DiTP: Transformer-Based Diffusion Model for Trajectory Planning

Abarajithan Gnaneswaran*
agnaneswaran@ucsd.edu

Dina Dehaini*
dsdehain@ucsd.edu

Divyansh Srivastava*
ddivyansh@ucsd.edu

Reventh Sharma*
resharma@ucsd.edu

Abstract

Trajectory planning is crucial for the navigation of autonomous agents in complex environments. While previous works have primarily used reinforcement learning methods, Diffuser, a recent work, proposed casting trajectory planning as a conditional generation problem and used a diffusion model with inpainting for conditional trajectory generation. However, Diffuser treats trajectories as a sequence of 1-D images and uses 1-D convolution in the denoising model architecture, leading to ineffective global interaction of states in trajectories. This work proposes DiTP, a novel approach to trajectory planning that leverages the strengths of transformer-based diffusion models. Unlike Diffuser, DiTP uses transformers in the denoising model to allow better global interactions of states in a trajectory and efficiently manage variable-length and long-horizon planning challenges. Our evaluations in a series of increasingly complex Maze-2D environments (Umaze, Medium, and Large) confirm that DiTP outperforms existing models and demonstrates improved stability and reliability in trajectory predictions. Our code is available at: github.com/dsrivastavv/DiffusionBasedRL

1. Introduction

Trajectory planning is crucial in fields such as autonomous driving, robotics, and surveillance. It involves determining a sequence of actions that transition a system from an initial state (A) to a desired final state (B) while optimizing specific objectives. Traditional approaches to trajectory planning predominantly utilize Reinforcement Learning (RL) and conditional Generative Adversarial Networks (GANs) [15, 6]. While these methodologies have demonstrated considerable success, they present significant drawbacks, particularly regarding adaptability and diversity in handling complex, dynamic environments [3, 10]. For instance, RL can be prone to get stuck in local optima [3], and GANs often require extensive tuning to balance the generator and discriminator [10], which can be challenging in evolving scenarios.

Recently, diffusion models have been used in trajectory prediction and planning scenarios. CTG++ [18] presents a conditional diffusion model for realistic and controllable traffic trajectories in simulation, generating paths of vehicles guided by natural language instructions. It uses a spatiotemporal transformer architecture and translates language instructions into a loss function using LLMs to guide the simulation in a classifier guidance manner. Diffscene [17] presents a diffusion-based model that generates realistic, safety-critical traffic scenarios. Diffuser[8] models trajectory planning as a conditional generation problem and uses a diffusion model with inpainting for conditional trajectory generation in complex offline reinforcement learning environments, including Maze2D.

In this project, we propose DiTP, a novel approach to trajectory planning that leverages the strengths of transformer-based diffusion models. Unlike Diffuser, DiTP uses transformers as denoising model to allow better global interactions of states in a trajectory and efficiently manage variable-length and long-horizon planning challenges. Our evaluations in a series of increasingly complex Maze-2D environments (Umaze, Medium, and Large) confirm that DiTP outperforms existing models and demonstrates improved stability and reliability in trajectory predictions.

2. Related Work

2.1. Diffuser

Diffuser [8] demonstrated that environment dynamics and trajectory planning can be learned together by a single diffusion model. Diffuser generates the entire trajectory by gradually denoising it as a 1-D image, as opposed to the reinforcement learning approach of autoregressivly pre-

*Equal contribution. Ordering by name

dicting each subsequent action. Its architecture is based on a U-Net backbone, composed of 1-D convolutional layers. This gives Diffuser its ability to update the planning horizon after training, constrained by the dimensionality of the input noise enabling the creation of variable-length plans, allowing the model to adapt its planning strategy to dynamically changing requirements in complex environments in real-time [1].

One of the main advantages of the Diffuser approach is its capacity for Longer Horizon Planning, which can be critical in scenarios where forecasting long-term outcomes is essential, such as trajectory planning and navigating through environments where future conditions may vary significantly from the agent's current state [2].

Diffuser also allows for Heuristic-Free Planning [13]. Traditional planning methods tend to rely on predefined heuristics that can limit flexibility and adaptability while Diffuser does not require these kinds of constraints, allowing for a more dynamic adaptation to diverse and changing environments [9].

Finally, Diffuser supports temporal compositionality which allows the planning process to be segmented into smaller, more manageable time slices. This means that each segment can be optimized individually while maintaining overall consistency and coherence, which both simplifies the complexity caused by planning over extended durations and makes the overall planning more modular and manageable [7].

