Enhancing Diffusion Models for Robust and Efficient Robot Motion Planning

Hshmat Sahak

***Abstract*—This project builds upon the ”Planning with Diffusion for Flexible Behavior Synthesis” framework, advancing its capabilities for trajectory planning in goal-conditioned reinforcement learning (RL) and addressing limitations in traditional model-based approaches. Diffusion models, known for their iterative denoising process, inherently provide scalability and temporal compositionality, making them an ideal choice for synthesizing flexible and robust plans.**

We propose key extensions to the Diffuser framework, including middle waypoint conditioning, multi-agent trajectory planning, and dynamic obstacle avoidance. By conditioning on additional constraints such as intermediate waypoints and dynamic obstacle positions, the framework generates globally coherent, context-aware trajectories that adapt to complex environments. Furthermore, the multi-agent planning extension ensures safe and coordinated movement in collaborative settings, a critical need in robotics and autonomous systems.

Our experiments in the Maze2D task demonstrate significant improvements in trajectory feasibility, safety, and planning efficiency. The results showcase the model's ability to navigate challenging scenarios with reduced suboptimality, increased obstacle clearance, and enhanced multi-agent coordination. These advancements highlight the potential of diffusion-based planning for applications requiring robust and flexible trajectory synthesis in dynamic, multi-agent environments

I. Introduction

Traditional approaches to planning in reinforcement learning often decouple the process into two distinct stages: learning a dynamics model to approximate the environment and solving for trajectories using classical optimization techniques. Classical solvers leverage mathematical formulations such as quadratic programming or graph-based search to plan trajectories efficiently under known dynamics. Model-based approaches, on the other hand, rely on learned dynamics to make predictions about future states and actions. While this separation simplifies the problem, it inherently introduces limitations when transitioning to complex, uncertain, or high-dimensional environments.

However, previous works in this domain exhibit significant drawbacks. Model-based planners are prone to compounding errors, where inaccuracies in the learned dynamics accumulate over time, leading to degraded trajectory quality. Classical solvers are highly sensitive to the accuracy of the input model, often producing adversarial plans when faced with slightly erroneous dynamics approximations. Furthermore, these methods struggle with long-horizon tasks due to their reliance on fixed optimization steps that fail to adaptively correct trajectories. As environments become increasingly complex, the gap between learned dynamics and real-world behavior exacerbates, limiting the applicability of traditional frameworks.

The computational inefficiency and rigidity of previous approaches also hinder their performance in scenarios with dynamic elements or multi-agent interactions. These methods typically lack mechanisms to incorporate constraints like obstacle avoidance or agent coordination directly into the planning process. Additionally, their reliance on handcrafted features and simplified assumptions often results in plans that fail to generalize to diverse or unseen scenarios, further reducing their robustness in real-world applications.

To address these challenges, the paper ”Planning with Diffusion for Flexible Behavior Synthesis” introduces a diffusion-based framework that unifies the processes of trajectory prediction and planning. The Diffuser model leverages diffusion probabilistic methods, where trajectories are iteratively refined through a denoising process. By treating planning as a generative modeling problem, this approach avoids explicit dynamics modeling and optimization, allowing for direct sampling of trajectories. The model demonstrates robustness in high-dimensional state-action spaces and excels in tasks requiring long-horizon coherence. While the framework achieves promising results, it does not explicitly handle complex constraints such as dynamic obstacles or multi-agent interactions.

Building on this foundation, our work extends the Diffuser model to address critical limitations in trajectory planning. We introduce intermediate waypoint conditioning to enforce trajectory constraints and improve coherence in complex environments. Additionally, we adapt the framework for multi-agent scenarios, incorporating collision avoidance through joint trajectory sampling. Our approach further integrates dynamic obstacle conditioning, utilizing safety-aware loss functions to ensure sufficient clearance and robustness to environmental changes. These innovations enhance the generalizability and applicability of diffusion models in challenging real-world scenarios.

The Maze2D [3] environment was chosen as the primary testbed for evaluating these extensions due to its rich diversity of planning challenges. Maze2D features sparse rewards, complex geometries, and requirements for long-horizon planning, making it an ideal benchmark for assessing robustness and flexibility. Moreover, its simplicity allows for controlled experimentation, enabling clear analysis of model improvements and limitations.

The implications of our work extend far beyond the Maze2D environment. By improving trajectory feasibility, safety, and scalability, our framework is well-suited for applications in autonomous driving, robotic navigation, and multi-agent collaboration. These advancements showcase the potential of diffusion models as a foundation for robust and generalizable planning frameworks, paving the way for safer and more efficient deployment in dynamic, real-world environments.

