A Study on Enhancing the Generalization Ability of Visuomotor Policies via Data Augmentation

Hanwen Wang

***Abstract*— The generalization ability of visuomotor policy is crucial, as a good policy should be deployable across diverse scenarios. Some methods can collect large amounts of trajectory augmentation data to train more generalizable imitation learning policies, aimed at handling the random placement of objects on the scene's horizontal plane. However, the data generated by these methods still lack diversity, which limits the generalization ability of the trained policy. To address this, we investigate the performance of policies trained by existing methods across different scene layout factors via automate the data generation for those factors that significantly impact generalization. We have created a more extensively randomized dataset that can be efficiently and automatically generated with only a small amount of human demonstration. The dataset covers five types of manipulators and two types of grippers, incorporating extensive randomization factors such as camera pose, lighting conditions, tabletop texture, and table height across six manipulation tasks. We found that all of these factors influence the generalization ability of the policy. Applying any form of randomization enhances policy generalization, with diverse trajectories particularly effective in bridging visual gap. Notably, we investigated on low-cost manipulator the effect of the scene randomization proposed in this work on enhancing the generalization capability of visuomotor policies for zero-shot sim-to-real transfer.**

I. INTRODUCTION

The generalization ability of visuomotor policies is crucial for robotic manipulation, as it determines whether the model can be effectively deployed. A good policy should be able to handle variations in object layouts, scene changes, and even enable cross-embodiment deployment. Existing approaches enhance the generalization of policies by collecting diverse datasets. Imitation learning methods based on human demonstrations have been proven effective in robotic unstructured manipulation tasks. Typically, human expert demonstration data is collected, which includes action signals generated by human teleoperation [1], [2], [3], [4], [5], [6], [7], [8], [9]. robot observation information such as RGB images from different viewpoints, and the robot's state (usually represented by the end-effector pose). Some studies have also collected task-related language instructions, enabling the policy to generalize object categories using pre-trained encoders from Vision-Language Models (VLM). Generally, imitation learning policies take visual images and the robot's state as inputs and output the action the robot will perform for closed-loop position control [10], [11], [12], [13], [2], [14], [15], [16], [17], [18]. The cost in terms of both human effort and time for data collection is high; therefore, some methods attempt to leverage a small number of human demonstrations to generate large amounts of data for training more generalizable policies across scenes. These methods increase the diversity of trajectory data, enabling position generalization for objects on a desktop, and have achieved notable performance in tasks such as pick-and-place and simple assembly [19], [20], [21], [22], [23].

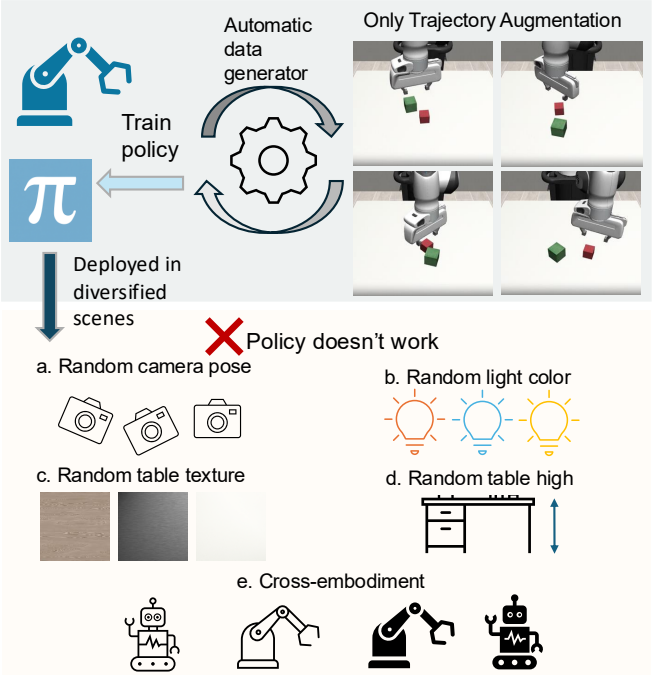

Fig. 1: Previous work has primarily focused on trajectory augmentation methods. However, to obtain a generalizable visuomotor policy, it is necessary to address other factors such as visual discrepancies. As the diversity of the scene increases, such as variations in camera poses, lighting colors, tabletop textures, table heights and cross-embodiment using data augmented by a single method becomes insufficient.

