

Stem-OB: GENERALIZABLE VISUAL IMITATION LEARNING WITH STEM-LIKE CONVERGENT OBSERVATION THROUGH DIFFUSION INVERSION

Anonymous authors

Paper under double-blind review

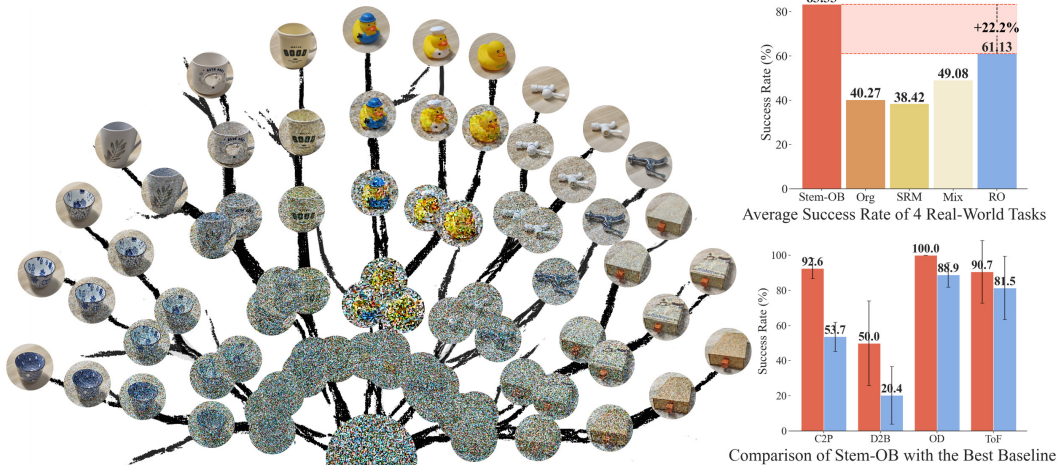


Figure 1: **Left:** The tree of *Stem-OB* inversion is composed of different objects progressively inverted through a diffusion inversion process. Moving downward along the tree’s branches, objects of different textures, appearances, and categories gradually get closer, eventually converging into the same root of Gaussian noise, where they are completely indistinguishable. **Right:** Real-world tasks success rate, where *Stem-OB* showcases a significant improvement.

ABSTRACT

Visual imitation learning methods demonstrate strong performance, yet they lack generalization when faced with visual input perturbations like variations in lighting and textures. This limitation hampers their practical application in real-world settings. To address this, we propose *Stem-OB* that leverages the inversion process of pretrained image diffusion models to suppress low-level visual differences while maintaining high-level scene structures. This image inversion process is akin to transforming the observation into a shared representation, from which other observations also stem. *Stem-OB* offers a simple yet effective plug-and-play solution that stands in contrast to data augmentation approaches. It demonstrates robustness to various unspecified appearance changes without the need for additional training. We provide theoretical insights and empirical results that validate the efficacy of our approach in simulated and real settings. *Stem-OB* shows an exceptionally significant improvement in real-world robotic tasks, where challenging light and appearance changes are present, with an average increase of **22.2%** in success rates compared to the best baseline. See our website for more videos.

1 INTRODUCTION

Visual Imitation Learning (IL), where an agent learns to mimic the behavior of the demonstrator by learning a direct mapping from visual observations to low-level actions, has gained popularity in recent real-world robot tasks (Chi et al., 2023; Zhao et al., 2023; Wang et al., 2023; Chi et al., 2024;

Ze et al., 2024). Despite the versatility demonstrated by visual IL, learned policies are often brittle and fail to generalize to unseen environments, even minor perturbations such as altering lighting conditions or changing the texture of the object may lead to failure of the learned policy (Xie et al., 2023; Yuan et al., 2024b). The underlying reason is that the high-dimensional visual observation space is redundant with virtually infinite variations in appearance that are irrelevant to the task and hard to generalize.

As human beings, we can easily manipulate objects that have different appearances. For example, we can pick up a cup of coffee regardless of its color, texture, or the lighting condition of the room. This is partially because our visual system is capable of abstracting the high-level semantics of the scene, such as the silhouette of the object, the structure and arrangement of different objects, etc in a hierarchical manner (Hochstein & Ahissar, 2002), effectively merging scenes with perceptual differences to similar “meta” observations.

