CS5478 - Motion Planning for 2D Navigation Final Report

Yuwei ZENG[*]
Department of Computer Science
National University of Singapore
inazurich@gmail.com

Shanhe ZHAO[†]
Department of Computer Science
National University of Singapore
e1275103@u.nus.edu

Abstract

In the realm of motion planning for 2D navigation, this research explores the integration of classic planners and a diffusion-based generative model. The demand for efficient and adaptive motion planning is driven by the increasing use of autonomous navigation and robotics applications. Historically, motion planning has seen advancements from optimization-based approaches to sampling-based techniques. More recently, learning-based methods, particularly diffusion models, have shown promise in capturing complex multi-modal distributions.

The main objective of this study is to frame 2D path planning as a planning-as-inference problem using diffusion models. The research aims to leverage the strengths of classic planners, such as RRT, RRT-Connect, RRT*, APF, and CHOMP, to provide successful paths for training the diffusion-based planner. The study explores the performance of each planner using comprehensive evaluation metrics, including completeness, optimality, and time complexity.

The research contributes to the field by showcasing the advantages of diffusion-based planners, such as fast computation and test-time composability. The diffusion model's ability to capture multi-modal solutions is highlighted, addressing a challenge often faced by other learning-based planners. Additionally, the study emphasizes the importance of task-conditioned guidance during inference, improving success rates and producing more reliable and optimal paths.

In conclusion, this work presents a unified approach that combines classic planners with diffusion-based models, demonstrating their complementary strengths. The findings suggest that diffusion-based planners offer a promising direction for 2D path planning, particularly when guided by task-specific constraints.

1 Introduction

With the increasing demand for autonomous navigation and robotics applications in various domains, motion planning for 2D navigation has emerged as a critical aspect of ensuring safe and efficient movement. Over the years, researchers and engineers have explored diverse techniques to address the complexities associated with navigating environments, avoiding obstacles, and reaching desired destinations. The field has witnessed significant advancements, driven by the need for more sophisticated and adaptive planning methods.

Historically, motion planning problems as one of the earliest robotics problems with methods including A* search[1], visibility graph[2]. Later the dominated pilgrim shifted towards computational

[*] *Yuwei ZENG – A0267614Y*
[†] *Shanhe ZHAO – A0287084U*

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

geometry and optimisation. A variety of optimization-based planners and algorithms were introduced ranging from potential field[3] to trajectory optimization approaches[4] and recent methods leveraging upon natural gradient[5] or stochastic update rules[6]. Optimization-based approaches benefits from being able to integrate desired properties such as smoothness as costs to be optimized. Nevertheless, they depend on a good initialization and can get trapped in local minima due to the non-convexity of complex problems. Specifically, they commonly require a good initialization prior and well-tuned hyperparameters to work well. As a parallel track, to address such shortcomings, the sampling-based approaches were developed. Some remains the most common planners used today, including Probabilistic Roadmap Method (PRM)[7], Expansive-Spaces Tree planner (EST)[8], Rapidly-exploring Random Tree (RRT) [9] [10] with its variants.

Recently, learning-based methods[11][12][13][14] have shown promising potential to improve classical motion planning by learning useful prior from the experience of previously successful plans. In particular, sampling from learned prior distributions, conditioned on contexts such as start/goal configurations and environmental variables, can provide good initializations for motion planners.

Among different manners to learn an efficient prior from data in the realm of machine learning, diffusion-based generative models showed powerful results in multi-modality and high-dimensional data modelling. It demonstrated superior performance over previous generative models such as GAN[15], VAE[16] and flow-based[17] models on multiple tasks such as image generation[18], material discovery[19]. The nature of generative model fits our target to learn an effective prior in a data-driven manner, In addition to this, diffusion model yields an attractive property as any other energy-based models that is test time composable[20], which means during the inference time, the learnt prior can be further combined with new environment or task constraints and make it especially promising for better generalization.

Based on such rationality, in this work, we set to study the 2D path planning problem with diffusion model framed as a planning-as-inference problem. Our work also presents a comprehensive exploration and integration of various common planners from both sampling-based and optimziation-based pilgrims into a unified python package. It amalgamates the strengths of different planners, such as the APF planner for simplicity, CHOMP for trajectory optimization, and RRT and its variants for efficient exploration. Such classic planners serve as demonstrators to provide successful paths for the diffusion-based planner training, and are compared against each other thoroughly with designed evaluation metrics and analysis on the strengths and limitations of each planner. The code for our implementation and evaluation can be accessed at `https://github.com/friolero/cs5478_path_planning`.