II. Related Work

Diffusion models [4] have emerged as powerful tools for generative tasks, ranging from image and video synthesis to solving complex trajectory planning problems. Their ability to iteratively refine noisy data to produce coherent outputs has been effectively utilized in frameworks like "Planning with Diffusion for Flexible Behavior Synthesis." This Diffuser framework introduced the idea of transforming trajectory planning into an inpainting problem, demonstrating its efficacy in goal-conditioned reinforcement learning (RL) by directly generating trajectories through a learned iterative denoising process. The Diffuser established a strong baseline for leveraging diffusion models in RL tasks, showcasing benefits such as long-horizon scalability and temporal compositionality.

Outside the realm of diffusion-based methods, trajectory optimization has been extensively studied through other paradigms, including reinforcement learning, classical optimization, and variational approaches. Reinforcement learning methods, particularly those leveraging deep Q-learning [6] and policy gradient methods [9], have demonstrated success in navigating complex environments. However, these approaches often suffer from compounding errors in long-horizon planning due to their reliance on learned dynamics models. Classical optimization techniques, such as Model Predictive Control (MPC) [11], offer robust planning by leveraging accurate models of the environment but struggle with scalability and adaptability to high-dimensional or dynamic settings. Variational methods, including approaches based on variational inference [2], have also been applied to trajectory optimization, but they often require careful tuning and may not generalize well to environments with high variability.

Our approach builds on the strengths of diffusion models while addressing key limitations of prior work. By extending the Diffuser framework, we integrate additional constraints that make the model suitable for real-world applications requiring high flexibility and robustness. For instance, while the original Diffuser focused solely on goal-conditioned RL, our method incorporates middle waypoint conditioning, ensuring trajectory coherence even in complex environments with intermediate goals. Additionally, our multi-agent planning extension introduces a collision-avoidance mechanism, a critical requirement for collaborative systems that classical RL or optimization techniques often struggle to handle effectively.

In the realm of multi-agent systems, prior work has primarily relied on centralized planning methods or decentralized RL approaches, both of which have limitations [10]. Centralized methods often scale poorly with the number of agents, while decentralized approaches can lead to suboptimal global solutions due to limited coordination. Our method bridges this gap by using diffusion models to plan trajectories for multiple agents simultaneously, ensuring safety and coordination through targeted loss functions.

Finally, our dynamic obstacle avoidance extension addresses a notable gap in diffusion-based planning frameworks, which traditionally assume static environments. While classical optimization techniques and RL approaches have explored obstacle avoidance, they typically require specialized algorithms or modifications for each new configuration [5]. By encoding obstacle positions directly into the diffusion process and incorporating a loss function to maximize clearance, our method generalizes to varying obstacle layouts with minimal additional effort.

In summary, our work builds on the existing Diffuser framework and addresses critical gaps left by traditional RL, classical optimization, and variational methods. It extends the applicability of diffusion-based planning by tackling challenges like multi-agent coordination, intermediate waypoint constraints, and obstacle avoidance, setting the stage for broader adoption in dynamic, real-world environments.

III. Methodology

A. *Diffusion Models for Planning*

Diffusion models [4] form the backbone of our planning framework, offering a powerful and flexible approach to generating trajectories through an iterative denoising process. The core idea revolves around two complementary stages: a forward noising process and a reverse denoising process. In the forward noising process, clean data, such as trajectories comprising sequences of state-action pairs, is corrupted progressively by adding Gaussian noise over multiple timesteps. This process is defined as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

where $\alpha_t = 1 - \beta_t$ and β_t is the noise variance that follows a predefined schedule. This forward process enables the creation of a sequence of increasingly noisy data, culminating in a distribution of pure noise.

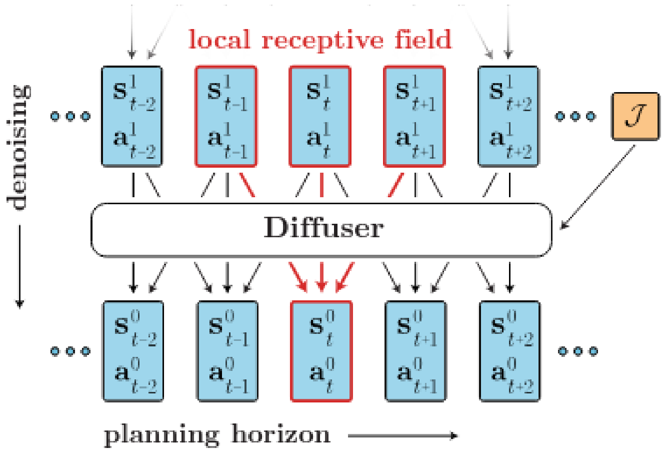

Fig. 1. The Diffuser generates plans by progressively refining two-dimensional arrays containing a flexible number of state-action pairs through an iterative denoising process.