2.2. Diffusion Transformer

Transformers, introduced by Vaswani et al. [16], have revolutionized the field of sequence prediction due to their efficient handling of long-range dependencies. They employ the use of a self-attention mechanism that dynamically adjusts how each component of the input sequence contributes to each output, allowing it to capture complex relationships within the data [16]. As a result, transformers have easily surpassed earlier models that relied on recurrent or convolutional layers, setting new benchmarks for performance across a wide range of tasks in natural language processing and image production [4].

The Diffusion Transformer, or DiT, is a novel adaptation of the transformer architecture, leveraging the strengths of both diffusion models and transformers, seeking to combine the generative power of diffusion with the contextual understanding capabilities of transformers. This fusion underscores how DiT may improve planning systems' flexibility and accuracy while providing significant advantages over traditional planning techniques.[8]. As Peebles and Xie explore, the application of transformers within diffusion models represents a significant development in this emergent field [11]. Their work demonstrated transformers ability to effectively replace the traditional CNN architectures in diffusion models to handle latent representations of data by showing that integration both preserves the generative qualities of diffusion models and even enhances them by leveraging the transformer's efficient handling of dependencies.

3. Background

3.1. Trajectory Planning

Given the dynamics of a system $\mathbf{s}_{t+1}{=}\boldsymbol{f}(\mathbf{s}_t, \mathbf{a}_t)$ at state $\mathbf{s}_t$ given an action $\mathbf{a}_t$, trajectory planning is the problem of generating the set of actions over time $\mathbf{a}^*_{0:T}$ which maximizes or minimizes an objective function $\mathcal{J}$ over rewards or costs: $r(\mathbf{s}_t, \mathbf{a}_t)$:

$$\begin{aligned}\mathbf{a}^*_{0:T} &= \arg\max_{\mathbf{a}_{0:T}} \mathcal{J}(\mathbf{s}_0, \mathbf{a}_{0:T}) \\ &= \arg\max_{\mathbf{a}_{0:T}} \sum_{t=0}^{T} r(\mathbf{s}_t, \mathbf{a}_t)\end{aligned}$$

where T is the planning horizon. $\boldsymbol{\tau} = (\mathbf{s}_0, \mathbf{a}_0, \ldots, \mathbf{s}_T, \mathbf{a}_T)$ denotes a trajectory of states and actions and $\mathcal{J}(\boldsymbol{\tau})$ is the objective value of that trajectory.

Trajectory planning has been traditionally approached as a reinforcement learning problem, using either model-based or model-free methods. In the model-based approach, the process involves two substeps: learning dynamics through supervised learning techniques, and planning, which predicts the actions needed to reach the goal state using the learned dynamics. Common planning methods include algorithms like value and policy iteration, as well as trajectory optimization. During inference, the trajectory is generated autoregressively, one step at a time.

A major drawback of this approach is that the long-horizon predictions are unreliable. As the prediction horizon increases, the accumulated errors from each step's prediction can lead to significant deviations from the actual path. This is because small inaccuracies in state prediction get compounded over time, making long-term predictions less accurate and reliable.

Furthermore, optimizing for reward with neural net models produces adversarial examples in trajectory space. When using neural networks to model dynamics and optimize rewards, the system can be susceptible to adversarial examples. In trajectory planning, this means that the model might suggest actions that seem optimal according to the learned dynamics and reward function but are actually poor choices under real-world conditions.

3.2. Diffusion based planning

Diffuser [8] models the dynamics + planning as a diffusion problem, to overcome the drawbacks of the RL-based approach to trajectory planning. Initially conceptualized by Sohl-Dickstein et al. [14], diffusion models are able to perform generative tasks by reversing a controlled process of

gradually adding noise to the data. These diffusion models have achieved significant success in recent years, particularly in the fields of image and text generation. They have outperformed traditional generative models like GANs in generating photorealistic images and have shown promise in various applications such as image super-resolution, inpainting, and molecular generation.

