Diffusion for Long-Horizon Multi-Robot Path Planning in Human-Shared Environments

Vaibhav Sanjay[1], Yorai Shaoul[1], and Jiaoyang Li[1]

***Abstract*— Multi-robot path planning in human-shared environments requires a delicate balance between robust inter-robot coordination and socially aware behavior. While diffusion models excel at generating predictable, human-like paths, existing generative planners are often restricted to paths of fixed duration and high computational latency, limiting their adaptability to varying goal distances and hindering real-time deployment. We present Multi-Robot Rolling Diffusion (MRRD), a novel framework that enables real-time, long-horizon navigation for large robot teams through dense crowds. MRRD combines a rolling-horizon scheme to accommodate the limited prediction horizon of human motion, parallelized diffusion inference for scalable generation of human-like paths, and a conflict-based-search mechanism for resolving inter-robot collisions. It further incorporates urgency-based temporal conditioning to generate paths with varying speeds and employs differentiated guidance terms to maximize both social awareness around humans and efficient coordination between robots. Experimental results in crowded environments demonstrate that MRRD successfully scales to 15 robots in real-time, significantly outperforming existing baselines in both safety and mission success rates.**

I. INTRODUCTION

Multi-robot autonomous systems are successfully used in scenarios like warehouse automation, drone fleets, and inspection systems. However, their use in human-shared environments remains largely underexplored. This requires ensuring not only collision-free coordination among robots but also safe interaction with surrounding humans.

Single-robot navigation in human-shared environments has been explored extensively [1]–[12], with a wide range of methods including human motion prediction and reinforcement learning. In this work, we focus on methods that directly generate human-like paths because they enable robots to move in a predictable and socially acceptable manner. Diffusion models [13], which excel at learning from demonstrations, are therefore a natural choice for this setting. However, these generative planners are often subject to a "fixed-time problem," where they are restricted to produce paths of fixed duration regardless of the actual distance to the goal, leading to erratic velocities and reduced navigation efficiency.

Research on multi-robot path planning in human-shared environments is not as extensive and can be largely summarized in a few distinct approaches. Prioritized planning methods [14], [15] anticipate human motions and plan robot paths in a predefined sequence, avoiding humans and higher-priority robots. Although quite fast, planning in sequence may result in uncooperative behavior since high-priority robots do not take the paths of low-priority robots into account. Hierarchical methods [16], [17] plan global paths ignoring collisions and resolve them during execution by maneuvering robots around each other. Since collisions are resolved by local planning, these methods may lack the ability to perform long-term planning while maintaining social awareness. RL-based methods [18] can directly learn robot coordination policies. However, they often rely on carefully engineered reward functions, which may not capture the full range of desirable human-like behaviors.

This paper introduces Multi-Robot Rolling Diffusion (MRRD), a rolling-horizon approach utilizing diffusion models to generate real-time, human-like, paths for a team of robots in human-shared environments. Rather than generating a single, globally "pinned" path of rigid duration, our diffusion model produces localized path segments conditioned on a diverse range of urgency values. This gives MRRD significant freedom in planning for complex, dynamic scenes; it can adaptively scale velocities to suit immediate crowd density and distance to the goal. Inspired by [19], MRRD scales single-robot diffusion models trained from human walking data to a multi-robot setup via Conflict-Based Search (CBS) [20], a classical multi-agent path-finding technique. This way, MRRD distinguishes between robot-robot interactions (handled by CBS) and robot-human interactions (handled by the diffusion model). We demonstrate its effectiveness in various simulated environments, showing real-time navigation with up to 15 robots in empty environments and 10 in cluttered ones.

II. RELATED WORKS

We briefly review related work on multi-robot path planning and robot navigation in human-shared environments.

A. Multi-Robot Path Planning (Without Humans)

Multi-robot path planning is a challenging problem of planning collision-free paths for a team of robots in a shared environment. One promising approach is to abstract the problem as Multi-Agent Path Finding (MAPF) [21], which assumes that robots move in discrete timesteps on a graph, typically a 2D grid. Solutions to MAPF range from optimal search-based methods to efficient heuristics. Prioritized Planning (PP) [22] is a fast, suboptimal algorithm that assigns fixed priorities to robots, plans their paths sequentially, and treats higher-priority robots as moving obstacles. However, PP may force lower-priority robots into circuitous routes due to sequential dependencies. The optimal

[1]Vaibhav Sanjay, Yorai Shaoul, and Jiaoyang Li are with the Robotics Institute, Carnegie Mellon University. {vsanjay,yshaoul,jiaoyanl}@andrew.cmu.edu

and scalable standard is Conflict-Based Search (CBS) [20], a two-level algorithm that combines single-robot path planning with a high-level constraint tree to resolve collisions, imposing constraints only where collisions occur. Multi-Agent Reinforcement Learning (MARL) [23] has emerged as a prominent decentralized alternative, where robots learn local navigation policies—often via actor-critic architectures.

