Neural Dynamics Augmented Diffusion Policy

Ruihai Wu[1*], Haozhe Chen[2*], Mingtong Zhang[2*], Haoran Lu[1], Yitong Li[1], Yunzhu Li[3]

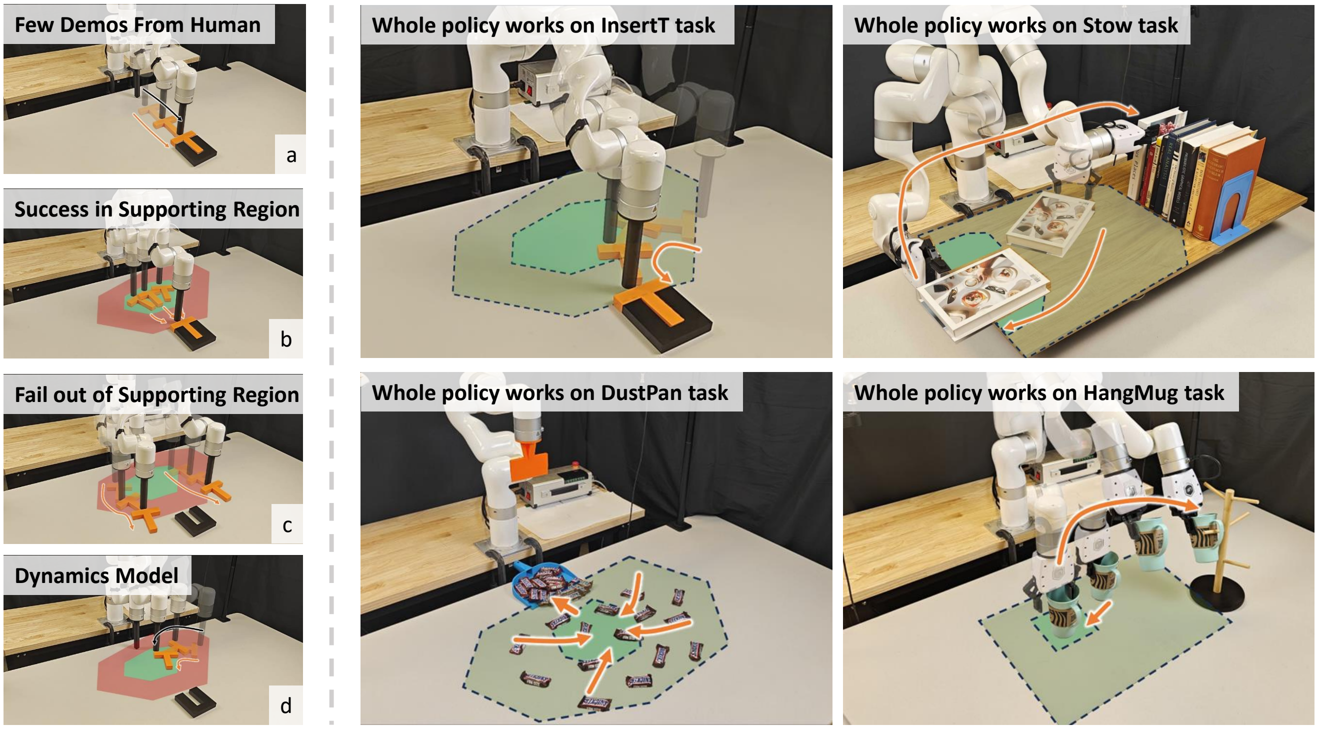

Fig. 1: **(Left)** We propose a neural dynamics-augmented diffusion policy, where a few-shot diffusion policy is enhanced with model-based planning to generalize to broader scene configurations. The **green** region represents the supporting region covered by the few-shot diffusion policy, while the **red** region denotes the space outside this region, which can be covered through model-based planning using the learned dynamics models. **(Right)** The proposed method demonstrates strong performance across various tasks. The **deep green** region represents the area covered by the few-shot diffusion policy, while the **light green** region shows the expanded coverage after augmentation.

