation $\boldsymbol{z}_{\phi}$ and the global state $\boldsymbol{s}$ through a dot-product attention as

$$\lambda^{\phi}_{h,i} = \frac{\exp\left((\boldsymbol{w}_{\boldsymbol{z}_\phi}\boldsymbol{z}_\phi)^\top \mathrm{ReLU}(\boldsymbol{w}_s \boldsymbol{s})\right)}{\sum_{i=1}^{N}\exp\left((\boldsymbol{w}_{\boldsymbol{z}_\phi}\boldsymbol{z}_\phi)^\top \mathrm{ReLU}(\boldsymbol{w}_s \boldsymbol{s})\right)}, \tag{4}$$

where $\boldsymbol{w}_{\boldsymbol{s}}$, $\boldsymbol{w}_{\boldsymbol{z}_\phi}$ are learnable weight matrices, and ReLU is the element-wise rectified-linear activation. $\lambda_{h,i}$ is positive with softmax operation to ensure monotonicity and h is the number of attention heads. The softmax ensures each $\lambda^{\phi}_{h,i}$ is positive and that $\sum_i \lambda^{\phi}_{h,i} = 1$. Then the joint action function Q^{Φ}_{tot} of the subtask selector can be estimated as

$$Q^{\Phi}_{tot} = c_\phi(s) + \sum_{h=1}^{H} w^{\phi}_h \sum_{i=1}^{N} \lambda^{\phi}_{h,i} Q^{\phi^j}_i(\tau_i, \phi^j_i), \tag{5}$$

where $c_\phi(s)$ is learned by a neural network with the global state s as the input. The joint action function Q^{Φ}_{tot} of the subtask selector can be optimized by minimizing TD loss:

$$\mathcal{L}_{ss}(\theta_{\tau_\phi}, \xi_\phi) = \mathbb{E}_{\mathcal{D}}\Big[\Big(\sum_{\Delta t=0}^{\Delta T-1} r_{t+\Delta t} + \gamma \max_{\boldsymbol{\Phi}'} \bar{Q}^{\Phi}_{tot}(s_{t+\Delta T}, \boldsymbol{\Phi}') - Q^{\Phi}_{tot}(s_t, \boldsymbol{\Phi}_t)\Big)^2\Big], \tag{6}$$

where ξ_ϕ denotes the parameters of the mixing network, $\boldsymbol{\Phi} = \langle \phi^1, \phi^2, \ldots, \phi^N \rangle$ is the joint subtask of all agents, and the expectation is taken over mini-batches sampled uniformly from the replay buffer $\mathcal{D}$.

During the timestep of ΔT, each agent follows the subtask assigned by the high-level selector. When the agent is assigned to the corresponding subtask, it selects its action in the action space of the subtask. Therefore, each subtask has a corresponding policy $\pi_{\phi^j} : \mathcal{T} \times \mathcal{A}_{\phi^j} \to [0, 1]$, which is defined in the restricted subtask action space and updates such a policy network. To take full advantage of the action representation of the corresponding subtask space, we also use the mechanism to compute the joint function Q_{tot}. Here, we use the shared MLP layer and GRU layer parameterized by θ_τ in the same way to compile the local action-observation information history τ into a vector h_τ. For each subtask policy, we use a fully connected network $f_{\phi'}(h_\tau; \zeta_\phi)$ parameterized by ζ_ϕ to represent it. Thus we estimate the individual value of agent i by choosing a primitive action a^m_i as $Q_i(\tau_i, a^m_i) = \boldsymbol{z}^T_{\tau_i} \boldsymbol{z}_{a^m_i}$.

Similar to the value function factorization of the subtask selector, the action representation $\boldsymbol{z}_a$ and the global state $\boldsymbol{s}$ are still fed into the intervention function to estimate credits. Here, the action representations are restricted in the action space of the subtasks assigned to the agent, rather than the subtask representations in the subtask selector, so the credit $\lambda_{h,i}$ for the subtask policy is computed as

$$\lambda_{h,i} = \frac{\exp\left((\boldsymbol{w}_{z_a}\boldsymbol{z}_a)^\top \mathrm{ReLU}(\boldsymbol{w}_s \boldsymbol{s})\right)}{\sum_{i=1}^{N}\exp\left((\boldsymbol{w}_{z_a}\boldsymbol{z}_a)^\top \mathrm{ReLU}(\boldsymbol{w}_s \boldsymbol{s})\right)}, \tag{7}$$

where $\boldsymbol{w}_s$, $\boldsymbol{w}_{z_a}$ are the learnable parameters, and ReLU is the activation function. $\lambda_{h,i}$ is positive with softmax operation to ensure monotonicity and h is the number of attention heads. The joint value function of the subtask policy is predicted based on the credits and factorized Q-values

$$Q_{tot} = c(s) + \sum_{h=1}^{H} w_h \sum_{i=1}^{N} \lambda_{h,i} Q_i(\tau_i, a^m_i), \tag{8}$$

where $c(s)$ is learned by a neural network with the global state s as the input. The formulation gives the TD loss for subtask policies:

$$\mathcal{L}_s(\theta_\tau, \xi) = \mathbb{E}_{\mathcal{D}}[(r + \gamma \max_{\boldsymbol{a}'} \bar{Q}_{tot}(s', \boldsymbol{a}') - Q_{tot}(s, \boldsymbol{a}))^2], \tag{9}$$

where the parameters of the mixing network are denoted by ξ and $\bar{Q}_{tot}$ is a target network. Training samples are drawn uniformly from the same replay buffer $\mathcal{D}$ used for the high-level selector. Under the CTDE paradigm, the selector, subtask policies, and individual utility networks are used jointly at execution time, but only local information is required for each agent to act. Pseudocode for the complete CD^{3}T algorithm is given in Appendix C.