Augmentation techniques such as Spectrum Random Masking (SRM) (Huang et al., 2022) and Mixup (Zhang et al., 2018) remove details from observations to encourage the model to focus on structural features; however, they lack the ability to distinguish between low-level and high-level features. It is preferable if we can sweep the photometrical differences while maintaining the high-level structure for the scene. Achieving this requires a semantic understanding of the observations, and naively perturbing the data with Gaussian noise can lead to irreversible information loss.

Pretrained large image diffusion models, such as Stable Diffusion (Rombach et al., 2022; Esser et al., 2024), embed essential world knowledge for visual understanding. Apart from synthesizing new images from random noise, these models are capable to perform a reverse procedure called inversion (Song et al., 2022), which converts an image back to the space of random noises. A recent study (Yue et al., 2024) indicates that this inversion process selectively eliminates information from the image. Rather than uniformly removing information from different semantic hierarchies, it will push those images with similar structures closer in the early stages of the inversion process. Inversion is like the reprogramming of a differentiated cell back to a stem cell, which bears the totipotency to differentiate into any cell type. This characteristic aligns perfectly with our will of enhancing the robustness and generalizability of visual IL algorithms to visual variations. To distill such property into a visual IL policy, we propose an imitation learning pipeline which applies diffusion inversion to the visual observations. We name our method *Stem-OB* to highlight the similarity between the inversed observation and the stem cell in biology, as illustrated in Figure 1.

To be specific, our method is as simple as inverting the image for reasonable steps before sending them to the downstream visual IL algorithms, effectively serving as a preprocessing step that converges low-level appearance differences into similar high-level structures. From this perspective, our approach fundamentally distinguishes from generative augmentation methods, which aim to enrich the training dataset with more unseen objects and appearances (Yu et al., 2023; Mandelkar et al., 2023). Moreover, *Stem-OB* is indifferent to many unspecified appearance changes, in contrast to augmentation-based methods that must concentrate on a few selected types of generalization, thereby introducing inevitable inductive biases.

We provide theoretical analysis and a user study to support our claim that *Stem-OB* can effectively merge scenes with perceptual differences to similar “stem observations”. Empirical study demonstrates the effectiveness of our approach in a variety of simulated and real-world tasks and a range of different perturbations. *Stem-OB* proves to be particularly effective in real-world tasks where appearance and lighting changes hamper the other baselines, establishing an overall improvement in the success rate of **22.2%**. What’s better, no inference time inversion is required for *Stem-OB* to take effect, making the deployment of our method virtually free of computational cost.

2 PRELIMINARY

2.1 THE INVERSION OF DIFFUSION MODELS

We begin by outlining the fundamentals of Diffusion Inversion. A diffusion model operates with two passes: a backward denoising pass, which generates an image from noise, and a forward pass, where noise is incrementally added to an image until it becomes pure Gaussian noise. This forward process is a Markov chain that starts with x_0 , and gradually adds noise to obtain latent variables

$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T$. Each step in this process is a Gaussian transition following the common form

$$\mathbf{x}_t = \sqrt{\alpha_t} \mathbf{x}_{t-1} + \sigma_t \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{x}_t | \sqrt{\alpha_t} \mathbf{x}_{t-1}, \sigma_t^2 \mathbf{I}) \quad (1)$$

where $\alpha_t \in (0, 1)$ represents the scheduler parameter at each step t , while σ_t characterizes the variance of the Gaussian noise $\boldsymbol{\epsilon}_t$ introduced at each step. In Denoising Diffusion Probabilistic Models (DDPM) (Ho et al., 2020), $\sigma_t = \sqrt{1 - \alpha_t}$. Consequently, equation Eq. (1) can be reformulated as Eq. (2) by applying the cumulative product $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{x}_t | \sqrt{\bar{\alpha}_t} \mathbf{x}_{t-1}, (1 - \bar{\alpha}_t) \mathbf{I}) \quad (2)$$

Diffusion inversion is similar to the forward process in that they both maps an image to a noise, however, inversion tries to preserve the image’s information and obtain the specific noise that can reconstruct the image during a backward denoising process.

