

---

# ReDiG: Reinforced Diffusion on Graphs for Decentralized Coordinated Multi-Robot Navigation with Smooth Formation Adaptation

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 Coordinated navigation is a fundamental capability for multi-robot teams to traverse  
2 complex unstructured environments. During navigation, robots are often required  
3 to maintain mission-specific formations, such as wedge formations for enhanced  
4 visibility and area coverage. However, rigid formations can hinder navigation in  
5 challenging scenarios like narrow corridors, which demand formation adaptation.  
6 Reinforcement learning (RL) is commonly used for coordinated multi-robot navi-  
7 gation due to its ability to learn through interaction with the environment. However,  
8 its step-wise decision-making process often results in jerky motion. In contrast,  
9 diffusion models generate smoother trajectories through probabilistic denoising,  
10 but rely heavily on high-quality demonstrations. Collecting such demonstrations is  
11 challenging in multi-robot systems due to the coordination and synchronization  
12 required among individual robots. To address these issues, we introduce a novel  
13 method named *Reinforced Diffusion on Graphs* (ReDiG) to enable decentralized co-  
14 ordinated multi-robot navigation with smooth formation adaptation. Under a unified  
15 learning framework, ReDiG integrates: (1) graph learning for decentralized coordi-  
16 nation to enable formation adaptation, (2) diffusion models for generating smooth  
17 individual robot trajectories, and (3) RL to refine noisy demonstrations by leverag-  
18 ing feedback from environment interaction, which enables robot synchronization  
19 and guides effective diffusion training. We evaluate ReDiG through extensive  
20 experiments in both indoor and outdoor environments using physical robot teams  
21 and robotics simulations. Experimental results show that ReDiG enables smooth  
22 formation adaptation and achieves state-of-the-art performance in coordinated  
23 multi-robot navigation within complex unstructured environments. More details  
24 are available on the project website: <https://anonymous23885.github.io/ReDiG/>.

25 

## 1 Introduction

26 Multi-robot systems have gained significant attention in recent years due to their advantages, such as  
27 redundancy [1], operational efficiency [2], and scalability [3], which make them well-suited for a  
28 wide range of real-world large-scale applications, including search and rescue [4-6], transportation  
29 [7-9], and space exploration [10-12]. Coordinated multi-robot navigation is an essential capability,  
30 which enables teams of robots to traverse environments in a synchronized manner. While navigating  
31 as a coordinated team, robots are typically required to maintain task-specific formations, such as a  
32 wedge formation, to enhance visibility and area coverage in the direction of movement. To support  
33 this, multi-robot coordination is critical, which enables robots to share and integrate information  
34 with their teammates, particularly in a decentralized manner. However, rigid formations can hinder  
35 progress and impede navigation in complex environments, such as narrow corridors. To overcome

36 this, multi-robot synchronization is essential, allowing robots to align their actions in both time and  
37 space to cohesively maintain and adapt formations. Furthermore, such constrained scenarios often  
38 require frequent motion adjustments for formation adaptation, which can lead to non-smooth and  
39 inefficient traversal, which results in an additional challenge that must be addressed.

40 Due to the importance of coordinated multi-robot navigation, a wide range of methods have been  
41 developed. Traditional formation control methods, including leader-follower approaches [13-15] and  
42 virtual region methods [16-19], often rely on preset rigid formation shapes. However, such rigid  
43 formations lack the flexibility required to adapt to complex environments. Learning-based methods,  
44 such as reinforcement learning (RL) [20-24], address this limitation by optimizing robot actions  
45 through interaction with the environment to maximize rewards. However, due to the step-wise nature  
46 of RL decision-making, robots stop or adjust their motion frequently in pursuit of higher rewards,  
47 resulting in jerky trajectories. Generative models such as diffusion models [25-28] have shown  
48 promising results recently for generating smooth trajectories in single robot navigation by iteratively  
49 denoising demonstrations. However, collecting expert demonstrations in multi-robot systems is  
50 challenging due to the need for precise coordination and synchronization among individual robots.

51 To address the above limitations in the current state of the art, we introduce a novel approach called  
52 *Reinforced Diffusion on Graph* (**ReDiG**) to enable a new multi-robot capability of decentralized  
53 coordinated multi-robot navigation with smooth formation adaptation. Specifically, ReDiG represents  
54 a robot team as a graph, where each node denotes a robot along with its attributes such as position,  
55 velocity, goal, and obstacle proximity, and each edge encodes the spatial relationship between robot  
56 pairs. ReDiG integrates three learning components into a unified framework to enable coordinated  
57 multi-robot navigation with adaptive formation control: First, a decentralized graph neural network  
58 computes team-level embeddings from the graph representation, which captures the team context  
59 necessary for effective coordination. Second, a diffusion model on each robot learns a navigation  
60 policy that generates smooth, collision-free trajectories conditioned on the team embedding. Third,  
61 an actor-critic RL module synchronizes individual robot actions through iterative interaction with the  
62 environment, enabling formation adaptation in dynamic and constrained scenarios.

63 Our primary contribution is the introduction of the ReDiG approach to enable coordinated multi-robot  
64 navigation with smooth formation adaptation. The specific novelties include:

- 65 • From the perspective of robot capability, we develop one of the first learning-based solutions  
66 for decentralized coordinated multi-robot navigation in challenging scenarios. RiDiG not  
67 only enables a new capability of coordinated navigation with formation adaptation, but also  
68 improves motion trajectory smoothness for individual robots, particularly when traversing  
69 narrow corridors that require frequent adjustments.
- 70 • From the perspective of algorithmic novelty, we introduce a unified framework in ReDiG  
71 that integrates decentralized graph learning for team-level coordination, diffusion models for  
72 smooth individual trajectory generation, and actor-critic reinforcement learning to iteratively  
73 refine these trajectories through interaction with the environment, ensuring coordinated and  
74 synchronized multi-robot adaptive formation control across the robot team.

## 75 2 Related Work

76 **Coordinated Multi-Robot Navigation.** Traditional coordinated navigation with formation control  
77 often relies on manually designed strategies, such as the leader-follower structure [13-15], where  
78 follower agents are programmed to maintain formation by tracking a designated leader. Virtual region  
79 methods [16-19] allow teams to adjust their formation within predefined spatial constraints. However,  
80 these formations are often rigid and lack the ability to adapt based on environments. For learning  
81 methods, Graph Neural Networks (GNNs) have been introduced to improve team coordination  
82 and communication in a decentralized manner [29-32]. Recently, RL has been widely applied in  
83 multi-agent systems [23, 24, 10, 21], which enables robot teams to learn complex, coordinated  
84 behaviors that are difficult to manually design. Despite these advantages, RL often suffers from step-  
85 wise decision-making, which can lead robots to frequently stop or adjust their motion to maximize  
86 immediate rewards, which result in jerky trajectories and reduced motion smoothness.

87 **Diffusion Models for Robot Policy Learning.** Diffusion models have gained significant attention  
88 in robotics for generating smooth trajectories through iterative denoising. For single-robot planning,

diffusion models are used to sample motion plans conditioned on task objectives [33, 26] and environmental context [34, 37]. Hierarchical diffusion models have been proposed to handle long-horizon planning problems [38, 39]. Recent works extend diffusion models to multi-robot systems to enable coordinated trajectory generation. Motion Diffuser enables trajectory prediction for multi-robot through cost function [27], Resilient Distributed Diffusion enables resilient distributed control under adversarial conditions based on the centerpoint concept [40], MMD generates collision-free multi-robot trajectories based on single-robot data [28]. GSC samples from skill model to generate long-horizon plans. However, applying diffusion models to multi-robot systems remains challenging due to the need for large-scale, well-synchronized expert demonstrations, which are difficult to obtain.

**Diffusion RL.** Diffusion models have been integrated with RL to improve policy through generative sampling guided by RL signals. Diffusion-QL [41] biases diffusion sampling toward high-value actions using Q-learning. CEP [42] defines contrastive energy scores to steer denoising. SRDP [43] enhances out-of-distribution (OOD) generalization by reconstructing state representations. CPQL [44] introduces consistency modeling for stable policy learning. Diffuser [45] applies reward signals at the trajectory level, while Simple Hierarchical [39] extends this to multi-task settings using hierarchical diffusion policies. MTDiff [46] further supports multi-task planning through transformer-based conditioning. However, offline RL struggles to generate optimal behaviors when expert demonstrations are limited or missing. DIPO [47] is the first to employ Online RL by updating actions through gradient ascent. QVPO [48] introduces a Q-weighted variational loss to ensure robust policy improvement. However, none of these methods have been applied to multi-robot systems, particularly those requiring multi-robot coordination and synchronization.

### 3 Approach

#### 3.1 Problem Definition

We represent a team of  $n$  robots as an undirected graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ . Each robot is represented as a node within the node set  $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ . The attributes of each robot  $i$  are represented by  $\mathbf{v}_i = [\mathbf{p}_i, \mathbf{q}_i, \mathbf{g}_i]$ , where  $\mathbf{p}_i = [p_i^x, p_i^y]$  denotes its position,  $\mathbf{g}_i = [g_i^x, g_i^y]$  represents its goal position, and  $\mathbf{q}_i = [q_i^x, q_i^y]$  defines its velocities along the x and y directions. The edge matrix  $\mathbf{E} = \{a_{i,j}\}_{n \times n}$  represents the spatial adjacency between robots, where  $a_{i,j} = 1$ , if the  $i$ -th robot and the  $j$ -th robot are within a radius; otherwise  $a_{i,j} = 0$ . We define the state of the  $i$ -th robot as the concatenation of its attributes,  $\mathbf{s}_i = [\mathbf{p}_i, \mathbf{q}_i, \mathbf{g}_i, d_i]$ , where  $d_i$  represents the distance from the nearest obstacle to the  $i$ -th robot. We further define the action of the  $i$ -th robot as  $\mathbf{a}_i = [v_i^x, v_i^y]$ , where  $v_i^x$  and  $v_i^y$  represent the output velocities in the  $x$  and  $y$  directions, respectively.