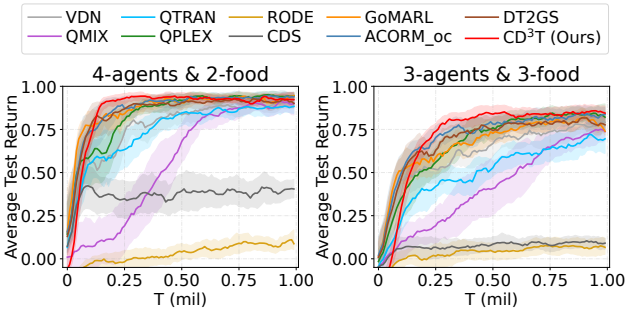


Figure 2: Performance comparison with baselines on LBF.

Experiments

We evaluate CD^{3}T on three challenging benchmarks, including LBF (Christianos, Schäfer, and Albrecht 2020), SMAC (Samvelyan et al. 2019) and SMACv2 (Ellis et al. 2023). The baselines we select are five classic value-decomposition methods (VDN (Sunehag et al. 2018), QMIX (Rashid et al. 2020), QTRAN (Son et al. 2019), QPLEX (Wang et al. 2021a), and CDS (Li et al. 2021)), and four subtask-related methods (RODE (Wang et al. 2021b),

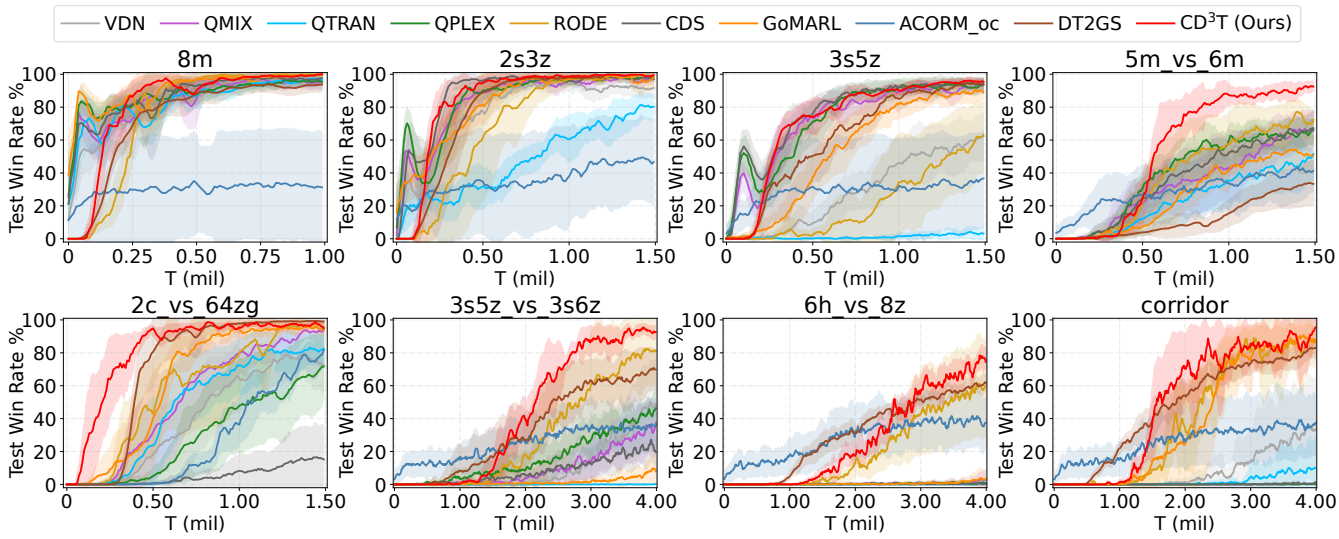


Figure 3: Performance comparison with baselines on *easy*, *hard*, and *super hard* scenarios.

GoMARL (Zang et al. 2023), ACORM (Hu et al. 2024) and DT2GS (Tian et al. 2023)). ACORM performs k-means clustering at each timestep to obtain commendable results, which undoubtedly imposes a substantial computational burden. Therefore, to improve efficiency and maintain consistency with our algorithm, we likewise apply clustering for ACORM only once at $50K$ timesteps. We use "ACORM_oc" to represent ACORM with once clustering in our experiments. The implementation details of all experiments are provided in Appendix D. All learning curves report the mean $\pm$ standard deviation over five random seeds.