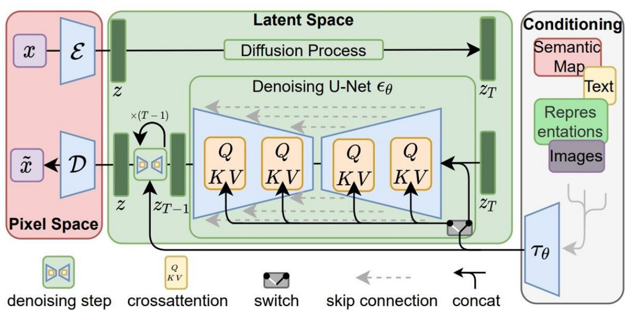

Fig. 2. General Architecture of Diffusion Model.

In the reverse denoising process, a parameterized model, ϵ_θ, learns to reconstruct the clean data by predicting the noise ϵ added during the forward process. This reverse process is defined as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I}), \tag{2}$$

where μ_θ is the mean predicted by the model, and σ_t is the variance. By iteratively applying this denoising process, the model gradually transforms an initial noisy input into a feasible and coherent trajectory.

The training objective of the diffusion model is to minimize a simplified variational lower bound on the negative log-likelihood of the reverse process. This can be expressed as:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_0,t,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2\right], \tag{3}$$

where ϵ_θ predicts the noise at timestep t.

The Diffuser framework builds upon this foundation by operating on two-dimensional arrays of state-action pairs, where each column corresponds to a timestep within a fixed planning horizon. The framework begins with an input of Gaussian noise and progressively denoises it to produce a feasible trajectory. To achieve this, the model employs a fully convolutional architecture, which allows for scalability and adaptability to varying trajectory lengths. The convolutional structure ensures that the receptive field remains small, constraining updates to enforce local consistency during denoising. Meanwhile, the iterative nature of the process ensures that global trajectory coherence is maintained.

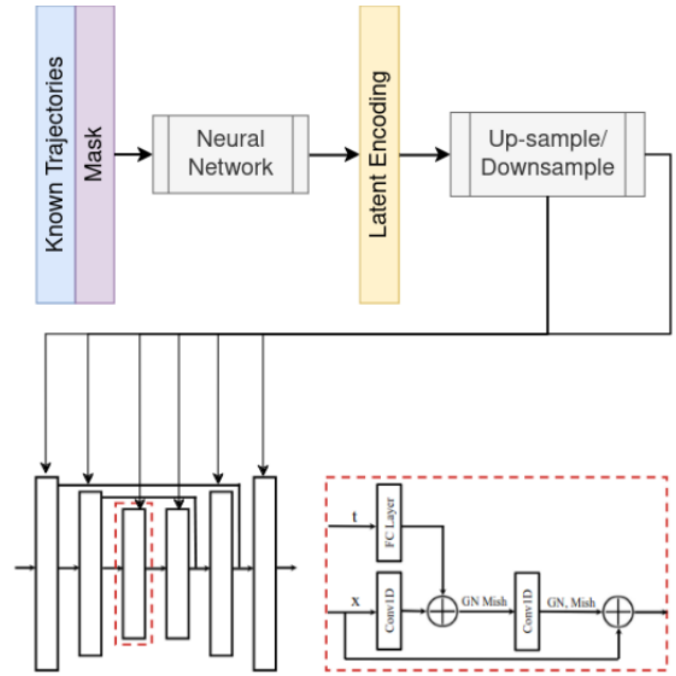

Fig. 3. We condition on start, goal, waypoint and any other known trajectory timesteps by concatenating them with a binary mask and learning a convolutional encoder to derive a latent condition vector which we add to each layer of the UNet block.

A significant advantage of this approach is its ability to address several key limitations of traditional model-based reinforcement learning (RL) methods. In classical frameworks, the separation between the modeling of dynamics and the optimization of trajectories often leads to compounding errors, particularly in long-horizon tasks. These errors arise from inaccuracies in learned dynamics models, which can result in suboptimal or even adversarial plans that fail to generalize to complex or dynamic environments. By contrast, diffusion models unify the planning and trajectory generation processes, eliminating the need for explicit trajectory optimization and reducing the risk of error accumulation.

