Navigating 2D environments with diffusion policy

Nathan Cloos **Kavya Anbarasu**

Abstract

Humans demonstrate remarkable abilities in rapid learning and adaptive behavior when faced with novel environments, leveraging a range of priors and inductive biases. Recent progress in AI has shown that large neural networks trained on large and diverse datasets can exhibit rapid in-context learning on tasks that differ from their training objective. However, these models still lack the ability to rapidly learn to interact in grounded environments in a generalizable way. In this work, we focus on a simple path planning behavior and ask our models to robustly generalize across a wide range of visually diverse environments. Our approach leverages diffusion models, commonly used in generative tasks, to produce path trajectories in 2D mazes. We propose a pipeline for generating procedurally diverse environments and evaluate the model's performance on unseen textures and configurations, highlighting its potential for scalable and efficient training of grounded agents. We find that the diffusion model generalizes to unseen test textures. Additionally, the model can sometimes generalize to textures generated from a different text-to-image model than the one used for training. We identify the main source of errors as the model confusing the agent and the goal textures, resulting in trajectories from the goal to the agent instead of the intended direction. Preliminary results suggest that this issue may stem from the quality of the textures generated by the text-to-image model, where some goal textures depict characters that can be easily confused with agent textures.

1 Introduction

When humans encounter novel environments, they rapidly leverage a wealth of priors to understand the goal, identify the agent they control, and determine how to manipulate the agent to achieve the goal [Dubey et al., 2018]. This capability extends even to environments never seen before, aided by our visual priors—such as associating red with danger and green with safety—and the ability to quickly infer control dynamics through minimal interactions. For instance, pressing keys to observe resulting movements allows us to identify the controllable agent in an environment [De Freitas et al., 2023]. Additionally, humans can rapidly learn the rules of a game from a few interactions and explore systematically to determine the type of various objects such as enemy and goal objects[Tsividis et al., 2021].

In contrast, state-of-the-art deep reinforcement learning (RL) agents require extensive training to adapt to new environments. A significant reason for this disparity is that humans approach new environments armed with substantial prior knowledge, whereas RL agents often start from scratch. Recent work has shown promise in enabling RL agents to rapidly adapt and learn from interactions using meta-RL trained on large sets of environments, such as AdA from DeepMind [Team et al., 2023]. After training, these agents can solve novel environments with unseen rules within a few episodes, progressively refining their strategies from exploration to optimal solutions while leveraging memory from previous episodes. However, this worked considered generalization where test environments visually resemble training environments and differ only for underlying interaction rules.

Preprint. Under review.

The challenge remains to develop models that can generalize and quickly learn to navigate visually diverse environments and various action spaces. In this work, we aim to address this by focusing on a simple yet robust behavior across a wide range of environments.

Concurrently, the success of large neural networks trained on vast text datasets has demonstrated emergent abilities in in-context learning [Brown et al., 2020]. By describing a novel task in a prompt and providing a few examples, these models can generalize effectively. However, large language models have been shown to struggle with spatial planning tasks [Momennejad et al., 2023]. Another domain where scale has played a crucial role for the recent progress is robotics. Models trained on extensive human demonstration datasets with behavior cloning methods have shown unprecedented motor control abilities.

Our approach draws inspiration from these advances, aiming to train large models on extensive datasets for interacting and navigating 2D environments. A significant challenge is dataset collection. We propose using classical methods like path planning (e.g., A*) or deep RL to generate demonstration trajectories for given environments and tasks, combined with a pipeline for procedurally generating visually diverse environments. With a dataset of image observations and target action trajectories, we train a behavior cloning model, specifically a diffusion policy model [Janner et al., 2022, Chi et al., 2024], to reproduce path planning trajectories in maze environments and test its ability to generalize to new, unseen visual environments.

Our pipeline leverages large language models to generate diverse prompts for creating 2D environments, encoding human inductive biases into the visual rendering. For example, agents are often depicted as characters and goals as valuable items, reflecting human data patterns. We aim to distill these priors by generating varied data and training large models on it, differing from previous approaches that encoded priors as architectural inductive biases. Instead, we focus on scalable models that learn useful priors from diverse data with randomized irrelevant aspects.

