GenPlanner: From Noise to Plans - Emergent Reasoning in Flow Matching and Diffusion Models

Agnieszka Polowczyk[1][0009−0008−1583−4493], Alicja Polowczyk[1][0009−0001−3110−8255], and Michał Wieczorek[1][0000−0002−5319−3366]

Silesian University of Technology, Faculty of Applied Mathematics, Gliwice, 44-100, Poland

Abstract. Path planning is a fundamental component of agent-based systems, where individual agents must navigate complex environments while satisfying spatial constraints. This makes the problem relevant not only in classical navigation tasks but also in broader settings such as multi-agent simulations and decision-making systems. In this paper, we explore the potential of using generative models as planning and reasoning mechanisms. We propose GenPlanner, an approach based on diffusion models and flow matching, along with two variants: DiffPlanner and FlowPlanner. We demonstrate the application of generative models to find and generate correct paths in mazes. A multi-channel condition describing the structure of the environment, including an obstacle map and information about the starting and destination points, is used to condition trajectory generation. Unlike standard methods, our models generate trajectories iteratively, starting with random noise and gradually transforming it into a correct solution. Experiments conducted show that the proposed approach significantly outperforms the baseline CNN model.

Keywords: Generative planning · Diffusion Models · Flow Matching · Visual reasoning · Maze navigation

1 Introduction

Recent years have seen a rapid development of generative models and large language models (LLMs), which demonstrate significant capabilities in many domains. Despite these successes, tasks requiring logical reasoning and spatial planning still pose significant challenges [2]. Standard Vision-Language Models (VLMs) often hallucinate, failing to maintain logical consistency over long planning horizons [10]. There are comprehensive machine learning-based planning approaches that avoid sequential methods [8]. Additionally, diffusion-based solutions have been proposed that utilize computationally expensive inference-time optimizations [7]. In navigation tasks, global trajectory modeling using additional guidance mechanisms is crucial [9].

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_29

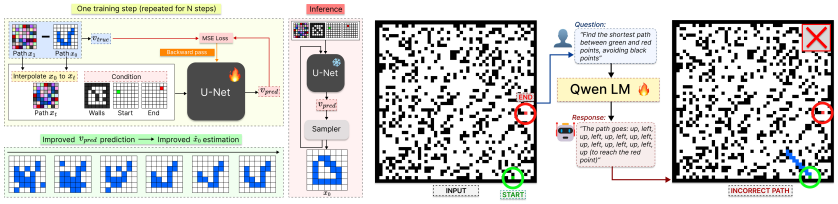

Fig. 1: **Comparison of approaches.** Left: **Overview of FlowPlanner training and inference.** Right: **Failure case of logical and spatial reasoning in a vision–language model.** Given a maze image with start and goal locations, the VLM (Qwen) generates a plausible sequence of directional moves, yet the resulting path is invalid.

In this work, we interpret reasoning as the ability to generate globally consistent trajectories that satisfy spatial constraints, such as obstacle avoidance and connectivity between start and goal, without explicit search. We propose a novel approach GenPlanner that solves the problem of planning in a maze with varying mesh sizes by formulating it as an image generation task. Instead of using expensive optimization during inference, we developed the FlowPlanner and DiffPlanner algorithms based on generative models: Flow Matching and Diffusion. Our model takes as input a noisy mesh and a conditioning tensor, whose individual channels encode the starting position, goal, and wall layout. Experiments show that our method achieves a path generation efficiency of 89% for solving 48×48 mazes. In summary, our principal contributions are as follows:

- We introduce a new representation of the planning problem in the form of a multi-channel conditional tensor (start, end, and walls), which allows the generative model to learn obstacle avoidance.
- We introduce GenPlanner in two variants: DiffPlanner and FlowPlanner, for planning and reasoning tasks.
- We experimentally demonstrate that our model effectively handles the growing solution space for larger meshes.