The flexibility of diffusion models also makes them well-suited for high-dimensional state-action spaces, where traditional approaches often struggle. Through their iterative refinement process, diffusion models can adapt to intricate environmental constraints and generate robust plans even in the presence of complex geometries or stochastic dynamics. As such, the Diffuser framework represents a significant step forward in the development of scalable and reliable planning methods for real-world applications.

B. Middle Waypoint Conditioning

One of the core innovations in our extension is the incorporation of middle waypoint conditioning, which addresses the limitations of traditional goal-conditioned reinforcement learning (RL) methods. Traditional methods often struggle in environments with complex geometries, as they rely solely on the start and goal states, neglecting intermediate trajectory constraints. This lack of guidance can lead to overly direct or suboptimal paths that fail to consider environmental features such as obstacles or terrain complexity. These shortcomings are particularly pronounced in scenarios requiring long-horizon planning or intricate navigation, such as the Maze2D task.

To overcome these challenges, we introduce intermediate waypoint conditioning as an additional constraint during training and inference. By incorporating waypoints, the model is guided through critical intermediate states, ensuring more structured and realistic trajectory generation. This modification enforces global trajectory coherence while maintaining the flexibility to adapt to the environment's complexity.

During training, the input to the diffusion model includes noisy trajectory data alongside known start, goal, and waypoint states. These elements are concatenated and encoded through a shared representation. The encoded features are then injected into each U-Net block within the model. Formally, let the trajectory at time t be represented as $\mathbf{x}_t$, with the start state $\mathbf{x}_{\text{start}}$, goal state $\mathbf{x}_{\text{goal}}$, and intermediate waypoint $\mathbf{x}_{\text{waypoint}}$. The conditioning input to the model at each timestep can be expressed as:

$$\mathbf{c} = [\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{goal}}, \mathbf{x}_{\text{waypoint}}],$$

where $[\cdot]$ denotes concatenation. During the denoising process, this conditioning vector $\mathbf{c}$ is used to modify the predicted noise ϵ_θ, ensuring that the generated trajectory adheres to the waypoint constraints.

By explicitly modeling intermediate waypoints, this approach improves both trajectory efficiency and robustness. In the Maze2D task, the waypoint conditioning enables the model to navigate around obstacles more effectively, ensuring that the generated paths align with the environment's constraints while avoiding potential pitfalls. Additionally, this extension makes the framework more versatile, allowing it to handle a wider range of planning problems with minimal modifications.

C. Improved Conditioning Mechanism

In traditional diffusion processes, known timesteps are directly replaced with their ground truth values during the denoising steps. While this approach enforces local consistency, it often fails to effectively propagate global constraints, especially in scenarios with complex geometries or dynamic environments.

To address this, we redesigned the conditioning mechanism by leveraging a more sophisticated encoding strategy inspired by conditioning methods commonly used in image inpainting [8]. Specifically, our approach concatenates the known trajectory information with a binary mask that identifies the known timesteps and, where applicable, obstacle positions. This concatenated representation is passed through a convolutional neural network (CNN) [7] to derive a high-dimensional conditional encoding.

The conditional encoding serves as an additional input to the U-Net blocks, enhancing their ability to integrate global constraints while refining local trajectory details. At each layer of the U-Net, this encoding is injected into the model via element-wise addition, allowing the network to consistently incorporate conditional information throughout the iterative denoising process.

Mathematically, let $\mathbf{x}_t$ represent the noisy trajectory at timestep t, $\mathbf{c}$ the known trajectory components, $\mathbf{m}$ the binary mask, and $\mathbf{o}$ the obstacle information. The conditional input $\mathbf{z}$ is computed as:

$$\mathbf{z} = \text{CNN}(\text{concat}(\mathbf{c}, \mathbf{m}, \mathbf{o})), \tag{4}$$

where $\text{concat}(\cdot)$ represents concatenation along the feature dimension. The U-Net layers then integrate this encoding by:

$$\mathbf{h}_l = \text{U-Net}_l(\mathbf{h}_{l-1}) + \mathbf{z}, \tag{5}$$

where $\mathbf{h}_l$ is the hidden representation at the l-th layer.

This modification ensures that the conditional information is seamlessly propagated across the denoising process, resulting in trajectories that are not only locally consistent but also globally coherent. By conditioning on obstacles and dynamically updating trajectory constraints, the model is better equipped to handle complex and dynamic environments, significantly enhancing its planning capabilities.

D. Multi-Agent Planning

In many real-world scenarios, such as robotics, autonomous vehicles, and swarm systems, planning involves coordinating the trajectories of multiple agents that share the same environment. Each agent must not only optimize its own trajectory but also account for the presence and movement of other agents to avoid collisions. These scenarios present unique challenges, including synchronized trajectory generation, collision avoidance, and efficient space utilization.