***Abstract*— Imitation learning has been proven effective in mimicking demonstrations across various robotic manipulation tasks. However, to develop robust policies, current imitation methods, such as diffusion policy, require training on extensive demonstrations, making data collection labor-intensive. In contrast, model-based planning with dynamics models can effectively cover a sufficient range of configurations using only off-policy data. Yet, without the guidance of expert demonstrations, many tasks are difficult and time-consuming to plan using the dynamics models. Therefore, we take the best of both model learning and imitation learning, and propose neural dynamics augmented imitation learning that covers a large scene configurations with few-shot demonstrations. This method trains a robust diffusion policy in a local support region using few-shot demonstrations and rearranges objects outside this region into it using offline-trained neural dynamics models. Extensive experiments across various tasks in both simulations and real-world scenarios, including granular manipulation, contact-rich task and multi-object interaction task, have demonstrated that trained with only 1 to 30 demonstrations, our proposed method can robustly cover a significantly larger area than the policy trained purely from the demonstrations. Our project page is available at: `https://dynamics-dp.github.io`.**

*Equal contribution. [1]Peking University [2]University of Illinois Urbana-Champaign [3]Columbia University

I. Introduction

Imitation learning [1–4] has been a powerful paradigm for teaching robots to perform complex tasks by mimicking expert demonstrations. However, the efficacy of imitation learning is often contingent on the availability of a substantial number of demonstrations. For instance, current methods like Diffusion Policy [5–7] and Action Chunking with Transformers [8–10], typically require over 200 demonstrations in tasks such as inserting T (Figure 1 left). When the number of demonstrations is limited, shown in Figure 1 (a, b, c), the performance becomes restricted to small regions.

On the other hand, model-based planning can effectively generalize across diverse configurations [11–13]. Learned models offer a number of advantages. First, training data for dynamics models is easy to obtain, from offline sources, self-play or simulations, enabling offline training of dynamics models. Second, it only requires task-agnostic data, eliminating the need for task-specific demonstrations. However, even with the model, planning for complex long-horizon tasks can remain challenging, as sampling precise trajectories is often difficult and time-consuming. In such cases, expert demonstrations can be particularly helpful in guiding manipulation.

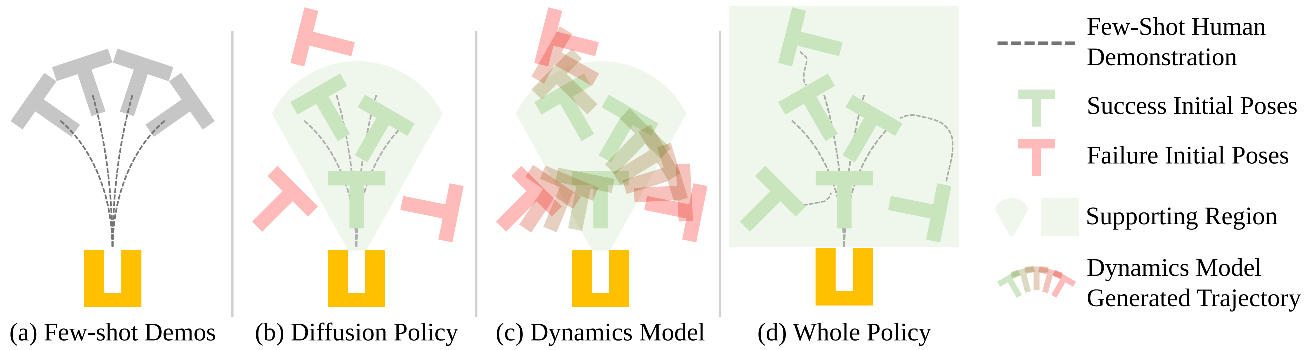

Fig. 2: **Our Proposed Framework.** (a) Collecting few-shot human demonstrations that could cover a small region in the task space. (b) Diffusion policy trained on the few-shot human demonstrations is robust in the local supporting region, but lacks robustness in outside configurations. (c) Model-based planning, guided by learned dynamics models, generates actions to rearrange objects from diverse initial poses to the supporting region. (d) The whole policy, using trajectories from (c), enables robust manipulation across a much larger space.