2 Related Works

Contemporary research confirms that, despite great advances in vision tasks, VLM models encounter fundamental difficulties in tasks requiring reasoning, see Fig. 1. These models struggle to abstract rules and perform worse than humans even on real-world images [8], [10]. Furthermore, benchmarks such as VisuLogic [13] and ENIGMATA [2] reveal that models often generate semantically correct but logically inconsistent solutions. Recent studies based on the DynaMath [14] and PARTNR [1] benchmarks confirm VLM's sensitivity to small visual variances in mathematical tasks and its shortcomings in long-term spatial planning. The application of diffusion models to planning is a rapidly developing area, employing global attention mechanisms [9] or costly constrained optimization

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_29

to enforce collision-free behavior [7]. In the context of robotics, datasets such as RoboCerebra [4] and WOMB [6] emphasize the need to integrate navigation with the physics of the environment. Unlike methods based on decomposition [11], our model solves the problem through a denoising process. Furthermore, we are also inspired by the observations from the work [10], which shows that longer reasoning correlates with correctness on difficult tasks. In our case, this role is played by an iterative diffusion process.

3 GenPlanner

This section introduces our GenPlanner method, a generative approach to planning based on conditional models, see Fig. 1. It includes two variants: DiffPlanner, which utilizes a diffusion model, and FlowPlanner, which is based on flow matching.

Input Data Representation The input to the model comprises two components: a noisy path map and a condition that describes the maze structure. The binary path map $x_0 \in \{0,1\}^{H\times W}$ is initially mapped to the range of $[-1,1]$ and then noisy by linear interpolation with a random noise sample $\epsilon \sim \mathcal{N}(0,I)$. For the diffusion model, the noised sample is constructed for a timestep $t \in \{0,\ldots,T-1\}$ as follows:

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon. \tag{1}$$

where $T = 1000$ in all experiments and $\alpha_t \in [0,1]$ is a timestep-dependent coefficient controlling the balance between the clean signal and the injected noise. For the flow model, a sample is created for a randomly selected t in the range $[0,1]$:

$$x_t = (1-t)x_0 + tx_1 \tag{2}$$

where in the flow matching formulation, the noise variable is denoted by x_1 instead of ϵ. The condition $c \in \{0,1\}^{3\times H\times W}$ is the concatenation of three binary masks: the walls map, the starting point mask, and the destination point mask. The final input to the network is obtained by concatenating x_t and c along the channel dimension, resulting on a four-channel input tensor. In the inference phase, pure noise ϵ is supplied instead of x_t, and the model generates a path map conditioned solely on the maze structure.

DiffPlanner Training During training, a forward diffusion process is simulated, in which Gaussian noise ϵ is gradually added to the binary path map x_0 for a specified timestep t, according to Eq. 1. The model, conditioned on the maze representation c, learns to predict noise based on the triplet (x_t, c, t), i.e., it estimates the noise $\epsilon_\theta(x_t, c, t)$. Training is accomplished by minimizing the mean square error:

$$\mathcal{L}_{\text{DiffPlanner}} = \mathbb{E}_{x_0,t,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t,c,t)\|_2^2\right]. \tag{3}$$

FlowPlanner Training FlowPlanner instead of iteratively adding and removing noise, as in diffusion, a flow matching approach is used, in which the

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_29

model directly learns the vector field that controls this transformation. For each sample, a noise map $x_1 \sim \mathcal{N}(0, I)$ is sampled, and the ground-truth path mask x_0 is used. For a randomly sampled scalar t, an intermediate point is computed according to Eq.2. The target vector field is defined as the time derivative of the interpolation trajectory (x_t is defined in Eq.2):

$$\frac{dx_t}{dt} = x_1 - x_0 \tag{4}$$

The model, conditioned on the maze representation c, estimates the function $v_\theta(x_t, c, t)$, which describes the local direction and rate of this transformation. The network parameters are optimized by minimizing the mean square error between the predicted and target velocities.

$$\mathcal{L}_{\text{FlowPlanner}} = \mathbb{E}_{x_0, x_1, t}\left[\|(x_1 - x_0) - v_\theta(x_t, c, t)\|_2^2\right] \tag{5}$$

FlowPlanner inference Path generation is formulated as a solution to an ordinary differential equation describing continuous dynamics transforming random noise into a solution. In the inference phase, FlowPlanner starts generation with a random noise sample $x_1 \sim \mathcal{N}(0, I)$. Then, the model, conditional on the maze structure c, estimates the vector field $v_\theta(x_t, c, t)$ describing the instantaneous evolution rate of the sample. Based on this velocity, the Sampler (see Fig. 1) updates the state using the explicit Euler schema:

$$x_{t+\Delta t} = x_t + \Delta t\, v_\theta(x_t, c, t) \tag{6}$$

where $\Delta t < 0$ corresponds to backward integration in time from $t = 1$ to $t = 0$, with the interval $[0, 1]$ discretized into T uniform integration steps. This process is repeated iteratively until a final state is obtained corresponding to the path map.