While MAPF algorithms were originally designed for discrete settings, many have been extended to continuous time and space. In particular, Multi-Robot Multi-Model Planning Diffusion (MMD) [19] combines CBS with diffusion-based planners. By training on large datasets of successful single-robot paths, diffusion-based planners can generate paths that effectively capture the complex spatial dependencies of the environment and movement patterns in the data. MMD uses diffusion models to generate single-robot paths while borrowing the constraint tree structure from CBS to designate constraints to the diffusion-based planners to resolve collisions. It defines spherical constraints that push diffusion-generated paths out of collision regions without over-constraining the entire path like PP would.

B. Robot Navigation in Human-Shared Environments

Single-robot path planning in human-shared environments has been extensively studied, utilizing approaches such as A* search [1], model predictive control [2]–[4], reinforcement learning [5]–[8], vision language models [9], [10], and generative adversarial networks [11]. We are mainly concerned with algorithms that learn from human data or input, directly embedding socially aware navigation in the planning process. Navi-STAR [12] uses preference learning with human oversight for training, but gathering preferences can be costly. CoBL-Diffusion [24] learns from human walking data using a diffusion model to emulate movement and avoid paths via guidance; however, it requires knowing the full human paths beforehand, which is unrealistic. However, diffusion models have also been shown to be particularly useful for predicting human movements throughout a scene [25].

Multi-robot path planning in human-shared environments is less studied than single-robot path planning and requires balancing high-level coordination with socially acceptable local reactivity. This general multi-robot navigation problem [26] can lead to constrained situations, where robots and humans may have to vie for space. Existing methods can be divided into three categories: PP, hierarchical planning, and MARL. PP methods employ prioritized planning with A* [14] or model predictive control [15] to avoid predicted paths of humans and higher-priority robots. PP limitations persist and can be worsened by humans cutting off paths. Hierarchical methods generate global paths without considering humans or other robots, relying on a local planner for collision avoidance during execution. Recent work integrates neural control barrier functions for safety guarantees [16] or uses a Lloyd-based planner with Voronoi-partitioning to shift paths away from others and predicted human zones [17]. These controllers lack the long-horizon foresight needed for complex social dynamics due to prioritizing local safety. MARL has been adapted to address the specific complexities of social navigation. A primary distinction from standard MARL—which typically focuses on robot-robot coordination and fast goal reaching—is the additional requirement to model asymmetric human-robot interactions and maintain social awareness. SAMARL [18] achieves this by utilizing a hybrid spatial-temporal transformer to capture and align high-order dependencies in both human-robot and robot-robot interactions. Such methods are computationally efficient during inference, but developing a reward function that truly emulates human-like behavior remains an active research area.

III. Preliminaries

We now introduce the problem definition and background on diffusion models for multi-robot path planning.

A. Problem Definition

Our task is multi-robot navigation in a 2D workspace $\mathcal{W} \subset \mathbb{R}^2$ containing N robots $\{X_i\}_{i=1}^N$, M humans $\{Y_j\}_{j=1}^M$, and Q static obstacles $\{O_k\}_{k=1}^Q$ where each $O_k \subseteq \mathcal{W}$. Time is discretized into uniform timesteps separated by Δt (set to 0.1 seconds in our experiments). We assume access to a perfect prediction function that provides $\mathcal{H}_j = \{Y_j^0, Y_j^1, \cdots, Y_j^{H-1}\}$, the predicted path of human Y_j over the next H timesteps, where H is the planning horizon (set to 32 in our experiments), and $Y_j^t \in \mathcal{W}$ is the position of human Y_j t timesteps after the prediction is made. Each robot X_i is assigned a start and goal position $s_i, g_i \in \mathcal{W}$. A full path of robot X_i is written as a time-indexed sequence of waypoints $\tau_i = \{\tau_i^1, \tau_i^2, \cdots, \tau_i^{L_i}\}$ with $\tau_i^1 = s_i$ and $\tau_i^{L_i} = g_i$, where L_i is referred to as the length of the path, and $\tau_i^t \in \mathcal{W}$ is the position of robot X_i at time t. Our objective is to find a set of full paths $\{\tau_i\}_{i=1}^N$ that do not collide with each other (robot-robot collision), with any humans (robot-human collision), or with any obstacles (robot-obstacle collision).

B. Diffusion Models

Diffusion models learn to emulate complex patterns from a dataset through an iterative denoising process. In this framework, motion planning is formulated as a probabilistic inference problem where the model learns a prior over expert paths. The generated paths should also satisfy task objectives, such as avoiding obstacles. The goal is to maximize the posterior probability of a path τ_i given task objectives $\mathcal{O}$:

$$\arg\max_{\tau_i} \log p(\tau_i|\mathcal{O}) = \arg\min_{\tau_i}(\mathcal{J}(\tau_i) - \log p(\tau_i)), \quad (1)$$

where $\mathcal{J}(\tau_i)$ is a cost associated with the task objectives, and $\log p(\tau_i)$ represents the prior to the learned data distribution. In our case, relevant task objectives are robot collision avoidance with humans, obstacles, and other robots.