We address decentralized coordinated multi-robot navigation, while ensuring both smooth individual robot motion and adaptive formation control for the entire robot team. Specifically, we aim to address:

- **Decentralized Coordination** enables each robot to coordinate with teammates by sharing and integrating state information in a fully decentralized manner, ensuring coherent multi-robot coordination without centralized control.
- **Smooth Trajectory Generation** enables each robot to generate smooth, collision-free trajectories that enable stable motion and efficient goal-reaching.
- **Adaptive Formation Synchronization** enables robots to align their motion in both time and space, thus maintaining mission-specific formations while dynamically adjusting their spatial team configuration to navigate through constrained environments.

Given this problem definition, we discuss our ReDiG approach, which is illustrated in Figure I

##### 3.1.1 Decentralized Multi-Robot Coordination

Given the robot team's graph representation  $\mathcal{G}$  and robot states  $\mathbf{s}_i$  of the  $i$ -th robot, we develop a graph neural network  $\phi$  to encode spatial relationships among robots in the team and then compute the embedding  $\mathbf{h}_i = \phi(\mathbf{s}_i, \mathcal{G})$  of the team state for each  $i$ -th robot. The graph network  $\phi$  consists of linear layers that first map the robot state  $\mathbf{s}_i$  to the individual embedding  $\mathbf{z}_i$  of the  $i$ -th robot by  $\mathbf{z}_i = \mathbf{W}^z \mathbf{s}_i$ , where  $\mathbf{W}^z$  denotes the learnable weight matrix of the linear layers. By using message passing,  $\phi$  then aggregates  $\mathbf{z}_i$  with the embeddings of all other teammates within a spatial radius to compute the final team state embedding  $\mathbf{h}_i$  for the  $i$ -th robot, which is defined as  $\mathbf{h}_i = \mathbf{W}^h \mathbf{z}_i + \sum_{j \in \mathcal{N}(i)} \mathbf{W}^h (\mathbf{z}_j - \mathbf{z}_i)$ ,

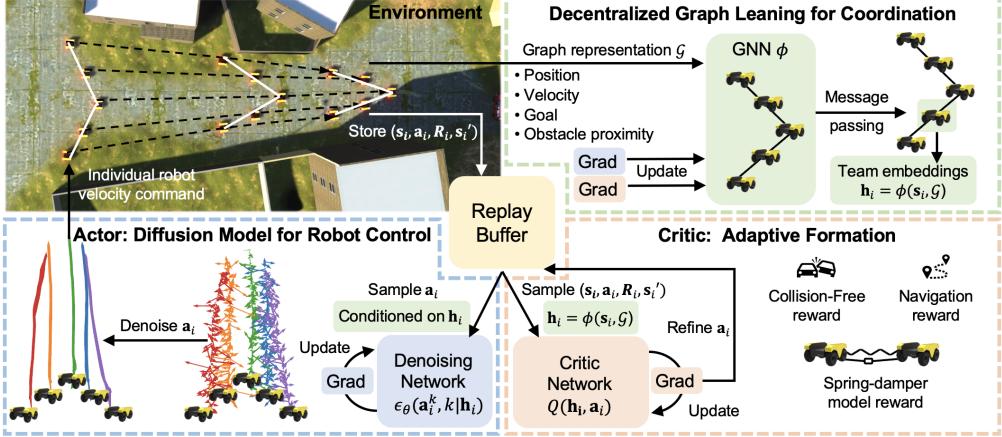


Figure 1: Under a unified learning framework, ReDiG integrates decentralized graph learning for multi-robot coordination, diffusion models for individual trajectory generation, and reinforcement learning for adaptive formation synchronization.

140 where  $\mathbf{W}^h$  is the learnable weight matrix. For each  $i$ -th robot, the first term captures its individual  
 141 state, while the second term encodes its relative spatial relationships with teammates, representing  
 142 the team-level context. The learnable weight matrix  $\mathbf{W}^h$  enables each robot to determine which  
 143 information from teammates is most critical for its decision-making. The graph neural network  $\phi$  is  
 144 agnostic to the number of robots, which is able to aggregate arbitrary embeddings from neighborhood  
 145 robots within a spatial radius, thus enabling decentralized team-level context embedding.

### 146 3.1.2 Smooth Action Generation for Individual Robots

147 We develop a diffusion model  $\psi$  conditioned on the team-level embedding  $\mathbf{h}_i$  to generate actions  
 148  $\mathbf{a}_i$  for each robot, while maintaining awareness of its teammates' states. Formally, we model  
 149 the individual action probabilistically as a conditional distribution  $p(\mathbf{a}_i|\mathbf{h}_i)$ . However, due to the  
 150 high dimensionality of the continuous action space, directly modeling or sampling from  $p(\mathbf{a}_i|\mathbf{h}_i)$   
 151 is intractable. To address this, we introduce a diffusion model  $\psi$  to approximate the conditional  
 152 distribution via a parameterized denoising process.

153 **Smooth Trajectory Generation.** The diffusion model is built upon a probabilistic diffusion process  
 154 that consists of a forward and a reverse process. In the forward process, Gaussian noise is progressively  
 155 added to the ground-truth action  $\mathbf{a}_i^0$ , which can be provided through expert demonstrations. A noise  
 156 variance schedule  $\{\beta^k\}_{k=1}^K$  with  $\beta^k \in (0, 1)$  determines the variance of the noise added at each  
 157 step  $k$ . This results in a sequence of noisy actions  $\mathbf{a}_i^1, \mathbf{a}_i^2, \dots, \mathbf{a}_i^k$ . Formally, the forward process is  
 158 defined as  $q(\mathbf{a}_i^k|\mathbf{a}_i^{k-1}) = \mathcal{N}(\mathbf{a}_i^k; \sqrt{1 - \beta^k} \mathbf{a}_i^{k-1}; \beta^k \mathbf{I})$ . In the reverse process, the diffusion model  $\psi$   
 159 iteratively reconstructs the ground-truth action  $\mathbf{a}_i^0$ . At each step  $k$ , the model learns to denoise  $\mathbf{a}_i^k$   
 160 to recover  $\mathbf{a}_i^{k-1}$ , which progressively refines the trajectory toward the demonstrated behavior. This  
 161 reverse process is governed by three key coefficients that jointly ensure smooth transitions across  
 162 diffusion steps:  $\lambda^k = 1/\sqrt{1 - \beta^k}$  scales the denoised prediction,  $\alpha^k = \prod_{i=1}^k (1 - \beta^i)$  progressively  
 163 reduces the influence of noise, and  $\sigma^k$ , derived from  $\beta_k$ , regulates exploration by controlling the  
 164 amount of noise injected during sampling. To estimate the noise added during the forward process,  
 165 we train a neural network  $\epsilon_\theta(\mathbf{a}_i^k, k)$  parameterized by  $\theta$ . Then, the inverse process is defined as  
 166  $\mathbf{a}_i^{k-1} = \lambda^k (\mathbf{a}_i^k - \alpha_k \epsilon_\theta(\mathbf{a}_i^k, k)) + \sigma^k z$ , where  $z \sim \mathcal{N}(0, \mathbf{I})$  denotes standard Gaussian noise. The  
 167 unstructured Gaussian noise introduces perturbations to the ground-truth action, which can result in  
 168 jerky motion. The denoising network  $\epsilon_\theta(\mathbf{a}_i^k, k)$  learns to remove these high-frequency perturbations,  
 169 guided by three coefficients in the reverse process, thereby enables smooth trajectory generation.

170 **Incorporating Team Context.** To enable team-aware trajectory generation for individual robots,  
 171 we condition each robot's control on the team state embedding  $\mathbf{h}_i$ . Formally, this is expressed as:  
 172  $\mathbf{a}_i^{k-1} = \lambda^k (\mathbf{a}_i^k - \alpha^k \epsilon_\theta(\mathbf{a}_i^k, k | \mathbf{h}_i)) + \sigma^k z$ . We learn this conditioned reverse process by training the  
 173 model to predict the actual noise added during the forward process, progressively recovering the

174 ground-truth actions. The loss function is defined as

$$\min_{\epsilon_\theta} \mathbb{E}_{\mathbf{h}_i, \mathbf{a}_i^0} [\|\epsilon_k - \epsilon_\theta(\mathbf{a}_i^0 + \epsilon_k, k | \mathbf{h}_i)\|_2^2] \quad (1)$$

175 where  $\epsilon_k$  denotes the noise added at the  $k$ -th step during the forward process. The denoising network  
 176  $\epsilon_\theta$  serves as the core learnable component of the diffusion model  $\psi$ , iteratively denoising a randomly  
 177 sampled noise through a sequence of reverse diffusion steps conditioned on the team-level embedding  
 178 to generate individual actions.

179 We theoretically analyze the upper bound of the loss defined in Eq. (1). The loss defined in Eq.  
 180 (1) measures how well the learned action distribution  $\hat{p}_\theta(\mathbf{a}_i^0 | \mathbf{h}_i)$  approximates the true distribution  
 181  $p(\mathbf{a}_i^0 | \mathbf{h}_i)$ . This upper bound consists of three components: 1). *Prior Mismatch* measures how closely  
 182 the marginal distribution  $q(\mathbf{a}_i^K)$  at the final diffusion step matches the Gaussian prior  $p(\mathbf{a}_i^K) =$   
 183  $\mathcal{N}(0, I)$ . A poor match degrades initial samples, reducing trajectory feasibility. 2). *Matching*  
 184 *Error* captures the cumulative noise prediction error across all diffusion steps. Inaccurate denoising  
 185 leads to poor action reconstruction, affecting trajectory smoothness. 3). *Discretization Error* arises  
 186 from approximating a continuous diffusion process with a finite number of steps. This may result  
 187 in suboptimal trajectories.

