Decentralized Multi-Goal Multi-Agent Pathfinding with Spatial Prior and Neighbor Intent Prediction

Xuening Wang[1] **Bin Wang**[1*] **Sihui Li**[1] **Xiaochun Yang**[1] **Yucen Gao**[2*]
[1]School of Computer Science and Engineering, Northeastern University
[2]School of Software, Northeastern University
Shenyang, China

Abstract

As a variant of the multi-agent path finding (MAPF) problem, multi-goal multi-agent path finding (MG-MAPF) requires planning conflict-free paths for a team of agents to visit sequences of pre-assigned goal vertices. Existing search-based methods suffer from poor scalability due to their reliance on centralized control, while decentralized learning-based approaches struggle with long-horizon planning under partial observability. Furthermore, existing paradigms rarely address joint goal ordering across agents, where uncoordinated goal sequences cause multiple agents to simultaneously converge on the same region, increasing congestion and conflicts. To address these challenges, we propose the Intent-aware multi-goal multi-agent path finding (IMMP), a hierarchical framework for decentralized MG-MAPF that explicitly models spatial congestion at the high level and neighbor intent at the execution level. The high-level planner encodes a spatial prior via spatially-balanced goal ordering and path planning, reducing conflict pressure passed to the low-level policy. The low-level module resolves real-time conflicts via a reinforcement learning policy augmented with a neighbor intent prediction module, trained end-to-end through a prediction-guided collaborative proximal policy optimization. Extensive experiments demonstrate that IMMP achieves strong scalability and generalization, reducing sum of costs (SoC) by an average of 19.77% compared to the best-performing decentralized baseline, while maintaining robust performance. Our code is available at `https://github.com/MayWang1007/MG-MAPF`.

1 Introduction

The multi-agent path finding (MAPF) problem requires planning conflict-free paths for a team of agents from their start positions to individual goal locations [1]. In many real-world scenarios such as warehouses and logistics systems [2, 3], each agent must visit all assigned goals in sequence rather than one-shot pathfinding, giving rise to the multi-goal multi-agent path finding (MG-MAPF) problem [4]. Solving MG-MAPF optimally is NP-hard, as it subsumes both the classical MAPF problem [5] and the Path Traveling Salesman Problem (TSP) with a fixed source for each agent [6, 7].

Existing approaches for MG-MAPF are search-based algorithms. Surynek [4] first formalized the problem and proposed Hamiltonian Conflict-Based Search (HCBS) and Satisfiability Modulo Theories-Hamiltonian Conflict-Based Search (SMT-HCBS), two solvers that decouple goal vertex ordering from single-agent pathfinding. Tang et al. [8] showed that this decoupling strategy cannot always achieve optimal solutions. To address this, they proposed Multi-Goal Conflict-Based Search (MGCBS), which ensures optimality and completeness by searching over the visiting orders of goal safe intervals (GSI) rather than discrete vertices. Despite these theoretical guarantees, these

*Corresponding authors: Bin Wang and Yucen Gao.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

search-based approaches require a centralized controller, with success rates dropping sharply as the number of agents increases under a limited time budget [8], limiting scalability (**Challenge 1**).

Decentralized learning-based methods, particularly those leveraging reinforcement learning (RL) and imitation learning (IL), have demonstrated significant scalability in solving the MAPF problem and its variants [9, 10, 11, 12, 13, 14]. Yet these methods operate under partial observability, where each agent observes only a local field of view (FOV) and has no access to the historical trajectories or future actions of neighboring agents, limiting its long-horizon planning capability (**Challenge 2**) [11]. Extending decentralized learning-based approaches to MG-MAPF introduces an additional challenge, as poorly ordered goal sequences can cause multiple agents to converge simultaneously on the same region, increasing the likelihood of conflicts [8]. Since the learning model must jointly address goal ordering and conflict-free pathfinding under partial observability constraints, this combinatorial problem has yet to be thoroughly investigated in the existing literature (**Challenge 3**).

To address these challenges, we propose the Intent-aware multi-goal multi-agent path finding (IMMP), a hierarchical framework for decentralized MG-MAPF, consisting of two levels: a high-level planner that addresses goal ordering and pathfinding, and a low-level RL module that handles real-time conflicts. Our contributions can be summarized as follows.

- To encode a spatial prior for the low-level policy, we introduce a spatially-balanced goal ordering module that jointly minimizes the spatial path overlap across agents.
- To enable proactive collision avoidance, we incorporate a neighbor intent prediction module that allows each agent to anticipate the movements of neighboring agents.
- To train the prediction module and policy network end-to-end, we adopt a Straight-Through Estimator (STE) with differentiable Gaussian rasterization, and propose a prediction-guided collaborative Proximal Policy Optimization (PPO) objective.
- We conduct comprehensive experiments to evaluate the performance of IMMP in varied scenarios. Results show that IMMP reduces sum of costs (SoC) by an average of 19.77% compared to the best-performing decentralized baseline, demonstrating strong performance in path efficiency.

2 Related Work

Congestion-aware Goal Ordering. To solve the goal vertex ordering problem in MG-MAPF, HCBS [4] formulates it as a Hamiltonian path problem for each agent, using A* to search over goal vertices. MGCBS [8] replaces this with GSI ordering, searching over which interval to visit at each goal vertex rather than the vertex alone, thereby avoiding suboptimal solutions. However, existing methods optimize goal vertex ordering solely in terms of individual path cost, causing paths to accumulate in the same region—a spatial distribution problem inherent to shortest path planning [15, 16]. Congestion-aware path planning methods [17, 18] have demonstrated that spatially dispersed paths can improve system throughput, but their problem formulations differ from MG-MAPF. To this end, we aim to incorporate a congestion-aware goal vertex ordering for MG-MAPF, generating spatially-balanced goal visiting orders that establish a spatial prior for the low-level RL policy.

Neighbor Intent Prediction under Partial Observability. A prominent line of decentralized MAPF research formulates the problem as a partially observable Markov decision process (POMDP) [19, 20] and solves it using learning-based methods. PRIMAL [21] showed that fully decentralized policies can be trained via RL and IL, where each agent observes a local FOV containing obstacle and neighbor positions. However, relying solely on a limited FOV often makes agents short-sighted, as agents lack the ability to perceive and plan for obstacles and agents beyond their local observations [11]. PRIMAL2 [13] and ALPHA [11] attempt to estimate neighbors' future actions using the A* algorithm, which is not learnable and incompatible with dynamic environments. Unlike these studies, we introduce a learnable module that predicts the action of each neighbor using its position and goal.

End-to-End Policy Training with PPO. PPO has been acknowledged to be effective in decentralized MAPF, enabling each agent to learn a local policy [22, 23, 24], and our method follows this line of research. However, integrating a neighbor intent prediction module introduces a discrete bottleneck that prevents end-to-end training with PPO. Unlike prior work that relies on separate supervised objectives for prediction modules with an RL policy [25], we adopt the STE [26] to jointly optimize the prediction module and policy network via the proposed Prediction-Guided Collaborative PPO, without any auxiliary loss. As summarized in Table 1, IMMP is the only method

Table 1: Comparison of MAPF methods. ✓ supported; ✗ not supported. **Learnable nbr pred.** refers to learnable neighbor intent prediction, and **E2E Training** refers to end-to-end training.

Method	Congestion-aware	Learnable nbr pred.	E2E Training	MG-MAPF
HCBS/MGCBS	✗	✗	✗	✓
PRIMAL	✗	✗	✓	✗
PRIMAL2/ALPHA	✗	✗	✓	✗
FOLLOWER	✓	✗	✓	✗
IMMP	✓	✓	✓	✓

that simultaneously supports congestion-aware goal ordering, learnable neighbor prediction, and end-to-end training, while also addressing the MG-MAPF problem.