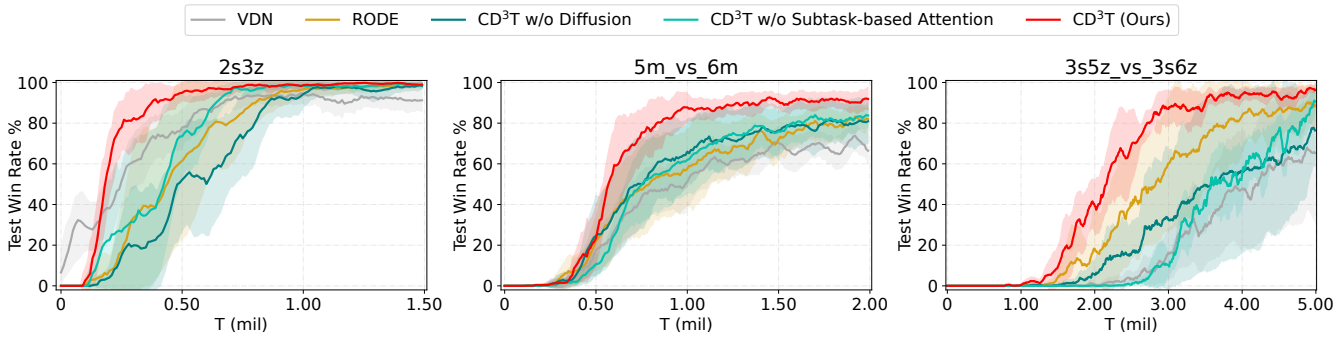
Performance on LBF

We first conduct experiments on two constructed LBF tasks to assess the performance of different algorithms under two distinct settings. *Fig. 2* illustrates a comprehensive comparison of performance against baselines on two specially crafted LBF tasks. Our approach shows competitive performance across LBF tasks, demonstrating its flexibility and effectiveness on both scenarios. The failure of CDS and RODE could originate from the inability of its heterogeneous agents to effectively explore and develop collaborative strategies. In contrast, VDN, QMIX, and QTRAN require more steps to uncover more refined policies, suggesting that they may struggle with the inherent limitations in solving spurious relationships between credit assignments and decomposed Q-values. QPLEX receives a lower return compared to CD^{3}T between 0.1M and 0.6M timesteps, potentially necessitating additional time for exploration due to its complex value decomposition design. CD^{3}T achieves slightly higher performance than GoMARL and ACORM, which implies that in terms of semantic representation, the action representation may have superiority over both group-based information and role representation. The marginally inferior performance of DT2GS compared to CD^{3}T further substantiates that enhancing the generalization of subtasks inevitably entails a trade-off with performance on single tasks.

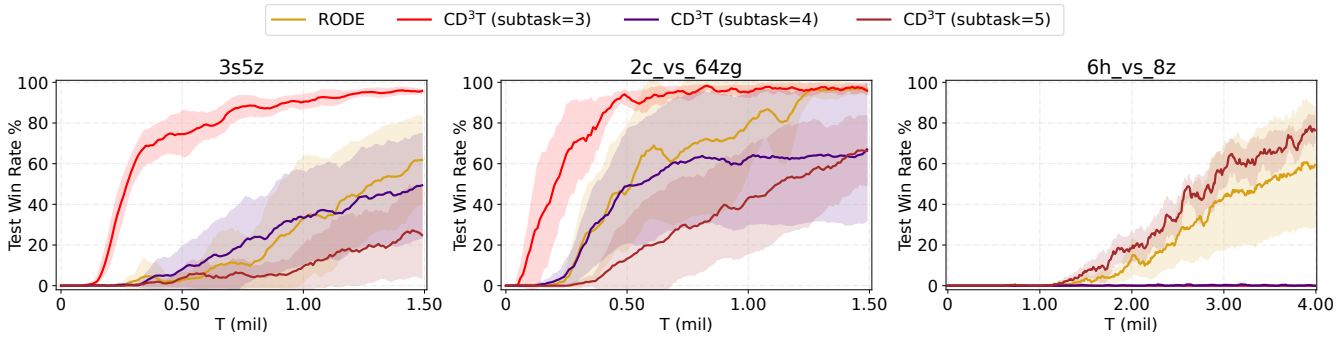
Performance on SMAC

To further evaluate CD^{3}T, we benchmark it on the more challenging SMAC benchmark, which is a testbed commonly used for MARL algorithms. We compare CD^{3}T with other baselines on 8 different scenarios, including *easy*, *hard*, and *super hard* scenarios.