Our contributions include an open-source pipeline for generating random textured mazes, a scalable and accessible method for collecting expert data to evaluate behavior cloning methods, and a framework for rigorously assessing the generalization abilities and scaling laws of these methods in 2D environments. We hope our work is a first step in building models that can reason spatially and rapidly learn to navigate new environments.

2 Related Work

The field of robotic navigation and path planning has seen significant advancements with the integration of machine learning techniques, particularly through the application of novel algorithms like diffusion models. Diffusion models [Ho et al., 2020], originally developed for applications in generative modeling of images and audio, have recently been adapted for sequential decision-making tasks in robotics. These models operate by gradually transforming random noise into a structured output through a reverse Markov process, a methodology that has proven effective in generating high-fidelity samples. In robotics, diffusion models [Wang et al., 2023, Chi et al., 2024] offer a promising approach to generating path trajectories, providing a novel alternative to traditional planners that rely on deterministic algorithms or reinforcement learning techniques. In a study, [Janner et al., 2022] explored the use of diffusion models within maze navigation contexts but did not address the challenge of generalization to new maze configurations. This study provides a valuable baseline by establishing the efficacy of diffusion models in a controlled maze environment, but it leaves open the question of how well such models can adapt to changes in environmental configurations. This gap points to a significant opportunity for extending diffusion models to more robustly handle diverse and unforeseen scenarios in real-world applications. Building on top of this work is the DiPPeR Liu et al. [2023] project which represents a pioneering application of diffusion models in robotic path planning. This approach leverages a diffusion model trained on 10,000 randomly generated maps using Kruskal's Minimum Spanning Tree Algorithm, with the model learning to generate 100 trajectories for each map using the A* algorithm as a baseline for comparison. A key feature of DiPPeR is its demonstrated ability to generalize to novel environments not encountered during training, a significant advancement that addresses the critical challenge of out-of-distribution generalization in robotic path planning. This capability is crucial for practical applications where robots must operate in unpredictable or dynamically changing environments.

Another work in this area is "Simple Hierarchical Planning with Diffusion" Chen et al. [2024] which introduces the Hierarchical Diffuser, a novel framework that enhances diffusion-based planning methods by incorporating hierarchical planning strategies. This model adopts a "jumpy" planning strategy at a higher level, allowing it to maintain a larger receptive field while reducing computational costs. The high-level diffuser generates subgoals, which guide a low-level planner responsible for detailed action sequences. This approach significantly improves planning efficiency and generalization capabilities on long-horizon tasks. The Hierarchical Diffuser outperforms both traditional diffusion models and other hierarchical planning methods, particularly in compositional out-of-distribution tasks. This work is directly relevant to our project as it addresses key challenges in generalization and efficiency, providing a robust foundation for developing adaptable path planning strategies in visually diverse environments.

In this project, we build upon these works by enhancing the environmental diversity in the training of diffusion models. By incorporating randomized visual textures and testing model performance on new renderings, this extension seeks to explore the limits of generalizability and robustness in diffusion-based path planning. This effort aligns with the broader trend in robotics research that emphasizes adaptability and resilience in autonomous systems, which are essential qualities for the deployment of robots in complex, real-world settings.

3 Method

3.1 Procedural environment generation

To enable the agent to navigate to the goal using visual observations, we employ a text-to-image model to randomly create visual textures for the environment. The process begins with a language model generating 10 different high-level visual theme prompts. We use ChatGPT-4 as the language model to generate the prompts for the textures model. The generated themes used here are: Medieval Castle, Futuristic City, Jungle Ruins, Arctic Expedition, Underwater Atlantis, Desert Oasis, Haunted Mansion, Space Station, Steampunk Factory, Candy Land. For each theme, we ask the language model to generate 10 subthemes, each with a description of the agent texture, wall texture, and goal texture[1]. These texture prompts are fed into an image generation model to create the visual textures used to render the maze as image observations for the agent. The text-to-image model runs five times for each set of prompts, resulting in 500 texture sets. We hold out 50 textures corresponding to the "Desert Oasis" theme, leaving 450 textures for training.