3 Problem Formulation

The multi-goal multi-agent path finding (MG-MAPF) problem can be formulated as a five-element tuple $\langle G, \mathcal{A}, \mathcal{T}, \mathcal{C}, \Pi \rangle$, comprising the environment, agents, task list, collision constraints, and the joint path for all the agents.

We consider a set of k agents, $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$, as shown in Figure 1. At any time step t, the state of agent a_i is denoted by its location $v_i^{(t)} \in V$. Consider an undirected graph $G = (V, E)$, where V represents the set of vertices and E represents the connectivity between them. We assume time to be discretized into uniform steps. At $t = 0$, each agent starts at $v_i^{(0)} \in V$. At each time step, an agent can perform an action from the set $\mathcal{A}ct = \{Move(u, v) \mid (u, v) \in E\} \cup \{Wait(v)\}$, transitioning from the current vertex u to an adjacent vertex v or remaining at u.

The system operates under decentralized constraints with limited observability and no explicit communication, which means agents cannot exchange messages with other agents. Each agent a_i possesses global knowledge of the static map G and the task list $\mathcal{T}$. Then at time step t, a_i receives a local observation $o_i^{(t)}$, which is limited to a $(2r + 1) \times (2r + 1)$ region centered at the agent's current position.

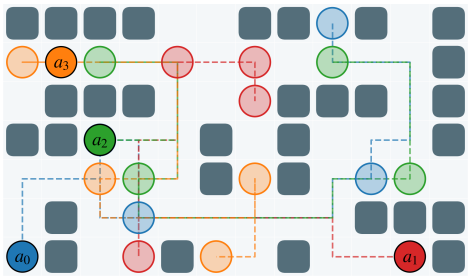


Figure 1: An example of a decentralized MG-MAPF environment we consider, where each agent is tasked with visiting multiple goals.

We formulate the MG-MAPF task as a sequential objective fulfillment problem. Specifically, the task for $\mathcal{A}$ is defined as a set of sub-tasks $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_k\}$. Notably, the task allocation mechanism is considered independent of the pathfinding system, with the goals in $\mathcal{T}$ pre-assigned. Let $\tau_i = \{g_i^1, g_i^2, \ldots, g_i^l\} \subseteq V$ denote the task assigned to agent a_i, where l is the number of goals and each g_i^k is a vertex that the agent must visit.

The solution to MG-MAPF is to find a set of conflict-free paths Π for the agents in $\mathcal{A}$ that satisfy the following collision constraints for all agents $a_i, a_j \in \mathcal{A}$ $(i \neq j)$ at all times t:

- **Vertex Conflict:** Agents cannot occupy the same location at the same time.

$$v_i^{(t)} \neq v_j^{(t)} \tag{1}$$

- **Edge Conflict:** Agents cannot traverse the same edge in opposite directions simultaneously.

$$v_i^{(t)} = v_j^{(t+1)} \wedge v_j^{(t)} = v_i^{(t+1)} \implies \text{False} \tag{2}$$

The primary objective of MG-MAPF is to minimize the SoC, which means minimizing the cost of the joint path $\Pi = \{\pi_{a_i} \mid a_i \in \mathcal{A}\}$.

$$SoC = \sum_{a_i \in \mathcal{A}} Cost(\pi_{a_i}) \tag{3}$$

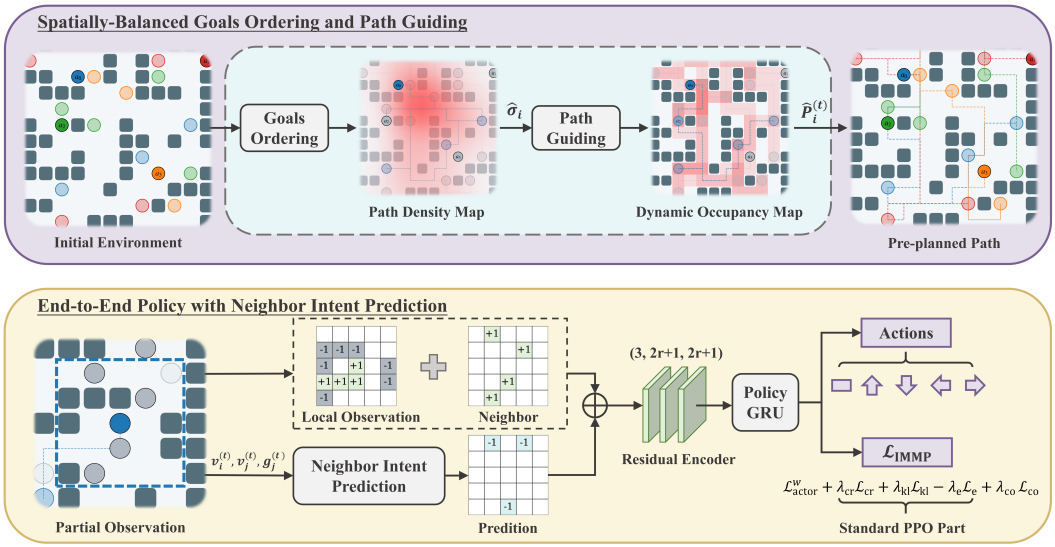


Figure 2: The hierarchical architecture of IMMP, consisting of a high-level planner that injects a spatial prior into guiding paths, and a low-level end-to-end RL policy for real-time conflict resolution.

4 Method

As shown in Figure 2, we propose a hierarchical framework for decentralized MG-MAPF. The high-level planner encodes a spatial prior into a guiding path. The low-level module is a decentralized reinforcement learning policy that utilizes the guiding path and avoids conflicts.

4.1 Spatially-Balanced Goal Ordering and Path Planning

Rather than solving the path TSP independently for each agent, the high-level planner generates spatially-balanced guiding paths $\{\hat{P}_i^{(t)}\}$ that represent a global spatial prior as an inductive bias for the low-level RL policy.

4.1.1 Global Spatially-Balanced Goal Ordering

We therefore maintain a path density map $\mathbf{H} \in \mathbb{R}^{|V|}$ to record the spatial density of pre-planned paths for all agents. Let $d(g_i^m, g_i^n)$ denote the shortest-path distance between $g_i^m, g_i^n \in V$, computed by Breadth-First Search (BFS) on G. We denote by P_i the pre-planned path of agent a_i, generated without taking other agents into account. The map $\mathbf{H}$ is initialized to zero and updated as each agent a_i determines its visiting order, with each vertex $v \in V$ updated as

$$\mathbf{H}(v) \leftarrow \mathbf{H}(v) + \mathbb{1}[v \in P_i]. \tag{4}$$

We define the spatial path density $C_{\text{density}} = \sum_{v \in V} \mathbf{H}(v)^2$ and formulate the joint optimization objective as

$$\min_{\{\hat{\sigma}_i\}} \sum_{i=1}^{k} |P_i| + \mu \cdot C_{\text{density}}, \tag{5}$$

where $\mu > 0$ is the congestion penalty weight and the quadratic structure is justified in Lemma 1, detailed in Appendix A.1. As agents with longer pre-planned paths are likely to occupy more vertices, we assign them higher priority in deciding visiting orders. For agent a_i, the edge cost between two consecutive goals is defined as

$$\text{Cost}(g_i^m, g_i^n) = d(g_i^m, g_i^n) + \mu \sum_{v \in \rho(g_i^m, g_i^n)} \big(2\mathbf{H}(v) + 1\big), \tag{6}$$

where the second term is the exact marginal increment of C_{density} induced by sub-path $\rho(g_i^m, g_i^n)$. The visiting order $\hat{\sigma}_i$ is obtained by minimizing the sequence cost

$$\mathcal{L}_{\text{seq}}(\sigma_i) = \sum_{m=1}^{l-1} \text{Cost}\left(\sigma_i^m,\, \sigma_i^{m+1}\right), \tag{7}$$

initialized via greedy search from τ_i and refined through local search over permutations. By Theorem 1, this process is guaranteed to terminate in a finite number of iterations, yielding a visiting order $\hat{\sigma}_i$ for each agent, which is then passed to the local path-guiding module.