DDPM inversion. We follow the DDPM inversion proposed in Huberman-Spiegelglas et al. (2024),

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \tilde{\boldsymbol{\epsilon}}_t \quad (3)$$

The DDPM inversion we consider here differs slightly from Eq. (2), as $\tilde{\boldsymbol{\epsilon}}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ are mutually independent distributions, in contrast to the highly correlated $\boldsymbol{\epsilon}_t$ in Eq. (2). As mentioned by Huberman-Spiegelglas et al. (2024), the independence of $\tilde{\boldsymbol{\epsilon}}_t$ results in a sequence of latent vectors where the structures of \mathbf{x}_0 are more strongly imprinted into the noise maps. An error reduction step is conducted in reverse order after the diffusion forward process to improve image reconstruction accuracy during the denoising process:

$$\mathbf{z}_t = \mathbf{x}_{t-1} - \hat{\boldsymbol{\mu}}(\mathbf{x}_t) / \sigma_t, \quad \mathbf{x}_{t-1} = \hat{\boldsymbol{\mu}}(\mathbf{x}_t) + \sigma_t \mathbf{z}_t \quad (4)$$

DDIM inversion. We follow the DDIM inversion proposed in Song et al. (2022), where at each forward diffusion step

$$\mathbf{x}_t = \sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}} \mathbf{x}_{t-1} + \left(\sqrt{\frac{1 - \bar{\alpha}_t}{\bar{\alpha}_t}} - 1 - \sqrt{\frac{1 - \bar{\alpha}_{t-1}}{\bar{\alpha}_{t-1}}} - 1 \right) \boldsymbol{\epsilon}_\theta(\mathbf{x}_{t-1}, t, \mathcal{C}) \quad (5)$$

Note that the noise $\boldsymbol{\epsilon}_\theta(\mathbf{x}_{t-1}, t, \mathcal{C})$ is now generated by a network trained to predict the noise based on the previous vector \mathbf{x}_{t-1} and the text embedding \mathcal{C} which, in our case, is \emptyset .

2.2 VISUAL IMITATION LEARNING

Visual Imitation Learning (VIL) is a branch of Imitation learning (IL) that focuses on learning action mappings from visual observations. Given a dataset of observation \mathcal{O} and action \mathcal{A} pairs, the goal of VIL is to learn a policy $\pi_\theta(\mathcal{A}|\mathcal{O})$ that maps observations to actions. The policy is typically parameterized by a neural network with parameters θ , and is trained to minimize the negative log-likelihood of the actions taken by the expert.

3 RELATED WORKS

3.1 VISUAL IMITATION LEARNING AND GENERALIZATION

Visual imitation learning typically follows two approaches: directly imitating expert policies, as in behavior cloning and DAgger (Ross et al., 2011), or inferring a reward function that aligns the agent’s behavior with expert demonstrations, like inverse reinforcement learning (Ng & Russell, 2000) and GAIL (Ho & Ermon, 2016). The former approach, favored in recent works due to its scalability and practicality in complex, real-world tasks, has led to several advancements. Notable methods include Diffusion Policy (Chi et al., 2023), which leverages a diffusion model (Ho et al., 2020) to maximize the likelihood of expert actions, and Action Chunk Transformer (Zhao et al., 2023) that uses a Transformer (Vaswani, 2017).