188 **Theorem 1.** *Let  $\hat{p}_\theta(\mathbf{a}_i^0 | \mathbf{h}_i)$  be the learned action distribution and  $p(\mathbf{a}_i^0 | \mathbf{h}_i)$  the true distribution, with  
 189  $K$  finite diffusion steps approximating the continuous process. The upper bound of the loss of the  
 190 Conditional Diffusion Policy on Graph is bounded by the combination of prior mismatch, matching  
 191 error, and discretization error:*

$$\text{KL}(\hat{p}_\theta(\mathbf{a}_i^0 | \mathbf{h}_i) \| p(\mathbf{a}_i^0 | \mathbf{h}_i)) \leq \underbrace{\text{KL}\left(q(\mathbf{a}_i^K) \| p(\mathbf{a}_i^K)\right)}_{\text{Prior mismatch}} + \underbrace{\sum_{k=1}^K \mathbb{E}_{\mathbf{a}_i^0, \epsilon, k} \left[\left\|\epsilon - \epsilon_\theta(\mathbf{a}_i^k, k | \mathbf{h}_i)\right\|_2^2\right]}_{\text{Matching error}} + \underbrace{\mathcal{O}\left(\sum_{k=1}^K (\beta^k)^2\right)}_{\text{Discretization error}}$$

192 Proof: See Appendix A.1.

### 193 3.1.3 Adaptive Formation Synchronization

194 To maintain and adapt formations as robot teams navigate through constrained environments, it is  
 195 essential to synchronize individual robot motions in both spatial and temporal dimensions. However,  
 196 the diffusion model is trained using expert demonstrations, which are often imperfect and noisy (e.g.,  
 197 when they are provided by human experts or algorithms with access to privilege information). To  
 198 address this limitation, we design a new actor-critic RL framework that uses the unsynchronized  
 199 actions generated by the diffusion model as initial guidance and refines the actions through reward-  
 200 driven feedback, while simultaneously promoting formation-aware synchronization across the team.

201 **Synchronization for Formation Adaptation.** To enable multi-robot synchronization for adaptive  
 202 formation control, we design a reward function inspired by the spring-damper model [49, 50]. The  
 203 spring component maintains a balance between keeping pairs of robots close enough to navigate  
 204 through constrained environments (e.g., narrow corridors) and maintaining sufficient distance to avoid  
 205 collisions, thus providing the flexibility necessary for adaptive formation control. The spring effect is  
 206 modeled as  $|d_{i,j} - p_{i,j}|$ , where  $d_{i,j}$  represents the expected distance in the original rigid formation  
 207 and  $p_{i,j}$  is the actual distance between the  $i$ -th and  $j$ -th robots, computed as  $\|\mathbf{p}_i - \mathbf{p}_j\|_2$ . The damper  
 208 component mitigates oscillations and prevents overshooting by regulating the relative velocities  
 209 between robot pairs, which is defined as  $q_{i,j} = \|\mathbf{q}_i - \mathbf{q}_j\|_2$ . By integrating both components, we  
 210 formulate the spring-damper model as a reward function, defined as  $R^{adp} = \sum_{\mathbf{v}_i, \mathbf{v}_j \in \mathcal{V}} -\omega|d_{i,j} -$   
 211  $p_{i,j}| - (1 - \omega)q_{i,j}$ , where  $\omega$  is a hyperparameter that balances the contributions of the spring and  
 212 damper components. The final reward  $R = R^{adp} + R^{collision}$ , where  $R^{collision}$  denotes the obstacle  
 213 avoidance reward for each individual robot [51].

214 **Individual Action Refinement.** To refine the actions generated by the diffusion model  $\psi$ , we treat  
 215 it as the actor network. A deep Q network  $Q(\mathbf{h}_i, \mathbf{a}_i)$  is designed to serve as a critic network to  
 216 evaluate the actor. The gradient from the critic is then backpropagated through both the decentralized  
 217 graph neural network  $\phi$  and the diffusion model  $\psi$  to refine individual robot actions conditioned on  
 218 the team-level context.

219 Formally, the critic network  $Q(\mathbf{h}_i, \mathbf{a}_i)$  consists of an encoder followed by a MLP to generate the  
 220 Q-value of a state-action pair  $(\mathbf{h}_i, \mathbf{a}_i)$ . The target value (i.e., TD target) of the critic network is

---

**Algorithm 1:** Unified Training for Reinforced Diffusion on Graph (ReDiG)

---

**Input** : Team graph representation  $\mathcal{G}$ , individual robot state  $\mathbf{s}$ , replay buffer  $\mathcal{M}$

**Output** : Trained graph neural network  $\phi$ , diffusion model  $\psi$ , and critic networks  $Q_1, Q_2$

```
1 Initialize all learnable components in ReDiG, including  $\mathbf{W}^z, \mathbf{W}^h, \epsilon_\theta, Q_1, Q_2$ ;
2 Generate individual ground-truth actions using individual expert policies, and store  $(\mathbf{s}_i, \mathbf{a}_i, R_i, \mathbf{s}'_i)$  in  $\mathcal{M}$ ;
3 while Not converged do
4   Use the decentralized graph neural network to generate team embeddings  $\mathbf{h}_i = \phi(\mathcal{G})$ ,
5   Use diffusion to generate actions  $\{\mathbf{a}_i\}^n = \{\psi(\mathbf{h}_i)\}^n$ ;
6   Compute the gradient of diffusion loss  $\nabla_{\mathbf{a}_i^k} \epsilon_\theta(\mathbf{a}_i^k, k | \mathbf{h}_i)$  according to Eq. (1);
7   Update denoising network  $\epsilon_\theta$  in the diffusion model  $\psi$  according to the gradient  $\nabla_{\mathbf{a}_i^k} \epsilon_\theta(\mathbf{a}_i^k, k | \mathbf{h}_i)$ ;
8   Apply  $\{\mathbf{a}_i\}^n$  in the environment and store the transition  $(\mathbf{s}_i, \mathbf{a}_i, R_i, \mathbf{s}'_i)$  to  $\mathcal{M}$ ;
9   Sample batch  $(\mathbf{s}_i, \mathbf{a}_i, R_i, \mathbf{s}'_i)$  from  $\mathcal{M}$ ;
10  Compute the gradient of the critic loss  $\nabla_{\mathbf{a}_i} Q(\mathbf{h}_i, \mathbf{a}_i)$  according to Eq. (2)
11  Refine the action  $\mathbf{a}_i = \mathbf{a}_i + \eta \nabla_{\mathbf{a}_i} Q(\mathbf{h}_i, \mathbf{a}_i)$ 
12  Update GNN  $\mathbf{W}^z$  and  $\mathbf{W}^h$  according to the gradient  $\nabla_{\mathbf{a}_i^k} \epsilon_\theta(\mathbf{a}_i^k, k | \mathbf{h}_i)$  and  $\nabla_{\mathbf{a}_i} Q(\mathbf{h}_i, \mathbf{a}_i)$ ;
13  Update  $Q_1$  and  $Q_2$  according to the gradient  $\nabla_{\mathbf{a}_i} Q(\mathbf{h}_i, \mathbf{a}_i)$ 
14 end
15 return  $\mathbf{W}^z, \mathbf{W}^h, \epsilon_\theta, Q_1, Q_2$ 
```

---

221 computed using the Bellman equation by combining the immediate reward with a discounted estimate  
222 of future value, defined as  $\mathbf{y}_i = R(\mathbf{s}_i, \mathbf{a}_i) + \gamma Q(\mathbf{h}'_i, \mathbf{a}'_i)$ , where  $\gamma$  is the discount factor, and  $\mathbf{h}'_i, \mathbf{a}'_i$   
223 denote the state embedding and action at the next time step for robot  $i$ . To train the critic network,  
224 we define the critic loss as  $\mathbb{E}[(Q(\mathbf{h}_i, \mathbf{a}_i) - \mathbf{y}_i)^2]$ , which minimizes the square error between the  
225 predicted value and the target value. To stabilize training, we introduce two identical critic networks  
226  $Q_1(\mathbf{h}_i, \mathbf{a}_i)$  and  $Q_2(\mathbf{h}_i, \mathbf{a}_i)$ , and modify the loss function as follows:

$$\mathbb{E}[(Q_1(\mathbf{h}_i, \mathbf{a}_i) - \mathbf{y}_i)^2 + (Q_2(\mathbf{h}_i, \mathbf{a}_i) - \mathbf{y}_i)^2] \quad (2)$$

227 The use of two identical critic networks mitigates overestimation bias by encouraging consistent value  
228 estimates and reducing variance in target value predictions, which leads to stable and reliable policy  
229 learning. Then, we compute the gradient of the minimum estimated value from  $Q_1$  and  $Q_2$  with  
230 respect to the action as  $\nabla_{\mathbf{a}_i} Q(\mathbf{h}_i, \mathbf{a}_i)$ , where  $Q = \min(Q_1, Q_2)$ . This gradient indicates the direction  
231 of the action  $\mathbf{a}_i$  refinement, which increases its estimated value. The refined action is defined as  
232  $\mathbf{a}_i = \mathbf{a}_i + \eta \nabla_{\mathbf{a}_i} Q(\mathbf{h}_i, \mathbf{a}_i)$ , where  $\eta$  is the step size controlling how large to modify the action.