Theorem 1 (Finite Convergence)**.** *The iterative minimization of $\mathcal{L}_{\text{seq}}(\sigma_i)$ over visiting orders $\{\sigma_i\}$ terminates in a finite number of iterations.*

Proof. See Appendix A.2. □

4.1.2 Spatial-Aware Local Path Guiding

While the pre-planned paths are spatially dispersed, real-time actions of agents introduce dynamic congestion that cannot be anticipated. Inspired by the static and dynamic costs proposed in FOLLOWER [12], we define the A* cost function as

$$f(v) = g(v) + h(v) + w_s \cdot \Phi(v) + w_d \cdot \mathbf{D}^{(t)}(v), \tag{8}$$

where $g(v)$ is the accumulated cost from the sub-path start to v, $h(v)$ is a heuristic based on Manhattan distance to the current sub-goal, and w_s, w_d are the static and dynamic weights respectively. Unlike [12], the static term $\Phi(v)$ penalizes vertices proximate to obstacles rather than topological bottlenecks, computed as

$$\Phi(v) = \alpha_s \cdot \exp\left(-\frac{\min_{v_o \in \mathcal{O}} d(v, v_o)}{2}\right), \quad \min_{v_o \in \mathcal{O}} d(v, v_o) \le 3, \tag{9}$$

where $\mathcal{O}$ denotes the set of obstacle vertices and α_s is the maximum static penalty. The dynamic occupancy map $\mathbf{D}^{(t)}$ counts the number of agents currently occupying vertex v. For vertex v, we have

$$\mathbf{D}^{(t)}(v) = \sum_{i=1}^{k} \mathbb{1}\left[v_i^{(t)} = v\right], \quad \forall v \in V. \tag{10}$$

The dynamic occupancy map $\mathbf{D}^{(t)}$ is recomputed at every step, ensuring that the guiding path $\hat{P}_i^{(t)}$ always reflects the most recent agent distribution.

4.2 End-to-End Intent-Aware Policy Learning

Under partial observability, agents cannot access neighbors' next-step actions, leading to potential collisions and suboptimal coordination. We introduce a learnable neighbor intent prediction module, which predicts each neighbor's most likely next-step action.

Observations. At each step t, agent a_i receives a local observation $o_i^{(t)}$ that consists of a local observation map encoding static obstacles and the guiding path $\hat{P}_i^{(t)}$, a neighbor map indicating the positions $\left\{v_j^{(t)}\right\}_{j=1,\ldots,|\mathcal{N}_i|}$ of neighbors within FOV, where $\mathcal{N}_i$ denotes the set of observable neighbors.

Neighbor Intent Prediction. To capture goal-directed intent among neighbors rather than treating each neighbor independently, we encode each neighbor a_j into a position embedding $\mathbf{q}_j$ and a goal embedding $\mathbf{u}_j$, computed by applying learned projections φ_{pos} and φ_{goal} to the normalized relative position $(v_j^{(t)} - v_i^{(t)})/S$ and relative goal $(g_j^{(t)} - v_i^{(t)})/S$,

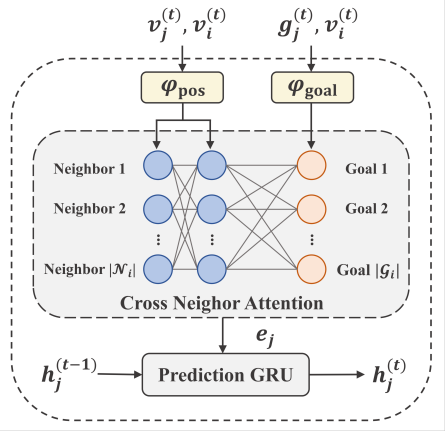


Figure 3: Architecture of the proposed Neighbor Intent Prediction module.

respectively, where S is the map size and $g_j^{(t)} \in V$ is the current sub-goal of a_j. As shown in Figure 3, to further exploit the interdependencies among neighbor intents, a cross-neighbor attention mechanism takes $\mathbf{q}_j$ as the query, $\{\mathbf{q}_{j'}\}_{j' \in \mathcal{N}_i}$ as keys, and $\{\mathbf{u}_{j'}\}_{j' \in \mathcal{N}_i}$ as values, yielding an attention output $\mathbf{e}_j$ that encodes the spatial relationships among neighbors within FOV. $\mathbf{e}_j$ is then passed through a prediction Gated Recurrent Unit (GRU) to jointly model the spatio-temporal dynamics of neighbor behavior,

$$\mathbf{h}_j^{(t)} = \mathrm{GRU}_{\mathrm{pred}}\Big(\mathbf{e}_j,\, \mathbf{h}_j^{(t-1)}\Big), \tag{11}$$

where $\mathbf{h}_j^{(t)} \in \mathbb{R}^{d_h}$ encodes the action history of neighbor a_j up to step t, preserving each neighbor's historical state to obtain more accurate predictions. An action head then maps $\mathbf{h}_j^{(t)}$ to a probability distribution $\mathbf{P}_j^{(t)}$ over the five-action space $\mathcal{A}ct$.

Differentiable Spatial Rendering with STE. To address the **non-differentiability bottleneck** introduced by the discrete action space in the prediction module, we apply the Straight-Through Estimator (STE) during training. The predicted position of neighbor a_j is computed as a soft expected coordinate weighted by the 5-action probability distribution

$$(\hat{x}_j, \hat{y}_j) = (x_j, y_j) + \sum_{a=0}^{4} \mathbf{P}_{j,a}^{(t)} \cdot \delta_a, \tag{12}$$

where (x_j, y_j) is the current coordinate of neighbor a_j, and $\delta_a \in \mathcal{A}ct$ denotes the predicted action vector. The predicted position is then rendered into a differentiable spatial occupancy map via Gaussian rasterization,

$$M_i^{(t+1)}(x, y) = \exp\left(-\frac{(x - \hat{x}_j)^2 + (y - \hat{y}_j)^2}{2\sigma_g^2}\right), \tag{13}$$

where σ_g is the Gaussian kernel width that controls the spatial spread of prediction uncertainty. The end-to-end differentiability will be established in Theorem 2. During inference, STE is replaced by hard argmax, and the occupancy map is rendered as a discrete indicator at the predicted position.

End-to-end Reinforcement Learning. The low-level module of IMMP adopts an actor-critic architecture consisting of a neighbor intent prediction module and a policy learning module. At each step t, the prediction map $M_i^{(t+1)}$ is jointly encoded with the local observation map and the neighbor map by the residual encoder,

$$\mathbf{z}_i^{(t)} = f_{\mathrm{enc}}\Big(o_i^{(t)} \oplus M_i^{(t+1)}\Big), \tag{14}$$

where $\oplus$ denotes channel-wise concatenation. The representation is then passed through a policy GRU,

$$\mathbf{s}_i^{(t)} = \mathrm{GRU}_{\mathrm{pol}}\Big(\mathbf{z}_i^{(t)},\, \mathbf{s}_i^{(t-1)}\Big), \tag{15}$$

where $\mathbf{s}_i^{(t)} \in \mathbb{R}^{d_{\mathrm{pol}}}$ is the policy GRU hidden state, shared by the actor and critic heads. During training, the actor head outputs the action distribution $\pi_\theta(a_i^{(t)} \mid o_i^{(t)})$, while the critic head produces a scalar value estimate $V_\phi(o_i^{(t)})$ trained against the reward signal detailed in Appendix B.1.