However, prior work has primarily focused on automating the generation of diverse trajectories without accounting for visual discrepancies [19], [22], [23]. For instance, methods like MimicGen [19] record the relative positions between the robot's end effector and the object to be manipulated based on a small number of human demonstrations. Even when the object's placement pose on a horizontal tabletop changes arbitrarily, the robot can still perform the task by continuously tracking both the object and the end effector poses. Nonetheless, relying on this single-type data augmentation proves to be quite limited in enhancing the generalization capability of the policy. Some studies have explored the policy's performance under extensive domain randomization,

University of Chinese Academy of Sciences School of Artificial Intelligence & Institute of Automation, Chinese Academy of Sciences & NLPR

aiming to bridge the visual gap and potentially facilitate sim-to-real policy deployment. Yet, these studies have not systematically assessed the impact of various domain randomization factors, and the conditions they investigate lack sufficient diversity. Moreover, cross-embodiment deployment using a single policy trained with such data has not been sufficiently explored.

Therefore, we systematically studied the impact of different domain randomization factors on the generalization ability of the policy. We focused on factors such as random camera poses, random lighting colors, random tabletop textures, and random table heights, and identified these as key factors influencing the policy's generalization capability. The data generated by existing methods does not cover these randomization factors. In addition, we explore how cross-embodiment data contribute to improving the generalization of visuomotor policies. For lighting and tabletop textures, we randomly sample between them. For camera poses, we ensured that each configuration sampled the same number of poses to maintain a balanced distribution. For cross-embodiment data, we used five manipulators with similar workspaces. To evaluate whether these scene randomization factors facilitate zero-shot sim-to-real deployment in real-world manipulation, we conducted experiments using a low-cost SO-101 manipulator. The results show that policies trained with scene randomization achieve stronger performance when objects are placed in more diverse configurations, effectively reducing the sim-to-real gap. Our contribution can be summarized as follows:

(1) The limitations of different randomization factors on the generalization ability of the policy trained by previous methods were systematically studied.
(2) We systematically investigate whether different randomization augmentation factors can mutually enhance each other to improve the generalization performance of the policy.
(3) We conducted real-world experiments, confirming that policies trained with randomization augmentation facilitate zero-shot sim-to-real transfer.

II. RELATED WORK

A. Manipulation data collection

In the early stages, some works focused on collecting large-scale data for robotic learning. These efforts concentrated on self-supervised learning through trial and error to acquire robot grasping data[24], [25], [26], [27], [28]. Later, some works, such as GraspNet, did not collect robot trajectories but instead used analytical methods to label grasp poses directly [29], [30]. The goal of robotic learning is not only to perform grasps but also to manipulate objects. Some related studies have collected large-scale operation datasets over extended periods of time using a large number of human operators [31], [32], [33], [34], but these methods still consume a significant amount of human and material resources. Some works on automatic data generation aim to address the problem of low-cost collection of large-scale trajectory data [19], [23], [22], [21]. For example, MimicGen [19] efficiently generates a large number of trajectory data with only 10 human demonstration samples for a single task, which is essential for policy training. However, these automatic data generation methods focus solely on the diversity of trajectories, which still limits the generalization capability of imitation learning policies. The method we propose aims to expand the diversity of the data by adding variations such as different camera poses, lighting colors, and textures to address the lack of diversity in the dataset. Furthermore, while previous methods did not account for variations in the vertical placement of objects, we enhance the trajectory diversity by randomizing the tabletop height.

B. Data augmentation in manipulation benchmarks

Evaluating the generalization of a policy requires not only strong trajectory level generalization to handle diverse object configurations in the workspace, but also robust deployment under visual perturbations arising from variations in the scene. Some works have explored trajectory augmentation in simulators generating large amounts of trajectory data across various task settings [23], [23], [35], [36], [37], but they largely overlook visual augmentation. Other works investigate combining trajectory augmentation with multiple visual randomization factors [38], [39]. For example, RoboVerse [38] allows random variations in lighting conditions, background textures, and camera poses. However, these works have not systematically examined how such generalization factors affect the policy's generalization ability, and their randomization settings remain limited. In many cases, camera poses are either manually predefined or only slightly perturbed, which restricts the generalization performance of policies trained with such augmented data. In contrast, our work introduces more principled randomization designs and systematically investigates the impact of different randomization factors on policy generalization.