A significant limitation of current diffusion-based path planners, including MMD and CoBL-Diffusion, is that all generated paths contain a fixed number of waypoints. Because consecutive waypoints are separated by a constant time interval Δt, the robot's arrival time is fixed. This "fixed-time problem" introduces critical performance issues. In

Algorithm 1: MRRD: Rolling-Horizon Planner

Input: Start positions $\{s_i\}_{i=1}^N$, goal positions $\{g_i\}_{i=1}^N$, planning horizon H, execution horizon K

1: **while** $\exists i : ||g_i - s_i|| > \delta$ **do**
2: $\quad \{\mathcal{H}_j\}_{j=1}^M \leftarrow$ GETHUMANPATHS(H)
3: $\quad \{\pi_i\}_{i=1}^N \leftarrow$ PLANSEGMENT($\{s_i\}, \{g_i\}, \{\mathcal{H}_j\}$)
4: $\quad \{s_i\}_{i=1}^N \leftarrow$ EXECUTE($\{\pi_i\}$, K)
5: **end while**

particular, the robot's velocity is dictated primarily by its distance to the goal, leading to dangerously high velocities for distant goals or unnaturally slow, stuttering movements for nearby goals. This behavior is problematic in human-shared environments, where maintaining an acceptable pace is vital. Furthermore, the fixed-time problem prevents diffusion-based planners from optimizing execution time.

IV. METHOD

We introduce MRRD, a 3-level diffusion-based path planning algorithm for multi-robot navigation in human-shared environments. At the top level, our method plans robot paths iteratively using a rolling-horizon scheme [27], interleaving planning and execution to form a set of full long-horizon paths. In the middle level, we resolve robot-robot collisions within horizons and assign constraints for path generation. At the bottom level, we generate human-like paths for each robot from a single-robot diffusion policy learned from human path data.

A. Rolling-Horizon Planning

Algorithm 1 shows our top-level rolling-horizon planner. Unlike prior diffusion-based planners, which plan the full path from start to finish, MRRD generates short path segments in a rolling-horizon fashion. Since we do not generate full robot paths directly, the human path prediction function only needs to look ahead to the planning horizon H (line 2). In line 3, we generate path segments π_i of length H for all robots. During the execution step (line 4), we execute $K < H$ steps of this plan, where K is called the execution horizon. The positions of the robots after execution are set as their new start positions for the next iteration. This process is repeated (line 1) until all robots have reached their goals within a certain tolerance δ (set to 0.2 meters in our experiments). In practice, we cap the maximum number of iterations (set to 11 in our experiments) of the while loop such that the robot execution ends.

MRRD treats each rolling horizon as its own localized multi-robot planning problem, as shown in Algorithm 2. This level is inspired by MMD [19] and is meant to resolve robot-robot collisions. The process requires several key steps.

Segment Generation: The path segment generation process begins with the creation of a root node. Each node contains $\{\pi_i\}_{i=1}^N$, a set of path segments of length H for each robot, $\{C_i\}_{i=1}^N$, a set of constraint sets imposed on each robot (initially empty), and *colls*, a list of robot-robot collisions between the path segments. During each planning horizon, we start by calling the diffusion-based planner to plan a path

Algorithm 2: Multi-Robot Planner (PLANSEGMENT)

Input: Start positions $\{s_i\}_{i=1}^N$, goal positions $\{g_i\}_{i=1}^N$, human paths $\{\mathcal{H}_j\}_{j=1}^M$
Output: Collision-free paths $\{\pi_i\}_{i=1}^N$

Phase 1: Segment Generation
1: *root* $\leftarrow$ NODE()
2: *root*.$C_i \leftarrow \emptyset \ \forall i \in \{1 \cdots N\}$
3: $\{$*root*.$\pi_i\}_{i=1}^N \leftarrow$ DIFFPLANPARALLEL($\{s_i\}, \{g_i\}, \{\mathcal{H}_j\}$)
4: *root.colls* $\leftarrow$ GETCOLLISIONS(*root*)
5: OPEN $\leftarrow \{$*root*$\}$