2 Classic Planners Implemented

For the sampling-based planner, we selected Rapidly-exploring Random Trees (RRT)[21] as one of the most commonly used algorithms across different tasks due to its simple yet effective performance. RRT variants were implemented correspondingly as RRT, RRT-Connect[22] and RRT*[23]. We also selected 2 common optimization-based planners based on Artificial Potential Field (APF)[24] and Covariant Hamiltonian Optimization for Motion Planning (CHOMP)[5]. In this section, we briefly go through the main idea for each planner.

2.1 Sampling-based Planners

RRT Introduced by [21], RRT revolutionized the motion planning landscape by presenting a probabilistically complete approach to explore high-dimensional spaces efficiently. RRT incrementally builds a tree structure, biasing exploration towards uncharted regions. Its simplicity and effectiveness have made it a cornerstone in many robotics applications.

RRT-Connect A natural extension of RRT, RRT-Connect focuses on connecting two trees in configuration space. Conceived to enhance RRT's efficiency in solving complex problems, RRT-Connect excels in scenarios where the start and goal configurations may belong to different connected components.

RRT* Enhancing upon the RRT framework, RRT* introduces a re-wiring mechanism that optimizes the tree structure. Developed by Karaman and Frazzoli in 2011, RRT* guarantees asymptotic optimality, providing a more refined and efficient solution compared to its predecessors.

2.2 Optimization-based Planners

APF Originating from the field of control theory, the APF method leverages the concept of virtual forces to guide a robot through its environment. By treating obstacles as repulsive forces and the goal as an attractive force, APF is intuitive and effective in simple environments.

CHOMP CHOMP approaches motion planning as an optimization problem, integrating a cost function into the trajectory generation process. Developed by [25], CHOMP excels in optimizing smooth and feasible trajectories, particularly in cluttered and constrained spaces.

3 Diffusion for Motion Planning

In this section, we first introduce the background of diffusion model. Then followed by the formulation of path planning with diffusion model as planning as an inference problem. Lastly, we introduce the test time composable inference procedure with the cost gradient guide.

3.1 Diffusion Model

Diffusion models are inspired by non-equilibrium thermodynamics. They define a Markov chain of diffusion steps to slowly add random noise to data and then learn to reverse the diffusion process to construct desired data samples from the noise.[26][27]

The forward diffusion process models the process that iteratively corrupts a target data distribution into a final isotropic Gaussian distribution along time[20]. At each step, a small amount of Gaussian noise controlled by a variance schedule $\beta_t \in (0,1)_{t=1}^{T}$ is added to the sample[28]. For the convenience of notation, we denote $\alpha_t = 1 - \beta_t$, and $\bar{\alpha_t} = \prod_{i=1}^{t} \alpha_i$:

$$q(\tau_t|\tau_{t-1}) = \mathcal{N}(\tau_t; \sqrt{\alpha_t}\tau_{t-1}, \beta_t\mathcal{I});$$

Thus, we have:

$$q(\tau_t|\tau_0, t) = \mathcal{N}(\tau_t; \sqrt{\bar{\alpha_t}}\tau_0, (1 - \bar{\alpha_t}I))$$

Based on the forward model, the Denoising Diffusion Probabilistic Models (DDPMs) [29] is a class of generative model that reverses the aforementioned process with denoising steps $p(\tau_{t-1}|\tau_t, t)$ to recover a noiseless true data distribution τ^0 as the optimal path in our task from a Gaussian noise input τ^N. The diffusion models approximate the denoise posterior distribution with a parameterized Gaussian $p_\theta(\tau_{t-1}|\tau_t, t) = \mathcal{N}(\tau_{t-1}; \mu_t = \mu_\theta(\tau_t, t), \Sigma_t)$ with the $\mu_\theta(\tau_t, t)$ learned as the noise prediction network:

$$p_\theta(\tau^0) = \int p(\tau^T) \prod_{i=1}^{T} p_\theta(\tau^{i-1}|\tau^i) d\tau^{1:T}$$

To train the DDPMs, the objective is to train the model such that the error measured with mean square error on noise scheduled is minimized[29]:

$$\theta^* = arg\ min_\theta\ \mathbb{E}_{t\in[1,T],\tau_0,\epsilon_t} ||\epsilon_t - \epsilon_\theta(\tau_t, t)||^2$$