### 233 3.1.4 Unified ReDiG Training and Decentralized Execution

234 **Unified Training of Graph, Diffusion, and RL Networks.** ReDiG includes three learning com-  
235 ponents, including a decentralized graph neural network  $\phi$  for multi-robot coordination, a diffusion  
236 model  $\psi$  for smooth individual trajectory generation, and an actor-critic RL to refine robot actions. To  
237 train all these components in a unified learning framework, ReDiG computes the gradients from both  
238 the diffusion loss in Eq. (1) and the critic loss in Eq. (2). Then, the gradients are back-propagated  
239 to update the graph network weight matrix  $\mathbf{W}^z$  and  $\mathbf{W}^h$ , the denoising network  $\epsilon_\theta$  in the diffusion  
240 model  $\psi$ , and the critic network  $Q_1$  and  $Q_2$ . The unified training algorithm for ReDiG is presented in  
241 Algorithm I with detailed explanations provided in the Appendix.

242 **Decentralized Execution.** ReDiG performs fully decentralized execution on multi-robot systems.  
243 During execution, each robot uses a 2D occupancy map of the environment. During coordinated  
244 navigation, each robot broadcasts its state to nearby teammates within a distance radius via wireless  
245 communication (e.g., Wi-Fi), enabling it to determine the relative positions of neighboring robots.  
246 Each robot then independently applies the trained graph neural network  $\phi$ , with shared weights  $\mathbf{W}^z$   
247 and  $\mathbf{W}^h$ , to process information from its neighbors and compute its team embedding  $\mathbf{h}_i$ . Finally,  
248 each robot runs its own copy of the trained diffusion policy  $\psi$  and executes velocity commands  $\mathbf{a}_i$   
249 based on its local embedding  $\mathbf{h}_i$ .

250 **Time Complexities.** *Training time complexity* is dominated by  $O(n^2)$ , where  $n$  is the number of  
251 robots. The decentralized graph network generates state embeddings has an  $O(L_g n^2)$  complexity,

Table 1: Quantitative comparison of ReDiG and prior methods from Gazebo simulations in ROS2.

Method	Circle Formation						Wedge Formation						Line Formation					
	SR (%)	TT (sec)	$\delta < 0.5$	$\delta < 0.1$	$\delta < 0.03$	SR (%)	TT (sec)	$\delta < 0.5$	$\delta < 0.1$	$\delta < 0.03$	SR (%)	TT (sec)	$\delta < 0.5$	$\delta < 0.1$	$\delta < 0.03$	SR (%)	TT (sec)	
L&F [13]	60.00	15.80	75.11	70.10	66.04	60.00	17.12	81.44	66.70	62.86	60.00	13.35	63.76	55.59	55.59	60.00	13.35	
DGNN [23]	100.00	34.70	60.41	58.91	58.91	100.00	63.70	47.85	42.33	41.92	100.00	36.80	27.90	20.16	20.16	100.00	36.80	
AFOR [49]	100.00	30.50	92.89	90.40	88.66	100.00	51.50	91.39	90.35	87.96	100.00	183.50	88.60	85.13	72.90	100.00	183.50	
ReDiG (ours)	100.00	13.10	84.51	81.52	78.97	100.00	12.46	82.43	81.21	80.37	100.00	10.03	91.17	87.68	80.91	100.00	10.03	

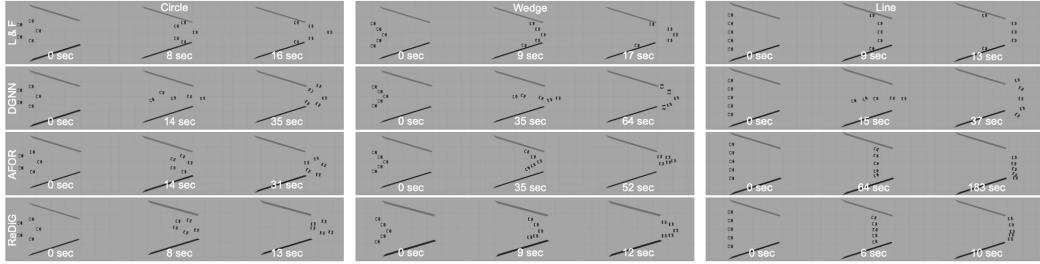


Figure 2: Qualitative results from Gazebo simulations on formation adaptation using Limo robots.

252 where  $L_g$  is the number of GNN layers,  $n^2$  is the number of edges computed for communications.  
253 The denoising network training has an  $O(T_d B_d L_d)$  complexity, where  $T_d$  is the number of denoising  
254 steps,  $B_d$  is the sampled batch size, and  $L_d$  is the number of denoising network layers. The critic  
255 network training has an  $O(B_c L_c n)$  complexity, where  $B_c$  is the sampled batch size,  $L_c$  is the number  
256 of critic network layers. The action refinement using gradient ascent has an  $O(G_a B_a n)$  complexity,  
257 where  $G_a$  is the number of gradient steps,  $B_a$  is the sampled batch size, and each gradient step  
258 requires critic evaluation with  $O(n)$  complexity. Combining all terms, the overall complexity for  
259 training is  $O(I(n^2 + T_d B_d L_d + B_c L_c n + G_a B_a n))$ , where  $I$  is the number of training iterations.  
260 *Execution time complexity* is dominated by  $O(n)$ . The complexities of embeddings generation from  
261 graph and actions generation from diffusion policy are  $O(L_g n)$  and  $O(T_d L_d)$ , the overall execution  
262 time complexity is  $O(n + T_s)$ .

## 263 4 Experiments

264 **Experimental Setups.** We comprehensively evaluate our ReDiG approach in three experimental  
265 settings: (1) a standard Gazebo simulation using ROS2, (2) a high-fidelity Unity-based 3D multi-  
266 robot simulator in ROS1, and (3) a physical robot team running ROS2. Each setup involves different  
267 differential-drive robot platforms (e.g., Limo and Warthog robots), and formation shapes (e.g.,  
268 circle, wedge, and line). To follow the identical trajectories, we convert the linear velocity  $a_i$  into  
269 corresponding wheel velocities. All scenarios feature narrow corridors, the robot teams are required  
270 to navigate through confined spaces while dynamically adjusting their formation and preserving its  
271 overall structure. In simulation, robot positions and environmental obstacles are obtained directly from  
272 the Gazebo and Unity simulation. For physical experiments, each robot performs state estimation  
273 and environment mapping using a SLAM-based method [52]. See Appendix A.3 for approach  
274 implementation and training details. All video demonstrations are available on our project website.

275 To demonstrate the effectiveness of ReDiG, we compare it with three prior methods for coordinated  
276 multi-robot navigation, including: (1) Leader and Follower method (**L&F**) [13], where one robot  
277 is designated as the “leader” to guide the team, while the remaining robots act as “followers” that  
278 maintain the formation by tracking the leader’s motion; (2) Decentralized GNN (**DGNN**) [23], which  
279 employs a RL framework to generate velocity commands for each robot, but does not account for  
280 formation control; and (3) Adaptive Formation with Oscillation Reduction (**AFOR**) [49], which is a  
281 RL-based method that incorporates a spring-damper model to enable adaptive formation control, but  
282 does not account for trajectory smoothness and efficiency.

283 To quantitatively evaluate and compare ReDiG with other methods, we employ three metrics, includ-  
284 ing: (1) Success Rate (**SR**) measures the proportion of robots in the team that successfully reach their  
285 goal without collisions, (2) Travel Time (**TT**) represents the total navigation time used by the entire  
286 team to reach their goals. (3) Contextual Formation Integrity (**CFI**) measures the real-time adherence  
287 of the robots to their designated formation, based on a threshold  $\delta$  that determines how strictly the  
288 formation shape must be preserved. See Appendix A.4 for details on CFI and its calculation.

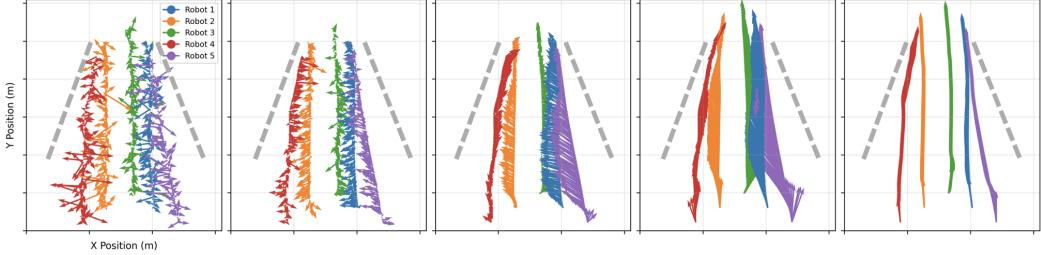


Figure 3: Visualization of ReDiG denoising for a wedge-formation team (left to right). Each colored arrow shows the velocity vector at a diffusion step of each robot. Dashed gray lines indicate obstacles.

289 **Results in Multi-Robot Simulations.** The qualitative results from the Gazebo simulation are  
 290 presented in Figure 2. The L&F method, which relies on a predefined rigid formation, fails to  
 291 navigate narrow corridors, as the outer robots collide with the walls. DGNN does not incorporate  
 292 formation control, resulting in robots passing through the corridor sequentially without coordination.  
 293 Both AFOR and our proposed ReDiG approach integrate a spring-damper model to enable formation  
 294 adaptation. However, AFOR is built upon RL with step-wise action decisions, which results in  
 295 jerky trajectories. In contrast, ReDiG generates significantly smoother trajectories by leveraging a  
 296 diffusion-based policy. Since visualizations alone may not fully capture the impact of jerkiness, we  
 297 further provide a quantitative analysis of motion trajectory smoothness in the discussion.