Phase 2: Collision Resolution
6: **while** OPEN $\neq \emptyset$ **do**
7: $\quad$ Sort nodes in OPEN by number of collisions
8: $\quad P \leftarrow$ OPEN.POP()
9: $\quad$ **if** $P.colls = \emptyset$ **then**
10: $\quad\quad$ **return** $\{P.\pi_i\}_{i=1}^N$
11: $\quad$ **end if**
12: $\quad p, t, X_i, X_j \leftarrow$ GETCONSTRAINT($P.colls[0]$)
13: $\quad$ **for** $k \in \{i, j\}$ **do**
14: $\quad\quad P' \leftarrow$ COPY(P)
15: $\quad\quad P'.C_k \leftarrow P'.C_k \cup \{\langle X_k, p, t\rangle\}$
16: $\quad\quad P'.\pi_k \leftarrow$ DIFFPLAN($s_k, g_k, \{\mathcal{H}_j\}, P'.C_k$)
17: $\quad\quad P'.colls \leftarrow$ GETCOLLISIONS(P')
18: $\quad\quad$ OPEN $\leftarrow$ OPEN $\cup \{P'\}$
19: $\quad$ **end for**
20: **end while**
21: **return** $\emptyset$

segment for each robot (line 3). If no returned segments are free of robot-human and robot-obstacle collisions, we terminate the algorithm and report failure (not shown in Algorithm 2). Then, we find the list of all (robot-robot) collisions between path segments in line 4 and add the root node to the OPEN list in line 5.

Since diffusion models are known to be computationally intensive, we implement parallelized path generation in line 3 to greatly increase inference speed. Unlike MMD, which creates its root node by generating the path segment of each robot sequentially, we parallelize the process by generating all path segments in one call. This change allows the system to scale to a larger number of robots while maintaining the high-frequency updates required for real-time navigation.

Collision Resolution: If the path segments generated in the root node contain collisions, MRRD initiates a tree search process to resolve them. In each iteration, the node P with the fewest collisions is popped from the OPEN list for expansion. Upon identifying a collision between two robots X_i and X_j at position p and time t (line 12), the search "splits" the problem by generating two child nodes. Each child node P' inherits the existing constraints of the parent, but in line 15, we add a new sphere constraint $\langle X_k, p, t\rangle$ with $k \in \{i, j\}$, preventing robot X_k from occupying a sphere centered at position p around time t. Each child replans its respective robot's path segment using the newly added constraint, along with all existing constraints, generating a new path segment guided around the constrained regions (line 16). The resulting child nodes are then evaluated for new collisions and added back to the OPEN list. This process

continues until a node containing zero collisions is retrieved (line 9), at which point the set of path segments is returned as a valid solution for the current planning horizon.

B. Single-Robot Diffusion

The bottom level of MRRD is the diffusion-based planner that generates a single-robot path segment of length H while taking into account human social norms, obstacle avoidance, and the sphere constraints imposed by Algorithm 2. The primary innovation in this level is the transition from rigid endpoint-pinned conditioning to more flexible temporal goal conditioning. Unlike previous approaches such as Motion Planning Diffusion (MPD) [13] and MMD, which typically perform global planning by fixing both the first and last waypoints to the start and goal positions at each denosing step, MRRD fixes only the first waypoint while allowing the last waypoint to remain flexible. Furthermore, MRRD conditions the diffusion model on the goal position g_i and an urgency term u_i, where u_i represents the total number of waypoints from the start to the goal. Intuitively, an urgency value u_i implies that the current generated path segment advances the agent by approximately H/u_i of the remaining distance toward the goal. Below, we provide more details on how this modification changes our model architecture, denoising, and inference.

1) Model Architecture and Training: Our architecture is based on the one utilized in MPD. Like MPD, we take a noisy path ${}^S\tau_i \sim \mathcal{N}(0, I)$ of length H and denoise it to generate ${}^0\tau_i$. The goal g_i and the urgency term u_i are new inputs to the model in addition to the random path and the denoising step s. We obtain a goal embedding using an MLP and an urgency embedding using an embedding dictionary. These inputs are concatenated and fed through a context layer to form the global context vector, providing key information to the model about the goal and desired speed.

Our diffusion-based planner is trained on human walking data, providing a set of paths $\{\psi_i\}$ of length $T > H$. For each path $\psi_i = \{\psi_i^1, \psi_i^2, \cdots, \psi_i^T\}$, we sample an integer u_i uniformly from $[H, T)$. Then, we use the first H waypoints of ψ_i as the expert path and set the corresponding goal input to $g_i = \psi_i^{u_i}$, effectively training the model to learn the urgency term u_i as a "time-to-goal."

2) Denoising and Guidance: Our diffusion-based planner is called in both line 3 and line 16 of Algorithm 2, where the inputs $\{\mathcal{H}_j\}$ and $P'.C_k$ are used in the guidance steps of the diffusion process. Specifically, the following guidance terms are summed to obtain our complete guidance at each denoising step, $\mathcal{J} = \sum_{o=1}^{O} \lambda_o \mathcal{J}_o$.

Constraint Costs: Like in MMD, robot-robot constraints are represented as spheres in the workspace, and paths with waypoints inside the spheres incur higher costs, resulting in gradient steps pushing the generation outside of the constrained regions. Specifically, we represent a sphere constraint for robot X_i as $\langle X_i, p, t\rangle$, where $p \in \mathcal{W}$ is the center of the sphere, and t is the associated timestep. The corresponding cost function is formulated as

$$\mathcal{J}_{\mathrm{c}}({}^s\tau_i) := \sum_{u=t-t_0}^{t+t_0} \max(r - d({}^s\tau_i^u, p), 0),$$

where t_0 specifies the half-length of the constrained time window, r is the fixed radius of the sphere, and $d({}^s\tau_i^u, p)$ is the distance of robot X_i to p at time u along path ${}^s\tau_i$.