3.2 Diffusion for Planning

Following the convention in motion planning, let $\mathcal{O}$ denote the task, which is a viable collision-free path between the start and goal position in our case. Meanwhile, the robot is denoted as a point mass in this task, and let $s = [q^\top, \dot{q}^\top]$ describe the current state as the planar position and velocity of the robot. The goal of the task is to find a trajectory $\tau = [s_0, ..., s_t]$ which is associated with and aims to minimize a set of cost $c(\tau) = \sum_i \omega_i c_i(\tau)$ such as collision penalty, energy consumption correlated to the path length, acceleration, etc. Formally this objective can be denoted as:

$$\tau^* = arg\ min_\tau \sum_i \omega_i c_i(\tau)$$

In path planning with a diffusion model, unlike traditional MDP settings where the trajectory is predicted forward in time autoregressively, diffusion treats planning nearly identical to sampling[20] and predicts waypoints across the horizon simultaneously. Thus learning an optimal plan is equivalent

to learning optimal sampling so the generative error will be minimized after a series of denoising steps $p(\tau_{t-1}|\tau_t, t)$. Again, diffusion models approximate the posterior distribution with a parametrized Gaussian $p_\theta(\tau_{t-1}|\tau_t, t)$. In practice, following the convention, we only learn the mean of the inverse process as shown in the objective function in section 3.1. The covariance is set to $\Sigma_t = \sigma_t^2 I = \frac{\beta_t(1-\bar{\alpha_{t-1}})}{1-\bar{\alpha_t}}$.

The state includes a fixed-length trajectory, which in our case is a sequence of planar coordinates [x, y]. During the diffusion process, the trajectory updates itself from an initial fixed-length trajectory $[s_0\ s_1\ ...\ s_T]$. By running the backward denoising process across K diffusion iteration steps, the output path can be retrieved as the updated input array.

3.3 Cost Gradient Guide During Inference

One attractive property of diffusion generative model thus diffusion-based planner is the test-time composibility. With general posterior $p(\tau_{t-1}|\tau_t, t)$ derived from section 3.2, applying Bayesian inference, the task-conditioned posterior distribution can be updated with:

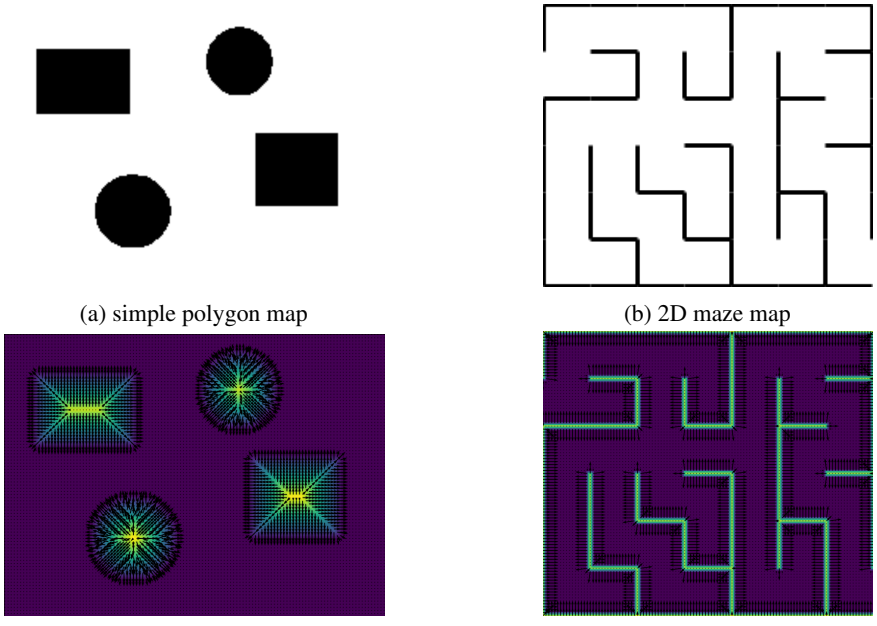
$$p(\tau_{t-1}|\tau_t, t, \mathcal{O}) \propto p(\tau_{t-1}|\tau_t, t)p(\mathcal{O}|\tau_{t-1}, \tau_t, t)$$

The $p(\mathcal{O}|\tau_{t-1}, \tau_t, t)$ is the likelihood of achieving the test task given test task constraints. As this solely depends on the current sample τ_{t-1}, the notation drops to $p(\mathcal{O}|\tau_{t-1})$.