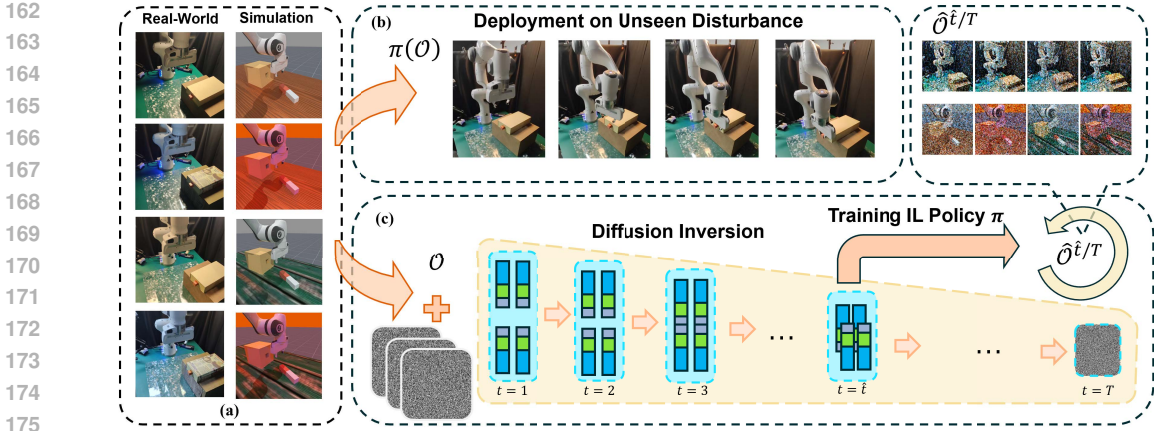


Figure 2: **Overview of Stem-OB:** (a). *Stem-OB* has been evaluated in both real-world and simulated environments. (b). The trained visual IL policies are directly applied to the original observation space \mathcal{O} , demonstrating robustness to unseen environmental disturbances. (c). We train the visual IL policy π on the diffusion-inversed latent space $\hat{\mathcal{O}}^{\hat{t}/T}$, where \hat{t} denotes a specific inversion step out of a total of T . Each composite rectangle in the diffusion inversion process, made up of three smaller sections, represents the latent vector of an image, with the smaller section depict the finer attributes (gray). During the inversion process, finer attributes converge earlier than coarser ones.

To enhance the generalization and robustness of visual imitation learning algorithms, various approaches have been explored (Cetin & Celiktutan, 2021). For instance, Li et al. (2023) leverage trajectory and step level similarity through an estimated reward function, while Wan et al. (2023) improve visual robustness by separating task-relevant and irrelevant dynamics models before applying the GAIL framework (Ho & Ermon, 2016). Zhang et al. (2023) use mutual information constraints and adversarial imitation learning to create compact representations that generalize to unseen environments. However, these approaches are not directly applicable to methods like diffusion policy, which focus on imitation without reward functions or dynamics models. Zheng et al. (2023) propose to filter extraneous action subsequence, yet their focus is not on visual perturbations. Most relevant to our setting, several works in robust visual reinforcement learning have explored adding noise in image or frequency domain to improve generalizability (Huang et al., 2022; Lee & Hwang, 2024; Yuan et al., 2024a), however, they lack the semantic understanding of the augmentation process.

3.2 INVERSION OF DIFFUSION MODELS AND ITS APPLICATION

Diffusion model inversion aims to recover an initial noise distribution from a given image, enabling image reconstruction from the same noise via backward denoising. A common approach is DDIM inversion (Song et al., 2022), which estimates the previous noise using predictions from the current diffusion step, though the approximation will introduce cumulative errors. To address this issue, several methods employ learnable text embeddings (Mokady et al., 2023; Miyake et al., 2023), fixed-point iterations (Pan et al., 2023; Garibi et al., 2024), or gradient-based techniques (Samuel et al., 2024) to refine the result. Another approach, based on the stochastic DDPM scheduler (Ho et al., 2020), reconstructs noisy images at each diffusion step (Huberman-Spiegelglas et al., 2024; Brack et al., 2024). In contrast to the DDIM inversion methods, the noise $\tilde{\epsilon}_t$ of each step is statistically independent, making them ideal for our application since we can obtain the noisy image of a certain step without the need to recover other steps, greatly reducing the calculation cost.

Diffusion Inversion is the crucial part of diffusion-based image editing methods (Meng et al., 2021; Kawar et al., 2023), which typically involves first inverting the diffusion process to recover the noise latent, then denoise the latent with desired editing conditions. Recent works also explore to apply attention control over the denoising process to improve the fidelity of the edited image Hertz et al. (2022); Tumanyan et al. (2023), and have shown promising application in robot learning tasks Gao et al. (2024). Beyond that, inversion is also used in tasks like concept extraction Huang et al. (2023) and personalization Gal et al. (2022). Most recently, Wang & Chen (2024) also propose the use of diffusion inversion to interpolate between image categories to improve classification performance.

4 METHOD

In this section, we introduce the intuition and implementation of our framework. We first propose the intuition of applying inversion on observations through theoretical analysis based on attribute loss, a diffusion-based measurement of image semantic similarity. Then, we conduct an illustrative experiment and a user study to validate our intuition. Finally, we explain how to practically implement *Stem-OB* and incorporate diffusion inversion into a visual imitation learning framework.

