Planning in large domains with continuous diffusion and discrete planners

Andreas Burger, Jack Sun, Yifan Ruan

February 2024

Abstract

Recent work has shown that diffusion probabilistic models can be used for planning by iteratively denoising trajectories. Practically these diffusion planners are restricted to small toy domains, where the training data covers the possible states and the denoising is reasonably fast. Classical trajectory planners can generalise rules to large domains, but work best in discrete domains. We propose to combine diffusion and classical planners to tackle large and continuous domains. In the proposed method a classical planner generates subgoals on a discretized domain, and the diffusion planner generates continuous trajectories between the subgoals. We validate our method on 2d mazes like in the original diffusion planning paper, but on a much bigger scale.

1 Introduction

While diffusion-based planners have been successful in smaller task, their performance is limited by the lack of diversity in their training data. In decision-making tasks, the cost of collecting and training on a diverse set of offline training data may be high. Insufficient diversity would impede the ability of the diffusion model to accurately capture the dynamics of the environment and the behavior policy. The problem of limited expert data is particularly prevalent when facing new tasks.

On the other hand, classical planners struggle to generalize past small toy domains with discrete action state spaces. There has been work to tackle this using algorithm-based approaches such as PDDLStream which as made significant progress. However, such approaches have so far shown limited scalability to longer and more complex tasks.

We see potential for combining the strenghts of the two approaches to cover the other's weaknesses. To overcome the problem of lacking data support in large problems, we restrict the diffusion planning to a smaller and simpler sub-regions. This enables learning from smaller amounts of data since the policy only needs to be able to reach local goals. This approach could be used to tackle problems with discrete decision-making under continuous environments such as block-stacking or perhaps navigation.

2 Experimental setup

Our goal is to generalise diffusion planning [Janner et al.,] to larger domains or more complex tasks. For a straight-forward comparison, we build on the same Maze2D environment as the original paper. We procedurally generate bigger mazes by combining the smaller mazes that were used previously. The diffusion planner is trained on the small sub-mazes. To be able to plan trajectories between two subgoals proposed by the planner, it is important that the sub-mazes in training locally look like pieces out of a large maze, and are similar to the gap between two subgoals.

Our diffusion planner is the same as in [Janner et al.,]. The original code of [Janner et al.,] has been published on github (https://github.com/jannerm/diffuser) and integrated into the Huggingface diffusers library. In their architecture, obstacles, start, and goal are described via classifier-guided sampling and image inpainting. In the Maze2D environments in particular goal-reaching is achieved via inpainting. It is possible to improve upon the original architecture, for example by using classifier-free guidance, but that is beyond the scope of this work.

We are still working out the exact experiments we will run. In the paper, they run a block stacking experiment where a particular goal stack only gets a reward in the final step. A potential experiment

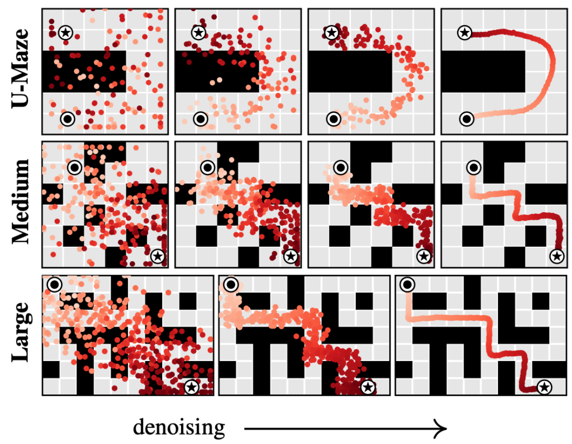

Figure 4. **(Planning as inpainting)** Plans are generated in the Maze2D environment by sampling trajectories consistent with a specified start ◉ and goal ⊛ condition. The remaining states are "inpainted" by the denoising process.

Figure 1: Diffusion planner of [Janner et al.,] in the Maze2D environment.

is to use a classical planning algorithm to generate intermediate configurations of blocks and have the diffusion planner generate the behavior to move from one intermediate subgoal configuration to the next.

3 Related work

Similar to our work, [Yang et al.,] decomposes a complex generation task into multiple simpler subregions, and uses the diffusion model for region-wise compositional generation. Instead of a classical planner, they use a Multimodal LLM as a global planner. They apply their method to image generation, whereas we target navigation.

AdaptDiffuser [Liang et al., a] improves over [Janner et al.,] in Maze2D. Similarly to our work the authors identify the lack of data coverage in complex tasks as a bottleneck for diffusion planners. Their strategy is to generate synthetic trajectories for unseen goals with the diffusion planner, using reinforcement learning rewards as guidance. They then train a discriminator to filter out high-quality trajectories, and train a self-improving, GAN-like fashion. This approach is orthogonal to our method.

Work like SkillDiffuser [Liang et al., b] and DecisionDiffuser [Ajay et al.,] combine a hierarchical planner over skills and a local diffusion planner to generate long-horizon trajectories. They respectively condition the diffusion planner on one-hot skill representations or vector-quantized latents by a sequence model. While their work focuses on abstract skills, we focus on concrete subgoals, and use a classical planner instead of a neural network.

4 Contributions

YR had the idea to combine diffusion planners with classical planners. AB and YR developed the initial idea through disscusions. AB wrote most of the proposal, YR wrote on limitations and the experimental setup of classical planners.

AB will primarily work on the diffusion planner, YR and JS will work more on integrating the classical planner. Generating the large Maze2D task, running experiments, analysing experiments and creating plots, writing the final report will be shared.

References

[Ajay et al.,] Ajay, A., Du, Y., Gupta, A., Tenenbaum, J., Jaakkola, T., and Agrawal, P. Is Conditional Generative Modeling all you need for Decision-Making?

[Janner et al.,] Janner, M., Du, Y., Tenenbaum, J. B., and Levine, S. Planning with Diffusion for Flexible Behavior Synthesis.

[Liang et al., a] Liang, Z., Mu, Y., Ding, M., Ni, F., Tomizuka, M., and Luo, P. AdaptDiffuser: Diffusion Models as Adaptive Self-evolving Planners.

[Liang et al., b] Liang, Z., Mu, Y., Ma, H., Tomizuka, M., Ding, M., and Luo, P. SkillDiffuser: Interpretable Hierarchical Planning via Skill Abstractions in Diffusion-Based Task Execution.

[Yang et al.,] Yang, L., Yu, Z., Meng, C., Xu, M., Ermon, S., and Cui, B. Mastering Text-to-Image Diffusion: Recaptioning, Planning, and Generating with Multimodal LLMs.