The experimental results for different scenarios are shown in *Fig. 3*. As we can see, CD^{3}T yields almost the highest win rate on all scenarios, especially on the *super hard* tasks. QTRAN performs poorly on almost all scenarios due to its soft constraints involving two ℓ_2-penalty terms. Although QPLEX behaves well on easy scenarios, resulting in its tendency to fall into local optima, its performance decreases on hard scenarios. Both VDN and QMIX can achieve satisfactory performance on some *easy* or *hard* maps, i.e., `8m`, `2s3z`, and `5m_vs_6m`, but they fail to cope with the tasks well on *super hard* maps. It should result from the fact that the super-hard task needs more efficient exploration to learn cooperation skills. RODE fails to learn efficient policies for subtasks, which implies that its reliance solely on the simple MLP structure hinders the accurate learning of role semantics. CDS fails to learn efficient policies since it may require more steps to explore, which celebrates diversity among agents, especially on the map `corridor` and `6h_vs_8z`. GoMARL attains comparable performance with CD^{3}T on part of easy and hard maps but underperforms on two *super hard* maps, possibly owing that the automatic grouping method primarily places excessive emphasis on the relative contribution of each agent to the entire group while ignoring the mutual contributions among agents within the group. ACORM achieves strong early learning, yet once per-step clustering is removed (ACORM-oc), performance collapses on nearly all maps, implying that its contrastive role representations rely heavily on continuous reclustering. One



(a) Ablation studies regarding each component of CD^{3}T.



(b) The performance of CD^{3}T with different number of subtask clusters.

Figure 4: Ablation studies of CD^{3}T on SMAC benchmark.

possible explanation for the consistently suboptimal performance of DT2GS is its excessive emphasis on generalization across a limited set of tasks. Overall, our approach achieves impressive performance across all scenarios, which validates the advantages of CD^{3}T with its attentive design. Additional experiments on SMACv2 in the Appendix E further confirm the effectiveness of CD^{3}T.

Ablation Studies

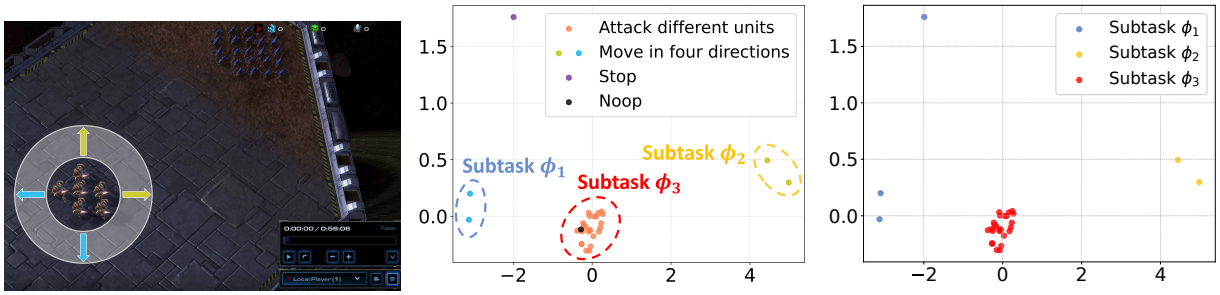
To quantify the contribution of each component, we perform three ablations and address the following questions: (a) How does the diffusion model enhance subtask representation and improve overall performance? (b) What role does the attention mechanism leveraging subtask representation play in credit assignment? (c) Does the number of subtasks affect the capability of the model? To test components (a), we replace the diffusion model with a vanilla MLP structure and denote it as *CD^{3}T w/o diffusion*. For (b), we replace the subtask-based attention mechanism in our mixing network, which is substituted with the QMIX method, and denote it as *CD^{3}T w/o Subtask-based Attention*. To test component (c), we test it with the number of subtasks formed during clustering, named *CD^{3}T (subtask=3)*. Considering an excessive number of subtasks could not affect the performance for the finite action space, we set the number of subtasks to $3 \leq subtasks \leq 5$.

The results on three scenarios with different difficulties are shown in *Fig. 4*. *CD^{3}T w/o diffusion* attains the lowest win rates on all maps—particularly the hard and super hard ones—highlighting the critical role of diffusion-based subtask representations in high-dimensional state–action spaces. Especially on the hard and super hard maps, it becomes clear that CD^{3}T achieves a larger margin than replacing the diffusion with a simple MLP. The reason is that associating subtask policy with its proven powerful representational capacity in such expansive spaces benefits the performance in complex tasks. *CD^{3}T w/o Subtask-based Attention* is lower than that of CD^{3}T, which indicates the importance of subtask representation for estimating credits. As shown in *Fig. 4(b)*, the performance of CD^{3}T consistently improves as the number of subtasks increases. Generally, moderate order terms (e.g., $subtask \leq 5$) are enough for an appropriate trade-off between performance improvement and computation. In summary, the superior performance of CD^{3}T is conditioned on all parts, where it is largely due to the efficient subtask assignment representation.

Generation of Subtask Representation

To learn action representations, we collect samples and train the diffusion model for $50K$ timesteps, guided by the loss function specified in Eq. (3). In *Fig. 5*, we provide an in-depth illustration of how the action representations derived from the diffusion model are strategically harnessed to enable subtask decomposition.

The `corridor` scenario features homogeneous agents and enemies—specifically, 6 Zealots versus 24 Zer-



(a) (b) (c)

Figure 5: The process of subtask generation through subtask representations. (a) illustrates that moving northward or eastward has a comparable effect on the environment by directing agents toward the enemies, whereas moving southward or westward moves agents away. (b) depicts the distribution of action representations in a two-dimensional space after PCA projection. (c) visualizes the formation of subtasks derived from action representations following clustering.