III. PREREQUISITES

Policy learning. All manipulation tasks in this paper are defined as a Markov Decision Process (MDP), where the robot needs to learn an observation based policy π to model the relationship between visual inputs and robotic action outputs. The collected dataset consists of N demonstrations, denoted as $\mathcal{D} = \{(s_0^i, o_0^i, a_0^i, s_1^i, o_1^i, a_1^i, ..., s_{H_i}^i)\}_{i=1}^N$, where $s \in \mathcal{S}$ denotes the state, $o \in \mathcal{O}$ denote the observations, $a \in \mathcal{A}$ denote the actions. At the beginning of each episode, the initial state s_0^i is sampled from the distribution $D \subseteq S$. When employing imitation learning, we use Behavioral Cloning (BC) [40], which learns a policy by maximizing the likelihood objective.

$$\arg\max_{\theta} \mathbb{E}_{(s,o,a)\sim\mathcal{D}}[\log \pi_\theta(a \mid s, o)].$$

When training online reinforcement learning, we employ the generalized Proximal Policy Optimization (PPO) algorithm. Given an estimate of the advantage $\hat{A}(s, a)$, the PPO update is the sample approximation to

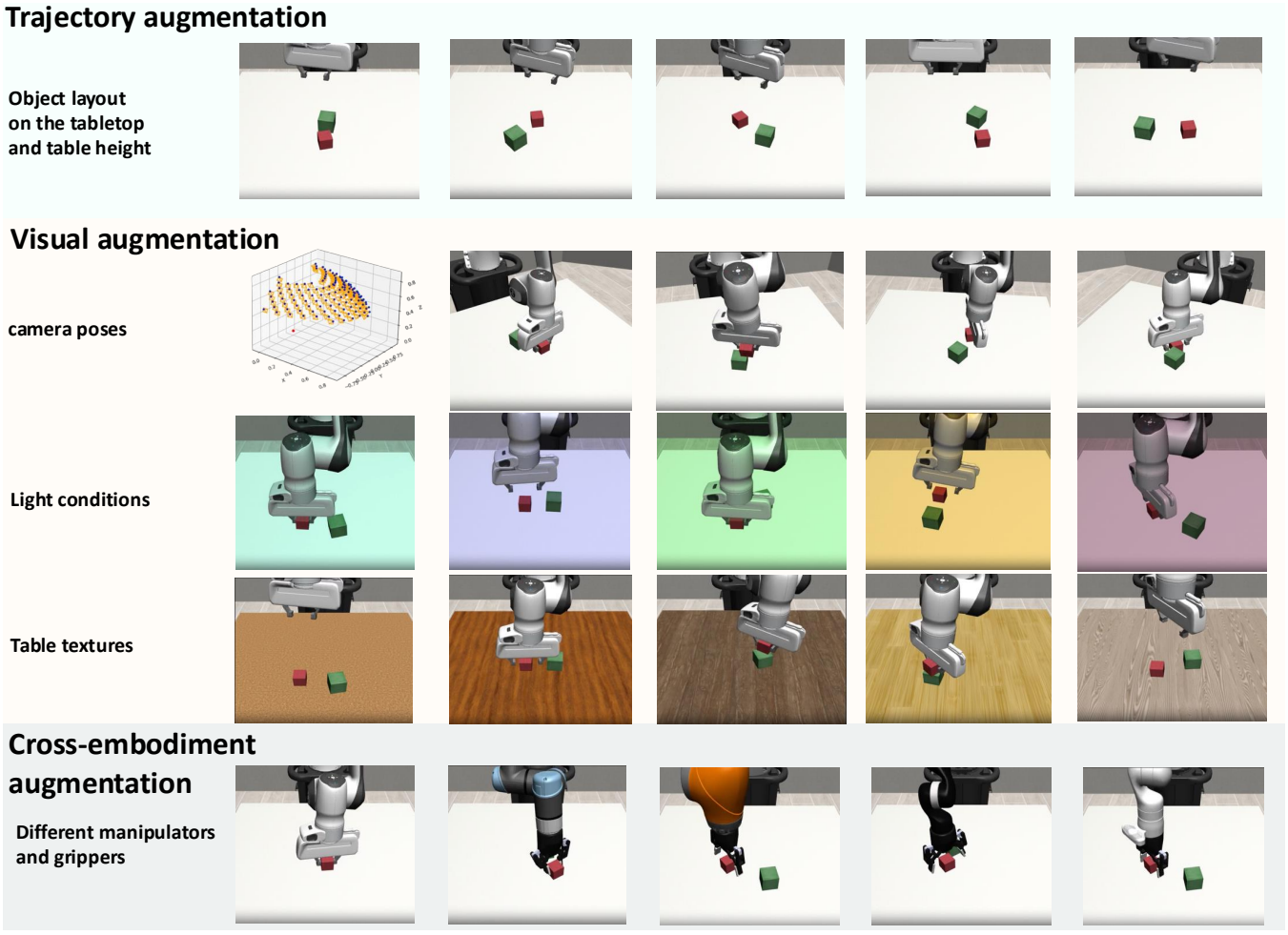

Fig. 2: We focus on applying extensive scene randomization during the collection of robotic manipulation data in order to obtain a broad distribution of demonstrations. Compared to prior work, our trajectory augmentation introduces randomization over table height, as well as visual factors such as camera pose, lighting color, and tabletop texture. Furthermore, our data collection spans five types of manipulators and two types of grippers.

$$\nabla_\theta \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}} \min\Big(\hat{A}^{\pi_{\theta_{\text{old}}}}(s_t,a_t)\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)},$$
$$\hat{A}^{\pi_{\theta_{\text{old}}}}(s_t,a_t)\,\text{clip}\Big(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)},1-\varepsilon,1+\varepsilon\Big)\Big)$$

The clipping ratio ε limits how much the policy can change from the previous policy.

Assumptions. The trajectory augmentation method we used during data collection is inspired by MimicGen [19]. Assumption 1: The action space A for each manipulator includes two components: a pose command to control the end effector and an actuation command for the gripper, which operates on a one-dimensional open/close basis. Assumption 2: We assume that a task can be decomposed into multiple object-centric subtasks, each defined with respect to the coordinate frame of the corresponding object, and that the sequence of subtasks is known. Assumption 3: During data collection, the pose of each object is assumed to be known at the beginning of every subtask.

IV. METHOD