298 We visualize the denoising process of our ReDiG approach for a robot team in wedge formation, as  
 299 shown in Figure 3. Each arrow represents a velocity vector generated at different denoising steps.  
 300 Starting from pure Gaussian noise, the action of each individual robot is progressively denoised into  
 301 smooth, coordinated motion. This demonstrates ReDiG’s ability to iteratively reconstruct meaningful  
 302 actions that enable both smooth navigation and formation adaptation.

303 The quantitative results are shown in Table 1. The L&F method achieves a 60% success rate due to  
 304 its inability to adapt formations. DGNN, which lacks formation awareness, performs worst in the CFI  
 305 metric. AFOR shows the longest travel time, especially in the line formation, due to the step-wise  
 306 nature of multi-robot RL, which results in inefficient progress caused by jerky actions. Our proposed  
 307 method addresses these limitations and achieves above 82% CFI with the shortest travel time across  
 308 all three formation shapes. Although ReDiG has slightly lower CFI scores compared to AFOR, this  
 309 is a reasonable trade-off for the significant gain in efficiency. These results highlight the effectiveness  
 310 of ReDiG in enabling smooth and efficient formation adaptation in complex environments.



Figure 4: Qualitative results from Unity3D simulations using a team of differential-drive Warthog  
 robots that follow circle, wedge, and line formations while navigating unstructured narrow corridors.

311 In addition to Gazebo simulation, we further evaluate our approach in a Unity3D-based simulator  
 312 integrated with ROS1 for multi-robot perception and control. These outdoor environments feature  
 313 extended narrow pathways formed by closely spaced buildings and uneven flooded terrain, requiring  
 314 the robot team to navigate long, curved trajectories while maintaining adaptive formation control.  
 315 As shown in Figure 4, ReDiG enables Warthog robot teams to dynamically adapt their formation in  
 316 response to environmental constraints, navigating smoothly and reaching goals without collisions.  
 317 In circle formation scenarios, the robot team successfully traverses multiple narrow corridors by  
 318 continuously adapting the formation to accommodate tighter spaces. The results demonstrate the  
 319 effectiveness of ReDiG in achieving smooth and adaptive formation control in complex environments.

320 **Validation on Physical Robot Teams.** We validate ReDiG through case studies using physical  
 321 differential-drive Limo robots, each equipped with an onboard Intel NCU i7 processor for real-time  
 322 execution. The robots run ROS2 and coordinate via Wi-Fi-based broadcasting for decentralized  
 323 communication. Experiments were conducted in both indoor and outdoor environments. As shown in

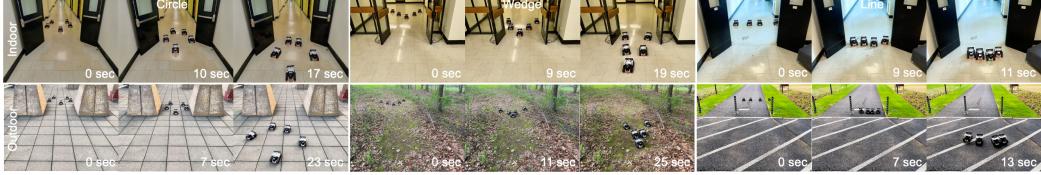


Figure 5: Qualitative results from real-world indoor and outdoor environments using varying numbers of Limo robots running ROS2 and communicating via Wi-Fi broadcasting.

324 Figure 5, our method enables teams of varying sizes to smoothly adapt formation while navigating  
 325 narrow indoor spaces. The outdoor experiments include narrow passages bordered by bollards,  
 326 scattered trees, and roadblocks. Notably, in wedge formation scenarios, ReDiG effectively guided  
 327 the team through uneven forest terrain where wheel slippage introduced movement uncertainty. The  
 328 results further demonstrate the effectiveness of our approach in enabling coordinated navigation with  
 329 formation adaptation, as well as its adaptability to unfamiliar environments.

## 330 5 Discussion

### 331 Importance of Inter-Robot Message Passing for Coor- 332 dination.

333 To understand how graph learning facilitates  
 334 team coordination, we conduct a message importance anal-  
 335 ysis. The graph network uses learnable weight  $\mathbf{W}^h$  to  
 336 compute team embeddings  $\mathbf{h}_i$  through message passing.  
 337 We quantify the importance of messages exchanged be-  
 338 tween robots by calculating the Euclidean norm of the  
 339 message vectors  $\|\mathbf{W}^h(\mathbf{z}_j - \mathbf{z}_i)\|_2$ . Figure 6 visualizes  
 340 message importance between robot pairs during training.  
 341 Early in training, robot 4 distributes attention uniformly  
 342 across teammates, lacking awareness of team structure, which eventually leads to a collision with the  
 343 wall. By the end of training, robots 3 and 4, as the outermost in the formation and key to controlling  
 344 team size, assign the highest importance to each other's messages. This verifies the effectiveness of  
 345 graph learning in enabling team coordination and formation adaptation.

### 346 Analysis of Motion Trajectory Smoothness.

347 To quantitatively evaluate the smoothness of trajectories en-  
 348 abled by the diffusion model, we use  
 349 the jerk metric [53], which measures  
 350 the rate of change of acceleration over  
 351 time. For each robot  $i$ , the jerk is de-  
 352 fined as  $j_i(t) = \frac{d^3\mathbf{p}_i(t)}{dt^3}$ . Figure 7  
 353 shows the jerk profiles over time for a team of five robots under three configurations. The first  
 354 subfigure represents ReDiG with RL only, without the diffusion or spring-damper component, the  
 355 resulting jerk is high. The second shows ReDiG with spring-damper but no diffusion; the damper  
 356 reduces jerk, but still remains due to step-wise RL control. The third includes both diffusion and  
 357 spring-damper; by iteratively refining robot actions, this setup achieves the lowest jerk. These results  
 358 highlight the effectiveness of diffusion model in producing smooth individual robot trajectories.

## 359 6 Conclusion

360 In this paper, we propose ReDiG to enable decentralized coordinated multi-robot navigation with  
 361 smooth formation adaptation. ReDiG is built upon a unified learning framework, including graph  
 362 learning for decentralized coordination to enable formation adaptation, diffusion models for gener-  
 363 ating smooth trajectories for individual robot, and RL to refine noisy demonstrations using feedback  
 364 from environment interaction, which enables robot synchronization and guides effective diffusion  
 365 training. Results from extensive experiments show that ReDiG enables smooth formation adaptation  
 366 and achieve state-of-the-art performance in coordinated multi-robot navigation.

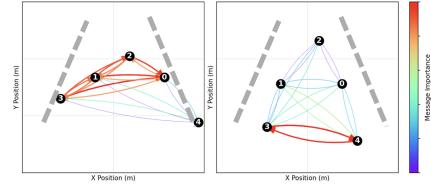


Figure 6: Comparison of inter-robot mes-  
 sage importance over training.



Figure 7: Trajectory smoothness analysis through Jerk pro-  
 files of five robots under three different settings.

367 **References**

- 368 [1] P. Gao, S. Siva, A. Micciche, and H. Zhang, “Collaborative scheduling with adaptation to failure  
369 for heterogeneous robot teams,” in *IEEE International Conference on Robotics and Automation*,  
370 2023.
- 371 [2] I. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, “Efficient boustrophedon multi-robot  
372 coverage: an algorithmic approach,” *Annals of Mathematics and Artificial Intelligence*, vol. 52,  
373 pp. 109–142, 2008.
- 374 [3] T. Balch and M. Hybinette, “Social potentials for scalable multi-robot formations,” in *IEEE  
375 International Conference on Robotics and Automation*, 2000.
- 376 [4] J. P. Queraltá, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj,  
377 J. Raitoharju, and T. Westerlund, “Collaborative multi-robot search and rescue: Planning,  
378 coordination, perception, and active vision,” *IEEE Access*, vol. 8, pp. 191 617–191 643, 2020.
- 379 [5] Q. Yang and R. Parasuraman, “Needs-driven heterogeneous multi-robot cooperation in rescue  
380 missions,” in *IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2020.
- 381 [6] J. L. Baxter, E. Burke, J. M. Garibaldi, and M. Norman, “Multi-robot search and rescue: A  
382 potential field based approach,” *Autonomous Robots and Agents*, pp. 9–16, 2007.
- 383 [7] A. Amanatiadis, C. Henschel, B. Birkicht, B. Andel, K. Charalampous, I. Kostavelis, R. May,  
384 and A. Gasteratos, “Avert: An autonomous multi-robot system for vehicle extraction and  
385 transportation,” in *IEEE International Conference on Robotics and Automation*, 2015.
- 386 [8] D. Koung, O. Kermorgant, I. Fantoni, and L. Belouaer, “Cooperative multi-robot object trans-  
387 portation system based on hierarchical quadratic programming,” *IEEE Robotics and Automation  
388 Letters*, vol. 6, no. 4, pp. 6466–6472, 2021.
- 389 [9] H. Farivarnejad, A. S. Lafmejani, and S. Berman, “Fully decentralized controller for multi-robot  
390 collective transport in space applications,” in *IEEE Aerospace Conference*, 2021.
- 391 [10] R. Han, S. Chen, and Q. Hao, “Cooperative multi-robot navigation in dynamic environment with  
392 deep reinforcement learning,” in *IEEE International Conference on Robotics and Automation*,  
393 2020.
- 394 [11] V. Indelman, “Cooperative multi-robot belief space planning for autonomous navigation in  
395 unknown environments,” *Autonomous Robots*, vol. 42, pp. 353–373, 2018.
- 396 [12] B. Martinez Rocamora Jr, C. Kilic, C. Tatsch, G. A. Pereira, and J. N. Gross, “Multi-robot  
397 cooperation for lunar in-situ resource utilization,” *Frontiers in Robotics and AI*, vol. 10, p.  
398 1149080, 2023.
- 399 [13] B. Reily, C. Reardon, and H. Zhang, “Leading multi-agent teams to multiple goals while  
400 maintaining communication,” in *Robotics Science and Systems*, 2020.
- 401 [14] T. Wu, K. Xue, and P. Wang, “Leader-follower formation control of usvs using APF-based  
402 adaptive fuzzy logic nonsingular terminal sliding mode control method,” *Journal of Mechanical  
403 Science and Technology*, vol. 36, no. 4, pp. 2007–2018, 2022.
- 404 [15] H. Xiao and C. P. Chen, “Leader-follower consensus multi-robot formation control using  
405 neurodynamic-optimization-based nonlinear model predictive control,” *IEEE Access*, vol. 7, pp.  
406 43 581–43 590, 2019.
- 407 [16] D. Roy, A. Chowdhury, M. Maitra, and S. Bhattacharya, “Multi-robot virtual structure switch-  
408 ing and formation changing strategy in an unknown occluded environment,” in *IEEE/RSJ  
409 International Conference on Intelligent Robots and Systems*, 2018.
- 410 [17] N. Abujabal, R. Fareh, S. Sinan, M. Baziyad, and M. Bettayeb, “A comprehensive review of the  
411 latest path planning developments for multi-robot formation systems,” *Robotica*, vol. 41, no. 7,  
412 pp. 2079–2104, 2023.