Human Distance Costs: We also penalize robots for being in range of human personal space. We take inspiration from CoBL-Diffusion [24] in designing the cost. The human distance cost at time t for human Y_j is modeled as $P_j(t) = \max(0, R_{out} - D) + \max(0, R_{in} - D)^2$, where R_{in} and R_{out} (set to 0.1 meters and 0.7 meters in our experiments) specify a core radius with squared penalty and a close radius with linear penalty, and $D = d({}^s\tau_i^t, Y_j^t)$ is the distance between robot X_i and human Y_j at time t. To efficiently manage dense crowds, for each time t, we identify the m (set to 3 in our experiments) humans with the highest penalties $P_j(t)$ and sort them in descending order: $P_{(1)}(t) \geq P_{(2)}(t) \geq \cdots \geq P_{(m)}(t)$. The final cost $\mathcal{J}_h({}^s\tau_i)$ is the rank-weighted sum of these top m penalties over the planning horizon H:

$$\mathcal{J}_h({}^s\tau_i) = \sum_{t=0}^{H-1} \sum_{j=1}^{m} (m - j + 1) P_{(j)}(t).$$

Obstacle Costs: This cost (similar to the one in MPD [13]) evaluates how deeply each waypoint of the generated path penetrates into static obstacles, using the environment's Signed Distance Field (SDF). The SDF returns negative values for points inside obstacles and positive values for those outside. The cost is weighted by a precision factor $\kappa = 1/\sigma_{coll}^2$, where σ_{coll} (set to 1.0 in our experiments) controls the penalty strength. The per-timestep obstacle error is the clamped SDF value (with $M_{coll} = 0.02$ in the experiments): $e(t) = \max(M_{coll},\ \mathrm{SDF}({}^s\tau_i^t))$. The total collision cost over the path is:

$$\mathcal{J}_{coll}({}^s\tau_i) = \kappa \sum_{t=0}^{H-1} e(t).$$

After the guidance is applied, we set the first waypoint ${}^s\tau_i^0$ of the generated path to s_i, so that our final path ${}^0\tau_i$ starts from the current position of the robot.

3) Model Inference: During inference, the diffusion model is queried in batches. We vary u_i linearly from a range $[u_{min}, u_{max})$ to generate a batch of B path segments with varying speeds. This diversity ensures that the planner has multiple options, such as moving quickly through a transient opening or slowing down to wait for a pedestrian to pass. In our experiments, we set $u_{max} = 200$ and $u_{min} = \lambda \min(||s_i - g_i||, \delta_{\text{close}})$ with $\delta_{\text{close}} = 4$ and $\lambda = 10$.

Then, among the generated paths, we first discard infeasible ones, namely those that lead to robot-human or robot-obstacle collisions, and then select the path with the smallest u_i, corresponding to the highest speed.

TABLE I: Collision Rates and Goal Scores.

	6 Robots				9 Robots				12 Robots				15 Robots			
Method	R-R↓	R-H↓	SR↑	GR↓	R-R	R-H	SR	GR	R-R	R-H	SR	GR	R-R	R-H	SR	GR
CoBL-Diffusion	.10	.22	.68	1.00	.10	.34	.56	1.80	.24	.38	.44	1.28	.28	.54	.32	1.46
Navi-STAR	.38	.76	.16	.90	.58	.85	.04	1.00	.90	1.00	.00	.92	.98	.96	.00	1.12
Ours (No Parallelization)	**.00**	.04	.96	.36	**.00**	**.04**	**.96**	.36	**.00**	.12	.88	.40	**.00**	.14	.86	.44
Ours	**.00**	**.00**	**1.0**	**.29**	**.00**	**.04**	**.96**	**.24**	**.00**	**.08**	**.92**	**.34**	**.00**	**.08**	**.92**	**.37**

TABLE II: Collision Rates and Goal Scores (Obstacles).

	Empty					Simple					Spheres Sparse					Spheres				
Method	R-R↓	R-H↓	R-O↓	SR↑	GR↓	R-R	R-H	R-O	SR	GR	R-R	R-H	R-O	SR	GR	R-R	R-H	R-O	SR	GR
CoBL-Diffusion	.04	.70	**.00**	.30	1.20	.16	.42	.20	.42	1.09	.10	.64	.48	.18	1.56	.26	.60	.94	.00	2.08
Ours	**.00**	**.04**	**.00**	**.96**	**.50**	**.00**	**.10**	**.18**	**.72**	**.42**	**.00**	**.02**	**.12**	**.88**	**.54**	**.00**	**.10**	**.30**	**.56**	**.52**