4.3 Prediction-Guided Collaborative PPO

Since the confidence of the prediction module varies across neighbors and time steps, treating all predictions equally risks allowing unreliable estimates to degrade the policy. To encourage the policy network to make more effective use of neighbor predictions, we weight the standard clipped PPO surrogate objective $\mathcal{L}_{\mathrm{actor}}$ using the average confidence $\bar{w}^{(t)}$, which is calculated by

$$\mathcal{L}_{\mathrm{actor}}^{w} = \mathcal{L}_{\mathrm{actor}} \cdot \mathrm{clip}\Big(1 + \alpha_c \bar{w}^{(t)},\ 1,\ 1 + \alpha_c\Big), \tag{16}$$

where α_c is a confidence weight. The average prediction confidence is

$$\bar{w}^{(t)} = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} w_j^{(t)}, \quad w_j^{(t)} = 1 - \frac{H\Big(\mathbf{P}_j^{(t)}\Big)}{\log |\mathcal{A}ct|}, \tag{17}$$

where $H(\cdot)$ denotes the Shannon entropy and a higher $w_j^{(t)}$ indicates greater prediction confidence. This bounded modulation focuses optimization on high-confidence predictions, while safely reverting to standard PPO when predictions are unreliable, as formally stated in Proposition 1 in Appendix A.4.

Furthermore, we introduce a collaborative penalty that penalizes the ego agent for moving toward positions that neighbors are predicted to occupy,

$$\mathcal{L}_{\mathrm{co}} = \sum_{x,y} M_i^{(t+1)}(x,y) \cdot M_{\mathrm{ego}}^{(t+1)}(x,y) \tag{18}$$

where $M_{\mathrm{ego}}^{(t+1)}$ is the occupancy map of the ego agent's next position, constructed directly from the action sampled by the actor head. Finally, the overall training objective is defined as

$$\mathcal{L}_{\mathrm{IMMP}} = \mathcal{L}_{\mathrm{actor}}^{w} + \lambda_{cr}\mathcal{L}_{\mathrm{cr}} + \lambda_{\mathrm{kl}}\mathcal{L}_{\mathrm{kl}} - \lambda_{\mathrm{e}}\mathcal{L}_{\mathrm{e}} + \lambda_{\mathrm{co}}\,\mathcal{L}_{\mathrm{co}} \tag{19}$$

where $\mathcal{L}_{\mathrm{cr}}$ is the clipped value function squared-error loss, $\mathcal{L}_{\mathrm{kl}}$ is the Kullback-Leibler (KL) divergence between the current and old policies, $\mathcal{L}_{\mathrm{e}}$ is the entropy bonus to encourage exploration, and $\lambda_{cr}, \lambda_{\mathrm{kl}}, \lambda_{\mathrm{e}}, \lambda_{\mathrm{co}}$ are hyperparameters.

Theorem 2 (End-to-End Differentiability)**.** *Under the soft expected position and Gaussian rasterization defined in Eq. 13, $\mathcal{L}_{\mathrm{IMMP}}$ is continuously differentiable with respect to both the actor parameters θ and the prediction module parameters ϕ.*

Proof. See Appendix A.3. □

5 Experiments

5.1 Implementation Details

The training environments consist of representative sub-grids from the Moving AI MAPF benchmark [27, 1] and the Rolling-Horizon Collision Resolution (RHCR) warehouse map collection [3], including den520d (referred to as Den520d), warehouse_large (referred to as Warehouse) and maze-32-32-2 (referred to as Maze). To rigorously assess the model's zero-shot generalization capability, the test maps include three seen maps from the training set and two unseen maps, which are Paris_1_256 (referred to as Paris) and random-32-32-20 (referred to as Random).

The policy is trained with an observation radius of 5, giving each agent an 11×11 local FOV. All models are trained for 2×10^8 environment steps on a single NVIDIA A800 GPU. Throughout the training phase, each agent is assigned four goals on training maps with a maximum episode duration of 1,024 steps. To ensure a rigorous comparison, all methods are evaluated under identical task configurations. Further hyperparameter details are provided in Appendix B.2.

We evaluate the performance of IMMP against the following baselines:

- **MGCBS** [8]: A centralized MG-MAPF solver that provides optimal solutions.
- **FOLLOWER** [12]: A decentralized method specifically designed for high-density environments and partially observable settings.
- **SCRIMP** [19]: A recent learnable approach that utilizes a transformer-based communication mechanism with partial observability.
- **SYLPH** [10]: A learning-based hierarchical MAPF framework that introduces Social Value Orientation (SVO) as a temporally extended latent variable to mitigate homogeneous agent behavior caused by parameter sharing.

In addition to SoC, we report the collision rate (CR), success rate (SR), and makespan (MS). The collision rate and success rate are calculated as follows:

$$\mathrm{CR} = \frac{\sum_{t=1}^{T} N_{\mathrm{col}}^{(t)}}{k \cdot T}, \qquad \mathrm{SR} = \frac{\sum_{i=1}^{k} N_{\mathrm{com}}^{(i)}}{k \cdot l}, \tag{20}$$

where $N_{\mathrm{col}}^{(t)}$ denotes the number of agents involved in either vertex or edge conflicts at time step t, and $N_{\mathrm{com}}^{(i)}$ denotes the number of completed goals of agent a_i. The makespan (MS) represents the time step at which all agents complete their l assigned goals.

Table 2: Performance on five maps across varying agent counts (k), measured by SoC, CR, MS, and SR, where an instance is considered a failure if the makespan exceeds 2,000 steps.

Map	Methods	SoC (CR (%))				MS (SR (%))			
		$k = 8$	$k = 16$	$k = 32$	$k = 64$	$k = 8$	$k = 16$	$k = 32$	$k = 64$
Maze	MGCBS	**952** (0.00)	-	-	-	**143**	-	-	-
	FOLLOWER	1660 (0.00)	3613 (0.24)	6282 (0.42)	12971 (1.27)	310	318	330	347
	SCRIMP	1813 (0.00)	2425 (0.00)	6322 (0.00)	**9441** (0.00)	327	267	316	**324**
	SYLPH	1871 (0.69)	3948 (1.33)	6684 (1.16)	25700 (7.03)	332	358	351	1286
	IMMP	1074 (0.00)	**2042** (0.00)	**4613** (0.16)	10467 (0.63)	152	**200**	**240**	340
Den520d	MGCBS	**1026** (0.00)	-	-	-	**171**	-	-	-
	FOLLOWER	1404 (0.10)	3817 (0.15)	7438 (0.18)	15414 (0.66)	242	328	392	408
	SCRIMP	1309 (0.00)	3549 (0.00)	6225 (0.00)	13497 (0.00)	218	308	367	399
	SYLPH	1297 (0.08)	3516 (0.05)	6123 (0.07)	13587 (1.17)	219	305	348	346
	IMMP	1215 (0.00)	**2853** (0.05)	**5253** (0.14)	**11431** (0.23)	191	**249**	**259**	**299**
Warehouse	MGCBS	**624** (0.00)	-	-	-	**113**	-	-	-
	FOLLOWER	985 (0.00)	1940 (0.07)	4365 (0.14)	8398 (0.32)	176	178	223	207
	SCRIMP	874 (0.00)	1816 (0.00)	4040 (0.00)	7830 (0.00)	139	168	209	184
	SYLPH	865 (0.05)	1802 (0.17)	4314 (0.72)	8508 (0.92)	141	167	216	208
	IMMP	794 (0.18)	**1386** (0.19)	**3111** (0.00)	**6370** (0.26)	140	**132**	**159**	**147**
Random	MGCBS	**464** (0.00)	-	-	-	**67**	-	-	-
	FOLLOWER	719 (0.22)	1498 (0.11)	3021 (0.18)	8492 (0.40)	115	119	175	(99.61)
	SCRIMP	738 (0.00)	1517 (0.00)	2931 (0.00)	6743 (0.00)	111	123	141	166
	SYLPH	728 (0.64)	1552 (0.38)	3048 (0.58)	7573 (1.24)	118	128	147	174
	IMMP	519 (0.00)	**1173** (0.13)	**2448** (0.31)	**5666** (0.63)	75	**93**	**102**	**159**
Paris	MGCBS	**969** (0.00)	**2132** (0.00)	-	-	**147**	**228**	-	-
	FOLLOWER	1505 (0.00)	3088 (0.12)	6652 (0.13)	13312 (0.34)	238	320	339	363
	SCRIMP	1428 (0.00)	3014 (0.00)	6271 (0.00)	12438 (0.00)	210	310	308	**328**
	SYLPH	1414 (0.00)	3024 (0.12)	6303 (0.19)	13218 (0.36)	208	312	332	341
	IMMP	1054 (0.00)	2468 (0.00)	**4825** (0.12)	**10503** (0.22)	156	232	**207**	342

