BlindBot: Diffusion Policy with Scene-Based Visual Occlusion

Kevin Nguyen & Albert Sun

***Abstract*—The process of training robots to manipulate objects based on visual obstruction remains unevaluated on a wide array of benchmarks. In our study, we aimed to address a facet of performance with the following question: to what extent do diffusion policies remain effective for object manipulation under partial scene observability? To answer the proposed research question, we created hypothetical scenes in simulation with varying levels of scene obstruction. Then, we developed a diffusion policy training pipeline to evaluate performance based on success rate and accuracy. Our results give insight into the limits and spaces for improvement in these models.**

I. Introduction

Diffusion Policy is a relatively new policy that generates robot action trajectories after a denoising process. The authors tested their new policy across 15 different tasks using 4 different benchmarks. In a clean environment, diffusion policy outperformed existing state-of-the-art robot learning methods with an average improvement of 46.9%[1].

However, in the original paper introducing diffusion policy, the authors never rigorously tried to test their policy in a "dirty environment", where the objects that they needed to pick up were never visually occluded, which never provided any problems for the robot in a visual manner. In this paper, we set out to try to test diffusion policy in an environment where the object that the robot needs to use is visually occluded, to see how effective diffusion policy really is in the world of robot learning. We test this under simulation, using RoboSuite to construct and render the environment. We also chose to test scene-based visual occlusion as opposed to sensor-based visual occlusion (adding gaussian noise to the camera) as we believe scene-based visual occlusion is more applicable to real world environments and more important to test rather than just adding noise to the camera.

In this paper, we introduce BlindBot, a model that incorporates diffusion policy and attempts to find a specific color cube that's hidden by multiple other colored cubes. In our research, we seek to evaluate the model on 3 different scenes.

I **Clean Environment** - We first test our model in a clean environment, mainly to compare how the model performs in a clean environment vs a dirtier environment.

II **Partially-Occluded Environment** - This is where the real research begins, to see how the model performs under partial visual occlusion. We evaluate this by partially covering the block we want to pick up with different color blocks.

III **Fully-Occluded Environment** - Here, the block the model is looking to pick up is fully surrounded by different colored blocks. The robot has to push blocks out of the way in order to find the block it's looking for, and pick it up.

This work expands on the paper: *Diffusion Policy: Visuomotor Policy Learning via Action Diffusion* by Cheng Chi et. al.

Authors are with The University of Texas at Austin

II. Related Works

BlindBot is a more narrow approach to a specific problem, that is scene-based visual occlusion. As stated before, BlindBot is just an expansion off of the original paper's model by Cheng chi et al.

Current approaches to solving partial observations are very few and far between in the current world of robotics. Only a few papers have mentioned this topic, and even then they rarely focus on scene-based partial observability[1]. For example, in the original article concerning *Diffusion Policy*, Cheng chi et al. introduced and tested the policy on 15 different tasks. However, when it came to testing it against visual occlusion, we find that their method was limited because their version of visual occlusion was just waving a hand in front of the robot's camera for a few seconds, which tests sensor-based visual occlusion and not scene-based. Many studies have also been conducted with respect to diffusion policy and its role in generalization. Li et al[2] proposed Lan-o3dp which utilized segmented point clouds to improve obstacle-free robustness, Ma et al[3] identified shortcomings in contextual awareness and created a Hierarchal Diffusion Policy agent to streamline long-horizon planning and low-level actions. Hou et al[4] used their system, Dita, to improve the scalability for generalization with diffusion and a transformer backbone. Although all of these studies were able to optimize and improve certain aspects of generalization training, they did not fully evaluate the effects of visual occlusion on diffusion models and instead manipulated camera views.