We focus on exploring the use of extensive data augmentations in the scene, As shown in Fig.2, such that after policy learning, the policy acquires generalization capabilities. These generalization abilities manifest mainly as robustness to visual variations in the scene, resistance to background and lighting perturbations, the capability to manipulate objects arranged randomly on a tabletop, and the ability to be deployed across different embodiments. we describe how we introduce diverse randomization settings and present the imitation learning model we employ.

A. Randomized Augmentation

Trajectory augmentation. As stated in Assumption 2, each task can be decomposed into a sequence of object-centric subtasks, where each subtask is defined with respect to the coordinate frame of the manipulated object. Between subtasks, manually specified rule-based signals are available, which allow the automatic detection of subtask termination and thus enable the segmentation of each trajectory into sub-trajectories $\{\tau_i\}_{i=1}^{M}$, with each τ_i corresponding to a distinct subtask S_i. Thus, each trajectory $\tau \in \mathcal{D}_{\text{src}}$ can be represented

as a sequence of consecutive sub-trajectories, D_{src} denotes the small set of demonstration data collected via human teleoperation. Specifically, During trajectory augmentation, suitable subtask segment are first selected from the pool of segments based on the pose of the manipulated object. Next, we treat the subtask segment τ_i as a sequence of end-effector poses in the world frame, $\tau_i = (T_W^{C_{i,0}}, T_W^{C_{i,1}}, \ldots, T_W^{C_{i,K}})$, where C_t is the controller target pose frame at timestep t, W is the world frame, and K is the length of the segment. According to Assumption 3, even if the object is placed arbitrarily on the table, we can transform τ_i based on the object's pose $T_W^{O_0}$ to obtain a new segment, $\tau_i' = (T_W^{C'_{i,0}}, T_W^{C'_{i,1}}, \ldots, T_W^{C'_{i,K}})$, where $T_W^{C'_t} = T_W^{O_0}(T_W^{O_0})^{-1}T_W^{C_t}$. When generating a new trajectory segment, to ensure continuity, we adds an interpolation segment at the beginning of the transformed subtask segment. With this trajectory generation method, we can produce diverse trajectories simply by changing the placement of objects on the table. It is worth noting that previous work only randomized object positions on the tabletop without allowing vertical displacement, which greatly limits the generalization ability of the policy. To ensure that the policy can learn trajectories for objects at different heights, we vary the table height, allowing objects to be arranged at different heights relative to the robot's coordinate frame.