5.2 Experimental Results

Table 2 presents a comprehensive comparison of our method against existing approaches.

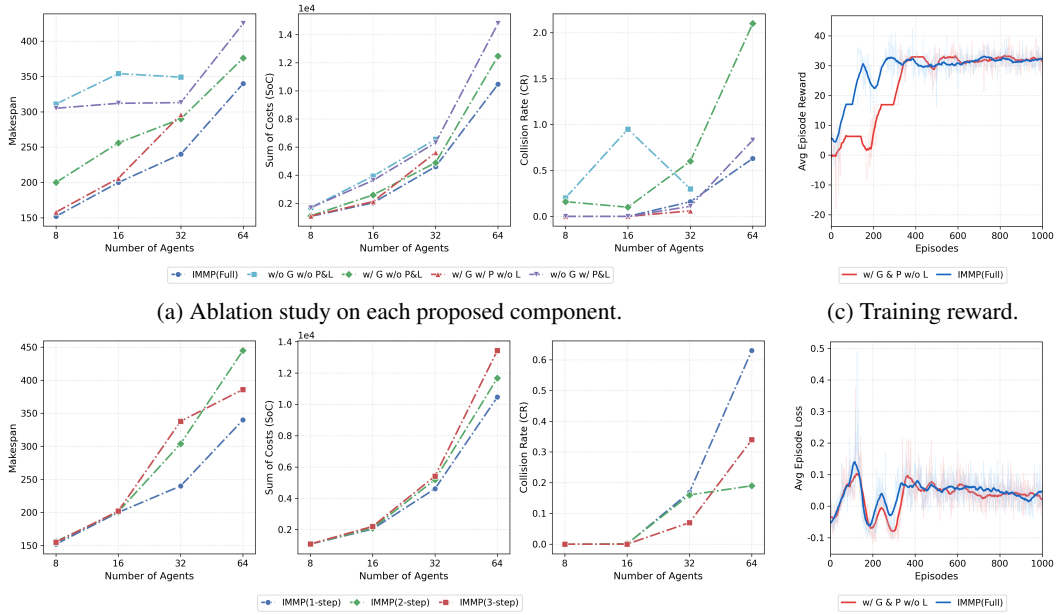
Superior Efficiency in High-Density Coordination. Among all decentralized methods, IMMP achieves superior performance in SoC and makespan across all maps, with the advantage becoming most pronounced at $k = 64$. On the three training maps, IMMP reduces SoC by an average of 25.89%, 13.68%, and 18.24% on Maze, Den520d, and Warehouse respectively, compared with the best-performing decentralized baseline at each configuration. While MGCBS obtains the lowest SoC owing to its optimality guarantee, it suffers from scalability limitations and fails to find solutions within the time limit as the number of agents increases, whereas IMMP remains efficient throughout.

Strong Zero-Shot Generalization. On the two unseen maps (Random and Paris), IMMP demonstrates strong generalization capability, reducing SoC by an average of 20.49% and 20.55% respectively, compared with the best-performing decentralized baseline at each configuration. Notably, in the Random map at $k = 64$, FOLLOWER fails to complete all goals within the step limit, whereas IMMP successfully completes all tasks, confirming that the prediction module learns generalizable interaction patterns rather than map-specific behaviors.

The Trade-off Between Efficiency and Conservatism. IMMP incurs a slightly higher collision rate than FOLLOWER in certain configurations on the Warehouse and Random maps, reflecting a trade-off between collision avoidance and path efficiency. IMMP adopts a more aggressive pathfinding strategy that prioritizes shorter paths and a lower makespan, whereas methods such as SCRIMP achieve a zero collision rate across all configurations by relying on built-in rule-based collision avoidance at the cost of a higher makespan.

5.3 Analysis

Ablation Study. To investigate the contribution of each component in IMMP, we conduct an ablation study on three components: goal vertex ordering (referred to as G), prediction module (referred to as P) and the modified training objective (referred to as L), where w/o L denotes replacing our modified objective with the classic PPO optimization. We evaluate their effects on makespan, SoC



(a) Ablation study on each proposed component.

(c) Training reward.

(b) Parameter study on prediction steps.

(d) Training loss.

Figure 4: Experimental analysis of IMMP.

and collision rate on the Maze map. As shown in Figure 4a, removing the goal vertex ordering (w/o G w/ P&L) leads to a substantial increase in both makespan and SoC, indicating that without spatially-balanced ordering, agents converge on the same regions and accumulate path conflicts. Removing the prediction module and modified training objective (w/ G w/o P&L) results in a sharp rise in collision rate, highlighting that they are essential for collision avoidance. Furthermore, removing the modified training objective (w/ G w/ P w/o L) and removing all three components (w/o G w/o P&L) both result in a failure to complete all goals when the number of agents reaches 64. Consequently, only three data points are reported for these two variants in Figure 4a. This suggests that the modified training objective is especially critical in high-density settings.

Parameter Study. We further conduct a parameter study to investigate how the number of prediction steps affects the performance of IMMP. The number of steps is varied from 1 to 3, corresponding to three variants of IMMP. As shown in Figure 4b, IMMP with 1-step prediction yields the lowest SoC and makespan among all variants. When the number of agents is 8 or 16, the three variants exhibit nearly identical performance. However, as the number of agents increases, the collision rate of IMMP with 1-step prediction rises sharply. Specifically, with 64 agents, IMMP with 1-step prediction reaches a CR of 0.63%. In contrast, although IMMP with 2-step or 3-step prediction incurs higher SoC and makespan, the collision rates of these two variants under 64 agents are lower than that of IMMP with 1-step prediction. In conclusion, increasing the number of prediction steps enhances the collision avoidance capability of IMMP, at the cost of higher SoC and makespan.

Training Convergence and Stability. As illustrated in Figures 4c and 4d, IMMP achieves faster policy convergence than the ablated variant (w/ G & P w/o L). Guided by the prediction-based collaborative PPO objective, the average episode reward increases rapidly from the onset of training. In contrast, the ablated variant exhibits notably slower convergence, requiring approximately 300–400 episodes to reach a comparable reward level. Correspondingly, the training loss for both methods converges toward zero, consistent with the reward trajectories. During the early training phase (episodes 0–200), both variants exhibit substantial loss fluctuations, a behavior commonly observed in PPO-based optimization.