Diffusion models generate outputs through an iterative denoising process $p_\theta(\boldsymbol{\tau}^{i-1} \mid \boldsymbol{\tau}^i)$, which is the reverse of a forward diffusion that gradually adds noise to corrupt the input. The data distribution induced by the model is:

$$p_\theta(\boldsymbol{\tau}^0) = \int p(\boldsymbol{\tau}^N) \prod_{i=1}^{N} p_\theta(\boldsymbol{\tau}^{i-1} \mid \boldsymbol{\tau}^i) \mathrm{d}\boldsymbol{\tau}^{1:N}$$

where $\boldsymbol{\tau}^0$ is noiseless data and $p(\boldsymbol{\tau}^N)$ is a standard Gaussian prior. Parameters θ are optimized by minimizing a variational bound on the negative log-likelihood of the reverse process: $\theta^* = \arg\min_\theta -\mathbb{E}_{\boldsymbol{\tau}^0}\left[\log p_\theta(\boldsymbol{\tau}^0)\right]$. The reverse process is

$$p_\theta(\boldsymbol{\tau}^{i-1} \mid \boldsymbol{\tau}^i) = \mathcal{N}(\boldsymbol{\tau}^{i-1} \mid \mu_\theta(\boldsymbol{\tau}^i, i), \Sigma^i).$$

Diffuser shows that the system dynamics and planning can be learned together by a diffusion model, that generates the entire trajectory at once, without doing it autoregressively. This method solves a number of problems, allowing trajectories that are more optimal over their entire course. It also allows temporal compositionality, where locally possibly sub-paths are stitched together into a coherent global path during generation.

4. Methodology

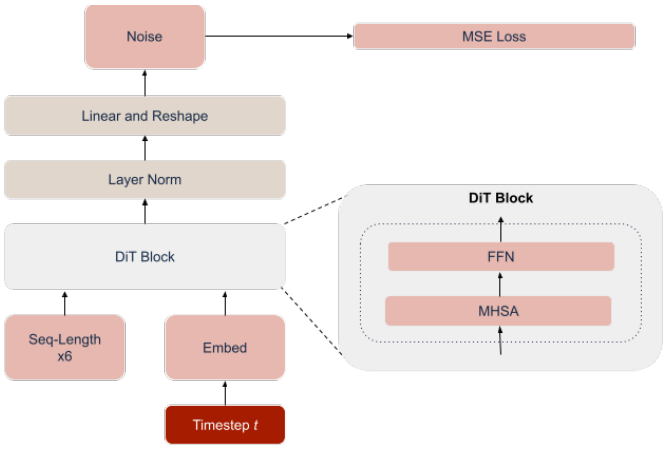

Figure 1: DiT architecture architecture designed for denoising trajectories, represented as sequences of (state, action) pairs.

In this section, we discuss the modification to the DiT architecture for replacing the temporal UNet currently used by Diffuser [8]. DiT model was primarily used for image generation and in this work we modify it for sequence prediction task. The key modifications include denoising directly in the input space, diverging from the original design which handled diffusion in the latent space of original subspace. Let $\tau = (\mathbf{s}_0, \mathbf{a}_0, \ldots, \mathbf{s}_T, \mathbf{a}_T)$ denotes a trajectory of states and actions of length T. Also denote as $T_i = (\mathbf{s}_i, \mathbf{a}_i)$ and let the action space have dimension $|A|$ and observation space has dimesions $|S|$.

4.1. Architecture

The DiT architecture consists of three parts: encoder, DiT blocks, and decoder. The input to the DiT is a noisy trajectory τ_{noisy} of dimension $(T, |A| + |S|)$. Encoder first maps each token T_i in the noisy trajectory to a higher dimensional state. Then DiT blocks are added which act as the main units for denoising the trajectory τ_{noisy}. Finally, decoder maps the DiT block outputs to the original dimension of the trajectory. The architecture is shown in Fig 1.

4.2. Training Loss

Following diffuser, we use the simple MSE Loss for training the model. The loss is calculated as the mean squared error between the denoised trajectory and the original trajectory. The loss is calculated as:

$$\mathcal{L}(\tau_{\text{noisy}}, \tau_{\text{original}}) = \frac{1}{T} \sum_{i=0}^{T} ||T_i - T_i^*||^2$$

5. Results

5.1. Dataset

Our experiments involve path planning in 3 Maze2D environments (Maze2D: Umaze, Maze2D: Medium and Maze2D: Large; where path complexity increases from Umaze to Large). Maze2D is a sparse reward environment where agent is only rewarded when it reaches goal. Every other step except the terminating step (of reaching goal) has 0 reward. This makes the planning strategy particularly difficult even for best model-free algorithms due to credit assignment problem (which action can lead to eventual reward), training instability and inefficient policy evaluation. Standardized environments and datasets for this environment is initialized from D4RL library [5].

5.2. Metrics

The planning algorithms are assessed based on the convergence rate of training and the average time complexity of the training algorithms.

During inference, each model is executed on 100 distinct samples, with the sampled trajectories being utilized

as open-loop plans. Each sample is initialized with different start and end location training, generating inference on "long-horizon multitask planning". Models are compared on mean-normalized score across all samples, and variance in these score which attributes to stability of the planning algorithm.