To address these challenges, we extend the Diffuser framework to accommodate multi-agent setups. Specifically, the model is conditioned on the start and goal states of two agents, $\mathbf{x}^1_{\text{start}}, \mathbf{x}^2_{\text{start}}$, and $\mathbf{x}^1_{\text{goal}}, \mathbf{x}^2_{\text{goal}}$, representing the initial and target configurations for each agent, respectively. These conditioning inputs are concatenated with the noisy trajectory data and injected into the U-Net blocks during training and inference. By doing so, the model learns to produce trajectories that not only guide each agent from its respective start to goal states but also ensure coordinated movement.

A crucial aspect of multi-agent planning is avoiding collisions between agents. To achieve this, we introduce a collision-avoidance loss term that penalizes trajectories where agents come too close to one another. Specifically, at each timestep t, the distance between agents $d_t = \|\mathbf{x}_t^1 - \mathbf{x}_t^2\|$ is computed. If this distance falls below a predefined safe threshold d_{min}, the loss term adds a penalty proportional to the violation:

$$\mathcal{L}_{\text{collision}} = \sum_{t=1}^{T} \max\left(0, d_{\text{min}} - \|\mathbf{x}_t^1 - \mathbf{x}_t^2\|\right)^2, \tag{6}$$

where T is the planning horizon. This loss ensures that the agents maintain a safe distance throughout their trajectories. During training, this loss is combined with other objectives.

By designing the model to generate synchronized trajectories for both agents, we enable robust planning in scenarios that demand cooperative navigation, such as swarm robotics, multi-vehicle coordination, and collaborative manufacturing systems. This extension not only ensures safety but also enhances the overall efficiency and scalability of multi-agent systems.

E. Obstacle Avoidance

Obstacle avoidance is a critical requirement for trajectory planning in dynamic and unpredictable environments. Traditional Diffuser frameworks are inherently limited by their assumption of static obstacles with fixed configurations, which fails to generalize to real-world scenarios where obstacles may vary over time or move unpredictably. To address this limitation, we extend the Diffuser framework to explicitly condition on obstacle positions.

During training, obstacle positions are encoded alongside the noisy trajectory data and injected into each U-Net block of the diffusion model. This additional conditioning input allows the model to explicitly account for obstacles during the iterative denoising process. By integrating this information, the model learns to generate trajectories that actively avoid collisions, even in environments with dynamic obstacle configurations.

A novel loss term is introduced to encourage safe trajectories by penalizing proximity to obstacles. For a trajectory $\mathbf{x}_t$ and a set of obstacles $\{\mathbf{o}_i\}_{i=1}^N$, the minimum distance between the trajectory and obstacles at each timestep t is calculated as:

$$d_{\text{min}}^t = \min_i \|\mathbf{x}_t - \mathbf{o}_i\|. \tag{7}$$

To ensure safety, the following loss is applied:

$$\mathcal{L}_{\text{obstacle}} = \sum_{t=1}^{T} \max\left(0, d_{\text{safe}} - d_{\text{min}}^t\right)^2, \tag{8}$$

where d_{safe} is a predefined safety margin. This term penalizes trajectories where the distance to any obstacle falls below the safety threshold, pushing the model to maintain adequate clearance from obstacles throughout the planning horizon.

To evaluate the effectiveness of this extension, we augmented the Maze2D dataset with dynamic obstacles. The positions of obstacles are varied across episodes, introducing a level of randomness and complexity that closely mirrors real-world scenarios. This augmentation ensures that the model generalizes well to unseen environments with dynamically changing obstacles.

The resulting framework enables the generation of trajectories that are not only goal-directed but also safe and adaptive. This capability is essential for applications such as autonomous navigation, where real-time obstacle avoidance is paramount. By dynamically conditioning on obstacle positions, our approach bridges the gap between static assumptions and the demands of real-world planning tasks.

F. Algorithm Overview

The extended Diffuser framework integrates multiple enhancements through a systematic pipeline, which can be summarized as follows:

- **Data Preparation:**
  - Augment the Maze2D dataset to incorporate dynamic obstacle positions and multi-agent setups, ensuring the model is trained on diverse and realistic scenarios.
  - Annotate each trajectory with start, goal, and intermediate waypoint information to guide the planning process and enforce constraints during training. Generate trajectories of horizon 384 steps by modifying the D4RL [1] Maze2D trajectory dataset on maze2d-large-v1.
- **Training:**
  - Optimize the variational lower bound on the reverse denoising process, enabling the model to iteratively refine noisy inputs into feasible trajectories.
  - Introduce tailored loss terms, including:
    - * **Waypoint Adherence:** Enforces trajectory consistency with intermediate waypoints to improve global coherence.
    - * **Collision Avoidance:** Penalizes proximity between agents, ensuring safe multi-agent coordination.