Computational Efficiency. We further analyze runtime and memory scaling with respect to the number of agents and goals per agent in Appendix B.5.

6 Conclusion

In this paper, we proposed IMMP, a hierarchical framework for decentralized MG-MAPF that decouples high-level goal ordering from low-level action execution. At the high level, we introduced a Spatially-Balanced Goal Ordering and Pathfinding strategy that jointly optimized the visiting sequences of all agents to minimize spatial path overlap, thereby reducing congestion pressure on the low-level RL policy. At the low level, we designed a neighbor intent prediction module based on cross-attention that enabled each agent to predict the next-step actions of its neighbors, enhancing collision avoidance capability. Extensive experiments demonstrated that IMMP achieves strong scalability and generalization, outperforming decentralized baselines in solution quality and collision avoidance across varying agent scales and unseen maps. We also discuss the limitations of our method in Appendix B.6. Future work includes extending the framework with an inter-agent communication mechanism to improve collision avoidance without relying on explicit rules, and scaling the approach to larger agent teams and more complex map structures.

Acknowledgments and Disclosure of Funding

The work was partially supported by the National Key Research and Development Program of China (Grant No. 2024YFF0617702), the National Natural Science Foundation of China (Grant Nos. U22A2025, 62232007, U23A20309, and 62602105), the Project (No. 2026RJ02XKZX01), the 111 Project (No. B16009), the Fundamental Research Funds for the Central Universities (Grant No. N26XQD031), the Postdoctoral Innovative Talent Support Program (Grant No. BX2026292), and the Liaoning Provincial Natural Science Foundation for the Doctoral Research Initiation Project (Grant No. 2026-BS-0115).

References

[1] Roni Stern, Nathan Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, TK Kumar, et al. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Proceedings of the international symposium on combinatorial search*, volume 10, pages 151–158, 2019.

[2] Robert Morris, Corina S Pasareanu, Kasper Søe Luckow, Waqar Malik, Hang Ma, TK Satish Kumar, and Sven Koenig. Planning, scheduling and monitoring for airport surface operations. In *AAAI Workshop: Planning for Hybrid Systems*, pages 608–614, 2016.

[3] Jiaoyang Li, Andrew Tinka, Scott Kiesel, Joseph W. Durham, T. K. Satish Kumar, and Sven Koenig. Lifelong multi-agent path finding in large-scale warehouses. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11272–11281, May 2021.

[4] Pavel Surynek. Multi-goal multi-agent path finding via decoupled and integrated goal vertex ordering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12409–12417, 2021.

[5] Jingjin Yu and Steven LaValle. Structure and intractability of optimal multi-robot path planning on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, pages 1443–1449, 2013.

[6] Michael Jünger, Gerhard Reinelt, and Giovanni Rinaldi. The traveling salesman problem. *Handbooks in operations research and management science*, 7:225–330, 1995.

[7] Vera Traub, Jens Vygen, and Rico Zenklusen. Reducing path tsp to tsp. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 14–27, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369794.

[8] Mingkai Tang, Yuanhang Li, Hongji Liu, Yingbing Chen, Ming Liu, and Lujia Wang. Mgcbs: an optimal and efficient algorithm for solving multi-goal multi-agent path finding problem. *arXiv preprint arXiv:2404.19518*, 2024.

[9] Anton Andreychuk, Konstantin Yakovlev, Aleksandr Panov, and Alexey Skrynnik. Mapf-gpt: Imitation learning for multi-agent pathfinding at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(22):23126–23134, Apr. 2025.

[10] Chengyang He, Tanishq Duhan, Parth Tulsyan, Patrick Kim, and Guillaume Sartoretti. Social behavior as a key to learning-based multi-agent pathfinding dilemmas. *Artificial Intelligence*, 348:104397, 2025. ISSN 0004-3702.

[11] Chengyang He, Tianze Yang, Tanishq Duhan, Yutong Wang, and Guillaume Sartoretti. Alpha: Attention-based long-horizon pathfinding in highly-structured areas. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14576–14582. IEEE, 2024.

[12] Alexey Skrynnik, Anton Andreychuk, Maria Nesterova, Konstantin Yakovlev, and Aleksandr Panov. Learn to follow: Decentralized lifelong multi-agent pathfinding via planning and learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17541–17549, Mar. 2024.

[13] Mehul Damani, Zhiyao Luo, Emerson Wenzel, and Guillaume Sartoretti. Primal$_2$: Pathfinding via reinforcement and imitation multi-agent learning-lifelong. *IEEE Robotics and Automation Letters*, 6(2):2666–2673, 2021.

[14] He Jiang, Yutong Wang, Rishi Veerapaneni, Tanishq Duhan, Guillaume Sartoretti, and Jiaoyang Li. Deploying ten thousand robots: Scalable imitation learning for lifelong multi-agent path finding. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2025.

[15] Zikun Guo, Adeyinka P Adedigba, Rammohan Mallipeddi, and Heoncheol Lee. Structural-prior induced exploration for balanced and scalable multi-robot path planning. *Computers and Electrical Engineering*, 134:111141, 2026.

[16] Takuro Kato, Keisuke Okumura, Yoko Sasaki, and Naoya Yokomachi. Congestion mitigation path planning for large-scale multi-agent navigation in dense environments. *IEEE Robotics and Automation Letters*, 2025.

[17] Zhe Chen, Daniel Harabor, Jiaoyang Li, and Peter J Stuckey. Traffic flow optimisation for lifelong multi-agent path finding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20674–20682, 2024.

[18] Xiaolong Wang, Alp Sahin, and Subhrajit Bhattacharya. Coordination-free multi-robot path planning for congestion reduction using topological reasoning. *Journal of Intelligent & Robotic Systems*, 108(3):50, 2023.

[19] Yutong Wang, Bairan Xiang, Shinan Huang, and Guillaume Sartoretti. Scrimp: Scalable communication for reinforcement-and imitation-learning-based multi-agent pathfinding. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9301–9308. IEEE, 2023.

[20] Junjun Li, Zeyuan Ma, Ting Huang, and Yue-Jiao Gong. Learn to refine: Synergistic multi-agent path optimization for lifelong conflict-free navigation of autonomous vehicles. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 1400–1411, 2025.

[21] Guillaume Sartoretti, Justin Kerr, Yunfei Shi, Glenn Wagner, TK Satish Kumar, Sven Koenig, and Howie Choset. Primal: Pathfinding via reinforcement and imitation multi-agent learning. *IEEE Robotics and Automation Letters*, 4(3):2378–2385, 2019.

[22] Sili Huang, Hechang Chen, Haiyin Piao, Zhixiao Sun, Yi Chang, Lichao Sun, and Bo Yang. Boosting weak-to-strong agents in multiagent reinforcement learning via balanced ppo. *IEEE Transactions on Neural Networks and Learning Systems*, 36(5):9136–9149, 2025.

[23] Alexey Skrynnik, Anton Andreychuk, Anatolii Borzilov, Alexander Chernyavskiy, Konstantin Yakovlev, and Aleksandr Panov. POGEMA: A benchmark platform for cooperative multi-agent pathfinding. In *The Thirteenth International Conference on Learning Representations*, 2025.