Visual augmentation. We focus on three key factors: the pose of the third-person camera during data collection, lighting color, and tabletop texture, as shown in Fig.2, When deploying a policy, deviations in camera pose from those used during data collection can lead to performance degradation, as such visual discrepancies reduce the policy's generalization capability. To address this, we aim to expose the policy to a broader distribution of viewpoints during training. Specifically, we construct a sphere centered at the robot base coordinate frame and then extract a portion of this sphere by removing viewpoints that are unlikely to be used during deployment and ensuring that the robot's workspace is not heavily occluded. We then sample 100 camera poses uniformly on this surface using Fibonacci sampling. During data collection, to maintain a uniform distribution of viewpoints, we switch to the next camera pose only when trajectory augmentation results in a successful trajectory. For lighting condition randomization, we simulate dim indoor environments by randomizing the intensity of each RGB channel within the range of 0–0.5. For tabletop textures, we select 17 different patterns in RoboSuite [41] and the ARNOLD [42], including common wood and metallic textures.

Cross-embodiment augmentation. The robosuite simulator offers excellent scalability. Once a robot's URDF file is available, trajectory and visual augmentation data collection can be easily deployed on robots with different configurations. As long as the robot's workspace is not significantly different from that of the Franka Panda used in the original human demonstrations, deployment can generally be done directly, perhaps with minor adjustments to the placement of the robot base in space. We selected commonly used laboratory robots such as the UR5e, IIWA, Kinova3, and Jaco. These robots have 6 or 7 degrees of freedom and are capable of performing manipulation tasks. For the end-effector, we chose the Robotiq85Gripper, which is also widely used in the robotics community. In the simulator, the Robotiq85Gripper is represented by six degrees of freedom. To better align its motions with those of the PandaGripper, we mapped the Robotiq85Gripper actions to a single opening/closing degree of freedom, which was also applied to the PandaGripper when feeding these actions as proprioceptive inputs during policy training.

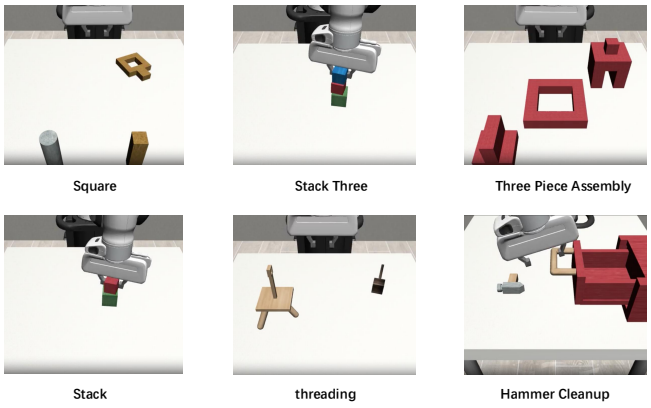

Fig. 3: This work utilizes six robotic manipulation tasks from RoboMimic, focusing primarily on assembly or pick-and-place, which vary in difficulty.

B. Imitation Learning Methodology

We formulate policy learning within the framework of denoising diffusion probabilistic model (DDPM). A diffusion model represents a data distribution $p(x_0)$ via the reverse process of a forward noising process $q(x_k \mid x_{k-1})$, where Gaussian noise is iteratively added to x_0. The reverse denoising process is parameterized by a neural network $\epsilon_\theta(x_k, k)$ that predicts the injected noise. Sampling begins from $x_K \sim \mathcal{N}(0, I)$ and iteratively generates denoised samples following

$$x_{k-1} \sim p_\theta(x_{k-1} \mid x_k) := \mathcal{N}\left(x_{k-1}; \mu_k(x_k, \epsilon_\theta(x_k, k)), \sigma_k^2 I\right),$$

where $\mu_k(\cdot)$ is a deterministic function and σ_k^2 follows a fixed variance schedule. To leverage this framework for control, we adopt Diffusion Policy (DP) [13], which conditions the denoising process on the current observation s. The policy parameterizes $p_\theta(a_{k-1} \mid a_k, s)$, and is trained via behavior cloning by fitting $\epsilon_\theta(a_k, s, k)$ to predict the noise injected into the action sequence. Unlike unimodal Gaussian policies, DP capture the multi-modal distribution of future trajectories without explicitly modeling $p_\theta(a_0 \mid s)$. For temporal consistency, the policy predicts an action chunk $a = a_t, \ldots, a_{t+T_p-1}$ of horizon T_p, from which the robot executes $T_a \leq T_p$ steps before replanning. We adopt a CNN-based implementation of DP, which provides more stable training and requires minimal hyperparameter tuning.