For the text-to-image model, we experimented with several models. Initially, we tried popular models like Stable Diffusion, but they didn't perform well due to the high dimensionality of the generated images, which didn't downscale effectively. We then explored models on HuggingFace, including two adapted from Stable Diffusion XL: one finetuned on Pokemon trainer sprites[2] and another on pixel art[3]. We selected the model finetuned to produce Pokemon trainer sprites for generating agent, wall, and goal textures, as it produced the best results. The other model had issues, such as generating textures with multiple instances of characters in the same image.

In addition to the agent, goal, and wall textures, we also experimented with generating textures for empty cells. We found that, often, the contrast between the empty cell texture and the wall texture was insufficient to distinguish the two. Therefore, we opted to use a uniform light gray background for the empty space. In future work, we could prompt the language model to ensure well-contrasted textures and request precise style descriptions, including color.

3.2 Maze generation

The generation of mazes is a fundamental aspect of this project, as it provides the environments necessary for testing the path planning capabilities of the diffusion models. We employed Kruskal's algorithm, a classical minimum spanning tree (MST) method, to generate random mazes, ensuring variability and complexity in the pathfinding challenges presented to the model.

[1] `https://chatgpt.com/share/cf2037f7-3bdb-4824-a3e1-ffc4d1b8f7b0?oai-dm=1`

[2] `https://huggingface.co/sWizad/pokemon-trainer-sprite-pixelart`

[3] `https://huggingface.co/nerijs/pixel-art-xl`

Kruskal's algorithm Kruskal [1956] was chosen for its simplicity and effectiveness in creating mazes with a single large connected component without loops, which resembles real-world navigation challenges. The algorithm begins by treating each cell in a grid as a separate set or tree. It randomly selects edges between adjacent cells and adds them to the spanning tree, provided they do not form a loop with the already added edges, thus connecting two separate trees. This process is repeated until all cells are connected, resulting in a maze that is both challenging and solvable.

The implementation details include initializing a graph where nodes represent grid cells and edges represent possible paths between these cells. Each edge is assigned a random weight to ensure the randomness of the maze generation process. The algorithm processes edges in ascending order of their weights, merging disjoint sets of nodes until all nodes are interconnected without cycles.

We use A* to generate optimal path trajectories from the agent to the goal location as it is a classic path planning approach used search-based path problems, 2D navigation, robotics, etc. Huang [2022] Kusuma et al. [2019]. Trajectories are cropped and padded by repeated the last time step to have a constant length of 64 steps.

Image maze observations are combined with the A* trajectories, which serves as the target outputs for the diffusion model. Notably, data samples are generated on the fly, by creating a random maze and selecting a random texture set from the 450 training textures at each training iteration. This approach enhances the diversity of the training data. Future work will focus on testing the importance of online data generation for the model's ability to generalize to new textures.