Another example of a paper that comes close to this topic is *Deep Visual Navigation under Partial Observability*, by Bo Ai et al[5]. This paper introduced the Deep Recurrent Controller for Visual Navigation (DECISION), which is used for visual navigation in both outdoor and indoor terrains. DECISION consists of two key structural components: Multi-scale temporal modeling and Multimodal memory. Multi-scale temporal modeling enlists the help of a CNN for generating control to help with partial observation in navigating an unknown terrain. This paper introduces a novel idea relevant

to our research, but there are too many differences in what we are trying to accomplish. Firstly, they use a quadruped robot for their work, while we are using a stationary Pandas Robot. Secondly, they do not choose to incorporate Diffusion Policy to train their Robot, instead opting for a convolutional neural network (CNN). Similarly, the investigation conducted in *AnyPlace* by Zhao et al[6] was related but different as focus was emphasized on placement generalization that could handle diverse geometries and configurations. Although their multimodal pipeline integrated diffusion policy and aimed to account for unseen objects, they did not experiment with occluded scenes and instead worked with set environments for the purpose of generalization.

Overall, current studies have shown the potential of diffusion-based models in generalization, and many improve on past inefficiencies. However, our aim is to propose our own evaluation of diffusion policy concerning a rather different area that has not been fully investigated into yet.

III. Method

In order to evaluate the viability of diffusion policy in scene-based occluded environments, we propose to gradually increase visual complexity and keep our model constant. To do this, three different configurations were chosen as baselines for testing: A clean environment with no scene occlusion, a partially-occluded environment with lower visibility, and a fully-occluded environment with no visibility. Our aim was to collect separate demonstrations in each environment and train using a set model based on diffusion.

A. Setting up the environment

We used the Robosuite simulator [7] to evaluate the success of diffusion policy in visually occluded environments. In order to test our policy under visual occlusion, we first needed to construct the environment in the simulation. The first environment consisted of multiple-colored blocks sitting on a table with the goal of picking up the red block every time. By keeping the blocks scattered, we were able to form a preliminary setting with little occlusion. We also randomized the positions of the blocks every time we collected a demonstration, as well as the position of the robot end-effector. Our intention behind this was to increase the robustness of our model and capture a variety of solutions. Trajectories were ultimately stored in an HDF5 file, with each containing state observations, end-effector poses, and the positions of the target cube. This process of collecting demonstrations was repeated a second time on a partially occluded environment induced by stacking blocks on top of each other. Whilst collecting demonstrations in this iteration, we knocked over the block covering the target in order to pick it up. Originally, we were planning to collect demonstrations with full occlusion (see *Fig 3.*)

B. Data Processing

Our demonstrations contained state observations including joint positions and end-effector poses. We used Robomimic's[8] infrastructure to slice demonstrations into fixed sequences, which was fed into Robomimic's diffusion policy to condition action predictions. Actions, rewards, and dones were also loaded. The utilization of Robomimic allowed us to separate configuration from execution, thus a JSON file was used to define parameters and build the dataset.

C. Model Details

Our diffusion policy revolved around conditional denoising with multi-step action trajectories. We configured a U-Net with diffusion and training timesteps. They were set to 256 and 100 steps respectively. In addition, we smoothed the Exponential Moving Average for increased sample quality. Finally, additional preparations included an AdamW optimizer for stability and efficiency. 20 total epochs ended up in our final result with a batch size of 32.

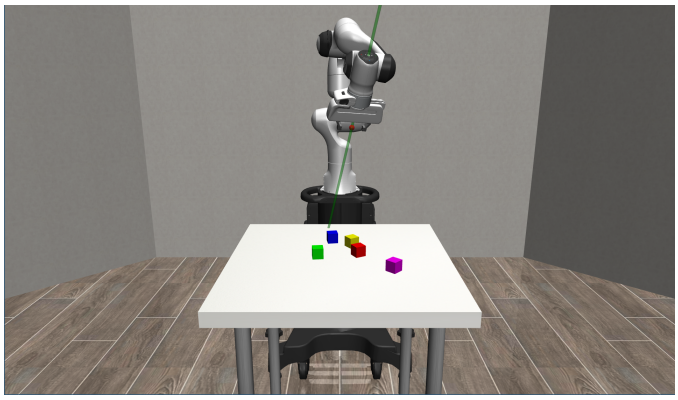

Fig. 1. Here is what our custom environment looks like in the Robosuite simulator. This is the clean environment to evaluate diffusion policy with no occlusion