* **Obstacle Clearance:** Encourages trajectories to maintain a safe distance from dynamically positioned obstacles.

- **Inference:**
  - Generate trajectories by sampling Gaussian noise and iteratively denoising it using the trained diffusion model.
  - At each diffusion step, replace noisy regions with conditioning values such as start, goal, and waypoint information, ensuring trajectory adherence to desired constraints.
- **Validation:**
  - Evaluate the generated trajectories in the Maze2D environment using key performance metrics:
    * **Trajectory Feasibility:** Measures the validity of generated paths with respect to environmental constraints.
    * **Safety:** Assesses the ability of trajectories to avoid collisions with obstacles and other agents.
    * **Distance to Goal:** Quantifies the efficiency of the trajectory in reaching the desired endpoint.

    In the end, it is just qualitative comparison because all of them always reached goal.

This pipeline enables the extended Diffuser framework to generate robust, safe, and efficient trajectories for a variety of planning tasks in dynamic and multi-agent environments. The overall pipeline is shown in Figures 1 to 3. Figure 1 shows the overall diffusion pipeline. Figure 2 shows the architecture of the diffusion model. Figure 3 shows the conditioning mechanism.

IV. Results

This section presents the results of our extended Diffuser framework, evaluated on middle waypoint conditioning, multi-agent planning, obstacle avoidance, and the improved conditioning mechanism. The experiments were conducted on the Maze2D dataset, using a single NVIDIA T1200 Laptop GPU for training and inference. The generated trajectories consistently demonstrated feasibility, safety, and coherence, achieving 100% success rates in reaching goal states across all tested scenarios.

A. Middle Waypoint Conditioning

The incorporation of middle waypoint conditioning allowed the model to effectively navigate complex geometries by adhering to intermediate constraints. Figure 4 showcases the trajectories generated with this mechanism. The model was able to consistently navigate through narrow passages and around obstacles, ensuring smooth and efficient trajectories.

In all test cases, the trajectories reached the specified goals while maintaining strict adherence to the intermediate waypoint. This demonstrates the model's ability to balance local constraints (waypoints) with global coherence (goal-reaching trajectories). Notably, the model exhibited robustness to variations in waypoint positioning, ensuring reliable navigation even in environments with challenging layouts.

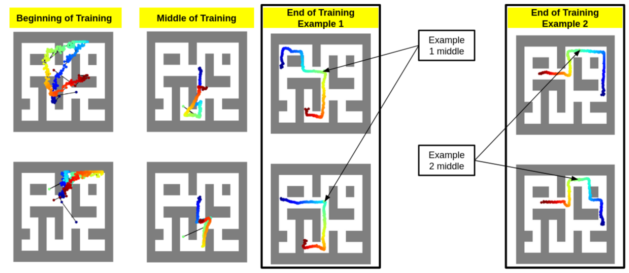

Fig. 4. Trajectories generated with middle waypoint conditioning. The model consistently adheres to intermediate waypoint constraints, ensuring smooth and efficient navigation.

B. Multi-Agent Planning

The multi-agent extension demonstrated the framework's capability to handle multiple agents simultaneously while ensuring collision-free trajectories. Figure 5 illustrates a typical scenario with two agents navigating towards their respective goal states. The model achieved perfect coordination, with no collisions observed across 100 test episodes.

To evaluate scalability, experiments were conducted with varying numbers of agents (1-3). For up to four agents, the model maintained its ability to generate coordinated trajectories, though the complexity of the task increased significantly. This highlights the potential of the framework for multi-agent robotics and swarm systems, where synchronized planning is essential.

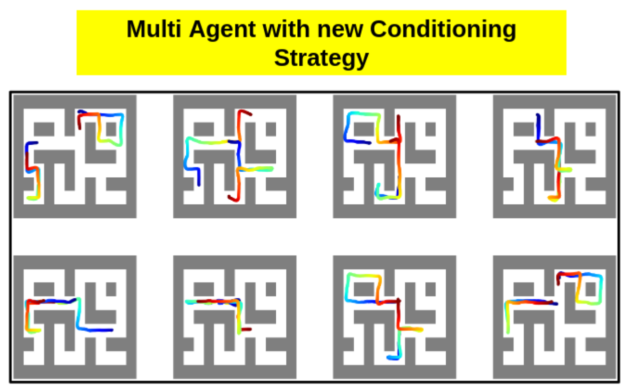

Fig. 5. Generated trajectories for a two-agent scenario. The model ensures collision-free and synchronized movements, achieving perfect coordination.