- 413 [18] D. Roy, A. Chowdhury, M. Maitra, and S. Bhattacharya, “Virtual region based multi-robot path  
 414 planning in an unknown occluded environment,” in *IEEE/RSJ International Conference on*  
 415 *Intelligent Robots and Systems*, 2019.
- 416 [19] J. Alonso-Mora, E. Montijano, T. Nägeli, O. Hilliges, M. Schwager, and D. Rus, “Distributed  
 417 multi-robot formation control in dynamic environments,” *Autonomous Robots*, vol. 43, pp.  
 418 1079–1100, 2019.
- 419 [20] R. Han, S. Chen, S. Wang, Z. Zhang, R. Gao, Q. Hao, and J. Pan, “Reinforcement learned  
 420 distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards,” *IEEE*  
 421 *Robotics and Automation Letters*, vol. 7, no. 3, pp. 5896–5903, 2022.
- 422 [21] N. Hacene and B. Mendil, “Behavior-based autonomous navigation and formation control  
 423 of mobile robots in unknown cluttered dynamic environments with dynamic target tracking,”  
 424 *International Journal of Automation and Computing*, vol. 18, no. 5, pp. 766–786, 2021.
- 425 [22] R. Han, S. Chen, and Q. Hao, “Cooperative multi-robot navigation in dynamic environment with  
 426 deep reinforcement learning,” in *IEEE International Conference on Robotics and Automation*,  
 427 2020.
- 428 [23] J. Blumenkamp, S. Morad, J. Gielis, Q. Li, and A. Prorok, “A framework for real-world multi-  
 429 robot systems running decentralized GNN-based policies,” in *International Conference on*  
 430 *Robotics and Automation*, 2022.
- 431 [24] Y. Hu, J. Fu, and G. Wen, “Graph soft actor–critic reinforcement learning for large-scale  
 432 distributed multirobot coordination,” *IEEE Transactions on Neural Networks and Learning*  
 433 *Systems*, 2023.
- 434 [25] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, “Motion planning diffusion: Learning  
 435 and planning of robot motions with diffusion models,” in *IEEE/RSJ International Conference*  
 436 *on Intelligent Robots and Systems*, 2023.
- 437 [26] X. Ma, S. Patidar, I. Haughton, and S. James, “Hierarchical diffusion policy for kinematics-aware  
 438 multi-task robotic manipulation,” in *Proceedings of the IEEE/CVF Conference on Computer*  
 439 *Vision and Pattern Recognition*, 2024.
- 440 [27] C. Jiang, A. Cornman, C. Park, B. Sapp, Y. Zhou, D. Anguelov, *et al.*, “Motondiffuser:  
 441 Controllable multi-agent motion prediction using diffusion,” in *Proceedings of the IEEE/CVF*  
 442 *Conference on Computer Vision and Pattern Recognition*, 2023.
- 443 [28] Y. Shaoul, I. Mishani, S. Vats, J. Li, and M. Likhachev, “Multi-robot motion planning with  
 444 diffusion models,” *arXiv*, 2024.
- 445 [29] M. Goarin and G. Loianno, “Graph neural network for decentralized multi-robot goal assign-  
 446 ment,” *IEEE Robotics and Automation Letters*, 2024.
- 447 [30] S. Zhang, K. Garg, and C. Fan, “Neural graph control barrier functions guided distributed  
 448 collision-avoidance multi-agent control,” in *Conference on Robot Learning*, 2023.
- 449 [31] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, “Graph neural networks for decentralized multi-robot  
 450 path planning,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.
- 451 [32] Z. Gao, G. Yang, and A. Prorok, “Co-optimization of environment and policies for decentralized  
 452 multi-agent navigation,” *arXiv*, 2024.
- 453 [33] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, “Motion planning diffusion: Learning  
 454 and planning of robot motions with diffusion models. in 2023 ieee,” in *RSJ International*  
 455 *Conference on Intelligent Robots and Systems*, 1916.
- 456 [34] X. Fang, C. R. Garrett, C. Eppner, T. Lozano-Pérez, L. P. Kaelbling, and D. Fox, “Dimsam:  
 457 Diffusion models as samplers for task and motion planning under partial observability,” in *2024*  
 458 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024.

- 459 [35] Z. Xian and N. Gkanatsios, “Chaineddiffuser: Unifying trajectory diffusion and keypose  
 460 prediction for robotic manipulation,” in *Conference on Robot Learning/Proceedings of Machine*  
 461 *Learning Research*, 2023.
- 462 [36] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion  
 463 policy: Visuomotor policy learning via action diffusion,” *The International Journal of Robotics*  
 464 *Research*, p. 02783649241273668, 2023.
- 465 [37] I. Kapelyukh, V. Vosylius, and E. Johns, “Dall-e-bot: Introducing web-scale diffusion models to  
 466 robotics,” *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3956–3963, 2023.
- 467 [38] W. Li, X. Wang, B. Jin, and H. Zha, “Hierarchical diffusion for offline decision making,” in  
 468 *International Conference on Machine Learning*, 2023.
- 469 [39] C. Chen, F. Deng, K. Kawaguchi, C. Gulcehre, and S. Ahn, “Simple hierarchical planning with  
 470 diffusion,” *arXiv*, 2024.
- 471 [40] J. Li, W. Abbas, M. Shabbir, and X. D. Koutsoukos, “Resilient distributed diffusion for multi-  
 472 robot systems using centerpoint.” in *Robotics: Science and Systems*, 2020.
- 473 [41] Z. Wang, J. J. Hunt, and M. Zhou, “Diffusion policies as an expressive policy class for offline  
 474 reinforcement learning,” *arXiv*, 2022.
- 475 [42] C. Lu, H. Chen, J. Chen, H. Su, C. Li, and J. Zhu, “Contrastive energy prediction for exact  
 476 energy-guided diffusion sampling in offline reinforcement learning,” in *International Conference*  
 477 *on Machine Learning*, 2023.
- 478 [43] S. E. Ada, E. Oztop, and E. Ugur, “Diffusion policies for out-of-distribution generalization in  
 479 offline reinforcement learning,” *IEEE Robotics and Automation Letters*, vol. 9, pp. 3116–3123,  
 480 2024.
- 481 [44] Y. Chen, H. Li, and D. Zhao, “Boosting continuous control with consistency policy,” *arXiv*,  
 482 2023.
- 483 [45] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, “Planning with diffusion for flexible behavior  
 484 synthesis,” *arXiv*, 2022.
- 485 [46] H. He, C. Bai, K. Xu, Z. Yang, W. Zhang, D. Wang, B. Zhao, and X. Li, “Diffusion model is  
 486 an effective planner and data synthesizer for multi-task reinforcement learning,” *Advances in*  
 487 *Neural Information Processing Systems*, vol. 36, pp. 64 896–64 917, 2023.
- 488 [47] L. Yang, Z. Huang, F. Lei, Y. Zhong, Y. Yang, C. Fang, S. Wen, B. Zhou, and Z. Lin, “Policy  
 489 representation via diffusion probability model for reinforcement learning,” *arXiv*, 2023.
- 490 [48] S. Ding, K. Hu, Z. Zhang, K. Ren, W. Zhang, J. Yu, J. Wang, and Y. Shi, “Diffusion-based  
 491 reinforcement learning via q-weighted variational policy optimization,” *arXiv*, 2024.
- 492 [49] Z. Deng, P. Gao, W. J. Jose, and H. Zhang, “Multi-robot collaborative navigation with formation  
 493 adaptation,” *arXiv*, 2024.
- 494 [50] C. Gabellieri, A. Palleschi, and L. Pallottino, “Force-based formation control of omnidirectional  
 495 ground vehicles,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*,  
 496 2021.
- 497 [51] Y. F. Chen, M. Liu, M. Everett, and J. P. How, “Decentralized non-communicating multiagent  
 498 collision avoidance with deep reinforcement learning,” in *IEEE International Conference on*  
 499 *Robotics and Automation*, 2017.
- 500 [52] Q. Zou, Q. Sun, L. Chen, B. Nie, and Q. Li, “A comparative analysis of LiDAR SLAM-based  
 501 indoor navigation for autonomous vehicles,” *IEEE Transactions on Intelligent Transportation*  
 502 *Systems*, vol. 23, no. 7, pp. 6907–6921, 2021.
- 503 [53] A. Gasparetto and V. Zanotto, “A new method for smooth trajectory planning of robot manipu-  
 504 lators,” *Mechanism and Machine Theory*, vol. 42, no. 4, pp. 455–471, 2007.
- 505 [54] D. P. Kingma, “Adam: A method for stochastic optimization,” *arXiv*, 2014.