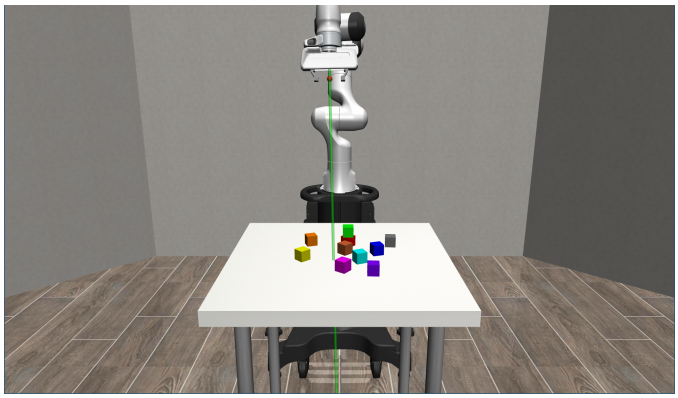

Fig. 2. Robosuite with our partially-occluded environment, in which we decided to stack blocks on top of each other

IV. Experiments

In evaluating the performance of our model, we observed 20 roll outs in our tests for the first environment and 10 for the second. The return of each is shown in the graphs of *Fig. 6* and *Fig. 7*. Our overall metric based on reward was how

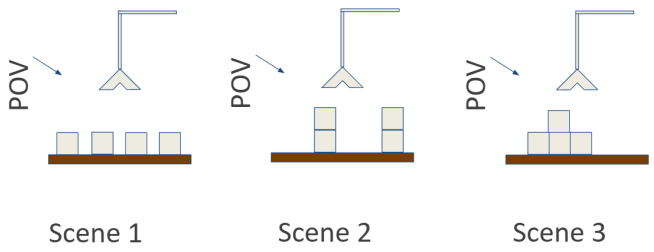

Fig. 3. Our original plan for all three environments, with each scene after the first has increased visual occlusion. Due to limited time, we did not run experiments on the last scene. The POV represents the camera viewing angle.

close the gripper and arm came to the block. Success was logged if the block was successfully gripped, with the return of 1.0 in graph 1 representing this. Results of our experiments on the second scene only generated partial reward, and no full success was achieved. Overall, we can see that diffusion policy has potential even through limited trials.

Fig. 4. An example of the final output from one of our epochs in scene 2. The robot arm consistently drifted away, mostly due to wrong environment configuration.

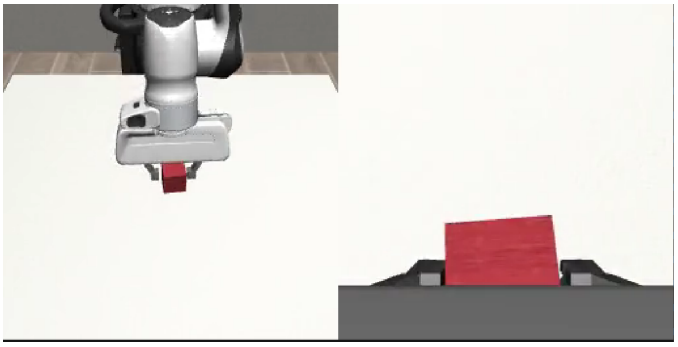

Fig. 5. One of our success trials from scene one epochs. Once receiving confirmation that the arm had gripped the cube, it was logged as a success.

V. Limitations

In configuring the environment and training, there was a big error. In our sleep deprived states we forgot to change the configuration of the JSON file that worked with RoboMimic,

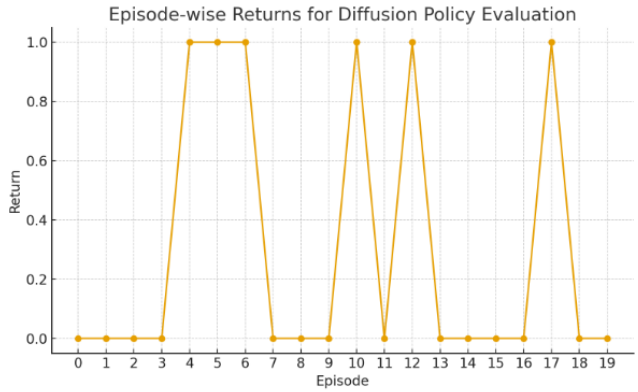

Fig. 6. The return and success rate of our results for the first environment with no visual occlusion. We had an overall success rate of about 30 percent accross all 20 epochs.