Evaluation env	Stack	StackThree	Square	HammerCleanup	Threading	ThreePieceAssembly
Normal	1.00/-	0.76/-	0.58/-	0.96/-	0.64/-	0.68/-
Camera pose randomization	0.82/**1.00**	0.48/**0.78**	0.10/**0.60**	0.06/**1.00**	0.02/**0.72**	0.00/**0.62**
Light condition randomization	1.00/0.94	0.50/**0.64**	0.46/**0.68**	0.92/**0.96**	0.26/**0.66**	0.22/**0.68**
Tabletop texture randomization	0.26/**0.90**	0.16/**0.74**	0.04/**0.48**	0.20/**1.00**	0.06/**0.58**	0.02/**0.80**
Table height randomization	0.96/0.94	0.58/**0.66**	0.42/**0.52**	0.58/**0.66**	0.10/**0.86**	0.36/**0.76**

TABLE I: Generalization of diffusion policies across randomization factors. Normal represents the original RoboMimic benchmark, The left side represents the performance of the model trained only with data generated by MimicGen as the baseline method, while the right side shows the performance of the model trained with data augmentation tailored to the corresponding test scenario.

Evaluation env	w/o augmentation	CPA	LCA	TTA	THA	CEA
Camera pose randomization	0.25	**0.77**	0.24	**0.28**	0.16	**0.30**
Light condition randomization	0.56	0.28	**0.76**	0.45	**0.63**	0.52
Tabletop texture randomization	0.12	0.08	**0.16**	**0.75**	**0.18**	**0.20**
Table height randomization	0.50	0.17	**0.53**	0.39	**0.73**	**0.56**

TABLE II: The ablation study evaluates whether introducing additional randomization factors during training, under a single-factor setting, improves the policy's generalization in test scenarios relative to a policy trained without augmentation. The table shows the average success rate over six tasks. CPA, LCA, TTA, THA, and CEA denote Camera Pose, Lighting Condition, Table Texture, Table Height, and Cross-Embodiment Augmentations, respectively.

V. EXPERIMENT

A. Experiment setup

We employed six tasks defined in the RoboMimic benchmark, as shown in Fig.3 and generated a large-scale dataset using trajectory augmentation, visual augmentation, and cross-agent augmentation. The dataset contains third-person and wrist-mounted RGB observations, proprioceptive states represented by the end-effector pose, and gripper states represented by the degree of opening. These synthetic data were used to train diffusion policies, which we conducted an in-depth investigation.

For real-world validation of the importance of randomization, we used a low-cost SO-101 manipulator, equipped with an Intel RealSense D435i camera. Online reinforcement learning in simulation was performed on an NVIDIA L20 GPU, while policy inference in real-world deployment was carried out on an NVIDIA GeForce RTX 3050 Ti GPU.

B. Performance Analysis of Policies Trained with Augmentation

We designed a more challenging benchmark compared to RoboMimic. Specifically, instead of only altering the object layout in each scene, we also introduced a set of randomized factors of our own design. We first examined whether models trained with data collected under the original method could handle these more complex scenarios. Then, we trained new models with data augmented by our randomization strategies and investigated whether they could succeed under the same conditions. As shown in Table I, for each scenario we introduced only one randomized factor at a time, and compared the task success rates between the original method [43] and ours. Each task was executed 50 rollouts with randomized scene configurations, and the success rate was reported. Experimental results indicate that models trained with the original method suffer from performance degradation once additional randomization factors are introduced. In contrast, models trained with our randomized augmentation demonstrate improved capability in these scenarios. These results validate the significance of studying the generalization ability of policies under such conditions, and further highlight that datasets generated by existing methods fail to sufficiently cover these challenging scenarios.

C. Ablation Study

We design a set of experiments to investigate how adding different randomization factors during data collection influences policy training, and whether these factors can promote complementary generalization performance across tasks. To investigate how different randomization factors contribute to the generalization of visuomotor policies, we design experiments where data is collected under scenes augmented with individual factors. Specifically, for each task, we introduce a single randomization factor to generate large-scale augmented trajectories. After training policies on these trajectories, we evaluate them in environments augmented with other randomization factors and report the task success rates. As shown in Table II, The success rate reported in the table represents the average success rate across six tasks.

The experimental results indicate that these randomization factors has a mutually reinforcing effect on improving the generalization capability of the policy. after trajectory augmentation, such as training with height randomization and cross-embodiment data, the policy exhibits a certain degree of generalization even in visually randomized test environments. The cross-embodiment data inherently contain visual diversity, which further enhances this generalization capability. Although policies trained with visual augmentation show limited generalization in scenarios with randomized table heights, a policy trained under one type of visual randomization still demonstrates reasonable performance when evaluated under another visual randomization condition.