We propose **neural dynamics augmented diffusion policy**, taking the best of both model learning and imitation learning to complement each other. Our framework trains neural dynamics models that apply to different object categories to explore the entire task space, and a diffusion policy on a few demonstrations to ensure robustness in a local supporting region, with detailed visualization and analysis in Figure 3 and 4. For any object in an arbitrary initial configuration, the dynamics model guides the object into this region, where the few-shot diffusion policy can robustly complete the rest of the task. Shown in Figure 1 (Left), to accomplish a complex task such as the contact-rich InsertT, we imitate few-shot demonstrations to robustly cover a small local region. Then, the neural dynamics model trained by offline interactions between the robot and the T-shape, manipulates the object from random initial configurations into the local region, allowing the robust imitation policy to handle the rest of the task. This manipulation strategy significantly expands the coverage area by augmenting the policy space with neural dynamics models.

II. Method

A. Problem Formulation and Preliminary Background

At each timestep t, policy takes the current observation o_t (*e.g.*, robot state, object point cloud, image and pose), optionally with k previous observations $(o_{t-k},\ o_{t-k+1},\ \dots,\ o_{t-1})$, and proposes the robot action a_t to execute. The policy is successful when the robot manipulates the object to achieve the goal after a sequence of actions.

With a set Q of m demonstrations composed of observation o and action a pairs, *i.e.*, $Q = \{(o_1,\ a_1),\ (o_2,\ a_2),\ \dots, (o_m,\ a_m)\}$, diffusion policy [5] aims to model the conditional distribution $P(A_t|\ O_t,\ \hat{A}_t)$ using diffusion-based models (Denoising Diffusion Probabilistic Models, DDPM). Here, A_t refers to the predicted action sequence $A_t = (a_t,\ \dots,\ a_{t+T_a})$, where T_a is the predicted action horizon. O_t refers to the observation history, including object states (*e.g.*, poses, 3D point cloud or its downsampled particles) and proprioception states, $O_t = (o_{t-T_o},\ \dots,\ o_t)$, where T_o is the history horizon. $\hat{A}_t$ refers to action history, $\hat{A}_t = (a_{t-T_o-1},\ \dots,\ a_{t-1})$.

B. Robust Few-Shot Imitation in a Local Supporting Region

The robustness of imitation methods requires extensive demonstrations. For example, for **InsertT** shown in Figure 2, the T shape can be placed in any positions and orientations, and thus extensive demonstrations with T shapes densely covering the large configuration space are required.

However, collecting extensive demonstrations is expensive. When the demonstration budget is limited, their distribution significantly impacts the policy robustness. We leverage the observation that even a small number of demonstrations can ensure policy robustness if they are concentrated within a small local supporting region with high local density. Specifically, given a set Q of m ($m \leq 30$) demonstrations, with densely distributed object initial configurations (including positions and poses) to be $\{x_1,\ x_2,\ \dots,\ x_K\}$, the local supporting region R that diffusion policy D_R trained on Q maintains high robustness can be represented as the convex region of these demonstrations:

$$R = \{\sum_{k=1}^{K} \alpha_k x_k | \sum_{k=1}^{K} \alpha_k = 1,\ \alpha_k \geq 0\ \forall k\}.$$

Shown in Figure 2 (a, b), trained on a small number of densely distributed demonstrations (each demonstration includes a trajectory from the initial configuration to the target), diffusion policy D_R works robustly on different object configurations (T colored in **green**) in the local supporting region. More visualizations and analysis on the robustness of the few-shot D_R in R are demonstrated in Figure 3.

C. Neural Dynamics Model with Better Spatial Coverage

While D_R demonstrates great robustness in the convex region R, it could not work well to configurations outside R (T colored in **red** as shown in Figure 2 (b)). To enhance robustness, we first introduce an offline learned task-agnostic dynamics model that captures object dynamics and interactions within the scene. We then demonstrate how this model extends the space coverage of few-shot diffusion policy.