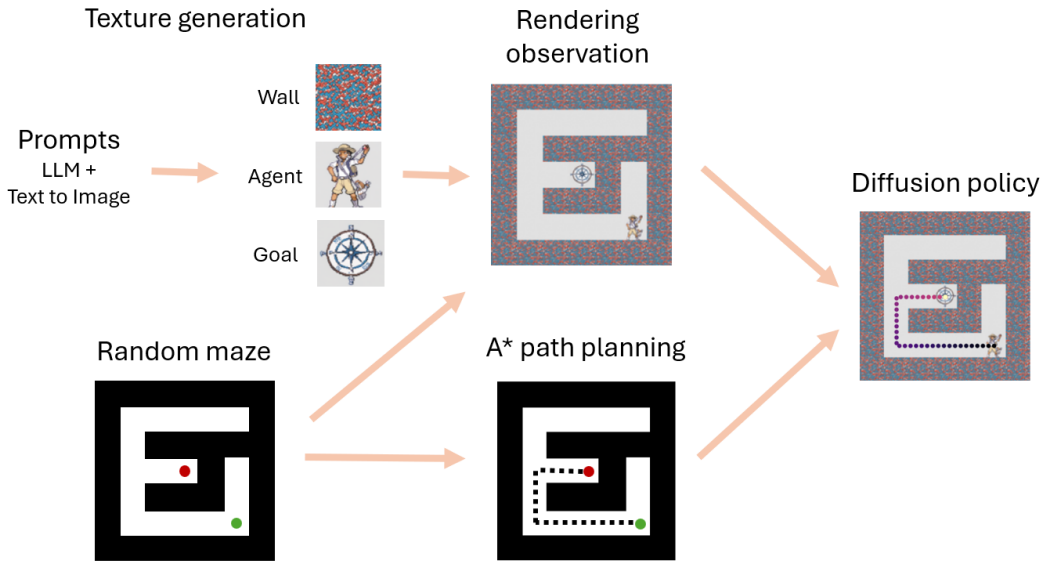

Figure 1: A procedural pipeline to create textured 2D mazes and optimal path trajectories. These mazes serve as training data for agents required to navigate from a randomly sampled start to a goal position based solely on visual input. Texture images are generated in 2 stages involving a pretrained large language model to generate a diverse set of prompts for an image generation model. First, the language model is prompted to generate 10 different theme descriptions for 2D environments and for each theme, to generate 10 subthemes with a description of the wall, the agent, and the goal textures. The wall, agent, and goal texture prompts are then fed to a text-to-image model to generate images. The text-to-image model is run 5 times on the same set of prompts to generate variations of the produced images. We generate a total of 500 texture sets, each comprising a texture image for the agent, the goal, and the walls.We generate random 7x7 maze grids with the Kruskal algorithm. For each of the maze configuration, we use A* to generate optimal path trajectory between a randomly sample a start and goal position. A diffusion policy model is trained to generate the optimal A* trajectories from rendered image observations using the generated textures.

3.3 Diffusion policy

We utilize the code from the diffusion policy paper [Chi et al., 2024]. We use the default hyperparameters of the PushT task. Unlike the original diffusion policy paper, our diffusion model is conditioned only on the image observation and does not include additional agent position input. We use a CNN UNet with 168px by 168px RGB image observations and a horizon action length of 64. In the original paper the diffusion policy is run for a short time horizon and is repeated multiple times within an episode. Here we find that it is enough to run the diffusion model only once and directly generate the entire target trajectory.

We train three models with varying numbers of total training iterations using the 450 texture sets, testing on the remaining 50 validation texture sets. The models are trained for 10,000, 150,000, and 500,000 total training steps, with compute times of 35 hours, 68 hours, and 200 hours, respectively. The models were trained on a single GPU, either an NVIDIA TITAN RTX or an NVIDIA GeForce GTX TITAN X.

The evaluation of our models' performance was carried out by computing the mean squared error (MSE) between the predicted and the target trajectories. To transform this MSE into a practical measure of model accuracy, we established a threshold error level. This threshold was determined visually, ensuring that trajectories falling below this threshold closely approximated the target trajectory in terms of their visual alignment. Accuracy assessments were performed across 20 randomly sampled maze configurations per texture set, with varying start and goal positions.