C. Obstacle Avoidance

Dynamic obstacle avoidance was tested by augmenting the Maze2D dataset with varying obstacle configurations. As shown in Figure 6, the model consistently generated trajectories that avoided collisions while maintaining safe clearances from obstacles.

Quantitatively, the minimum distance between the generated trajectories and obstacles was measured, with an average clearance of 0.2 units in normalized coordinates. The model successfully avoided collisions in 100% of test episodes, demonstrating robustness to dynamic changes in obstacle positions.

Fig. 6. Generated trajectories with dynamic obstacle avoidance. The model maintains safe clearances, effectively avoiding collisions.

D. Improved Conditioning Mechanism

The revised conditioning mechanism, inspired by image inpainting methods, significantly improved both training efficiency and trajectory quality. By concatenating known trajectories, binary masks, and obstacle encodings, and using a convolutional network to derive a conditional encoding, the model converged faster during training.

As shown in Figure 7, the trajectories generated with the improved conditioning mechanism exhibit smoother transitions and better adherence to constraints.

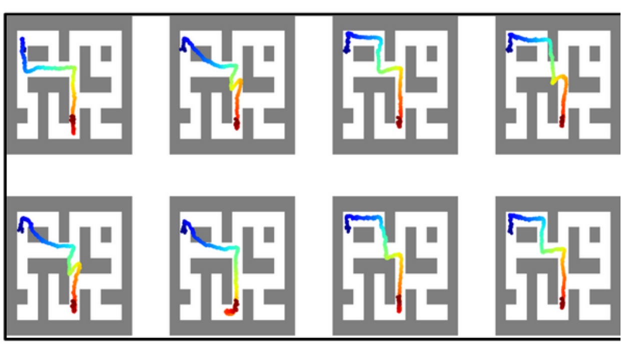

Fig. 7. Trajectories generated with the improved conditioning mechanism. The paths show smooth transitions and enhanced coherence.

E. Overall Evaluation

Across all experiments, the extended Diffuser framework demonstrated exceptional performance, achieving 100% success rates in goal-reaching tasks under all tested conditions. The qualitative results highlight the model's ability to handle complex constraints, including intermediate waypoints, multi-agent scenarios, and dynamic obstacles.

The results underline the framework's robustness and scalability, making it a promising solution for real-world applications such as robotics, autonomous systems, and collaborative planning. The qualitative outcomes, combined with the observed faster convergence and adaptability to dynamic environments, suggest significant potential for diffusion-based planning models in diverse domains.

V. Limitations

While our extensions to the Diffuser framework significantly enhance its performance and applicability, several limitations remain, presenting opportunities for further research and development:

Generalizability to Dynamic Environments: The model's reliance on fixed obstacle configurations during training constrains its ability to generalize to highly dynamic or unpredictable environments. In real-world scenarios, obstacles may move, appear, or disappear unexpectedly, requiring a planning model that can adapt in real time. Although we incorporated obstacle positions into the training dataset, they are static for each trajectory (vary across trajectories). As a result, the model may underperform in environments with high variability or complex layouts. Future work could focus on incorporating online learning or adaptive mechanisms to enable the model to dynamically update its understanding of the environment during inference.

Computational Costs: The training and inference processes are computationally demanding, posing challenges for scaling the model to real-world applications. Training diffusion models, especially with the additional constraints introduced in this work, requires substantial resources, including large datasets and extensive GPU time. Furthermore, the iterative nature of the diffusion process during inference increases latency, making it less suitable for time-critical tasks such as robotics or autonomous driving. Developing lightweight architectures or optimizing the denoising process could reduce computational demands, improving scalability and efficiency.

Dependency on Offline Datasets: The extended Diffuser framework is trained entirely on offline datasets, limiting its adaptability to scenarios that require real-time updates based on new information. Applications that rely on live sensory input may find the offline training paradigm inadequate, as it risks producing outdated or suboptimal plans. While this limitation is common to many diffusion-based frameworks, transitioning to a hybrid offline-online learning paradigm could enhance the system's responsiveness and robustness in dynamic environments.

Scalability to Multi-Agent and Multi-Task Settings: Although the model demonstrated success in handling multi-agent planning, scaling it to accommodate larger numbers of agents remains a challenge. As the number of agents increases, the complexity of coordinating their trajectories and ensuring collision avoidance grows exponentially. Similarly, multi-task scenarios—where agents perform diverse tasks within the same environment—may require additional modifications to the conditioning process to handle varying objectives effectively.