DiffPlanner inference Path generation is implemented via an inverse diffusion process, in which random noise is gradually transformed into a solution map. The process begins with a Gaussian random noise sample $x_T \sim \mathcal{N}(0, I)$. For the subsequent time steps $t = T - 1 \ldots, 0$, the model, conditioned on the maze structure c, predicts the additional noise $\epsilon_\theta(x_t, c, t)$ and then updates the state. During inference, a deterministic DDIM-sampler [12] is used instead of the stochastic DDPM sampler [5]. First, the clean sample is estimated as:

$$\hat{x}_{0,t} = \frac{x_t - \sqrt{1 - \alpha_t}\,\epsilon_\theta(x_t, c, t)}{\sqrt{\alpha_t}}. \tag{7}$$

Next, the transition to the subsequent timestep is performed according to:

$$x_{t-1} = \sqrt{\alpha_{t-1}}\,\hat{x}_{0,t} + \sqrt{1 - \alpha_{t-1}}\,\epsilon_\theta(x_t, c, t). \tag{8}$$

In our work, the DDIM sampler was adopted due to the possibility of selecting a smaller number of steps in the inference process. The effect of choosing the number of steps is shown in the Table. 3.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_29

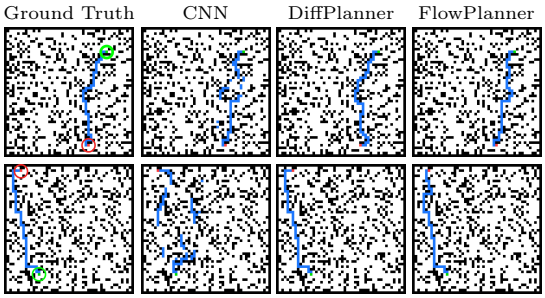

Fig. 2: **Qualitative comparison of paths generated by CNN, DiffPlanner, and FlowPlanner on** 48×48 **mazes.**

4 Experiments

In this section, we present both visual and quantitative experimental results and compare them with those obtained by other methods.

Dataset The dataset consists of randomly generated grid mazes with varying sizes and obstacle densities, where start and goal positions are sampled from free cells. Ground-truth paths are computed using the A* algorithm [3], and samples without valid solutions are discarded. While A* is efficient in this setting, we use 2D mazes as a testbed to evaluate whether generative models can learn spatial reasoning and generalize to more complex planning problems. For each valid sample, four binary masks (obstacles, start, goal, and solution path) are stored, and datasets are generated across multiple grid sizes (8×8 to 48×48),with 5,000/250 samples for 8×8, 10,000/500 for 16×16, and 20,000/1,000 for both 32×32 and 48×48 (train/test splits), and corresponding minimum path lengths of 1, 5, 10, and 20.

Metrics Four metrics assess the quality of the generated paths. *Validity* checks whether the generated path forms a continuous connection between the starting point and the goal. *Single-Path* metric assesses whether the generated trajectory forms a single, unbranched path. *Single-Path* equals 1 only if *Validity*=1, the path has no branches (*Branch-Rate*=0), and exactly two endpoints. *Branch-Rate* measures the percentage of the cells in the path with at least three neighbors (i.e., branching) with lower values indicating more regular trajectories. *Length Ratio* is the ratio of the length of the generated path to the length of the shortest path determined by the A* algorithm.