506 **NeurIPS Paper Checklist**

507 The checklist is designed to encourage best practices for responsible machine learning research,  
508 addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove  
509 the checklist: **The papers not including the checklist will be desk rejected.** The checklist should  
510 follow the references and follow the (optional) supplemental material. The checklist does NOT count  
511 towards the page limit.

512 Please read the checklist guidelines carefully for information on how to answer these questions. For  
513 each question in the checklist:

- 514 • You should answer [Yes] , [No] , or [NA] .
- 515 • [NA] means either that the question is Not Applicable for that particular paper or the  
516 relevant information is Not Available.
- 517 • Please provide a short (1–2 sentence) justification right after your answer (even for NA).

518 **The checklist answers are an integral part of your paper submission.** They are visible to the  
519 reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it  
520 (after eventual revisions) with the final version of your paper, and its final version will be published  
521 with the paper.

522 The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation.  
523 While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a  
524 proper justification is given (e.g., "error bars are not reported because it would be too computationally  
525 expensive" or "we were unable to find the license for the dataset we used"). In general, answering  
526 "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we  
527 acknowledge that the true answer is often more nuanced, so please just use your best judgment and  
528 write a justification to elaborate. All supporting evidence can appear either in the main paper or the  
529 supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification  
530 please point to the section(s) where related material for the question can be found.

531 **IMPORTANT**, please:

- 532 • **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”**,
- 533 • **Keep the checklist subsection headings, questions/answers and guidelines below**.
- 534 • **Do not modify the questions and only use the provided macros for your answers.**

535 **1. Claims**

536 Question: Do the main claims made in the abstract and introduction accurately reflect the  
537 paper's contributions and scope?

538 Answer: [Yes]

539 Justification: The abstract and introduction accurately state the paper's contribution and  
540 scope, see [1].

541 Guidelines:

- 542 • The answer NA means that the abstract and introduction do not include the claims  
543 made in the paper.
- 544 • The abstract and/or introduction should clearly state the claims made, including the  
545 contributions made in the paper and important assumptions and limitations. A No or  
546 NA answer to this question will not be perceived well by the reviewers.
- 547 • The claims made should match theoretical and experimental results, and reflect how  
548 much the results can be expected to generalize to other settings.
- 549 • It is fine to include aspirational goals as motivation as long as it is clear that these goals  
550 are not attained by the paper.

551 **2. Limitations**

552 Question: Does the paper discuss the limitations of the work performed by the authors?

553 Answer: [No]

554 Justification: Due to the length of the paper, we do not include limitations in the main paper.

555 Guidelines:

- 556 • The answer NA means that the paper has no limitation while the answer No means that  
557 the paper has limitations, but those are not discussed in the paper.
- 558 • The authors are encouraged to create a separate "Limitations" section in their paper.
- 559 • The paper should point out any strong assumptions and how robust the results are to  
560 violations of these assumptions (e.g., independence assumptions, noiseless settings,  
561 model well-specification, asymptotic approximations only holding locally). The authors  
562 should reflect on how these assumptions might be violated in practice and what the  
563 implications would be.
- 564 • The authors should reflect on the scope of the claims made, e.g., if the approach was  
565 only tested on a few datasets or with a few runs. In general, empirical results often  
566 depend on implicit assumptions, which should be articulated.
- 567 • The authors should reflect on the factors that influence the performance of the approach.  
568 For example, a facial recognition algorithm may perform poorly when image resolution  
569 is low or images are taken in low lighting. Or a speech-to-text system might not be  
570 used reliably to provide closed captions for online lectures because it fails to handle  
571 technical jargon.
- 572 • The authors should discuss the computational efficiency of the proposed algorithms  
573 and how they scale with dataset size.
- 574 • If applicable, the authors should discuss possible limitations of their approach to  
575 address problems of privacy and fairness.
- 576 • While the authors might fear that complete honesty about limitations might be used by  
577 reviewers as grounds for rejection, a worse outcome might be that reviewers discover  
578 limitations that aren't acknowledged in the paper. The authors should use their best  
579 judgment and recognize that individual actions in favor of transparency play an impor-  
580 tant role in developing norms that preserve the integrity of the community. Reviewers  
581 will be specifically instructed to not penalize honesty concerning limitations.

### 582 3. Theory assumptions and proofs

583 Question: For each theoretical result, does the paper provide the full set of assumptions and  
584 a complete (and correct) proof?

585 Answer: [Yes]

586 Justification: The theoretical proof can be found in Appendix A.1.

587 Guidelines:

- 588 • The answer NA means that the paper does not include theoretical results.
- 589 • All the theorems, formulas, and proofs in the paper should be numbered and cross-  
590 referenced.
- 591 • All assumptions should be clearly stated or referenced in the statement of any theorems.
- 592 • The proofs can either appear in the main paper or the supplemental material, but if  
593 they appear in the supplemental material, the authors are encouraged to provide a short  
594 proof sketch to provide intuition.
- 595 • Inversely, any informal proof provided in the core of the paper should be complemented  
596 by formal proofs provided in appendix or supplemental material.
- 597 • Theorems and Lemmas that the proof relies upon should be properly referenced.

### 598 4. Experimental result reproducibility

599 Question: Does the paper fully disclose all the information needed to reproduce the main ex-  
600 perimental results of the paper to the extent that it affects the main claims and/or conclusions  
601 of the paper (regardless of whether the code and data are provided or not)?

602 Answer: [Yes]

603 Justification: The information needed to reproduce the main experimental results can be  
604 found in Appendix A.3. The main paper also comprehensively provides the training and  
605 execution process in Section 3.1.4.

606 Guidelines:

- 607           • The answer NA means that the paper does not include experiments.
- 608           • If the paper includes experiments, a No answer to this question will not be perceived
- 609           well by the reviewers: Making the paper reproducible is important, regardless of
- 610           whether the code and data are provided or not.
- 611           • If the contribution is a dataset and/or model, the authors should describe the steps taken
- 612           to make their results reproducible or verifiable.
- 613           • Depending on the contribution, reproducibility can be accomplished in various ways.
- 614           For example, if the contribution is a novel architecture, describing the architecture fully
- 615           might suffice, or if the contribution is a specific model and empirical evaluation, it may
- 616           be necessary to either make it possible for others to replicate the model with the same
- 617           dataset, or provide access to the model. In general, releasing code and data is often
- 618           one good way to accomplish this, but reproducibility can also be provided via detailed
- 619           instructions for how to replicate the results, access to a hosted model (e.g., in the case
- 620           of a large language model), releasing of a model checkpoint, or other means that are
- 621           appropriate to the research performed.
- 622           • While NeurIPS does not require releasing code, the conference does require all submissions
- 623           to provide some reasonable avenue for reproducibility, which may depend on the
- 624           nature of the contribution. For example
- 625           (a) If the contribution is primarily a new algorithm, the paper should make it clear how
- 626           to reproduce that algorithm.
- 627           (b) If the contribution is primarily a new model architecture, the paper should describe
- 628           the architecture clearly and fully.
- 629           (c) If the contribution is a new model (e.g., a large language model), then there should
- 630           either be a way to access this model for reproducing the results or a way to reproduce
- 631           the model (e.g., with an open-source dataset or instructions for how to construct
- 632           the dataset).
- 633           (d) We recognize that reproducibility may be tricky in some cases, in which case
- 634           authors are welcome to describe the particular way they provide for reproducibility.
- 635           In the case of closed-source models, it may be that access to the model is limited in
- 636           some way (e.g., to registered users), but it should be possible for other researchers
- 637           to have some path to reproducing or verifying the results.

## 638        5. Open access to data and code

639        Question: Does the paper provide open access to the data and code, with sufficient instruc-

640        tions to faithfully reproduce the main experimental results, as described in supplemental

641        material?

642        Answer: [No]

643        Justification: We will release our code after paper acceptance.

644        Guidelines:

- 645           • The answer NA means that paper does not include experiments requiring code.
- 646           • Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- 647           • While we encourage the release of code and data, we understand that this might not be
- 648           possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not
- 649           including code, unless this is central to the contribution (e.g., for a new open-source
- 650           benchmark).
- 651           • The instructions should contain the exact command and environment needed to run to
- 652           reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- 653           • The authors should provide instructions on data access and preparation, including how
- 654           to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- 655           • The authors should provide scripts to reproduce all experimental results for the new
- 656           proposed method and baselines. If only a subset of experiments are reproducible, they
- 657           should state which ones are omitted from the script and why.
- 658           • At submission time, to preserve anonymity, the authors should release anonymized
- 659           versions (if applicable).

- 662           • Providing as much information as possible in supplemental material (appended to the  
663           paper) is recommended, but including URLs to data and code is permitted.

664           **6. Experimental setting/details**

665           Question: Does the paper specify all the training and test details (e.g., data splits, hyper-  
666           parameters, how they were chosen, type of optimizer, etc.) necessary to understand the  
667           results?

668           Answer: [Yes]

669           Justification: We provide sufficient training and execution detail in sec 3.1.4, and the  
670           approach implementation and training are further provided in Appendix A.3.

671           Guidelines:

- 672           • The answer NA means that the paper does not include experiments.  
673           • The experimental setting should be presented in the core of the paper to a level of detail  
674           that is necessary to appreciate the results and make sense of them.  
675           • The full details can be provided either with the code, in appendix, or as supplemental  
676           material.