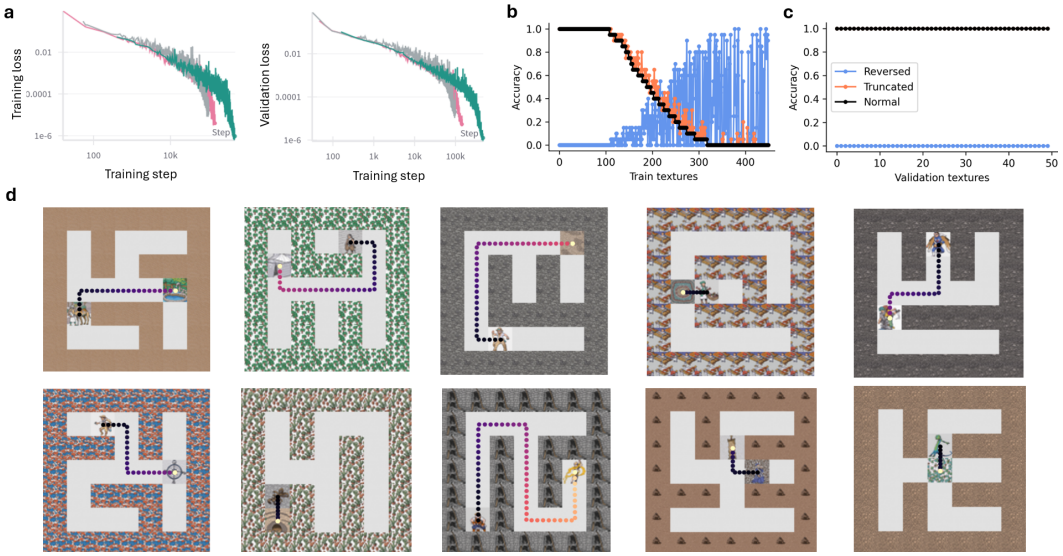

Figure 2: (a) We train 3 random seeds of a diffusion model conditioned on maze image observations to produce A* trajectories with varying number of total training steps (10000 in grey, 150000 in pink, and 500000 in green). The data includes 450 training texture sets and 50 held-out texture sets for model evaluation. We subsequently show analysis results for the 150000 training steps model. (b) Accuracy of generated trajectories when compared to groundtruth, ordered by accuracy value (black). Accuracy is evaluated as the fraction of generated trajectories that fall below an error threshold for a total of 20 mazes with random configuration, holding the visual textures constant. We also show in blue the accuracy if the target trajectories were reversed, going from goal to agent instead of agent to goal. The results show that, when ignoring errors in identifying goal and target, the model correctly predicts the path trajectories in the majority of the train textures. Additionally, we show in orange the accuracy when truncating the path trajectory when it reaches the target object. This shows that some of the errors come from the inability of the model to stop when reaching the target object. (c) We evaluate the trained model on held-out textures. We find a perfect generalization accuracy on the specific texture sets that were held-out here. (d) Examples of generated trajectories for each of the 10 subthemes in the held-out texture theme.

4 Results

We start by asking if a diffusion model trained to produce agent to goal path trajectories on a subset of the visual textures can generalize to unseen textures. We find that the model doesn't capture all the training data as shown by the low accuracy on more than half of the training texture sets (Figure 2, black curve). To our surprise, the model perfectly generalizes to the held-out test trajectories (Figure 2c). We show an example trajectory for each of the 10 subtheme texture sets in the held-out data (Figure 2d).

We examine the types of errors that the model makes on the training environments (Figure 2b, black curve). To solve the task, the model must infer which object is the agent and which is the goal solely from the image observation of the environment. A potential source of error is confusing the agent for the goal and vice versa. In this case, the model would produce the shortest path trajectory from the goal to the agent. We test this by comparing predicted trajectories to reversed trajectories and evaluating accuracy. We find that, for most texture sets where the model accuracy is low, the model predicts the reversed trajectory, as shown in blue in Figure 2b. Additionally, we evaluate another type of error where the model fails to stop the trajectory at the goal position. As described in **??**, when the A* trajectory reaches the goal before the maximum number of steps, the trajectory is padded by repeating the last position. This corresponds to the agent reaching the goal position and waiting there to ensure all trajectories have the same length. We find that truncating the target trajectories when they reach the goal position explains some fraction of the errors (Figure 2b, orange curve).