D. Real-World Experiment

To investigate whether introducing randomization during policy training facilitates zero-shot sim-to-real deployment, we set up a real-world experimental scene consisting of a low-cost SO-101 manipulator and a third-person Realsense

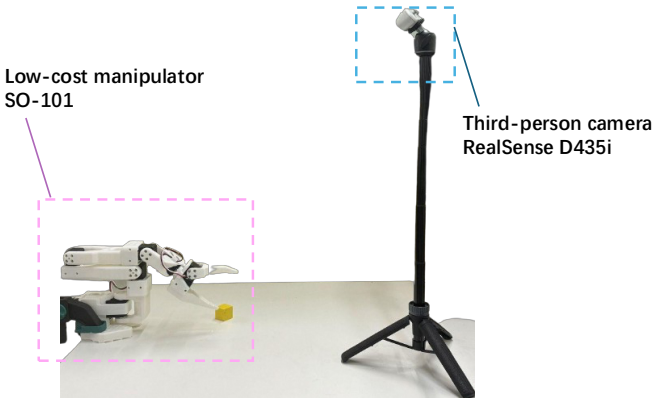

Fig. 4: Our real-world experimental platform consists of a low-cost SO-101 manipulators on the left side and a RealSense camera placed on the right side from a third-person perspective.

D435i camera, as illustrated in Fig.4. Following the Grasp Cube task defined in ManiSkill3 [44], we trained a multi-layer convolutional neural network policy with the PPO algorithm in simulation. In the training environment, we incorporated multiple randomization factors, including camera pose randomization, lighting condition randomization, object color randomization, and table height randomization. For comparison, we also trained a baseline policy in a separate environment without applying any randomization factors.

The experimental results, as shown in Table III, demonstrate that policies trained with visual randomization achieve higher success rates and perform better when objects are placed over a wider range or with varying poses. Such enhancements in visual and trajectory diversity contribute significantly to enabling zero-shot sim-to-real robotic manipulation.

Training Env	Grasp Cube
w/o randomize augmentation	0.28
randomize augmentation	**0.44**

TABLE III: Real-world experiment result.

VI. Conclusion

In this work, we systematically investigate the impact of scene and embodiment randomization on the generalization ability of visuomotor policies. By introducing trajectory augmentation and multiple visual randomization factors. On our proposed challenging benchmark, the policy trained with our method exhibits stronger generalization capability compared to prior approaches. We also found that the inclusion of these randomization factors has a mutually reinforcing effect on enhancing generalization capability. Real-world experiments show that randomized augmentation enhances the policy's zero-shot sim-to-real deployment capability.

VII. Limitation

If a policy is trained solely on the data provided in simulation in this work, it is difficult to perform complex manipulation tasks in the real world. In our simulation experiments, we did not explicitly align the simulated scenes with real-world scenarios, which results in a significant visual gap and some dynamics gap. Moreover, assembly tasks require not only diversity in visual and trajectory data but also an understanding of certain physical-world rules, which is challenging to achieve through purely vision-based imitation learning. A feasible approach is to first roughly align the visual configuration between the real and simulated environments, and reinforcement learning can be used to collect data while simultaneously applying both visual and trajectory augmentation., thereby acquiring more knowledge through interactions with the physical world.

References

[1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[2] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, "Open-television: Teleoperation with immersive active visual feedback," *arXiv preprint arXiv:2407.01512*, 2024.

[3] H. Fang, H.-S. Fang, Y. Wang, J. Ren, J. Chen, R. Zhang, W. Wang, and C. Lu, "Airexo: Low-cost exoskeletons for learning whole-arm manipulation in the wild," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 15 031–15 038.

[4] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

[5] A. Iyer, Z. Peng, Y. Dai, I. Guzey, S. Haldar, S. Chintala, and L. Pinto, "Open teach: A versatile teleoperation system for robotic manipulation," *arXiv preprint arXiv:2403.07870*, 2024.

[6] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[7] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik, "Learning visuotactile skills with two multifingered hands," *arXiv preprint arXiv:2404.16823*, 2024.

[8] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto, "On bringing robots home," *arXiv preprint arXiv:2311.16098*, 2023.

[9] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel, "Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 156–12 163.