With additional task constraint, we update the diffusion process to sample from task-conditioned posterior distribution instead of the general posterior. From work in [26], sampling from task-conditioned posterior is found to be equivalent to sampling from a Gaussian distribution with mean and covariance as $p(z) = \mathcal{N}(z; \mu_t + \Sigma_t g, \Sigma_t)$, where $g = -\sum_i \lambda_i \nabla_{\tau_{t-1}} c_i(\tau_{t-1})$ is the gradient of aggregated costs such as smoothness and collision costs we considered here w.r.t the trajectory.

4 Experiment & Results

4.1 Task Setup



(a) simple polygon map

(b) 2D maze map

(c) simple polygon map

(d) 2D maze map

Figure 1: (Upper) The map used in evaluation; (Lower) superimposed collision cost and gradient. The cost increases from low (purple) to high (yellow), where the arrow visualizes the cost gradient magnitude and direction.

For the experiment, we set the path planning on 2 different 2D occupancy map images as shown in fig 1 where the obstacle is marked in dark color and the free space is in white. The simple polygon

map has a more convex landscape while the 2d maze map is much more ragged and thin-peaked. The environment is discretized in an image format where our environment parser automatically parses the obstacle and free space grid by grid.

For test time guide cost, we implemented collision cost with distance to the nearest obstacle surface d by sweeping the neighbor grids, where d(x) < 0 indicates inside the obstacle and d(x) > 0 is in the freespace.

$$cost_{collision}(x) = \begin{cases} -d(x) + \epsilon, & \text{if } d(x) \leq \epsilon \\ 0, & \text{if } d(x) > \epsilon \end{cases}$$

We also implemented smoothness cost for guiding using L2 distance, where

$$cost_{smoothness}{}_t = ||\tau_t - \tau_{t-1}||_2$$

For cost gradient, we employ finite difference method to retrieve the gradient with respect to x and y over the map. The cost gradients of the two maps are visualized as the bottom two figures in fig 1.

4.2 Hyperparameters

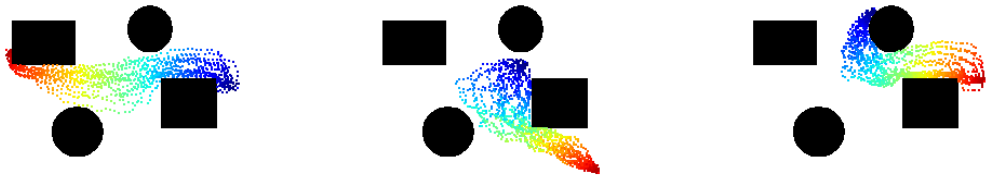


Figure 2: Training data visualization with blue being the starting point and red being the ending point. We generate 20 trajectories per context for the training data with RRT-Connect, and interpolate to fixed length of 64 waypoints.

We used temporal U-Net from[] for the diffusion model. To train the diffusion model, we randomly sampled 600 feasible start and end configurations and passed to RRT-Connect for path generation, with each configuration generating 20 trajectories as show in fig 2. Collision-free trajectories are stored and passed for diffusion training. Each trajectory is interpolated or downsampled to a fixed length of 64 waypoints. We implemented 25 steps of diffusion steps with exponential noise scheduling, and train it with the learning rate = 0.0001 on Adam optimizer until converge.

4.3 Evaluation Metric

To evaluate the performance of a planner, we design the evaluation metric from the three aspects:

Completeness - That is, given a feasible goal, if the planner is able to find a valid collision-free path. The success rate over multiple runs is reported for this.

Optimality - Given a starting and goal configuration, we check if the planner is able to find an optimal (shortest, smoothest) solution. Among all the successful paths, we report (i) the mean path length calculated as the sum of L2 distance between neighbor points and (ii) the curvature which are directly linked to the two properties.

Time complexity - Time complexity is another key aspect of planner selection as different tasks impose different computation requirements. We compared the average computation time for solving a feasible path. As various factors may affect this such as the computing device and level of optimization, we stick to the vanilla implementation based on numpy running on CPU and single-threaded.