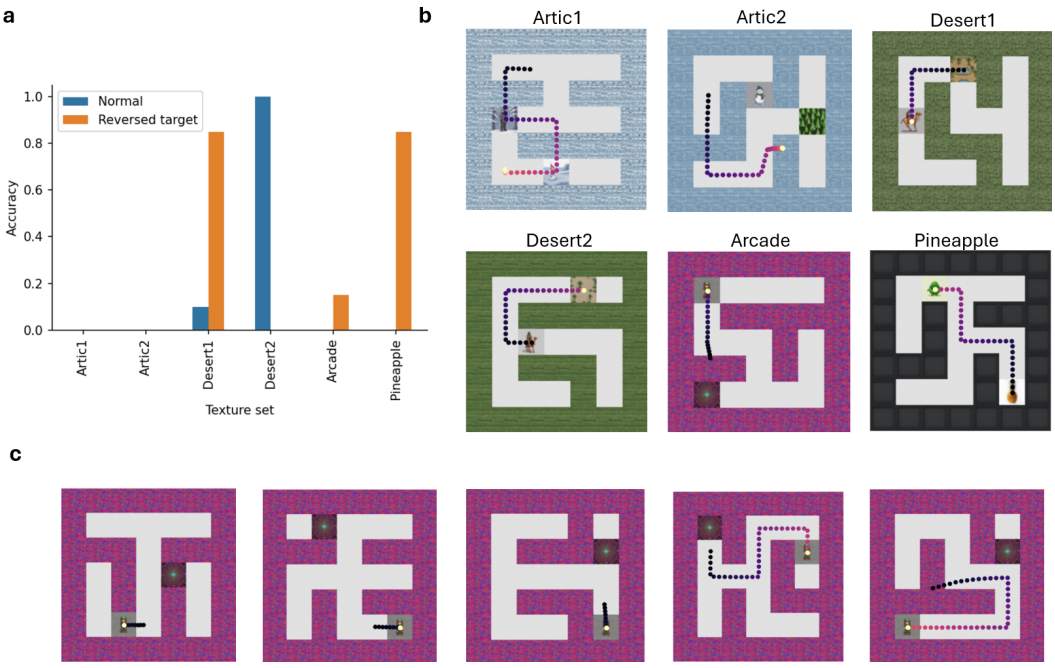

Figure 3: (a) We find that our trained diffusion policy model is able to generalize to novel textures generated by a different text-to-image model that the one used for training. The model matches to the target trajectories for the Desert2 texture set (blue), and the reversed trajectories for the Desert1 and Pineapple texture sets. The model fails on the remaining tested texture sets. (b) Example predictions of the model on a sample maze for each of the tested texture sets. (c) Illustration of an error pattern where the predicted trajectory ends at one object in the maze but fails to start at the other object.

To further assess the model's generalization capabilities, we test its performance in environments generated by a different text-to-image model. The training and test data shown in Figure 2 were both generated by the same text-to-image model but with different prompts. Specifically, the agent, goal, and wall textures were generated by a stable-diffusion model finetuned on Pokemon trainer sprites, which introduces biases such as specific character poses. To evaluate the model's ability to navigate diverse environments with varying visual biases, we generate new texture sets from different text-to-image models, including another stable-diffusion model finetuned on pixel art and a

model generating high-resolution images (Pineapple texture set in Figure 3). The diffusion policy model successfully generalizes to novel textures from a different text-to-image model, matching the target trajectories for the Desert2 texture set (blue) and the reversed trajectories for the Desert1 and Pineapple texture sets, but fails on the remaining tested texture sets. Example predictions on a sample maze for each tested texture set and an illustration of an error pattern are provided in Figure 3.