To be more specific, a perfect dynamics model f predicts the next state s' given the current state s and action a:

$$s' = f(s,\ a).$$

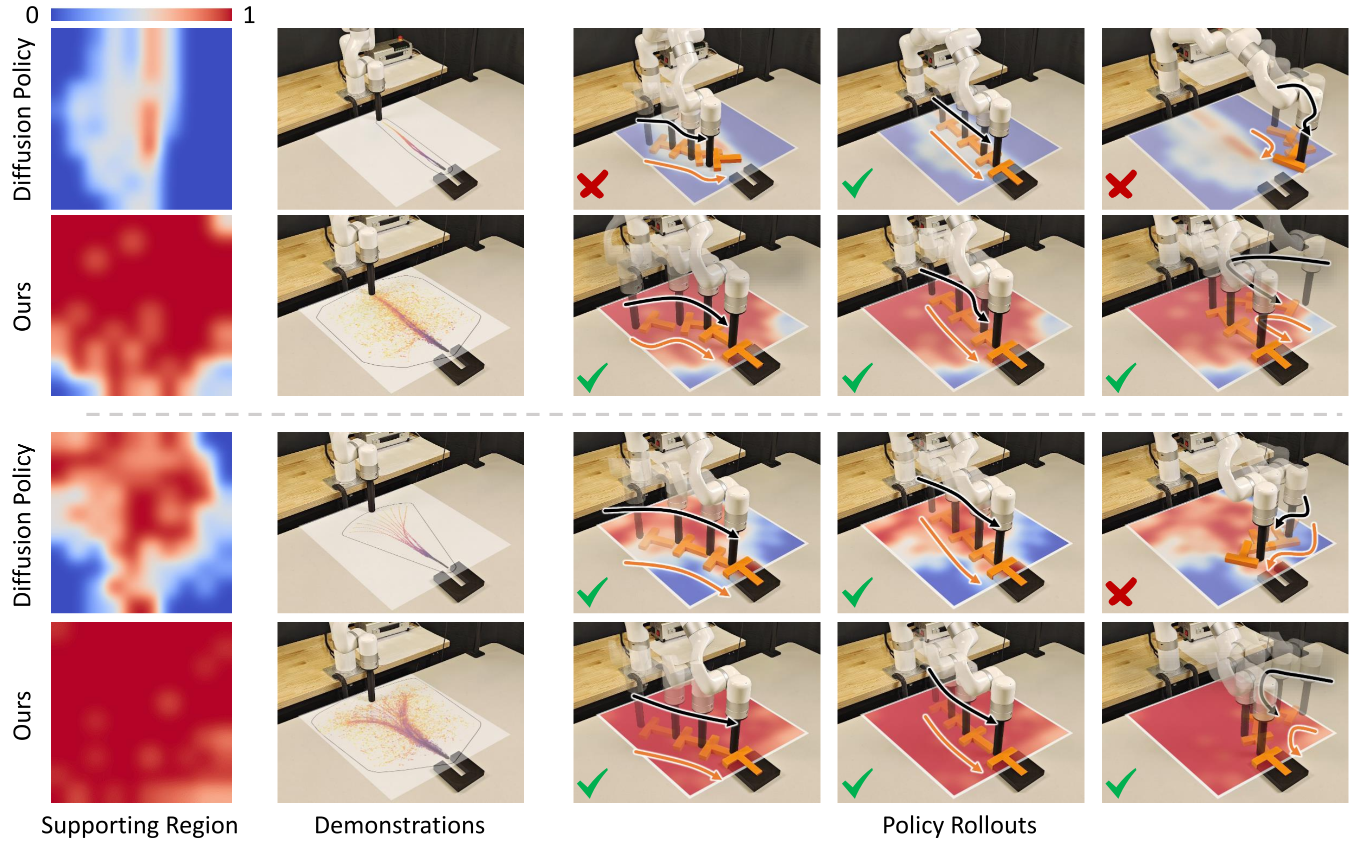

Fig. 3: **Qualitative Analysis on InsertT.** With few-shot demonstrations, while the original diffusion policy demonstrates robustness only in a certain local region, our proposed method enables generalization to a much wider space. Demonstrations in **Diffusion Policy** show the few-shot (10) demonstrations, and those in **Ours** show dynamics augmented demonstrations cover the large space.