5 Results

We compare three models: a CNN (U-Net), DiffPlanner, and FlowPlanner, see (Fig. 2). Preliminary experiments with VLMs show that their outputs are inconsistent and frequently fail to produce valid or coherent paths, which aligns with the qualitative failure case presented in Fig. 1. The CNN directly predicts

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_29

Table 1: **Quantitative comparison of CNN, DiffPlanner, and FlowPlanner across different grid sizes.** The best results are achieved by FlowPlanner, which leverages a learned vector field to guide the sample along a continuous trajectory toward the solution.

Maze Size	Model	Validity (%) ↑	Single-Path (%) ↑	Length Ratio ↓	Branch-Rate (%) ↓
8×8	CNN	92.80	89.20	**1.00**	0.29
	FlowPlanner	**94.00**	**92.40**	**1.00**	**0.26**
	DiffPlanner	90.40	77.20	1.02	1.95
16×16	CNN	74.00	65.20	**1.00**	0.82
	FlowPlanner	**88.60**	**86.20**	1.01	**0.19**
	DiffPlanner	84.60	67.60	1.03	2.15
32×32	CNN	49.60	45.20	**1.00**	0.42
	FlowPlanner	82.20	81.60	1.01	**0.05**
	DiffPlanner	**88.60**	**82.80**	1.03	0.44
48×48	CNN	38.70	28.40	**1.00**	0.95
	FlowPlanner	88.00	**86.10**	1.02	**0.09**
	DiffPlanner	**89.00**	76.10	1.04	0.47

Table 2: **Quantitative ablation results for FlowPlanner different conditioning channel combinations on** 48×48 **mazes.** Each condition channel is crucial for accurately generating the trajectory.

Config	Conditioning		Metrics			
	Start,End	Walls	Validity (%) ↑	Single-Path (%) ↑	Length Ratio ↓	Branch-Rate (%) ↓
1	✗	✗	0.00	33.70	N/A	0.18
2	✓	✗	6.40	6.30	1.06	**0.03**
3	✗	✓	0.10	73.30	**1.00**	0.22
Ours	✓	✓	**88.00**	**86.10**	1.02	0.09

Table 3: **Impact of the number of sampling steps on the performance of FlowPlanner and DiffPlanner for** 48×48 **mazes.** FlowPlanner is characterized by high stability regardless of the number of generation steps.

Model	Steps T	Validity ↑	Single-Path ↑	Length Ratio ↓	Branch-Rate ↓
FlowPlanner	50	**88.00**	**86.10**	1.02	**0.09**
	30	87.70	85.50	1.02	0.10
	20	86.40	84.60	1.02	0.10
	10	85.80	83.00	1.02	0.15
	5	79.10	75.90	1.01	0.14
	1	23.70	20.40	**1.00**	0.65
DiffPlanner	50	89.00	**76.10**	1.04	**0.47**
	30	88.00	71.10	1.04	0.62
	20	85.50	65.60	1.03	0.71
	10	81.30	52.00	1.03	1.27
	5	65.70	10.70	1.03	3.91
	1	**98.90**	0.00	**1.00**	71.05

the path, while DiffPlanner and FlowPlanner iteratively construct a solution, starting with random noise and gradually transforming it into a valid trajectory using a diffusion process and a trained vector field, respectively. Results in Table 1 demonstrate a clear advantage of generative methods, with FlowPlanner consistently outperforming the other methods. All models achieve *Length Ratio* values

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_29

close to one, indicating near-optimal path lengths, but only generative methods combine this with high correctness and structural regularity. FlowPlanner achieves the highest *Single-Path* and the lowest *Branch Rate*, producing clean, unbranched trajectories.

We conduct an ablation study on 48×48 mazes to assess the impact of conditioning components, comparing models with no condition, start-goal only, obstacles only, and the full conditioning, see Table 2. Removing any component lead to a significant drop in trajectory quality. The *N/A* value of the *Length Ratio* metric for the configuration without any conditioning results from the lack of valid trajectories generated (*Validity*=0).

Table 3 shows that reducing the number of generation steps degrades performance in both methods, but FlowPlanner remains significantly more stable, maintaining high *Validity* and *Single-Path* with low *Branch-Rate* even at 10 steps. In contrast, DiffPlanner is highly sensitive to step reduction, producing fragmented and branched trajectories and failing to generate meaningful paths at very low step counts, highlighting the efficiency advantage of FlowPlanner.