4.4 Result and Analysis

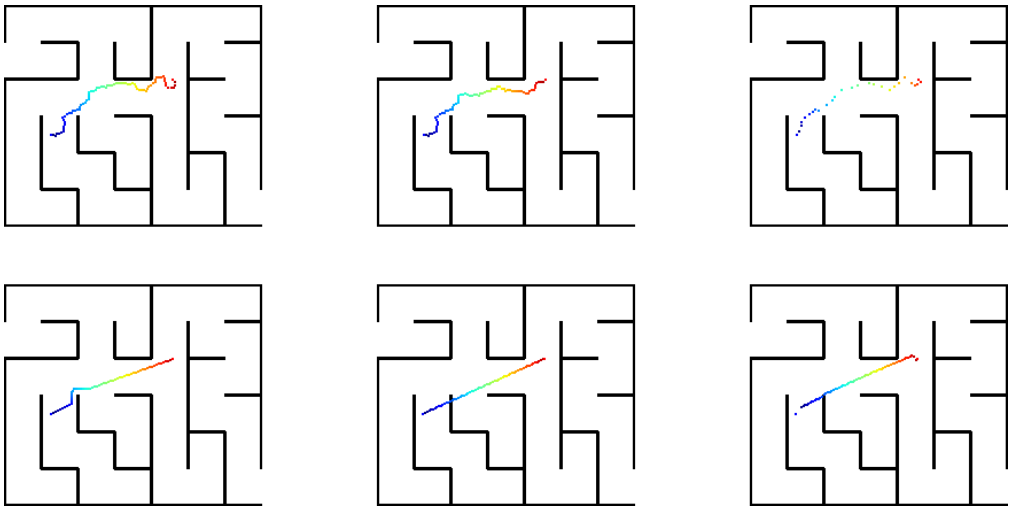
For each metric mentioned above, we run over 50 scenarios and report the results as follows in the table 1 and table 2:

Planner	Success rate	Mean path length (pixel)	Mean curvature	Mean time taken(s)
RRT	100%	162.49±92.60	0.14±0.03	24.44±33.29
RRT-Connect	100%	168.10±81.12	0.13±0.02	0.08±0.05
RRT*	100%	137.51±80.54	0.10 ± 0.065	34.36±60.07
APF	76%	94.16±54.27	0.03±0.02	1.01±0.58
CHOMP	74%	114.97±72.7	0.09±0.02	0.27±0.05
Unguided Diffusion	68%	171.58±58.75	0.12±0.07	0.13±0.12
Guided Diffusion	100%	161.32±54.35	0.10±0.05	0.18±0.12

Table 1: evaluation on the simple polygon map

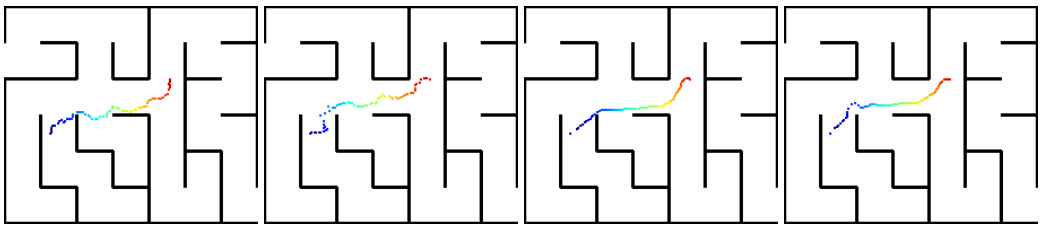
Planner	Success rate	Mean path length (pixel)	Mean curvature	Mean time taken(s)
RRT	100%	232.83±121.53	0.13±0.02	53.43±76.75
RRT-Connect	98%	224.87±114.46	0.12±0.02	1.44±2.31
RRT*	100%	189.02±91.18	0.09±0.03	74.14±108.65
APF	22%	66.05±29.9	0.03±0.02	0.84±0.29
CHOMP	32%	87.26±34.79	0.10±0.08	2.23±0.22
Unguided Diffusion	42%	208.38±84.93	0.09±0.05	0.13±0.12
Guided Diffusion	94%	207.18±78.60	0.13±0.09	0.17±0.12

Table 2: evaluation on the 2D maze map



(upper left) RRT (68) (upper middle) RRT-Connect (66) (upper right) RRT* (25)
(lower left) APF (66) (lower middle) CHOMP initial trajectory (64) (lower right) CHOMP (64)

Figure 3: path planning results with classic planners on the 2D maze. The blue dot indicates the starting point. The red dot indicates the goal. The number in the parenthesis indicates the number of waypoints in the planned trajectory.