[24] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

[25] Zhang-Wei Hong, Shih-Yang Su, Tzu-Yun Shann, Yi-Hsiang Chang, and Chun-Yi Lee. A deep policy inference q-network for multi-agent systems. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 1388–1396, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[26] Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley J. Osher, Yingyong Qi, and Jack Xin. Understanding straight-through estimator in training activation quantized neural nets. In *International Conference on Learning Representations*, 2019.

[27] Nathan R. Sturtevant. Benchmarks for grid-based pathfinding. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(2):144–148, 2012.

A Theoretical Analysis

A.1 Proof of Lemma 1

Lemma 1 (Marginal Density Cost)**.** *Let $\mathbf{H} \in \mathbb{R}^{|V|}$ denote the path density map prior to agent a_i deciding its visiting order, and let P_i denote the pre-planned path. Then:*

$$\Delta C_{\text{density}} = \sum_{v \in P_i} \big(2\mathbf{H}(v) + 1\big), \tag{21}$$

and for any sub-path $\rho(g_i^m, g_i^n) \subseteq P_i$, the marginal increment equals the penalty term of Cost defined in Eq. 6.

Proof. By definition, $C_{\text{density}} = \sum_{v \in V} \mathbf{H}(v)^2$. After incorporating P_i, the updated density map satisfies $\mathbf{H}'(v) = \mathbf{H}(v) + \mathbb{1}[v \in P_i]$. Therefore:

$$\begin{aligned}
\Delta C_{\text{density}} &= \sum_{v \in V} \big(\mathbf{H}'(v)^2 - \mathbf{H}(v)^2\big) \\
&= \sum_{v \in V} \mathbb{1}[v \in P_i]\big(2\mathbf{H}(v) + \mathbb{1}[v \in P_i]\big) \\
&= \sum_{v \in P_i} \big(2\mathbf{H}(v) + 1\big).
\end{aligned} \tag{22}$$

For the sub-path $\rho(g_i^m, g_i^n) \subseteq P_i$, the marginal increment is $\sum_{v \in \rho(g_i^m, g_i^n)} (2\mathbf{H}(v) + 1)$, matching the penalty term in $\text{Cost}(g_i^m, g_i^n)$. □

A.2 Proof of Theorem 1

Theorem 1 (Finite Convergence). *The iterative minimization of $\mathcal{L}_{\text{seq}}(\sigma_i)$ over visiting orders $\{\sigma_i\}$ terminates in a finite number of iterations.*

Proof. We establish finite convergence by identifying $\mathcal{J}$ as a Lyapunov-like potential function for the system.

(i) Strict Monotonicity. Recall $\mathcal{L}_{\text{seq}}$ defined in Eq. 7 and the joint objective:

$$\mathcal{J}(\{\sigma_j\}_{j=1}^k) = \sum_{j=1}^{k} |P_j| + \mu \cdot C_{\text{density}}. \tag{23}$$

Let σ_i' be the new visiting order for agent a_i, and denote by σ_{-i} the visiting orders of all other agents. By the acceptance criterion, σ_i' is adopted if and only if $\mathcal{L}_{\text{seq}}(\sigma_i') < \mathcal{L}_{\text{seq}}(\sigma_i)$. By Lemma 1, $\mathcal{L}_{\text{seq}}$ captures the exact marginal change of $\mathcal{J}$ with respect to σ_i. Consequently, any accepted update yields a strict decrease:

$$\mathcal{J}(\sigma_i', \sigma_{-i}) < \mathcal{J}(\sigma_i, \sigma_{-i}). \tag{24}$$

Hence $\mathcal{J}$ is strictly decreasing along any sequence of accepted updates, and no configuration can be revisited.

(ii) Finite State Space. Each agent a_i is assigned l goal vertices. The joint state space $\mathcal{S}$ across all k agents satisfies:

$$|\mathcal{S}| \leq \prod_{i=1}^{k} l! < \infty. \tag{25}$$

Conclusion. Since $\mathcal{J}$ is bounded below and strictly decreases at each accepted update over the finite state space $\mathcal{S}$, the algorithm terminates in at most $|\mathcal{S}| = \prod_{i=1}^{k} l!$ iterations. □

A.3 Proof of Theorem 2

Theorem 2 (End-to-End Differentiability). *Under the soft expected position and Gaussian rasterization defined in Eq. 13, $\mathcal{L}_{\mathrm{IMMP}}$ is continuously differentiable with respect to both the actor parameters θ and the prediction module parameters ϕ.*

Proof. Under hard argmax, $\hat{a}_j = \arg\max_a \mathbf{P}_{j,a}^{(t)}$ is a piecewise constant function of $\mathbf{P}_j^{(t)}$ with zero gradient everywhere it is defined, blocking gradient flow to the prediction module.

Under soft relaxation, the predicted position $(\hat{x}_j, \hat{y}_j)$ defined in Eq. 12 is an affine function of $\mathbf{P}_j^{(t)}$, hence differentiable. The spatial occupancy map $M_i^{(t+1)}$ in Eq. 13 is infinitely differentiable with respect to $(\hat{x}_j, \hat{y}_j)$, and $\mathcal{L}_{\mathrm{co}}$ is a linear functional of $M_i^{(t+1)}$. By the chain rule,

$$\frac{\partial \mathcal{L}_{\mathrm{co}}}{\partial \phi} = \frac{\partial \mathcal{L}_{\mathrm{co}}}{\partial M_i^{(t+1)}} \cdot \frac{\partial M_i^{(t+1)}}{\partial (\hat{x}_j, \hat{y}_j)} \cdot \frac{\partial (\hat{x}_j, \hat{y}_j)}{\partial \mathbf{P}_j^{(t)}} \cdot \frac{\partial \mathbf{P}_j^{(t)}}{\partial \phi} \tag{26}$$

where each factor exists and is continuous, with $\partial \mathbf{P}_j^{(t)} / \partial \phi$ following from the differentiability of the GRU and attention layers.

The confidence-weighted actor loss in Eq. 16 is differentiable with respect to θ, since $\mathcal{L}_{\mathrm{actor}}$ is differentiable by the standard PPO construction and the clip scaling factor is a bounded positive constant with respect to θ. Since all remaining terms in $\mathcal{L}_{\mathrm{IMMP}}$ are differentiable with respect to $\{\theta, \phi\}$ by construction, $\nabla_{\theta,\phi} \mathcal{L}_{\mathrm{IMMP}}$ exists and is continuous.

□

A.4 Proof of Proposition 1

Proposition 1 (Bounded Gradient Modulation)**.** *The confidence-weighted actor loss $\mathcal{L}_{\mathrm{actor}}^{w}$ satisfies: (i) when $\bar{w}^{(t)} = 0$, it reduces exactly to the standard PPO actor loss; (ii) when $\bar{w}^{(t)} = 1$, it amplifies the actor loss by a factor of $(1 + \alpha_c)$; (iii) the scaling factor $s_a = \mathrm{clip}(1 + \alpha_c \bar{w}^{(t)}, 1, 1 + \alpha_c)$ is bounded in $[1, 1 + \alpha_c]$ for all $\bar{w}^{(t)} \in [0, 1]$.*

Proof. Let $s_a = \mathrm{clip}(1 + \alpha_c \bar{w}^{(t)}, 1, 1 + \alpha_c)$. (i) When $\bar{w}^{(t)} = 0$, $s_a = 1$, so $\mathcal{L}_{\mathrm{actor}}^{w} = \mathcal{L}_{\mathrm{actor}}$. (ii) When $\bar{w}^{(t)} = 1$, $s_a = 1 + \alpha_c$, so $\mathcal{L}_{\mathrm{actor}}^{w} = (1 + \alpha_c)\mathcal{L}_{\mathrm{actor}}$. (iii) By definition of the clip operation, $s_a \in [1, 1 + \alpha_c]$ for all $\bar{w}^{(t)} \in [0, 1]$. □

Remark 1. *Although s_a varies across states as a function of $\bar{w}^{(t)}$, it remains bounded in $[1, 1 + \alpha_c]$, ensuring that the gradient of $\mathcal{L}_{\mathrm{actor}}^{w}$ never vanishes or diverges relative to the standard PPO gradient. The bounded gradient modulation serves as a necessary condition for training convergence and stability.*

B Supplementary Materials

B.1 Reward Structure

The reward structure is shown in Table 3. When agent a_i reaches its current sub-goal, it receives a goal reward, along with an extra consecutive reward to encourage it to complete goals in succession. When all agents complete the same number of goals, a team reward is given to encourage cooperative behavior. A movement penalty discourages unnecessary waiting, and an intrinsic reward provides a dense reward that encourages the agent to follow the guiding path $\hat{P}_i^{(t)}$.