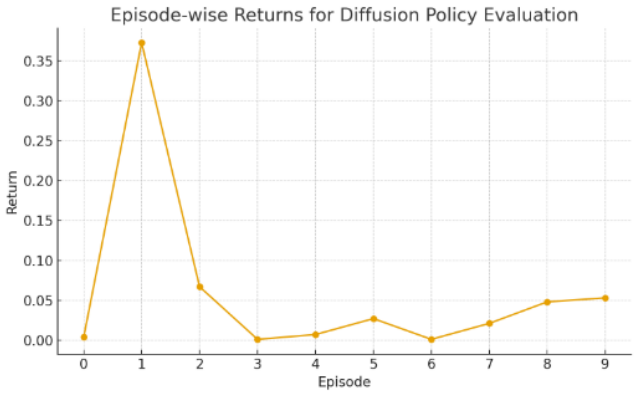

Fig. 7. The return and success rate of our results for the second environment with stacked visual occlusion. We didn't have any success trials overall, however the robot did come close, as shown by return values. This was also partially due to our own limitations.

so RoboMimic was trained on the base environment that it came with and not with the desired scenes it was supposed to train on. Also, we ran out of time to create and train on the 3rd environment (the fully occluded environment), so we were never able to test Diffusion Policy's capabilities on that scene. It should also be noted that there were tiny bugs in the code that we wrote that may be causing our results to be more inaccurate than the experiments actually are. In other words, we didn't focus on fine tuning our policy and this may have caused additional oversight in training.

VI. Conclusion

In this work, we proposed BlindBot, a model created with the purpose of evaluating the potential of Diffusion Policy in visually occluded environments. We used the Robosuite simulation and Robomimic framework to create environments to train our model in, which ultimately showed some success in our experiments. We found that in the end, Diffusion Policy

has the potential to perform well under partial occlusion, if little to no human errors are present. If presented with the opportunity to have more time, we think we could have gotten more meaningful results out of this project, since we could debug our issues, fix the JSON file, and test diffusion policy on the fully occluded environment. Additional implications for our research include further testing in a broader category of visually occluded environments with more complex navigation.

REFERENCES

[1] Cheng Chi, Siyuan Feng, Zhenjia Xu, Eric A Cousineau, Benjamin Burchfiel, Shuran Song, et al. Visuomotor policy learning via action diffusion, September 4 2025. US Patent App. 18/594,842.

[2] Hang Li, Qian Feng, Zhi Zheng, Jianxiang Feng, and Alois Knoll. Generalizable robotic manipulation: Object-centric diffusion policy with language guidance. In *Workshop on Embodiment-Aware Robot Learning*, 2024.

[3] Xiao Ma, Sumit Patidar, Iain Haughton, and Stephen James. Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18081–18090, 2024.

[4] Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, and Yuntao Chen. Dita: Scaling diffusion transformer for generalist vision-language-action policy, 2025. URL https://arxiv.org/abs/2503.19757.

[5] Bo Ai, Wei Gao, Vinay, and David Hsu. Deep visual navigation under partial observability. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9439–9446, 2022. doi: 10.1109/ICRA46639.2022.9811598.

[6] Yuchi Zhao, Miroslav Bogdanovic, Chengyuan Luo, Steven Tohme, Kourosh Darvish, Alán Aspuru-Guzik, Florian Shkurti, and Animesh Garg. Anyplace: Learning generalized object placement for robot manipulation. *arXiv preprint arXiv:2502.04531*, 2025.

[7] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

[8] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *arXiv preprint arXiv:2108.03298*, 2021.