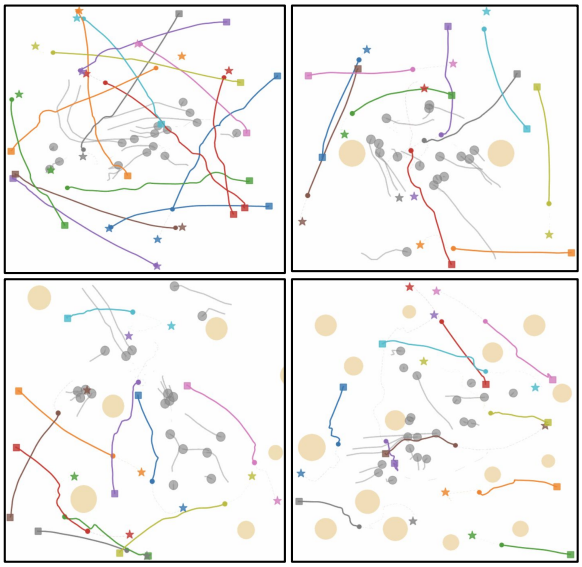


Fig. 1: Snapshot of ongoing runs. Robots (colored circles) move from starts (squares) to goals (stars). Humans are light gray; tan circles indicate obstacles. Layouts clockwise from top-left: Empty, Simple, Spheres, and Spheres Sparse.

TABLE III: Time and Acceleration Profiles.

Robots	Method	$T_P \downarrow$	$T_T \downarrow$	Acc↓
6 Robots	CoBL-Diffusion	25.2	**8.0**	.242
	Navi-STAR	7.1	19.1	.193
	Ours (No Parallel.)	14.5	17.0	**.043**
	Ours	**2.9**	16.3	.046
9 Robots	CoBL-Diffusion	71.7	**8.0**	.402
	Navi-STAR	10.1	20.0	.123
	Ours (No Parallel.)	34.6	16.8	**.047**
	Ours	**3.3**	16.0	.051
12 Robots	CoBL-Diffusion	49.2	**8.0**	.316
	Navi-STAR	13.4	20.0	.207
	Ours (No Parallel.)	33.3	17.6	**.046**
	Ours	**4.3**	17.4	**.046**
15 Robots	CoBL-Diffusion	60.2	**8.0**	.319
	Navi-STAR	17.3	20.0	.179
	Ours (No Parallel.)	38.6	17.6	.047
	Ours	**8.8**	17.6	**.046**

TABLE IV: Time and Acceleration Profiles (Obstacles).

Environment	Method	$T_P \downarrow$	$T_T \downarrow$	Acc↓
Empty	CoBL-Diffusion	52.3	**8.0**	.313
	Ours	**3.7**	17.6	**.049**
Simple	CoBL-Diffusion	49.6	**8.0**	.325
	Ours	**3.5**	17.6	**.051**
Spheres Sparse	CoBL-Diffusion	55.3	**8.0**	.348
	Ours	**4.3**	17.6	**.054**
Spheres	CoBL-Diffusion	54.1	**8.0**	.363
	Ours	**5.1**	17.4	**.065**

V. Experimental Analysis

We evaluate the performance of MRRD in maps with various human crowds, numbers of robots, and static obstacles. Experiments are performed on a machine with an AMD Threadripper 3990X 64-core processor and an NVIDIA RTX 3090Ti. Our implementation is in Python and is based on the official code for MMD [19].

A. Training Setup

The diffusion model is trained on 20,000 20-second paths, discretized into $T = 200$ waypoints from the ETH human pedestrian dataset [28] (making $\Delta t = 0.1$ seconds). Paths are augmented by randomizing their starting location and rotation to provide the model with a variety of human-like movements. Our model sets the planning horizon to $H = 32$ steps and the execution horizon to $K = 16$ steps, which correspond to 3.2 seconds and 1.6 seconds, respectively.

B. Baseline Setups

We compare our method to baselines that capture social awareness: CoBL-Diffusion [24], a diffusion-based planner, and Navi-STAR [12], an RL-based planner with preference learning. Since the SAMARL repository was unavailable, we used Navi-STAR, an earlier single-robot planner from the same group. We also include an ablation planner of MRRD that generates initial paths sequentially, without parallelization. CoBL-Diffusion and Navi-STAR, originally single-robot planners, were adapted for multi-robot planning using

a prioritized-planning approach, where paths are planned sequentially to avoid collisions with higher-priority robots.

C. Crowds in Empty Map

We use a 20 m $\times$ 20 m empty map with 10-20 human crowds from the UCY pedestrian dataset [29]. We tested 50 trials in 6, 9, 12, and 15 robots with randomly initialized start and goal positions (7-12 meters apart) and initial separation from humans (over 1 meter). We evaluated the generated paths using 6 metrics: The *robot-robot collision rate* (R-R) and the *robot-human collision rate* (R-H) are the fractions of trials that experience a robot-robot collision and a robot-human collision, respectively; *goal reaching* (GR) is the average distance from each robot to its goal at the end of the algorithm; the *success rate* is the fraction of trials that experience no collisions of any type; the *planning time* (T_P) is the overall planning time (summed over all rolling horizons for ours); the *travel time* (T_T) is the execution time for the robot paths; and the *average acceleration* (Acc) is a measure of smoothness. The upper left square in Fig. 1 shows an example run of our algorithm in progress.