(a) unguided path (b) collision cost guided (c) smoothness cost guided (d) combined cost guided

Figure 4: Path planned with the diffusion-based planner. We also conducted the ablation study on the cost gradient guide added during the inference time.

From the quantitative comparisons as shown in table 1 and table 2 and qualitative illustration in fig 3, there are some common patterns we observed. In terms of success rate, the sampling-based method in

general performs significantly better than the optimization-based method, especially in the maze cases which is a global optimization with very thin obstacles and a ragged landscape. The performance of the optimization-based method was observed to be easily affected by hyper-parameters and requires careful tuning. More complex optimization techniques are also expected to further improve its search results. However, on the contrary, in terms of optimality for both the smoothness of a trajectory and path length, optimization-based in general produces a more optimal, which is a smoother and shorter trajectory in a significantly shorter time. In terms of path length. When compared among 3 RRT* variant planners, RRT* generally improves the smoothness and path length but takes a much longer time which comes from constant tree restructuring to keep the node span from its nearest parent. RRT-Connect showed competitive computing time with even optimization-based planners and a strong success rate that shall be considered if the smoothness and trajectory path are not the primary concerns.

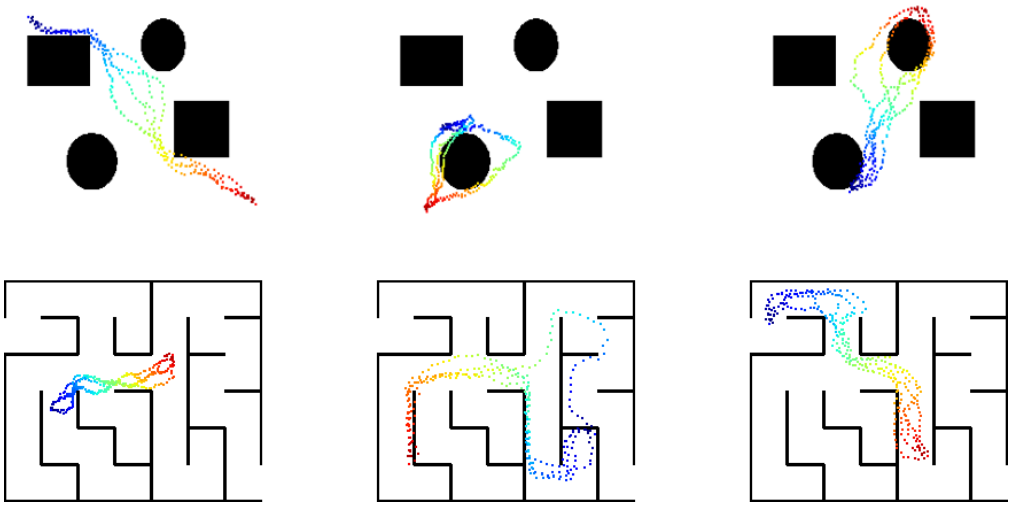


Figure 5: Path generated with the diffusion planner from the general posterior. Multi-modal solutions can be found in the upper middle, upper right and lower middle figures generated by the diffusion-based planner.

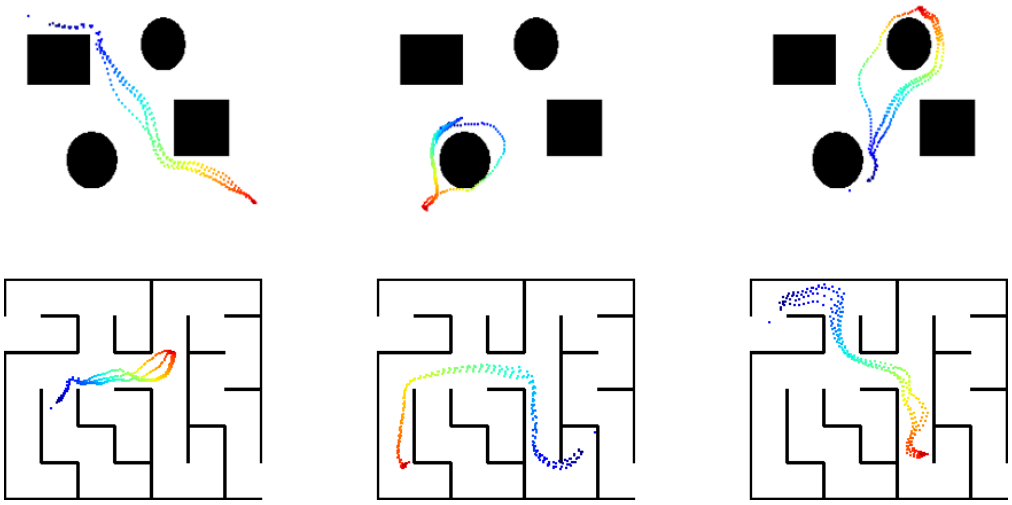


Figure 6: Path generated with the diffusion planner and test-time collision and smoothness cost guide. The smoothness and collision are greatly improved.