6 Conclusion

We presented GenPlanner, an approach to path planning with two variants: DiffPlanner and FlowPlanner. Diffusion and flow-based models can act as planning mechanisms, progressively refining solutions while maintaining global consistency and respecting constraints. Experiments showed that the proposed methods significantly outperform the baseline CNN model in terms of the correctness of generated trajectories, their regularity, and structural stability. In particular, FlowPlanner achieves the best results in all key metrics, maintaining high quality even when the number of generation steps is severely limited.

In future work, we plan to extend the framework with mechanisms for refining generated paths, especially in uncertain cases, and to explore applications in continuous and dynamic environments.

Acknowledgments. The authors acknowledge the contribution to this research from the Polish Ministry of Education and Science under grant No. PN/02/0084/2023.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Chang, M., Chhablani, G., Clegg, A., Cote, M.D., Desai, R., Hlavac, M., Karashchuk, V., Krantz, J., Mottaghi, R., Parashar, P., Patki, S., Prasad, I., Puig, X., Rai, A., Ramrakhya, R., Tran, D., Truong, J., Turner, J.M., Undersander, E., Yang, T.Y.: PARTNR: A benchmark for planning and reasoning in embodied multi-agent tasks. In: The Thirteenth International Conference on Learning Representations (2025)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_29

2. Chen, J., He, Q., Yuan, S., Chen, A., Cai, Z., Dai, W., Yu, H., Chen, J., Li, X., Yu, Q., Zhou, H., Wang, M.: Enigmata: Scaling logical reasoning in large language models with synthetic verifiable puzzles. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
3. Futuhi, E., Sturtevant, N.R.: Learning admissible heuristics for a*: Theory and practice. In: The Fourteenth International Conference on Learning Representations (2026)
4. Han, S., Qiu, B., Liao, Y., Huang, S., Gao, C., YAN, S., Liu, S.: Robocerebra: A large-scale benchmark for long-horizon robotic manipulation evaluation. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2025)
5. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
6. Li, Y., Fan, C., Ge, C., Zhao, S.Z., Li, C., Xu, C., Yao, H., Tomizuka, M., Zhou, B., Tang, C., Ding, M., Zhan, W.: WOMD-reasoning: A large-scale dataset for interaction reasoning in driving. In: Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., Zhu, J. (eds.) Proceedings of the 42nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 267, pp. 34288–34311. PMLR (13–19 Jul 2025), https://proceedings.mlr.press/v267/li25l.html
7. Liang, J., Christopher, J.K., Koenig, S., Fioretto, F.: Simultaneous multi-robot motion planning with projected diffusion models. In: Forty-second International Conference on Machine Learning (2025)
8. Lin, B.Y., Bras, R.L., Richardson, K., Sabharwal, A., Poovendran, R., Clark, P., Choi, Y.: Zebralogic: On the scaling limits of LLMs for logical reasoning. In: Forty-second International Conference on Machine Learning (2025)
9. Lu, H., Han, D., Shen, Y., Li, D.: What makes a good diffusion planner for decision making? In: The Thirteenth International Conference on Learning Representations (2025)
10. Małkiński, M., Pawlonka, S., Mańdziuk, J.: Reasoning limitations of multimodal large language models. a case study of bongard problems. In: Forty-second International Conference on Machine Learning (2025)
11. Ryu, H., Kim, G., Lee, H.S., Yang, E.: Divide and translate: Compositional first-order logic translation and verification for complex logical reasoning. In: The Thirteenth International Conference on Learning Representations (2025)
12. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2021)
13. Xu, W., Wang, J., Wang, W., Chen, Z., Zhou, W., Yang, A., Lu, L., Li, H., Wang, X., Zhu, X., Wang, W., Dai, J., Zhu, J.: Visulogic: A benchmark for evaluating visual reasoning in multi-modal large language models. In: The Fourteenth International Conference on Learning Representations (2026)
14. Zou, C., Guo, X., Yang, R., Zhang, J., Hu, B., Zhang, H.: Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. In: The Thirteenth International Conference on Learning Representations (2025)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_29