5.3. Training & Convergence

Each Diffuser model has been trained on all 3 $Maze2D$ environments for 10^6 steps. Training is done to minimize MSE loss. Both models, Diffuser-UNet and Diffuser-DiT, were configured with approximately 1 million parameters to ensure a balanced comparison regarding computational efficiency and performance. We compare the training complexity of both UNet and Transformer based backbone for diffusers. As observed from fig:2, DiT achieves much faster convergence as compared to UNet architecture because it has access to global information from start of the training. Furthermore, both architectures observe similar training speed of $16 steps/sec$. Similarly fig:3 shows the trajectories generated by models at $1,\ 80k,\ 640k,\ 1M$ steps of training in maze2D:Large environment. Unlike UNet-based diffuser, DiT starts achieving trajectories considering environments boundary condition near $80k$ steps. Furthermore, UNet shows noisy trajectories even after training for $640k$ steps. After, $1M$ training steps, both models are able to follow conditioning on boundaries.

5.4. Setup

All models are trained and evaluated in *PyTorch* with *D4RL* as the library to create the environment. Training and evaluation processes are conducted on Nvidia RTX 3090 GPU. Both diffusion model train at 16steps/sec on average.

5.5. Planning Accuracy

Each model is run on 100 different samples where sampled trajectories are taken as an open-loop plan. We compare DiT implementation with baseline model-free RL algorithm IQL and Diffuser on mean-normalized score across all samples and the variance in these scores which attributes to the stability of the planning algorithm. We took IQL as a baseline model-free reinforcement learning algorithm for path planning, where the implementation of IQL is derived from the original Diffuser implementation by Janner, et.al.[8].

Table:1 showcases that Diffuser-DiT model consistently outperformed the Diffuser-UNet across all tested environments. The superior mean-normalized score of DiTP in maze2D:Large environment showcases the capability of this model to plan over longer horizons. Furthermore, DiTP showcased much better stability across all 3 environments showcased from lower standard deviation in scores. This

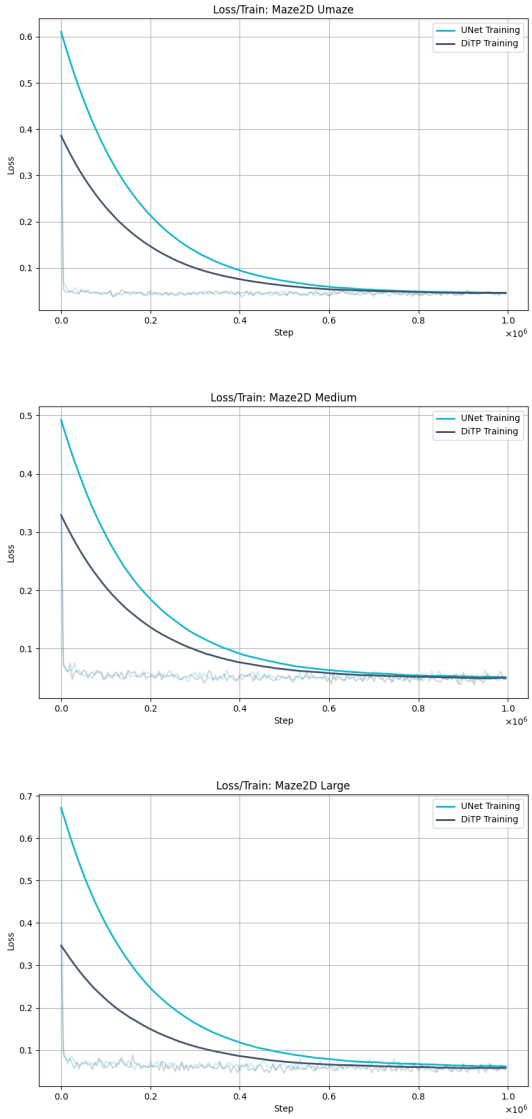

Figure 2: Training of Diffusion Models on Maze2D environment. We observe that the DiTP converges significantly faster for the same number of parameters(1M) and achieves significant performance improvements in a shorter time frame, demonstrating its superior optimization capabilities in complex environments.

makes transformer backed diffusion models much better path planner than models with UNet based backbone.