In various robotic scenarios where the ground truth object state is difficult to acquire, we use the observed object pose $\xi \in SE(3)$, keypoints $(p_1,\ p_2,\ ...,p_k)$ (k is the number of keypoints), or particles $pt \in R^{N\times 3}$ (N is the number of particles) sampled the scanned point cloud using Farthest Point Sampling (FPS) [14] as the object state representation $\hat{s}$ (in the following paragraphs, we use s to denote the estimated object state for simplicity), and capture the dynamics of the object in the scene using neural networks M. We train M with MLPs or GNNS as as backbones [11, 15]. We randomly initialize the object in the scene, with the estimated state s, execute a random action a, and estimate the new state s' of the object. With the new state predicted by M denoted as $\tilde{s}'$, we use the Mean Squared Error (MSE) distance between $\tilde{s}'$ and s' as the loss function for training M (d denotes the state dimension):

$$\tilde{s}' = M(s,\ a),$$
$$\mathcal{L}_M = \frac{1}{d}\sum_{i=1}^{d} ||\tilde{s}'_i - s'_i||_2^2\,.$$

D. Model-Based Planning with Neural Dynamics Models

Equipped with Model Predictive Path Integral (MPPI) [16] planning algorithm, the robot can manipulate the objects from any initial configuration s to target configuration s_T in the large space U, by optimizing the sampled action trajectories using the distance between s_T and M-predicted object state $\tilde{s}_P$ after $\{\mathbf{a}_t\}$ as the loss function $\mathcal{L}(s_T,\ \tilde{s}_P)$:

$$\{\mathbf{a}_t\} = \text{argmin}_{\{\mathbf{a}_t\}}\mathcal{L}(s_T,\ \tilde{s}_P),$$
$$\tilde{s}_P = M(s,\ \{\mathbf{a}_t\}),$$
$$\mathcal{L}(s_T,\ \tilde{s}_P) = \|s_T - \tilde{s}_P\|_2^2.$$

E. Neural Dynamics Augmented Few-Shot Diffusion Policy

To achieve a policy π with robustness in the large space, instead of directly planning the object to the direct target s_T, we change the target of the planner to the center state s_R of the local supporting region R and first plan the object to s_R:

$$s_R = \frac{1}{K}\sum_{i=1}^{K} x_i,$$
$$\{\mathbf{a}_t\} = \text{argmin}_{\{\mathbf{a}_t\}}\mathcal{L}(s_R,\ \tilde{s}_P),$$
$$\tilde{s}_P = M(s,\ \{\mathbf{a}_t\}),$$
$$\mathcal{L}(s_R,\ \tilde{s}_P) = \|s_R - \tilde{s}_P\|_2^2.$$

Although the actual object state s'_R after planning can deviate from the target s_R by a margin δ_{s_R} due to modeling and execution errors, this error δ_{s_R} is limited, and thus the local supporting region R can easily cover the planned s'_R, as shown in Figure 2 (c). Therefore, followed by D_R robust in R, the object manipulated from the initial s to s'_R can further be robustly manipulated from s'_R to the target s_T.