4.1 INTUITION DEVIATION BY ATTRIBUTE LOSS

Intuitively, given a source image and its variation, distinguishing between them becomes increasingly difficult during the diffusion inversion process. If there are two different variations, we want to show that as the inversion step increases, the pair with minor alterations will become indistinguishable sooner than the pair with larger and structural changes. We borrow the definition of attribute loss from Yue et al. (2024) to quantify the semantic overlapping of the two images at time step t during a inversion process:

$$loss(\mathbf{x}_0, \mathbf{y}_0, t) = \frac{1}{2} \text{OVL}(q(\mathbf{x}_t|\mathbf{x}_0), q(\mathbf{y}_t|\mathbf{y}_0)) \quad (6)$$

where \mathbf{x}_0 and \mathbf{y}_0 are the latent variables of the two images, and OVL is the overlapping coefficient quantifying the overlapping area of probability density functions. For an inversion process where each step follows a Gaussian transition, it takes the form $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sigma\epsilon \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, \sigma^2\mathbf{I})$. The OVL can be further calculated as the overlapping area of two Gaussian distributions, i.e.,

$$loss(\mathbf{x}_0, \mathbf{y}_0, t) = \frac{1}{2} \left[1 - \text{erf}\left(\frac{\|\sqrt{\bar{\alpha}_t}(\mathbf{y}_0 - \mathbf{x}_0)\|}{2\sqrt{2}\sigma}\right) \right] \quad (7)$$

where erf is the error function, which is strictly increasing. Given a source image \mathbf{x}_0 and its variations $\hat{\mathbf{x}}_0$ and $\tilde{\mathbf{x}}_0$, with $\tilde{\mathbf{x}}_0$ undergoing a larger variation than $\hat{\mathbf{x}}_0$, the following conclusion can be easily observed under the same diffusion scheduling:

$$\tau(\mathbf{x}_0, \hat{\mathbf{x}}_0, \rho) < \tau(\mathbf{x}_0, \tilde{\mathbf{x}}_0, \rho), \text{ s.t. } \|\hat{\mathbf{x}}_0 - \mathbf{x}_0\| < \|\tilde{\mathbf{x}}_0 - \mathbf{x}_0\| \quad (8)$$

Here, $\tau(\mathbf{x}_0, \mathbf{y}_0, \rho) = \inf\{t > 0 \mid loss(\mathbf{x}_0, \mathbf{y}_0, t) > \rho\}$ represents the earliest step where the loss between \mathbf{x}_0 and \mathbf{y}_0 exceeds the threshold ρ , and $\|\cdot\|$ measures the difference between an image and its variation. Eq. (8) provides a theoretical grounding for our intuition: images with fine-grained attribute changes tend to become indistinguishable sooner than those with coarse-grained modifications under identical diffusion schedules.

We can further derive the attribute loss for DDPM inversion

$$loss_{DDPM}(\mathbf{x}_0, \mathbf{y}_0, t) = \frac{1}{2} \left[1 - \text{erf}\left(\frac{\|\sqrt{\bar{\alpha}_t}(\mathbf{y}_0 - \mathbf{x}_0)\|}{2\sqrt{2(1 - \bar{\alpha}_t)}}\right) \right] \quad (9)$$

Additionally, we derive that the attribute loss for DDIM inversion exhibits a similar form under certain assumptions. The detailed derivation can be found in Appendix A.2.

$$loss_{DDIM}(\mathbf{x}_0, \mathbf{y}_0, t) = \frac{1}{2} \left[1 - \text{erf}\left(\frac{\|(\mathbf{y}_0 - \mathbf{x}_0)\|}{2\sqrt{2\sum_{i=1}^t \frac{1}{\bar{\alpha}_i} \left(\sqrt{\frac{1}{\bar{\alpha}_i} - 1} - \sqrt{\frac{1}{\bar{\alpha}_{i-1}} - 1}\right)^2}}\right) \right] \quad (10)$$

Because $\bar{\alpha}_t \in (0, 1)$ is strictly decreasing, the attribute loss tends to increase as the time step increases. Furthermore, as discussed in Yue et al. (2024), this attribute loss is equivalent to how likely the DM falsely reconstruct \mathbf{x}_t sampled from $q(\mathbf{x}_t|\mathbf{x}_0)$ closer to \mathbf{y}_0 instead of \mathbf{x}_0 , and vice versa.

4.2 ILLUSTRATIVE EXPERIMENT

In Sec. 4.1, we made a key assumption that similar images are closer in latent space, leading to the conclusion that such images exhibit higher attribute loss at a given inversion step (Eq. (8)). While this claim is intuitive, we conducted an illustrative experiment to further support it. To validate this

assumption, we used a set of images, denoted as \mathcal{I} , from the real-world task variations described in Sec. 5.1.1. Specifically, we selected several images of objects from 5 categories and calculated the pairwise distance between images on their latent representations, both for intra-category and cross-category image pairs. The computed image distances are presented in Tab. 1, where each entry represents the average similarity between two categories and the diagonal entries indicate intra-category similarity. As indicated in this table, intra-category images exhibit lower similarity compared to cross-category images, which justifies our claim.