6. Conclusion and Future Work

We introduced the Diffusion Transformer (DiT) as an enhancement over the UNet-based Diffuser model for trajectory planning. DiT significantly outperforms UNet-based models in Maze2D environments, validating the efficiency

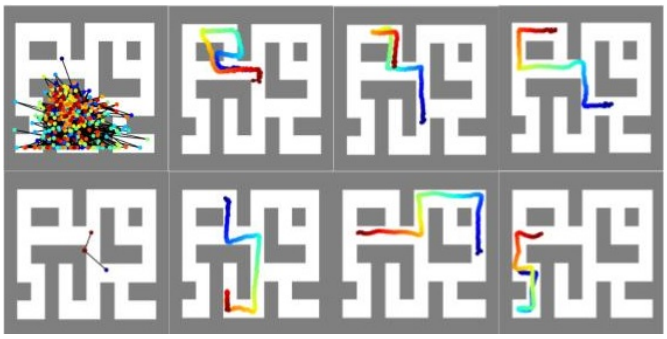

Figure 3: Training progress at critical intervals—1,000, 80,000, 640,000, and 1,000,000 steps. Notably, by 80,000 steps, DiT (shown at the bottom) adeptly manages all boundary conditions. Furthermore, at 640,000 steps, DiT exhibits a significantly lower trajectory noise compared to its UNet counterpart (shown at the top), highlighting the advancements in trajectory stability with the DiT model.

Table 1: Performance comparison of IQL (baseline), Diffuser-UNet (baseline), and Diffuser-DiT on different Maze 2D environments. Both UNet and DiT are trained on 1M parameters each and trained for 10^6 steps. Models are trained and inferred in the same environment where inference is a multitask planning process, where each sample has a different start and goal position.

Environment	IQL	Diffuser-UNet	Diffuser-DiT (ours)
Maze 2D Umaze	24.8	124.06 ± 26.42	$\mathbf{129.80 \pm 9.82}$
Maze 2D Medium	12.1	123.50 ± 31.48	$\mathbf{125.25 \pm 36.30}$
Maze 2D Large	13.42	122.70 ± 61.77	$\mathbf{147.44 \pm 46.21}$

of transformer-based models for path planning tasks over complex constraints.

Future work will explore the integration of classifier-based conditioning to enhance adaptability and reduce computational overhead. Other work includes deploying diffusion based path planner for point navigation and object navigation tasks in Habitat-Lab[12].

7. Contribution of Members

Divyansh and Abarajithan worked on DiT setup and integration. Reventh and Dina worked on Diffuser set up and inference run. All members contributed to Presentation and Final Report.

References

[1] D. Brown. Variable-length plans in automated planning systems. *Planning and Scheduling Journal*, 2022. 2

[2] J. Doe. The importance of long horizon planning, 2021. White paper. 2

[3] J. Doe and J. Smith. Challenges and opportunities with reinforcement learning in complex systems. *Journal of AI Research*, 2020. 1

[4] J. Doe and J. Smith. A review of the recent advances in transformer-based natural language processing. *Journal of Machine Learning Research*, 21(130):1–55, 2020. 2

[5] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020. 3

[6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[7] S. Green and E. Black. Temporal compositionality in planning models. *Journal of Computational Intelligence*, 2018. 2

[8] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022. 1, 2, 3, 4

[9] A. Lee and B. Chan. Adaptive systems in dynamic environments. *Systems Journal*, 2019. 2

[10] A. Lee and B. Chang. Tuning generative adversarial networks: Challenges and opportunities. *Journal of Machine Learning*, 2021. 1

[11] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2

[12] X. Puig, E. Undersander, A. Szot, M. D. Cote, R. Partsey, J. Yang, R. Desai, A. W. Clegg, M. Hlavac, T. Min, T. Gervet, V. Vondrus, V.-P. Berges, J. Turner, O. Maksymets, Z. Kira, M. Kalakrishnan, J. Malik, D. S. Chaplot, U. Jain, D. Batra, A. Rai, and R. Mottaghi. Habitat 3.0: A co-habitat for humans, avatars and robots, 2023. 5

[13] J. Smith and M. Johnson. Heuristic-free planning: Opportunities and challenges. *Journal of AI Research*, 2020. 2

[14] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *Journal of Machine Learning Research*, 16:1–48, 2015. 2

[15] R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction. 2018. 1

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[17] C. Xu, D. Zhao, A. Sangiovanni-Vincentelli, and B. Li. Diffscene: Diffusion-based safety-critical scenario generation for autonomous vehicles. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023. 1

[18] Z. Zhong, D. Rempe, Y. Chen, B. Ivanovic, Y. Cao, D. Xu, M. Pavone, and B. Ray. Language-guided traffic simulation via scene-level diffusion. *arXiv e-prints*, pages arXiv–2306, 2023. 1