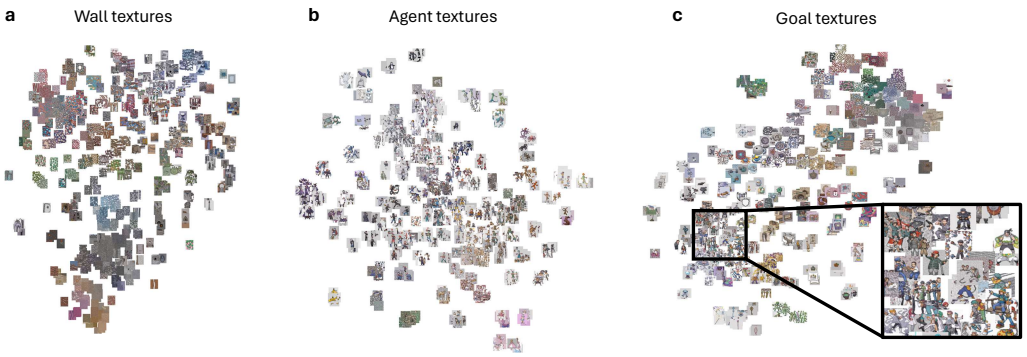

Figure 4: (a, b, c) Embeddings generated by DreamSim [Fu* et al., 2023] for images produced by an image generation model. DreamSim embeddings reflect human-judged similarity between images. (c) We focus on a subset of goal textures where the image generation model, finetuned on Pokemon trainer sprites, produced textures that include a character even if not prompted. For example, a texture prompt for a shield might generate an image with a knight holding the shield, leading to confusion between agent and goal textures.

To further investigate the source of errors, we visualized the generated textures using DreamSim [Fu* et al., 2023] and applied t-SNE to represent them in a 2D plane (Figure 4). The visualization includes 500 wall textures, agent textures, and goal textures. A significant subset of goal textures depicts characters, potentially confusing the model and hindering its ability to differentiate between agent and goal textures. Since the model must predict the target trajectory from a single observation without interactions, this confusion could lead to reversed trajectory predictions. This suggests that improving the texture generation pipeline could enhance model generalization. For example, by using an image-to-text model to filter and relabel textures appropriately.

5 Discussion

In this work, we presented a pipeline for generating visually diverse mazes and demonstrated that a diffusion model trained to reach a goal location can generalize to unseen environment textures. However, our findings indicate that the model does not generalize perfectly. A common mistake is confusing the goal with the agent textures. Improving the data generation pipeline could enhance the model's performance, ensuring the quality of generated textures and their faithful representation of the prompt. Additionally, scaling up the number of texture sets is a straightforward way to potentially improve performance, given our established pipeline.

One significant limitation of our model is its inability to learn from interactions with the environment. Humans rapidly adapt by learning from their interactions, such as observing how objects move when pressing keys to infer which part of the environment is controllable[De Freitas et al., 2023]. Extending our approach to allow for this type of learning could involve training the diffusion model on data generated by an exploration policy. This would also require conditioning the diffusion policy on the entire history of observations, enabling it to learn from interactions. The exploration behavior data could be generated by a meta-reinforcement learning policy that infers which object is the agent in the environment.

Another important aspect to consider is the potential benefit of incorporating hierarchical planning strategies, as demonstrated by Chen et al. [2024]. This approach could significantly enhance the efficiency and generalization capabilities of our model. By adopting a hierarchical planning framework,

we can improve the model's ability to manage long-horizon tasks and reduce computational costs. The integration of hierarchical diffusion models into our pipeline could provide a robust foundation for developing adaptable path planning strategies in visually diverse environments, further enhancing the practical applicability of our approach.

Further, another area of interest is the importance of generating data online, on the fly, which we utilized in this work. This approach increases the diversity of inputs received by the agent. If the sample space is sufficiently large, the agent will likely never encounter the same input twice, preventing task-solving by memorization and potentially improving generalization significantly. This approach aligns with our preliminary results, suggesting that online data generation could be a crucial factor in enhancing the robustness and adaptability of the model.

Another critical area for future research is exploring the scalability of our approach. While our current experiments have shown that diffusion models can generalize to new textures, it remains to be seen how well these models perform as the complexity of the environments increases. Investigating the limits of our model's scalability, both in terms of the size and complexity of the mazes and the diversity of visual textures, will be essential. This includes examining the model's performance in larger and more intricate maze configurations, as well as testing its ability to generalize to environments with more complex visual patterns and textures.