B.2 Implementation Details

Table 4 summarizes the hyperparameters used in all experiments.

Table 3: Reward structure.

Action	**Reward**
Goal reached reward	$+1.0$
Consecutive goals (c-th)	$+1.0 \times (c-1)$
Team reward	$+2.0$
Collision penalty	-1.0
Movement penalty	-0.001
Intrinsic reward	$+0.002$

Table 4: Hyperparameters of the IMMP framework.

Module	**Hyperparameter**	**Symbol**	**Value**
High-level Planner	Congestion weight	μ	0.5
	2-opt max iterations	–	30
Neighbor Intent Predictor	Gaussian kernel width	σ_g	0.8
	Confidence weight	α_c	0.3
Training Objective	Value loss coefficient	λ_{cr}	1.0
	KL divergence coefficient	λ_{kl}	1.0
	Entropy coefficient	λ_{e}	0.1
	Collaborative penalty coefficient	λ_{co}	0.1
	Learning rate	lr	1.46×10^{-4}
	Discount factor	γ	0.9756

Table 5: Experimental results of the ablation and parameter studies on the Maze (32×32) map.

	Ablation Study								Parameter Study					
	w/o G w/o P&L		w/ G w/o P&L		w/ G w/ P w/o L		w/o G w/ P&L		2-step		3-step		IMMP	
N	MS	SoC (CR)	MS	SoC (CR)	MS	SoC (CR)	MS	SoC (CR)	MS	SoC (CR)	MS	SoC (CR)	MS	SoC (CR)
8	311	1696 (0.20)	200	1118 (0.16)	158	1100 (0.00)	305	1692 (0.00)	156	1084 (0.00)	155	1085 (0.00)	**152**	**1074** (**0.00**)
16	354	3937 (0.95)	256	2602 (0.10)	206	2124 (0.00)	312	3636 (0.00)	201	2090 (0.12)	202	2208 (0.00)	**200**	**2042** (**0.00**)
32	349	6564 (0.30)	290	4892 (0.60)	296	5615 (0.06)	313	6312 (0.11)	304	5195 (0.16)	338	5419 (**0.07**)	**240**	**4613** (0.16)
64	-	-	376	12476 (2.10)	-	-	425	14788 (0.83)	445	11679 (0.19)	386	13435 (0.34)	**340**	**10467** (**0.63**)

B.3 Ablation Study and Parameter Study

Table 5 summarizes the numerical results of the ablation study and the parameter study.

B.4 IMMP under Hard Collision Constraints

To examine IMMP's performance under rule-based hard collision constraints, we conducted an additional experiment. The results are shown in Table 6.

All assigned goals are completed in the 8-, 16-, and 32-agent settings. As the number of agents increases from 8 to 32, makespan rises from 150 to 227 and SoC from 1076 to 4689. In the 64-agent setting, only 192 of 256 goals are completed within the evaluation budget. This result suggests that rule-based constraints may limit IMMP's performance in high-density settings, motivating future research into more effective learning-based collision-avoidance algorithms.

B.5 Runtime and Memory Scaling Analysis

We evaluate the computational scalability of IMMP by measuring runtime and GPU memory consumption under different agent numbers and per-agent goal counts. Table 7 reports the computational overhead under these configurations.

As shown in Table 7, IMMP exhibits manageable computational growth as the number of agents increases. With two goals per agent, guidance update time increases from 0.0295 to 0.1473 s/step when scaling from 8 to 64 agents, while episode time increases from 28.379 to 81.809 s. Increasing the number of goals mainly affects preprocessing and total execution time, with a smaller impact on

Table 6: Evaluation of IMMP under Hard Collision-Avoidance Constraints

Num. Agents	Makespan	SoC	Throughput	Success Rate
8	150	1076	0.213	100%
16	208	2050	0.320	100%
32	227	4689	0.564	100%
64	–	–	–	75%

Table 7: Runtime and memory scaling of IMMP. Order refers to one-time goal-ordering preprocessing, Guidance refers to average per-step guidance update, Episode refers to total episode execution time and GPU refers to peak GPU memory usage.

Agents	Goals/agent	Order (s)	Guidance (s/step)	Episode (s)	GPU (MB)
8	2	0.0371	0.0295	28.379	3216.750
8	4	0.0592	0.0341	30.503	3212.078
16	2	0.0556	0.0451	36.712	3211.875
16	4	0.1500	0.0425	37.418	3217.383
32	2	0.1424	0.0805	43.796	3205.316
32	4	0.3065	0.0956	56.634	3219.176
64	2	0.5557	0.1473	81.809	3233.391
64	4	0.4333	0.1618	88.318	3223.945

guidance updates. For example, at 32 agents, doubling the goals increases goal-ordering time from 0.1424 to 0.3065 s, whereas guidance time increases only from 0.0805 to 0.0956 s/step. Meanwhile, peak GPU memory remains approximately 3.2 GB across all configurations, indicating stable memory usage as the problem size increases.

B.6 Limitations

Performance Degradation at Scale. Despite achieving superior overall performance, IMMP exhibits two notable limitations when the number of agents scales. As shown in Table 2, both SoC and makespan grow super-linearly as the number of agents increases to 64, indicating that the increasingly complex environment poses a greater challenge for IMMP. Furthermore, unlike rule-based methods that can enforce strict collision avoidance, IMMP does not incorporate a strict collision prevention mechanism, resulting in a non-zero collision rate across most maps at $k = 64$, which may pose concerns in real-world applications where zero-collision guarantees are required.

Sensitivity to Prediction Steps. As shown in Figure 4b, 1-step prediction achieves the lowest SoC and makespan but incurs the highest collision rate at $k = 64$ (CR = 0.63%). Increasing the prediction step to 2 or 3 steps reduces the collision rate but introduces higher SoC and makespan, revealing a fundamental trade-off between path efficiency and collision avoidance.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We clearly claim our contributions and scope in the abstract and introduction, including a list of contributions in solving the MG-MAPF problem. The claims are supported by both theoretical analysis and comprehensive experiments.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Appendix B.6, covering scalability constraints and the trade-off between path efficiency and collision avoidance.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The Method or Appendix includes all assumptions and proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: All necessary information for reproduction, including algorithm details and hyperparameter settings, is fully disclosed in the paper and Appendix B.2

   Guidelines:

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

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Complete source code and step-by-step instructions are provided in the supplementary material and via an anonymous GitHub link. The repository will be made public once the paper is published.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details in the Experiments and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: All experimental results are obtained from five independent trials using different random seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provided the necessary computational resources, hardware setup, and duration required to reproduce the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We ensure compliance with all aspects of the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We have properly credited assets used in our paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We have provided a README file alongside our code.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

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

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The paper does not involve LLMs in method.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.