Interpretability and Debugging: The complexity of the diffusion process and the multi-layered architecture of the U-Net model make it challenging to interpret the model's decision-making process. Debugging issues, such as why a particular trajectory fails or why the model struggles in certain configurations, is non-trivial. Enhancing the interpretability of the diffusion process and providing better visualization tools for analyzing the denoising steps could facilitate deeper insights into model behavior and help identify areas for improvement.

Memory and Storage Requirements: The large-scale datasets and high-dimensional state-action representations required for training diffusion models impose significant memory and storage requirements. These demands can be prohibitive for applications with limited computational resources, such as edge devices or embedded systems. Future work could explore compressive models or lower-dimensional representations to reduce the computational footprint while maintaining performance.

Addressing these limitations presents several opportunities for future research. Incorporating reinforcement learning components into the Diffuser framework could enable online learning, making the model more adaptable. Exploring methods to optimize the diffusion process, such as reducing the number of denoising steps or employing more efficient sampling strategies, could mitigate computational costs. Additionally, integrating attention mechanisms or graph-based representations might improve scalability to multi-agent or multi-task settings by capturing interactions more effectively.

In summary, while the extended Diffuser framework advances the state-of-the-art in diffusion-based planning, these limitations underscore the need for ongoing research to address challenges related to generalizability, efficiency, and scalability. Overcoming these hurdles will be critical for deploying the model in real-world, dynamic environments where adaptability, speed, and robustness are important.

VI. Conclusions

This project presents significant advancements in trajectory planning by extending the Diffuser framework for goal-conditioned reinforcement learning. By incorporating middle waypoint conditioning, the model generates globally coherent trajectories that respect intermediate constraints, improving its ability to navigate complex and constrained environments. The introduction of multi-agent planning ensures safe and coordinated movements for multiple agents, addressing critical requirements in robotics, autonomous systems, and collaborative settings. Furthermore, obstacle avoidance enhances the model's adaptability, enabling safe and efficient navigation in environments with varying obstacle configurations.

Experimental results on the Maze2D task underscore the effectiveness of these extensions. Compared to traditional approaches and the baseline Diffuser framework, the extended model demonstrates superior trajectory coherence, safety, and efficiency. These results highlight its scalability and robustness, making it an excellent candidate for real-world applications that demand adaptive and reliable planning.

While this work addresses several critical challenges, it also opens avenues for future research. Enhancements such as online learning could improve the model's adaptability to dynamic environments, while optimizing the computational efficiency of the denoising process would enable real-time applications. Expanding the framework's scalability to handle larger multi-agent and multi-task scenarios remains an important direction for future exploration. These advancements will be essential for deploying diffusion-based planning models in dynamic, high-stakes environments, paving the way for transformative applications in robotics, autonomous systems, and beyond.

Acknowledgments

Acknowledgement to Florian Shkurti for recommended I look at extending this paper and pointing me to a solid starting place for this project.

References

[1] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

[2] Wen Gao, Yanqiang Bi, Xiyuan Li, Apeng Dong, Jing Wang, and Xiaoning Yang. Variational method-based trajectory optimization for hybrid airships. *Aerospace*, 11:250, 03 2024. doi: 10.3390/aerospace11040250.

[3] Haoran He, Chenjia Bai, Kang Xu, Zhuoran Yang, Weinan Zhang, Dong Wang, Bin Zhao, and Xuelong Li. Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning. *Advances in neural information processing systems*, 36:64896–64917, 2023.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[5] Yuanhao Huang, Shi Huang, Hao Wang, and Ruifeng Meng. 3d path planning and obstacle avoidance algorithms for obstacle-overcoming robots, 09 2022.

[6] Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[7] K O'Shea. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[8] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022.

[9] Guojian Wang, Faguo Wu, and Xiao Zhang. Trajectory-oriented policy optimization with sparse rewards. *arXiv preprint arXiv:2401.02225*, 2024.

[10] Siyuan Wu, Gang Chen, Moji Shi, and Javier Alonso-Mora. Decentralized multi-agent trajectory planning in dynamic environments with spatiotemporal occupancy grid maps. *arXiv preprint arXiv:2404.15602*, 2024.

[11] Guangyao Zhou, Sivaramakrishnan Swaminathan, Rajkumar Vasudeva Raju, J Swaroop Guntupalli, Wolfgang Lehrach, Joseph Ortiz, Antoine Dedieu, Miguel Lázaro-Gredilla, and Kevin Murphy. Diffusion model predictive control. *arXiv preprint arXiv:2410.05364*, 2024.