Additionally, our current approach relies heavily on synthetic data generated by text-to-image models. While this has allowed us to create a diverse set of training environments, incorporating real-world data could provide a more robust test of our model's generalization capabilities. Collecting and using real-world images of environments, possibly combined with synthetic data, could help bridge the gap between simulated training and real-world application. This would also involve addressing the challenges of variability in real-world data, such as lighting conditions, occlusions, and varying object appearances.

Lastly, the potential for applying our findings to other domains is significant. The principles of diffusion-based path planning and hierarchical modeling can be extended beyond 2D mazes to more complex robotic navigation tasks, autonomous driving, and even virtual reality environments. Exploring these applications could provide valuable insights into the versatility and robustness of our approach, paving the way for more advanced and adaptable AI systems capable of operating in a wide range of real-world scenarios.

Future work will focus on addressing these limitations and exploring these proposed extensions to enhance the robustness and generalizability of our models across diverse environments and tasks.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Chang Chen, Fei Deng, Kenji Kawaguchi, Caglar Gulcehre, and Sungjin Ahn. Simple hierarchical planning with diffusion. January 2024.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024.

Julian De Freitas, Ahmet Kaan Uğuralp, Zeliha Oğuz-Uğuralp, L. A. Paul, Joshua Tenenbaum, and Tomer D. Ullman. Self-orienting in human and machine learning. *Nature Human Behaviour*, 7 (12):2126–2139, December 2023. ISSN 2397-3374. doi: 10.1038/s41562-023-01696-5.

Rachit Dubey, Pulkit Agrawal, Deepak Pathak, Thomas L. Griffiths, and Alexei A. Efros. Investigating human priors for playing video games, 2018.

Stephanie Fu*, Netanel Tamir*, Shobhita Sundaram*, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv:2306.09344*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

Hongqian Huang. An improved a star algorithm for wheeled robots path planning with jump points search and pruning method. " complex engineering systems. *Complex Engineering Systems*, 2022.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. May 2022.

Joseph B Kruskal, join(' '. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Am. Math. Soc.*, 7(1):48, February 1956.

Mario Kusuma, Riyanto, and Carmadi Machbub. Humanoid robot path planning and rerouting using a-star search algorithm. In *2019 IEEE International Conference on Signals and Systems (ICSigSys)*. IEEE, July 2019.

Jianwei Liu, Maria Stamatopoulou, and Dimitrios Kanoulas. DiPPeR: Diffusion-based 2D path planner applied on legged robots. October 2023.

Ida Momennejad, Hosein Hasanbeig, Felipe Vieira Frujeri, Hiteshi Sharma, Nebojsa Jojic, Hamid Palangi, Robert Ness, and Jonathan Larson. Evaluating cognitive maps and planning in large language models with cogeval. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 69736–69751. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/dc9d5dcf3e86b83e137bad367227c8ca-Paper-Conference.pdf.

Adaptive Agent Team, Jakob Bauer, Kate Baumli, Satinder Baveja, Feryal Behbahani, Avishkar Bhoopchand, Nathalie Bradley-Schmieg, Michael Chang, Natalie Clay, Adrian Collister, Vibhavari Dasagi, Lucy Gonzalez, Karol Gregor, Edward Hughes, Sheleem Kashem, Maria Loks-Thompson, Hannah Openshaw, Jack Parker-Holder, Shreya Pathak, Nicolas Perez-Nieves, Nemanja Rakicevic, Tim Rocktäschel, Yannick Schroecker, Jakub Sygnowski, Karl Tuyls, Sarah York, Alexander Zacherl, and Lei Zhang. Human-timescale adaptation in an open-ended task space, 2023.

Pedro A. Tsividis, Joao Loula, Jake Burga, Nathan Foss, Andres Campero, Thomas Pouncy, Samuel J. Gershman, and Joshua B. Tenenbaum. Human-level reinforcement learning through theory-based modeling, exploration, and planning, 2021.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=AHvFDPi-FA.