[10] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, *et al.*, "Palm-e: an embodied multimodal language model," in *Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 8469–8488.

[11] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6892–6903.

[12] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, *et al.*, "*pi*_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

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

[14] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[15] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," *arXiv preprint arXiv:2502.19645*, 2025.

[16] J. Liu, H. Chen, P. An, Z. Liu, R. Zhang, C. Gu, X. Li, Z. Guo, S. Chen, M. Liu, *et al.*, "Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model," *arXiv preprint arXiv:2503.10631*, 2025.

[17] R. Zheng, Y. Liang, S. Huang, J. Gao, H. Daumé III, A. Kolobov, F. Huang, and J. Yang, "Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies," *arXiv preprint arXiv:2412.10345*, 2024.

[18] Y. Wang, X. Li, W. Wang, J. Zhang, Y. Li, Y. Chen, X. Wang, and Z. Zhang, "Unified vision-language-action model," *arXiv preprint arXiv:2506.19850*, 2025.

[19] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, "Mimicgen: A data generation system for scalable robot learning using human demonstrations," *arXiv preprint arXiv:2310.17596*, 2023.

[20] S. Deng, M. Yan, S. Wei, H. Ma, Y. Yang, J. Chen, Z. Zhang, T. Yang, X. Zhang, H. Cui, *et al.*, "Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data," *arXiv preprint arXiv:2505.03233*, 2025.

[21] Y. Mu, T. Chen, Z. Chen, S. Peng, Z. Lan, Z. Gao, Z. Liang, Q. Yu, Y. Zou, M. Xu, *et al.*, "Robotwin: Dual-arm robot benchmark with generative digital twins," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 27 649–27 660.

[22] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of everyday tasks for generalist robots," *arXiv preprint arXiv:2406.02523*, 2024.

[23] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. Fan, and Y. Zhu, "Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning," *arXiv preprint arXiv:2410.24185*, 2024.

[24] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[25] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on robot learning*. PMLR, 2018, pp. 651–673.

[26] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, "Robonet: Large-scale multi-robot learning," *arXiv preprint arXiv:1910.11215*, 2019.

[27] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 3406–3413.

[28] D. Kalashnikov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman, "Mt-opt: Continuous multi-task robotic reinforcement learning at scale," *arXiv preprint arXiv:2104.08212*, 2021.

[29] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3929–3945, 2023.

[30] H. Wang, Y. Zhang, Y. Wang, and J. Li, "6-dof grasp detection in clutter with enhanced receptive field and graspable balance sampling," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 3000–3007.

[31] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, *et al.*, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *Conference on Robot Learning*. PMLR, 2018, pp. 879–893.

[32] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[33] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[34] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, "Bridge data: Boosting generalization of robotic skills with cross-domain datasets," *arXiv preprint arXiv:2109.13396*, 2021.

[35] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, "Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations," *arXiv preprint arXiv:2107.14483*, 2021.

[36] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[37] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 3, pp. 7327–7334, 2022.

[38] H. Geng, F. Wang, S. Wei, Y. Li, B. Wang, B. An, C. T. Cheng, H. Lou, P. Li, Y.-J. Wang, Y. Liang, D. Goetting, C. Xu, H. Chen, Y. Qian, Y. Geng, J. Mao, W. Wan, M. Zhang, J. Lyu, S. Zhao, J. Zhang, J. Zhang, C. Zhao, H. Lu, Y. Ding, R. Gong, Y. Wang, Y. Kuang, R. Wu, B. Jia, C. Sferrazza, H. Dong, S. Huang, Y. Wang, J. Malik, and P. Abbeel, "Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning," 2025. [Online]. Available: https://arxiv.org/abs/2504.18904

[39] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Q. Liang, Z. Li, X. Lin, Y. Ge, Z. Gu, *et al.*, "Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation," *arXiv preprint arXiv:2506.18088*, 2025.

[40] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in neural information processing systems*, vol. 1, 1988.

[41] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, K. Lin, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," in *arXiv preprint arXiv:2009.12293*, 2020.

[42] R. Gong, J. Huang, Y. Zhao, H. Geng, X. Gao, Q. Wu, W. Ai, Z. Zhou, D. Terzopoulos, S.-C. Zhu, *et al.*, "Arnold: A benchmark for language-grounded task learning with continuous states in realistic 3d scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[43] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 1678–1690.

[44] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin, Y. Liu, T.-k. Chan, *et al.*, "Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai," *arXiv preprint arXiv:2410.00425*, 2024.