We then conducted a user study to validate that similar images exhibit higher attribute loss at a given inversion step. We recruited 51 participants and presented each with image pairs from \mathcal{I} after specific steps of diffusion inversion. The inversion steps were systematically sampled from 15 to 45, out of a total 50 steps inversion and with intervals of 5. Each participant was asked to determine whether the image pairs depicted the same object, and we recorded the proportion of incorrect responses.

The experimental results in Fig. 3 show that at inversion step $t_1 = 20$, the incorrect response rate within the same category starts to increase. In contrast, the incorrect responses for objects from different categories only started to rise at the inversion step $t_2 = 35$. This indicates that inversion makes objects within the same category harder to distinguish earlier than affecting the distinction between objects from different categories. Therefore, these results support our claim that similar images exhibit higher attribute loss at a given inversion step.

Table 1: **By-category image semantic similarity.**

Categories	Bowl	Cup	Drawer	Duck	Faucet
Bowl	156.65	172.20	172.81	172.82	167.73
Cup	172.20	154.63	165.43	167.83	167.37
Drawer	172.81	165.43	144.63	161.82	161.57
Duck	172.82	167.83	161.82	140.51	147.00
Faucet	167.73	167.37	161.57	147.00	145.34

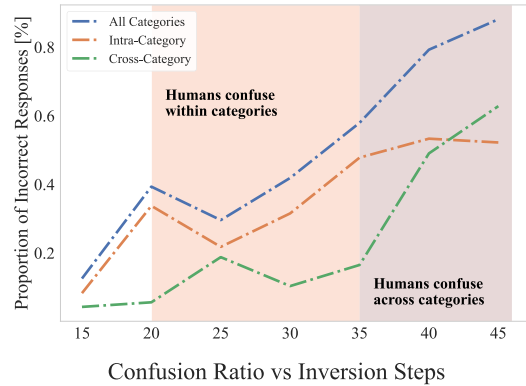


Figure 3: **User study confusion proportions.**

4.3 IMPLEMENTATION OF STEM-OB

Following the theoretical analysis and experimental validation, we propose a practical implementation of our framework. Our method applies diffusion inversion to converge different observations during training, and we find that the model gains robustness improvement in the original observation space during test time. The detailed implementation is described below.

Training. The training process begins by applying diffusion inversion to each observation. We define the following partial diffusion inversion process for an observation $\mathbf{o}_i \in \mathcal{O}$:

$$\hat{\mathbf{o}}_i^{t/T} = \mathbf{f}(\mathbf{o}_i, t, T) \tag{11}$$

Here, $\hat{\mathbf{o}}_i^{t/T} \in \hat{\mathcal{O}}^{t/T}$ denotes the observation after t out of T steps of diffusion inversion from the original observation $\mathbf{o}_i \in \mathcal{O}$. The function $\mathbf{f}(\cdot)$ applies t steps of diffusion inversion to \mathbf{o} using any inversion methods. We select the DDPM inversion method for its efficiency and effectiveness in our experiments, and the selection of t and total inversion step T is discussed in Sec. 5.5. The visual imitation learning algorithms are then trained on the inversion-altered space $\hat{\mathcal{O}}^{t/T}$. Note that we empirically find the error reduction step of DDPM inversion in Eq. (4) is not significant for the performance, so we omit it and approximate the partial inversion process with regard to Eq. (3) only. In this way, we avoid the full inversion process or reverse-order corrections that involve extensive Diffusion Model inference. This approach significantly reduces preprocessing time with minimal performance impact, achieving an average time for preprocessing of 0.2s per image.

Testing. Despite training is entirely conducted on the inversion-altered space, the imitation learning algorithm of our choice showcases surprising zero-shot adaptation ability to the original observation space. We adhere to the original observation space \mathcal{O} during testing, which means essentially no changes are made to the downstream visual imitation learning algorithms. This approach demonstrates improved robustness to environmental variations without any inference-time overhead, and is suitable for any other test-time augmentation techniques.