For diffusion-based planners, it shows the shortest planning time among all the planners. This can be further accelerated in real usage when GPU is available. Meanwhile, we observed that it is significant to include the task condition by utilizing the cost gradient guide, which improves the success rate on two maps significantly by around 40% and yields 100% and 94% correspondingly. The effect of

cost gradient guidance during inference time is studied individually and visualized in fig 4. We can observe the trajectory shifts towards the center of the maze path, thus further from the wall where the collision can be triggered when collision cost gradient is added. During this process, waypoints tend to be scattered more sparsely and this can be further improved by the effective smoothness guide as shown in (c) and (d). Such effectiveness can also been observed from the difference between fig 5 and fig 6, where we sampled 5 trajectories with the diffusion model per context, and the latter is generated with task information through collision and smoothness cost guide.

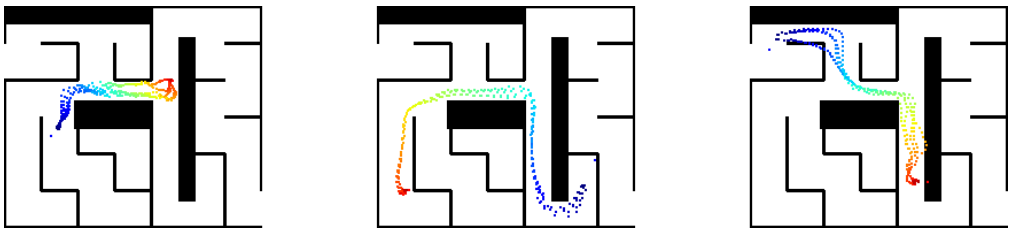


Figure 7: We also tested the test-time composiblity by updating the environment with new obstacles added, the paths generated are effectively updated with the majority being collision-free in the new environment.

As discussed above, the diffusion-based models yield multiple attractive properties, the first being fast; and the second being test time composable as validated above and shown to be significant for our 2D path planning task. There is another attractive property derived naturally from the stochasticity included in the diffusion process, which enables multi-modal solutions to be captured with diffusion models. This is usually challenging with a deterministic model. Such multi-modal solutions can be seen in (1) the upper middle (2) the upper right and (3) the lower middle figures of the polygon map in figure 5, which are further improved in figure 6. Both solutions of paths extending from the left and right of the circle can be found with a diffusion-based planner, which is usually considered to be a challenge for major learning-based planners, even for methods with multi-modal techniques such as using the Gaussian Mixture Model (GMM) to address this [].

Lastly, compared to sampling-based or optimization-based methods, we acknowledge one limitation for the diffusion-based planner is that it needs a database of feasbile paths to train. The performance is still constrained by the number, quality and coverage of such data. For instance, we believe the path length or curvature of the generated paths can be further improved with more optimal and smoother training data derived from for example APF planner.

5 Conclusion

Our work implemented and compared several classic motion planning algorithms, including RRT, RRT-Connect, RRT*, APF, and CHOMP. Additionally, we introduced a diffusion-based planner and analyzed its characteristics both with and without guidance during the inference process.

The implemented planners exhibit various strengths and weaknesses. RRT variants are effective in exploring these spaces, and RRT* ensures asymptotic optimality. APF and CHOMP, on the other hand, leverage optimization techniques, with APF focusing on simplicity and CHOMP excelling in optimizing smooth trajectories. Our evaluations across different scenarios revealed the diverse performance of these planners.

The diffusion-based planner stands out for its speed, test-time composability, and support for multi-modal solutions. Its efficiency is particularly notable when compared to other planners, making it a promising approach for real-time applications. However, it is crucial to acknowledge that the reliability and speed of the diffusion-based planner are contingent on the quality, quantity, and coverage of the training data. The planner's ability to provide multi-modal solutions is a significant advantage, especially in scenarios with complex environments.