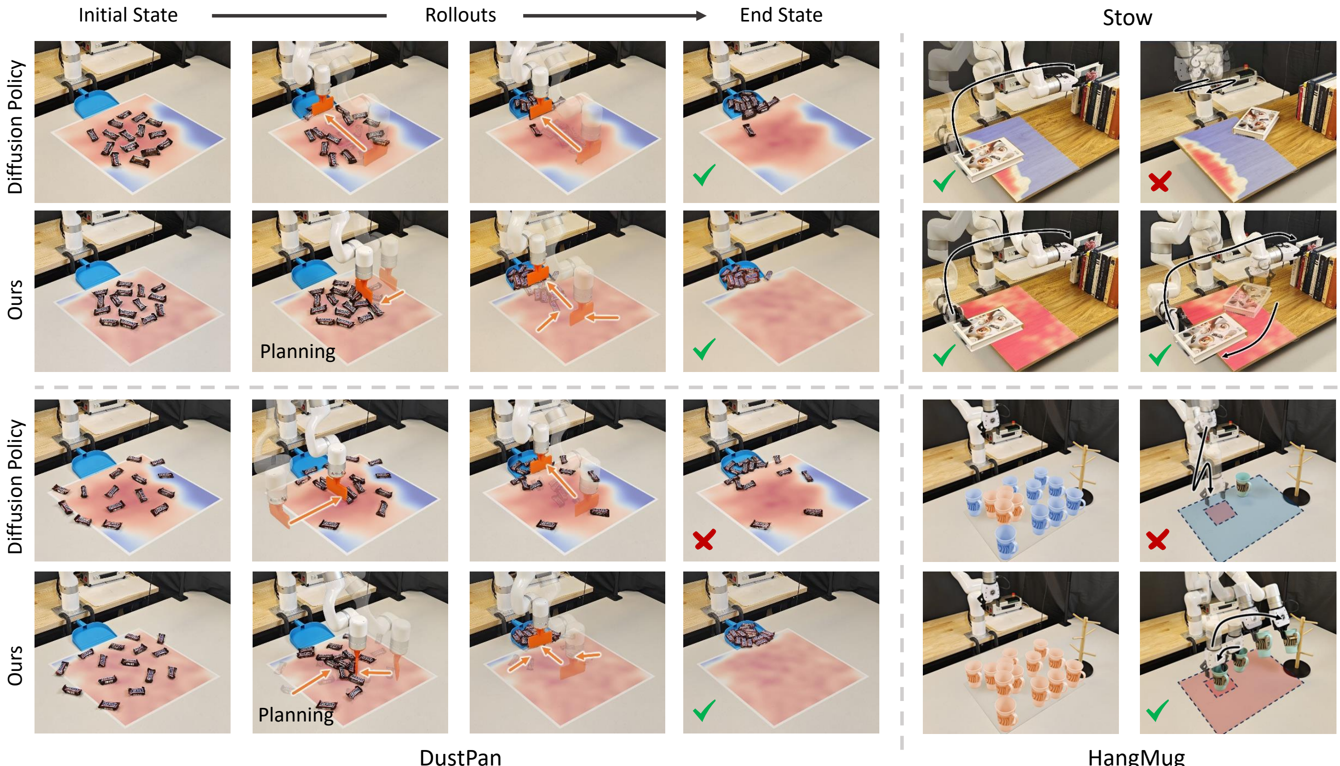

Fig. 4: **Qualitative Analyses.** While diffusion policy only covers specific regions, our method covers a significantly larger space with model-based planning to manipulate diverse objects into the local supporting region, followed by the few-shot diffusion policy. For **DustPan**, "Planning" denotes this step is fulfilled by model-based planning. For **HangMug**, red denotes success and blue denotes failure.

III. Experiments

A. Tasks and Baselines

We conduct evaluation over 4 long-horizon complex tasks:

- **InsertT** that requires inserting a T-Shape into a slot.
- **DustPan** that sweeps sparse granular into the dustpan.
- **Stow** that stows a book onto the bookshelf with books.
- **HangMug** that hangs a randomly located mug on rack.

Figure 1 (right) shows example trajectories of these tasks. We compare with 2 representative manipulation methods:

- **Model-Based Planning** that generates robot actions to achieve the final target using planning methods.
- **Diffusion Policy** trained on few-shot demonstrations.