Tables I and III show the performance of the four algorithms. The robustness of MRRD is most evident in the success rate metric. Although MRRD maintains a high success rate of 0.92 even with 15 robots, the baselines do not scale. CoBL-Diffusion's success rate drops to 0.32, and Navi-STAR fails to complete a single successful trial at the 15-robot scale. While these methods are individually very successful at navigating single robots through dense human crowds, they were not directly designed to handle a multi-robot scenario efficiently. If any of the generated paths results in a collision, the trial is automatically deemed a failure. Furthermore, the data confirms that the "fixed-time" constraint inherent in CoBL-Diffusion's architecture is a critical flaw. Because CoBL-Diffusion is forced to generate a path within a rigid temporal window, it cannot adapt to various crowds, distances, and robot-robot interactions. This results in higher R-R/R-O collision rates and failure to reach goals, evidenced by a 1.46-meter goal-reaching score.

D. Crowds in Obstacle Maps

Next, we test our algorithms on more complex scenarios with static obstacles. Although neither baseline natively supports planning in environments with obstacles, we provided a competitive comparison by augmenting CoBL-Diffusion with an obstacle-avoidance guidance term. We introduce a new metric *robot-obstacle collision rate*, which is the fraction of trials with a robot-obstacle collision. We set the number of robots to 10 and ran 50 trials on the empty map and the three 20 m $\times$ 20 m obstacle maps, shown in Fig. 1.

Tables II and IV present the results. Even as obstacle density increases, MRRD maintains a perfect 0.00 R-R collision rate across all tested scenarios, whereas CoBL-Diffusion reaches a high of 0.26. MRRD also effectively navigates robots around humans, keeping the R-H collision rate at or below 0.12 even in crowded obstacle fields. Table IV reveals that MRRD achieves a 10 to 15 times decrease in planning time, solving complex obstacle fields in just 5.1 seconds compared to CoBL-Diffusion's 54.1 seconds.

VI. Conclusion

In this paper, we introduced a 3-level diffusion-based algorithm MRRD for real-time, multi-robot path planning in human-crowded environments. MRRD integrates a rolling-horizon planner and a multi-robot coordinator with a diffusion model trained on human paths. By generating short, dynamically adaptable path segments rather than long, hard-conditioned paths, our method enables greater dynamic planning than recent diffusion-based works. Our experimental results quantitatively show fewer collisions and greater goal-reaching capability than other baselines trained on human data and input. One limitation of the evaluation is the absence of an ablation baseline isolating the impact of switching from rigid endpoint-pinned conditioning to temporal goal conditioning. While our performance against state-of-the-art models remains robust, including a "fixed-time" variant of MRRD could lead to a more thorough analysis. Another limitation of this work is its reliance on perfect prediction of human trajectories for a windowed horizon. This requirement could impact MRRD's performance in the real-world, due to human movement uncertainty. Future work could soften the requirement for perfect trajectory prediction by integrating human movement prediction modules directly into the diffusion denoising step, or could add invariance to the denoising process for guaranteed path compliance [30].

Acknowledgments

This work was partially supported by the National Science Foundation under grant numbers #2328671 and #2441629.

References

[1] P. D. C. Cheng, M. Indri, F. Maresca, A. Ragazzo, and F. Sibona, "Dynamic path planning in human-shared environments for low-resource mobile agents," in *IEEE International Symposium on Industrial Electronics (ISIE)*, pp. 1–6, 2023.

[2] J. R. Han, H. Thomas, J. Zhang, N. Rhinehart, and T. D. Barfoot, "Deep residual model predictive control for real-world social navigation," *IEEE Robotics and Automation Letters*, vol. 10, no. 4, pp. 4029–4036, 2025.

[3] S. Samavi, J. R. Han, F. Shkurti, and A. P. Schoellig, "Safe and interactive crowd navigation using model predictive control and bilevel optimization," *IEEE Transactions on Robotics*, vol. 41, pp. 801–818, 2025.

[4] A. Wang, C. Mavrogiannis, and A. Steinfeld, "Group-based motion prediction for navigation in crowded environments," in *Conference on Robot Learning (CoRL)*, vol. 164, pp. 871–882, 2022.

[5] Z. Long, X. Zhang, J. Mi, and J. Wang, "Human-risk-aware safe path planning based on reinforcement learning for autonomous mobile robots," *Sensors*, vol. 25, no. 23, 2025.

[6] Y.-J. Mun, M. Itkina, S. Liu, and K. Driggs-Campbell, "Occlusion-aware crowd navigation using people as sensors," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12031–12037, 2023.