(a) Timestep = 7 (b) Timestep = 22 (c) Timestep = 149

Figure 6: Visualizations of dynamic subtask selection in one episode on `corridor`. Blue denotes subtask ϕ^1, yellow denotes subtask ϕ^2, and red denotes subtask ϕ^3. (a), (b), and (c) depict game screenshots at $t = 7$, $t = 22$, and $t = 149$, respectively.

glings—where all attack actions produce similar effects due to enemy uniformity. Owing to the scenario's spatial symmetry, moving north or east similarly advances agents toward the enemies, while moving south or west leads them away. The diffusion model captures these structural regularities within the action space, as illustrated in *Fig.5(a)*. In *Fig.5(b)*, we apply PCA to project the high-dimensional action representations into a two-dimensional space, revealing clear clusters aligned with the primary action types. These clusters emerge consistently across random seeds, demonstrating that our diffusion model reliably learns subtask representations that reflect the underlying action effects.

Visualization of Dynamic Subtask Selections

To gain deeper insights into the subtask selection behavior of CD^{3}T, we visualize the agent-wise subtask assignments across a representative episode (Fig.6) and report the corresponding selection frequencies over time (Fig.8). Additional examples are included in Appendix G.

A key observation is that direct engagement under numerical disadvantage is strategically suboptimal. As shown in *Fig. 6(a)*, CD^{3}T assigns subtask ϕ^2 to Agent 1 early in the episode ($t = 7$), prompting it to move northward and draw nearly half of the enemies away. This diversion enables the remaining five agents to reposition eastward for a more coordinated attack. In *Fig. 6(b)*, Agent 1 switches to subtask ϕ^3 to engage the enemies directly until eliminated. Meanwhile, Agents 2 and 3 execute kiting behaviors under ϕ^1, while Agents 4–6 focus fire under ϕ^3. As the battle progresses, surviving enemies regroup in the bottom-left corner, out of sight of the Zealots. During mid-phase, CD^{3}T reassigns subtask ϕ^2 to Agent 2 to lure a subset of enemies away from the group and disrupt their formation. Meanwhile, the remaining agents continue alternating between evasive movement and focused fire to isolate and eliminate targets.

Empirical Visualization of Action Space Reduction

Fig. 7 presents a comparative visualization of the effective action space dimensions under three methods: CD^{3}T, RODE, and QMIX. CD^{3}T consistently achieves a more compact action space across diverse scenarios, as reflected in its lower average dimensionality and tighter confidence intervals. This suggests that its fine-grained subtask decomposition enables each sub-policy to operate within a task-relevant, reduced action set. By comparison, QMIX does not incorporate any subtask or role abstraction, and thus always

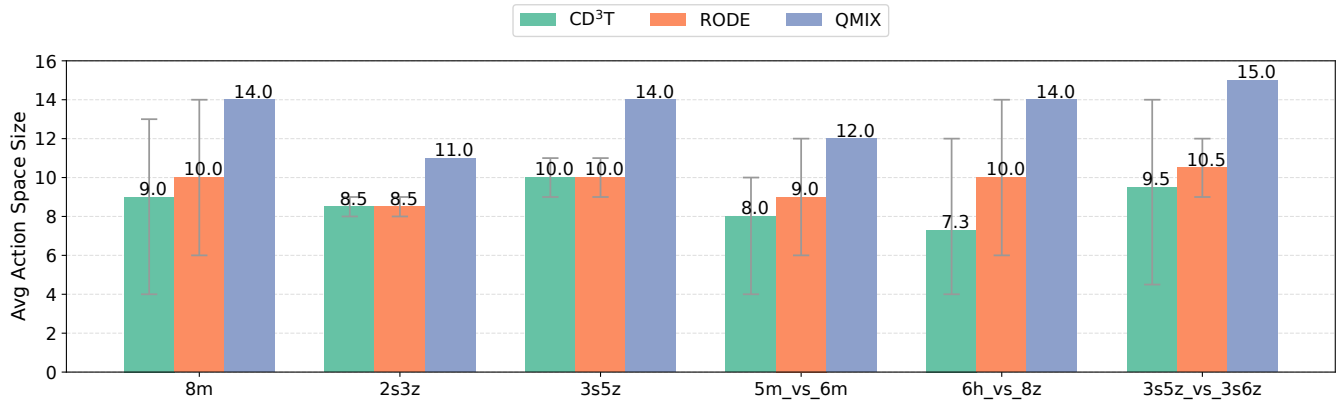


Figure 7: Average action space sizes of CD^{3}T, RODE, and QMIX across six SMAC scenarios. Each bar denotes the mean number of available actions per method, while the error bars capture variation across roles or agents, ranging from the minimum to the maximum number of available actions. QMIX takes actions over the full action space without any form of masking, and thus does not produce variation.