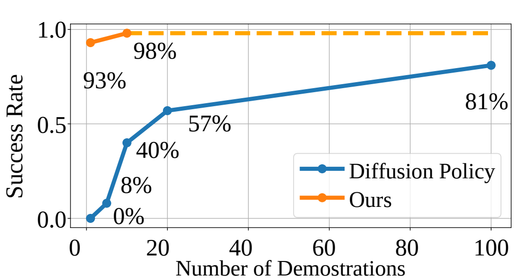

Fig. 5: **Data Efficiency.**

B. Quantitative and Qualitative Results and Analysis

Shown in Table I and II, our method outperforms baselines by a large margin. For **Diffusion Policy** trained on only a few demonstrations, it could only work in the local region. For **Model-Based Planning**, it is difficult and time-consuming to sample precise trajectories that satisfy these complex tasks.

TABLE I: **Quantitative Evaluation in Simulation.**

Method	InsertT	DustPan	Stow
Diffusion Policy	0.1472	0.4144	0.1563
Model-Based Planning	0.1087	0.4307	0.5172
Ours	**0.9827**	**0.8986**	**0.9655**

TABLE II: **Quantitative Evaluation in the Real World.**

Method	InsertT	HangMug	DustPan	Stow
Diffusion Policy	2 / 40	4 / 20	3 / 20	8 / 50
Model-Based Planning	1 / 40	9 / 20	5 / 20	22 / 50
Ours	**32 / 40**	**15 / 20**	**17 / 20**	**42 / 50**

We demonstrate qualitative results and analysis in Figure 3 and Figure 4. It is clear that, diffusion policy trained on few-shot demonstrations can only cover a specific small region with robustness, and will easily fail in manipulating objects initialized outside the region. With the augmentation of neural dynamics models, the whole policy will effectively cover a larger space by guiding the arbitrarily initialized object to the supporting region through model-based planning, followed by the few-shot diffusion policy in this region.

Furthermore, to analyze the data efficiency of our method, *i.e.*, how our policy maintains robustness when the demonstration number is continuously decreasing, we compare the original diffusion policy with our method by conducting experiments respectively using 1 to 100 human demonstrations in **InsertT**. As shown in Figure 5, the original diffusion policy continuously faces performance decrease when the number of demonstrations decreases, while our dynamics augmented method empowers diffusion policy with the robustness even when there exist only a few demonstrations.

References

[1] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[2] P. Bakker, Y. Kuniyoshi *et al.*, "Robot see, robot do: An overview of robot imitation," in *AISB96 Workshop on Learning in Robots and Animals*, vol. 5, 1996.

[3] S. Schaal, "Learning from demonstration," *NeurIPS*, 1996.

[4] S. Stefan, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[5] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[6] H. Ha, P. Florence, and S. Song, "Scaling up and distilling down: Language-guided robot skill acquisition," in *CoRL*, 2023.

[7] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," *arXiv preprint arXiv:2403.03954*, 2024.

[8] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *RSS*, 2023.

[9] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *CoRL*, 2024.

[10] L. X. Shi, A. Sharma, T. Z. Zhao, and C. Finn, "Waypoint-based imitation learning for robotic manipulation," in *CoRL*, 2023.

[11] Y. Wang, Y. Li, K. Driggs-Campbell, L. Fei-Fei, and J. Wu, "Dynamic-resolution model learning for object pile manipulation," *RSS*, 2023.

[12] H. Shi, H. Xu, Z. Huang, Y. Li, and J. Wu, "Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks," *arXiv preprint arXiv:2205.02909*, 2022.

[13] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu, "Robocook: Long-horizon elasto-plastic object manipulation with diverse tools," *arXiv preprint arXiv:2306.14447*, 2023.

[14] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *TIP*, 1997.

[15] H. Chen, Y. Niu, K. Hong, S. Liu, Y. Wang, Y. Li, and K. R. Driggs-Campbell, "Predicting object interactions with behavior primitives: An application in stowing tasks," in *CoRL*. PMLR, 2023, pp. 358–373.

[16] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, 2017.