[7] C. Mavrogiannis, A. M. Hutchinson, J. Macdonald, P. Alves-Oliveira, and R. A. Knepper, "Effects of distinct robot navigation strategies on human behavior in a crowded environment," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 421–430, 2019.

[8] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, "Decentralized structural-RNN for robot crowd navigation with deep reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3517–3524, 2021.

[9] D. Song, J. Liang, A. Payandeh, A. H. Raj, X. Xiao, and D. Manocha, "Socially aware robot navigation through scoring using vision-language models," *IEEE Robotics and Automation Letters*, vol. 10, no. 1, pp. 508–515, 2025.

[10] S. Narasimhan, A. H. Tan, D. Choi, and G. Nejat, "OLiVia-Nav: An online lifelong vision language approach for mobile robot social navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9130–9137, 2025.

[11] Y. Wang, J. Yu, Y. Kong, L. Sun, C. Liu, J. Wang, and W. Chi, "Socially adaptive path planning based on generative adversarial network," *IEEE Transactions on Intelligent Vehicles*, vol. 10, no. 8, pp. 4267–4278, 2025.

[12] W. Wang, R. Wang, L. Mao, and B.-C. Min, "Socially aware robot navigation with hybrid spatio-temporal graph transformer and preference learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11348–11355, 2023.

[13] J. Carvalho, A. Le, M. Baierl, D. Koert, and J. Peters, "Motion planning diffusion: Learning and planning of robot motions with diffusion models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1916–1923, 2023.

[14] A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, J. F. Fisac, S. Deglurkar, A. D. Dragan, and C. J. Tomlin, "A scalable framework for real-time multi-robot, multi-human collision avoidance," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 936–943, 2019.

[15] R. A. Knepper and D. Rus, "Pedestrian-inspired sampling-based multi-robot collision avoidance," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 94–100, 2012.

[16] S. Farell, C. Li, H. Yu, R. Yoshimitsu, S. Gao, and H. I. Christensen, "Safe human robot navigation in warehouse scenario," in *IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 548–555, 2025.

[17] M. Boldrer, A. Antonucci, P. Bevilacqua, L. Palopoli, and D. Fontanelli, "Multi-agent navigation in human-shared environments: A safe and socially-aware approach," *Robotics and Autonomous Systems*, vol. 149, p. 103979, 2022.

[18] W. Wang, L. Mao, R. Wang, and B.-C. Min, "Multi-robot cooperative socially-aware navigation using multi-agent reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12353–12360, 2023.

[19] Y. Shaoul, I. Mishani, S. Vats, J. Li, and M. Likhachev, "Multi-robot motion planning with diffusion models," in *International Conference on Learning Representations (ICLR)*, 2025.

[20] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 219, pp. 40–66, 2015.

[21] R. Stern, N. R. Sturtevant, A. Felner, S. Koenig, H. Ma, T. T. Walker, J. Li, D. Atzmon, L. Cohen, T. K. S. Kumar, R. Barták, and E. Boyarski, "Multi-agent pathfinding: Definitions, variants, and benchmarks," in *International Symposium on Combinatorial Search (SoCS)*, pp. 151–158, 2019.

[22] D. Silver, "Cooperative pathfinding," in *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 1, p. 117–122, 2005.

[23] G. Sartoretti, J. Kerr, Y. Shi, G. Wagner, T. K. S. Kumar, S. Koenig, and H. Choset, "Primal: Pathfinding via reinforcement and imitation multi-agent learning," *IEEE Robotics and Automation Letters*, vol. 4, pp. 2378–2385, 2018.

[24] K. Mizuta and K. Leung, "Diffusion-based conditional robot planning in dynamic environments using control barrier and lyapunov functions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13801–13808, 2024.

[25] S. Samavi, A. Lem, F. Sato, S. Chen, Q. Gu, K. Yano, A. P. Schoellig, and F. Shkurti, "Safe and interactive crowd navigation with diffusion trajectory predictions," *IEEE Robotics and Automation Letters*, vol. 10, no. 9, pp. 8738–8745, 2025.

[26] R. Chandra, S. Singh, W. Luo, and K. Sycara, "Multi-robot navigation in social mini-games: definitions, taxonomy, and algorithms," *Auton. Robots*, vol. 50, Apr. 2026.

[27] J. Li, A. Tinka, S. Kiesel, J. W. Durham, T. K. S. Kumar, and S. Koenig, "Lifelong multi-agent path finding in large-scale warehouses," in *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp. 1898–1900, 2020.

[28] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 261–268, IEEE, 2009.

[29] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Computer Graphics Forum*, vol. 26, no. 3, pp. 655–664, 2007.

[30] W. Xiao, T.-H. Wang, C. Gan, R. Hasani, M. Lechner, and D. Rus, "Safe planning with diffusion probabilistic models," in *International Conference on Learning Representations (ICLR)*, 2025.