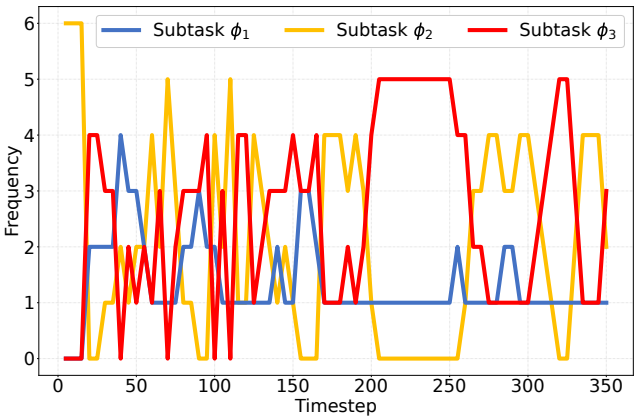


Figure 8: Demonstrates subtask selection frequencies of CD^{3}T on `corridor` map in the same episode. The curve represents the number of agents assigned the corresponding subtask at the timestep.

utilizes the full action space. Its action dimensionality remains fixed across scenarios, which may limit its adaptability to varying task complexity or coordination requirements. In relatively simple environments such as `8m`, `2s3z`, and `3s5z`, both CD^{3}T and RODE exhibit similar action space sizes, indicating that in low-complexity settings, the benefits of explicit decomposition may be less pronounced. However, in more challenging scenarios, `5m_vs_6m`, `6h_vs_8z`, and `3s5z_vs_3s6z`, CD^{3}T achieves more substantial reduction. This improvement may be attributed to its explicit subtask masking mechanism, which restricts each role to a compact, specialized action subset. In contrast, RODE's role abstraction is more implicit and does not enforce strict action sparsity, potentially resulting in less targeted compression.

These findings highlight CD^{3}T's ability to adaptively reduce the decision space in accordance with task complexity. By focusing each agent on a smaller, semantically meaningful set of actions, CD^{3}T not only improves computational efficiency but also facilitates more structured and scalable coordination in challenging multi-agent environments.

Conclusions

Task decomposition is a pivotal approach to simplifying complex multi-agent tasks, yet it remains a long-standing and unresolved challenge. To address this, we proposed leveraging latent representations extracted by a diffusion model to decompose tasks into multiple subtasks. This approach captures the relationship between subtasks and environmental dynamics more accurately. Agents are assigned to corresponding subtasks through subtask selectors, ensuring better compatibility between agents and subtasks. This compatibility enables agents to learn policies more efficiently in a shared learning framework. Furthermore, by clustering latent representations, similar agents can share experiences, accelerating training and enhancing overall performance. During training, the subtask-based attention mechanism in the mixing network effectively utilizes global state and semantic inference to guide the mixing of Q-values. Experimental results across three benchmarks demonstrate that our method achieves superior performance in nearly all scenarios, advancing the state of the art in MARL.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Nos. 72394363, 72394364 & 72394360), National Key Research and Development Program of China (2025YFA101690004), Major Science and Technology Project of Jiangsu Province under Grant BG2024041, and the Fundamental Research Funds for the Central Universities under Grant 011814380048, the China Postdoctoral Science Foundation under Grant Number 2025T180877, the Jiangsu Funding Program for Excellent Postdoctoral Talent 2025ZB267.

References

Al-Emran, M. 2015. Hierarchical reinforcement learning: a survey. *International Journal of Computing and Digital Systems*, 4(2): 137–143.

Butler, E. 2011. *The Condensed Wealth of Nations*. Adam Smith Institute.

Christianos, F.; Papoudakis, G.; Rahman, M. A.; and Albrecht, S. V. 2021. Scaling multi-agent reinforcement learning with selective parameter sharing. In *Proc. Int. Conf. Mach. Learn.*, volume 139, 1989–1998.

Christianos, F.; Schäfer, L.; and Albrecht, S. 2020. Shared experience actor-critic for multi-agent reinforcement learning. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 33, 10707–10717.

Daniel, C.; Neumann, G.; Kroemer, O.; and Peters, J. 2016. Hierarchical Relative Entropy Policy Search. *Journal of Machine Learning Research*, 17(93): 1–50.

Dayan, P.; and Hinton, G. E. 1992. Feudal Reinforcement Learning. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 5, 271–278.

Dilokthanakul, N.; Kaplanis, C.; Pawlowski, N.; and Shanahan, M. 2019. Feature control as intrinsic motivation for hierarchical reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.*, 30(11): 3409–3418.

Ellis, B.; Cook, J.; Moalla, S.; Samvelyan, M.; Sun, M.; Mahajan, A.; Foerster, J. N.; and Whiteson, S. 2023. SMACv2: An Improved Benchmark for Cooperative Multi-Agent Reinforcement Learning. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 36.

Gupta, T.; Mahajan, A.; Peng, B.; Böhmer, W.; and Whiteson, S. 2021. UneVEn: Universal Value Exploration for Multi-Agent Reinforcement Learning. In *Proc. Int. Conf. Mach. Learn.*, volume 139, 3930–3941.