As future work, enhancing the diffusion-based planner's robustness and generalization capabilities by incorporating more comprehensive training datasets could further improve its performance. Additionally, investigating methods to address its limitations and exploring ways to adapt the diffusion-based planner to different types of environments will contribute to the broader applicability of this approach in motion planning for 2D navigation.

In summary, our work contributes to the understanding of different motion planning approaches and highlights the strengths and limitations of each. The diffusion-based planner emerges as a promising solution with its speed, adaptability, and multi-modal support. Nevertheless, addressing challenges related to data requirements remains an area for further exploration and improvement in future research.

References

[1] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[2] Tomás Lozano-Pérez and Michael A Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, 1979.

[3] Oussama Khatib. The potential field approach and operational space formulation in robot control. In *Adaptive and Learning Systems: Theory and Applications*, pages 367–377. Springer, 1986.

[4] John Schulman, Jonathan Ho, Alex X Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. In *Robotics: science and systems*, volume 9, pages 1–10. Berlin, Germany, 2013.

[5] Matt Zucker, Nathan Ratliff, Anca D Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher M Dellin, J Andrew Bagnell, and Siddhartha S Srinivasa. Chomp: Covariant hamiltonian optimization for motion planning. *The International journal of robotics research*, 32(9-10):1164–1193, 2013.

[6] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. Stomp: Stochastic trajectory optimization for motion planning. In *2011 IEEE international conference on robotics and automation*, pages 4569–4574. IEEE, 2011.

[7] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.

[8] David Hsu, J-C Latombe, and Rajeev Motwani. Path planning in expansive configuration spaces. In *Proceedings of international conference on robotics and automation*, volume 3, pages 2719–2726. IEEE, 1997.

[9] BT Zhang and LIUSR LI JD. Rapidly-exploring random trees motion planning for non-holonomic robot with collision-test and regression mechanism. *Control Theory & Applications*, 33(7):870–878, 2016.

[10] Ignacio Pérez-Hurtado, Miguel Á Martínez-del Amor, Gexiang Zhang, Ferrante Neri, and Mario J Pérez-Jiménez. A membrane parallel rapidly-exploring random tree algorithm for robotic motion planning. *Integrated Computer-Aided Engineering*, 27(2):121–138, 2020.

[11] Brian Ichter, James Harrison, and Marco Pavone. Learning sampling distributions for robot motion planning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7087–7094. IEEE, 2018.

[12] Dorothea Koert, Guilherme Maeda, Rudolf Lioutikov, Gerhard Neumann, and Jan Peters. Demonstration based trajectory optimization for generalizable robot motions. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 515–522. IEEE, 2016.

[13] M Rana, Mustafa Mukadam, Seyed Reza Ahmadzadeh, Sonia Chernova, and Byron Boots. Towards robust skill generalization: Unifying learning from demonstration and motion planning. In *Intelligent robots and systems*, 2018.

[14] Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5923–5930. IEEE, 2023.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[17] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[18] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

[19] Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, pages 1–6, 2023.

[20] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

[21] Sertac Karaman, Matthew R Walter, Alejandro Perez, Emilio Frazzoli, and Seth Teller. Anytime motion planning using the rrt. In *2011 IEEE international conference on robotics and automation*, pages 1478–1483. IEEE, 2011.

[22] David Brandt. Comparison of a and rrt-connect motion planning techniques for self-reconfiguration planning. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 892–897. IEEE, 2006.

[23] Wang Xinyu, Li Xiaojuan, Guan Yong, Song Jiadong, and Wang Rui. Bidirectional potential guided rrt* for motion planning. *IEEE Access*, 7:95046–95057, 2019.

[24] Steven Byrne, Wasif Naeem, and Stuart Ferguson. Improved apf strategies for dual-arm local motion planning. *Transactions of the Institute of Measurement and Control*, 37(1):73–90, 2015.

[25] Nathan Ratliff, Matt Zucker, J Andrew Bagnell, and Siddhartha Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *2009 IEEE international conference on robotics and automation*, pages 489–494. IEEE, 2009.

[26] Joao Carvalho, An T Le, Mark Baierl, Dorothea Koert, and Jan Peters. Motion planning diffusion: Learning and planning of robot motions with diffusion models. *arXiv preprint arXiv:2308.01557*, 2023.

[27] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[28] Lilian Weng. What are diffusion models? *lilianweng.github.io*, Jul 2021.

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