677           **7. Experiment statistical significance**

678           Question: Does the paper report error bars suitably and correctly defined or other appropriate  
679           information about the statistical significance of the experiments?

680           Answer: [Yes]

681           Justification: The quantitative results can be found in Table 1. The qualitative results are  
682           shown in Figure 2-5.

683           Guidelines:

- 684           • The answer NA means that the paper does not include experiments.  
685           • The authors should answer "Yes" if the results are accompanied by error bars, confi-  
686           dence intervals, or statistical significance tests, at least for the experiments that support  
687           the main claims of the paper.  
688           • The factors of variability that the error bars are capturing should be clearly stated (for  
689           example, train/test split, initialization, random drawing of some parameter, or overall  
690           run with given experimental conditions).  
691           • The method for calculating the error bars should be explained (closed form formula,  
692           call to a library function, bootstrap, etc.).  
693           • The assumptions made should be given (e.g., Normally distributed errors).  
694           • It should be clear whether the error bar is the standard deviation or the standard error  
695           of the mean.  
696           • It is OK to report 1-sigma error bars, but one should state it. The authors should  
697           preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis  
698           of Normality of errors is not verified.  
699           • For asymmetric distributions, the authors should be careful not to show in tables or  
700           figures symmetric error bars that would yield results that are out of range (e.g. negative  
701           error rates).  
702           • If error bars are reported in tables or plots, The authors should explain in the text how  
703           they were calculated and reference the corresponding figures or tables in the text.

704           **8. Experiments compute resources**

705           Question: For each experiment, does the paper provide sufficient information on the com-  
706           puter resources (type of compute workers, memory, time of execution) needed to reproduce  
707           the experiments?

708           Answer: [Yes]

709           Justification: Training computational resources is clarified in Appendix A.3.

710           Guidelines:

- 711           • The answer NA means that the paper does not include experiments.

- 712           • The paper should indicate the type of compute workers CPU or GPU, internal cluster,  
 713            or cloud provider, including relevant memory and storage.  
 714           • The paper should provide the amount of compute required for each of the individual  
 715            experimental runs as well as estimate the total compute.  
 716           • The paper should disclose whether the full research project required more compute  
 717            than the experiments reported in the paper (e.g., preliminary or failed experiments that  
 718            didn't make it into the paper).

719           **9. Code of ethics**

720           Question: Does the research conducted in the paper conform, in every respect, with the  
 721           NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

722           Answer: [Yes]

723           Justification: This research fully adheres to the NeurIPS Code of Ethics.

724           Guidelines:

- 725           • The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.  
 726           • If the authors answer No, they should explain the special circumstances that require a  
 727            deviation from the Code of Ethics.  
 728           • The authors should make sure to preserve anonymity (e.g., if there is a special consid-  
 729            eration due to laws or regulations in their jurisdiction).

730           **10. Broader impacts**

731           Question: Does the paper discuss both potential positive societal impacts and negative  
 732           societal impacts of the work performed?

733           Answer: [Yes]

734           Justification: The paper discusses positive societal impacts in the abstract, introduction  
 735           [1], and related work [2], which include robot navigation applications and accessibility of  
 736           automation technology.

737           Guidelines:

- 738           • The answer NA means that there is no societal impact of the work performed.  
 739           • If the authors answer NA or No, they should explain why their work has no societal  
 740            impact or why the paper does not address societal impact.  
 741           • Examples of negative societal impacts include potential malicious or unintended uses  
 742            (e.g., disinformation, generating fake profiles, surveillance), fairness considerations  
 743            (e.g., deployment of technologies that could make decisions that unfairly impact specific  
 744            groups), privacy considerations, and security considerations.  
 745           • The conference expects that many papers will be foundational research and not tied  
 746            to particular applications, let alone deployments. However, if there is a direct path to  
 747            any negative applications, the authors should point it out. For example, it is legitimate  
 748            to point out that an improvement in the quality of generative models could be used to  
 749            generate deepfakes for disinformation. On the other hand, it is not needed to point out  
 750            that a generic algorithm for optimizing neural networks could enable people to train  
 751            models that generate Deepfakes faster.  
 752           • The authors should consider possible harms that could arise when the technology is  
 753            being used as intended and functioning correctly, harms that could arise when the  
 754            technology is being used as intended but gives incorrect results, and harms following  
 755            from (intentional or unintentional) misuse of the technology.  
 756           • If there are negative societal impacts, the authors could also discuss possible mitigation  
 757            strategies (e.g., gated release of models, providing defenses in addition to attacks,  
 758            mechanisms for monitoring misuse, mechanisms to monitor how a system learns from  
 759            feedback over time, improving the efficiency and accessibility of ML).

760           **11. Safeguards**

761           Question: Does the paper describe safeguards that have been put in place for responsible  
 762           release of data or models that have a high risk for misuse (e.g., pretrained language models,  
 763           image generators, or scraped datasets)?

764 Answer: [NA]

765 Justification: This paper posses no such risks.

766 Guidelines:

- 767 • The answer NA means that the paper poses no such risks.
- 768 • Released models that have a high risk for misuse or dual-use should be released with  
769 necessary safeguards to allow for controlled use of the model, for example by requiring  
770 that users adhere to usage guidelines or restrictions to access the model or implementing  
771 safety filters.
- 772 • Datasets that have been scraped from the Internet could pose safety risks. The authors  
773 should describe how they avoided releasing unsafe images.
- 774 • We recognize that providing effective safeguards is challenging, and many papers do  
775 not require this, but we encourage authors to take this into account and make a best  
776 faith effort.

## 777 12. Licenses for existing assets

778 Question: Are the creators or original owners of assets (e.g., code, data, models), used in  
779 the paper, properly credited and are the license and terms of use explicitly mentioned and  
780 properly respected?

781 Answer: [NA]

782 Justification: The paper does not use existing assets.

783 Guidelines:

- 784 • The answer NA means that the paper does not use existing assets.
- 785 • The authors should cite the original paper that produced the code package or dataset.
- 786 • The authors should state which version of the asset is used and, if possible, include a  
787 URL.
- 788 • The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- 789 • For scraped data from a particular source (e.g., website), the copyright and terms of  
790 service of that source should be provided.
- 791 • If assets are released, the license, copyright information, and terms of use in the package  
792 should be provided. For popular datasets, [paperswithcode.com/datasets](http://paperswithcode.com/datasets) has  
793 curated licenses for some datasets. Their licensing guide can help determine the license  
794 of a dataset.
- 795 • For existing datasets that are re-packaged, both the original license and the license of  
796 the derived asset (if it has changed) should be provided.
- 797 • If this information is not available online, the authors are encouraged to reach out to  
798 the asset's creators.

## 799 13. New assets

800 Question: Are new assets introduced in the paper well documented and is the documentation  
801 provided alongside the assets?

802 Answer: [NA]

803 Justification: The paper does not release new assets.

804 Guidelines:

- 805 • The answer NA means that the paper does not release new assets.
- 806 • Researchers should communicate the details of the dataset/code/model as part of their  
807 submissions via structured templates. This includes details about training, license,  
808 limitations, etc.
- 809 • The paper should discuss whether and how consent was obtained from people whose  
810 asset is used.
- 811 • At submission time, remember to anonymize your assets (if applicable). You can either  
812 create an anonymized URL or include an anonymized zip file.

## 813 14. Crowdsourcing and research with human subjects

814 Question: For crowdsourcing experiments and research with human subjects, does the paper  
815 include the full text of instructions given to participants and screenshots, if applicable, as  
816 well as details about compensation (if any)?

817 Answer: [NA]

818 Justification: The paper does not involve crowdsourcing nor research with human subjects.

819 Guidelines:

- 820 • The answer NA means that the paper does not involve crowdsourcing nor research with  
821 human subjects.  
822 • Including this information in the supplemental material is fine, but if the main contribu-  
823 tion of the paper involves human subjects, then as much detail as possible should be  
824 included in the main paper.  
825 • According to the NeurIPS Code of Ethics, workers involved in data collection, curation,  
826 or other labor should be paid at least the minimum wage in the country of the data  
827 collector.

## 828 15. Institutional review board (IRB) approvals or equivalent for research with human 829 subjects

830 Question: Does the paper describe potential risks incurred by study participants, whether  
831 such risks were disclosed to the subjects, and whether Institutional Review Board (IRB)  
832 approvals (or an equivalent approval/review based on the requirements of your country or  
833 institution) were obtained?

834 Answer: [NA]

835 Justification: The paper does not involve crowdsourcing nor research with human subjects.

836 Guidelines:

- 837 • The answer NA means that the paper does not involve crowdsourcing nor research with  
838 human subjects.  
839 • Depending on the country in which research is conducted, IRB approval (or equivalent)  
840 may be required for any human subjects research. If you obtained IRB approval, you  
841 should clearly state this in the paper.  
842 • We recognize that the procedures for this may vary significantly between institutions  
843 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the  
844 guidelines for their institution.  
845 • For initial submissions, do not include any information that would break anonymity (if  
846 applicable), such as the institution conducting the review.

## 847 16. Declaration of LLM usage

848 Question: Does the paper describe the usage of LLMs if it is an important, original, or  
849 non-standard component of the core methods in this research? Note that if the LLM is used  
850 only for writing, editing, or formatting purposes and does not impact the core methodology,  
851 scientific rigorousness, or originality of the research, declaration is not required.

852 Answer: [NA]

853 Justification: The core method development in this research does not involve LLMs as any  
854 important, original, or non-standard components.

855 Guidelines:

- 856 • The answer NA means that the core method development in this research does not  
857 involve LLMs as any important, original, or non-standard components.  
858 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.