He, H.; Bai, C.; Xu, K.; Yang, Z.; Zhang, W.; Wang, D.; Zhao, B.; and Li, X. 2023. Diffusion Model is an Effective Planner and Data Synthesizer for Multi-Task Reinforcement Learning. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 36, 64896–64917.

Hegde, S.; Batra, S.; Zentner, K.; and Sukhatme, G. 2023. Generating behaviorally diverse policies with latent diffusion models. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 36, 7541–7554.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 33, 6840–6851.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*, 9(8): 1735–1780.

Hu, Z.; Zhang, Z.; Li, H.; Chen, C.; Ding, H.; and Wang, Z. 2024. Attention-Guided Contrastive Role Representations for Multi-Agent Reinforcement Learning. In *Proc. Int. Conf. Learn. Represent.*

Hüttenrauch, M.; Šošić, A.; and Neumann, G. 2017. Guided deep reinforcement learning for swarm systems. *arXiv preprint arXiv:1709.06011*.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparameterization with Gumbel-Softmax. In *Proc. Int. Conf. Learn. Represent.*

Janner, M.; Du, Y.; Tenenbaum, J. B.; and Levine, S. 2022. Planning with Diffusion for Flexible Behavior Synthesis. In *Proc. Int. Conf. Mach. Learn.*, volume 162, 9902–9915.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *Proc. Int. Conf. Learn. Represent.*

Kraemer, L.; and Banerjee, B. 2016. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190: 82–94.

Lee, Y.; Yang, J.; and Lim, J. J. 2020. Learning to Coordinate Manipulation Skills via Skill Behavior Diversification. In *Proc. Int. Conf. Learn. Represent.*

Li, C.; Wang, T.; Wu, C.; Zhao, Q.; Yang, J.; and Zhang, C. 2021. Celebrating Diversity in Shared Multi-Agent Reinforcement Learning. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 34, 20888–20900.

Li, J.; Kuang, K.; Wang, B.; Liu, F.; Chen, L.; Fan, C.; Wu, F.; and Xiao, J. 2022. Deconfounded value decomposition for multi-agent reinforcement learning. In *Proc. Int. Conf. Mach. Learn.*, volume 162, 12843–12856.

Liu, Y.; Li, Y.; Xu, X.; Dou, Y.; and Liu, D. 2022. Heterogeneous Skill Learning for Multi-agent Tasks. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 35, 26412–26425.

Liu, Z.; Zhu, Y.; and Chen, C. 2023. NA^2Q: Neural Attention Additive Model for Interpretable Multi-Agent Q-Learning. In *Proc. Int. Conf. Mach. Learn.*, volume 202, 22539–22558. PMLR.

Liu, Z.; Zhu, Y.; Wang, Z.; Gao, Y.; and Chen, C. 2025. MIXRTs: Toward interpretable multi-agent reinforcement learning via mixing recurrent soft decision trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(5): 4090–4107.

Lu, C.; Ball, P.; Teh, Y. W.; and Parker-Holder, J. 2023. Synthetic Experience Replay. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 36.

Nair, S.; and Finn, C. 2020. Hierarchical Foresight: Self-Supervised Learning of Long-Horizon Tasks via Visual Subgoal Generation. In *Proc. Int. Conf. Learn. Represent.*

Oliehoek, F. A.; and Amato, C. 2016. *A Concise Introduction to Decentralized POMDPs*. Springer.

Oliehoek, F. A.; Spaan, M. T.; and Vlassis, N. 2008. Optimal and approximate Q-value functions for decentralized POMDPs. *J. Artif. Intell. Res.*, 32: 289–353.

Parisotto, E.; Song, F.; Rae, J.; Pascanu, R.; Gulcehre, C.; Jayakumar, S.; Jaderberg, M.; Lopez Kaufman, R.; Clark, A.; Noury, S.; Botvinick, M.; Heess, N.; and Hadsell, R. 2020. Stabilizing Transformers for Reinforcement Learning. In *Proc. Int. Conf. Mach. Learn.*, volume 119, 7487–7498.

Pham, H. X.; La, H. M.; Feil-Seifer, D.; and Nefian, A. 2018. Cooperative and distributed reinforcement learning of drones for field coverage. *arXiv preprint arXiv:1803.07250*.

Rashid, T.; Farquhar, G.; Peng, B.; and Whiteson, S. 2020. Weighted QMIX: Expanding Monotonic Value Function

Factorisation for Deep Multi-Agent Reinforcement Learning. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 33, 10199–10210.

Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *Proc. Int. Conf. Mach. Learn.*, volume 80, 4295–4304.

Samvelyan, M.; Rashid, T.; de Witt, C. S.; Farquhar, G.; Nardelli, N.; Rudner, T. G.; Hung, C.-M.; Torr, P. H.; Foerster, J.; and Whiteson, S. 2019. The StarCraft Multi-Agent Challenge. In *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, 2186–2188.

Sharma, A.; Gu, S.; Levine, S.; Kumar, V.; and Hausman, K. 2020. Dynamics-Aware Unsupervised Discovery of Skills. In *Proc. Int. Conf. Learn. Represent.*

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015a. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *Proc. Int. Conf. Mach. Learn.*, volume 37, 2256–2265.

Sohl-Dickstein, J.; Weiss, E. A.; Maheswaranathan, N.; and Ganguli, S. 2015b. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *Proc. Int. Conf. Mach. Learn.*, volume 37, 2256–2265.

Son, K.; Kim, D.; Kang, W. J.; Hostallero, D.; and Yi, Y. 2019. QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning. In *Proc. Int. Conf. Mach. Learn.*, volume 97, 5887–5896.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *Proc. Int. Conf. Learn. Represent.*

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; and Graepel, T. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning. In *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, 2085–2087.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1-2): 181–211.

Tian, Z.; Chen, R.; Hu, X.; Li, L.; Zhang, R.; Wu, F.; Peng, S.; Guo, J.; Du, Z.; Guo, Q.; and Chen, Y. 2023. Decompose a Task into Generalizable Subtasks in Multi-Agent Reinforcement Learning. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 36, 78514–78532.

Venkatraman, S.; Khaitan, S.; Akella, R. T.; Dolan, J.; Schneider, J.; and Berseth, G. 2024. Reasoning with Latent Diffusion in Offline Reinforcement Learning. In *Proc. Int. Conf. Learn. Represent.*

Wang, J.; Ren, Z.; Liu, T.; Yu, Y.; and Zhang, C. 2021a. QPLEX: Duplex Dueling Multi-Agent Q-Learning. In *Proc. Int. Conf. Learn. Represent.*

Wang, T.; Dong, H.; Lesser, V.; and Zhang, C. 2020a. ROMA: Multi-Agent Reinforcement Learning with Emergent Roles. In *Proc. Int. Conf. Mach. Learn.*, volume 119, 9876–9886.

Wang, T.; Gupta, T.; Mahajan, A.; Peng, B.; Whiteson, S.; and Zhang, C. 2021b. RODE: Learning Roles to Decompose Multi-Agent Tasks. In *Proc. Int. Conf. Learn. Represent.*

Wang, T.; Wang, J.; Zheng, C.; and Zhang, C. 2020b. Learning Nearly Decomposable Value Functions via Communication Minimization. In *Proc. Int. Conf. Learn. Represent.*

Wang, W.; Yang, T.; Liu, Y.; Hao, J.; Hao, X.; Hu, Y.; Chen, Y.; Fan, C.; and Gao, Y. 2023. ASN: Action Semantics Network for Multiagent Reinforcement Learning. *Autonomous Agents and Multi-Agent Systems*, 37(1): 1–28.

Wang, Z.; Hunt, J. J.; and Zhou, M. 2023. Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning. In *Proc. Int. Conf. Learn. Represent.*

Wen, M.; Kuba, J. G.; Lin, R.; Zhang, W.; Wen, Y.; Wang, J.; and Yang, Y. 2022. Multi-Agent Reinforcement Learning is a Sequence Modeling Problem. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 35, 24602–24616.

Xu, Q.; Zhu, Y.; Wu, X.; and Chen, C. 2025. High-order Interactions Modeling for Interpretable Multi-Agent Q-Learning. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 38.

Xu, Z.; Bai, Y.; Zhang, B.; Li, D.; and Fan, G. 2023. HAVEN: Hierarchical Cooperative Multi-Agent Reinforcement Learning with Dual Coordination Mechanism. In *Proc. AAAI Conf. Artif. Intell.*, volume 37, 11735–11743.

Yang, J.; Borovikov, I.; and Zha, H. 2020. Hierarchical Cooperative Multi-Agent Reinforcement Learning with Skill Discovery. In *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, 1566–1574.

Yang, M.; Zhao, J.; Hu, X.; Zhou, W.; Zhu, J.; and Li, H. 2022a. LDSA: Learning Dynamic Subtask Assignment in Cooperative Multi-Agent Reinforcement Learning. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 35, 11735–11743.

Yang, Y.; Chen, G.; Wang, W.; Hao, X.; Hao, J.; and Heng, P.-A. 2022b. Transformer-based Working Memory for Multiagent Reinforcement Learning with Action Parsing. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 35, 1–14.

Zang, Y.; He, J.; Li, K.; Fu, H.; Fu, Q.; Xing, J.; and Cheng, J. 2023. Automatic Grouping for Efficient Cooperative Multi-Agent Reinforcement Learning. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 36, 64896–64917.

Zhang, C.; and Lesser, V. 2011. Coordinated multi-agent reinforcement learning in networked distributed POMDPs. In *Proc. AAAI Conf. Artif. Intell.*, volume 25, 764–770.

Zhang, C.; Lesser, V. R.; and Abdallah, S. 2010. Self-Organization for Coordinating Decentralized Reinforcement Learning. In *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, 739–746.

Zhang, K.; Sun, T.; Tao, Y.; Genc, S.; Mallya, S.; and Basar, T. 2020. Robust multi-agent reinforcement learning with model uncertainty. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 33, 10571–10583.

Zhang, M.; Cai, Z.; Pan, L.; Hong, F.; Guo, X.; Yang, L.; and Liu, Z. 2024. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(6): 4115–4128.