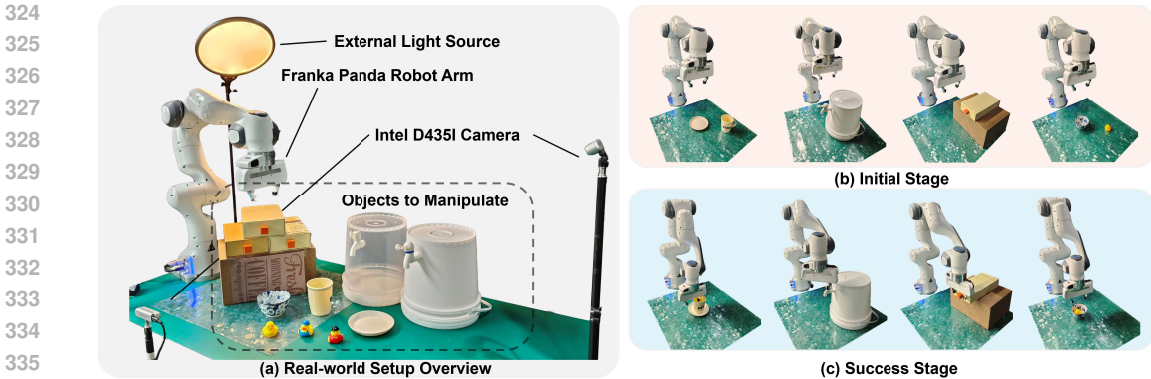


Figure 4: **Realworld Setup:** (a) Overview of the whole setup. (b)(c) These tasks are performed by the robot in a real-world environment, from left to right: *Cup2Plate*, *Turn on Faucet*, *Open Drawer*, and *Duck2Bowl*. The figure showcases the initial and final states of the tasks.

5 EXPERIMENTS

In this section, we present a comprehensive set of experiments using Diffusion Policy (DP) (Chi et al., 2023) as our visual imitation learning backbone. Since Stable Diffusion is pretrained on real-world images (Chi et al., 2023; Rombach et al., 2022), and a increasing interest in deploying visual imitation learning in real-world scenarios in the community (Paolo et al., 2024), we focus primarily on evaluating *Stem-OB* on real-world tasks. We extend the testing to simulated environments as well for further benchmarking. To assess the robustness of our method against visual appearance variations, we design experiments featuring different object textures and lighting conditions. We compare *Stem-OB* against several baselines to validate its effectiveness.

5.1 EXPERIMENT SETUP

5.1.1 REAL-WORLD TASKS

We conduct real-world experiments using a Franka Emika Panda Arm, and two RealSense D435I cameras positioned at different angles for RGB inputs. The setup is shown in Fig. 4. Our experiments focus on four tasks, as described below:

***Cup to Plate* (C2P):** The robot arm picks up a cup and places it on a plate, with variation introduced by changing tablecloth patterns.

***Duck to Bowl* (D2B):** The robot grasps a toy duck and places it into a bowl, with variations introduced by altering the duck’s appearance.

***Open Drawer* (OD):** The robot arm grabs a drawer handle and pulls it open, with variations introduced by modifying the drawer’s visual characteristics.

***Turn on Faucet* (ToF):** The robot arm turns on a faucet, with variations introduced by altering the appearance of the faucet and bucket.

In addition to the visual variations mentioned above, all four tasks above involve changes in *lighting conditions*, i.e., cool vs. warm light. All the variations only happen during testing time, with the training set contain only a basic setting. The object locations in training set are randomly initialized within a specified area, and 100 demonstrations are collected per task. For testing, nine predefined target positions are used. Further details of the environmental variations can be found in Appendix C.1.

5.1.2 SIMULATION TASKS

Our simulation experiments consider different tasks within two frameworks: a photorealistic simulation platform SAPIEN 3 (Xiang et al., 2020) and a less realistic framework MimicGen (Mandekar et al., 2023).

SAPIEN 3 provides physical simulation for robots, rigid bodies, and articulated objects. It delivers high-fidelity visual and physical simulations that closely approximate real-world conditions. We leverage the ManiSkill 3 dataset (Gu et al., 2023; Tao et al., 2024), collected on SAPIEN 3, for benchmarking. Specifically, we select four tasks from ManiSkill for evaluation:

PushCube: The robot arm pushes a cube to a target location.

PegInsertionSide: The robot arm inserts a peg into a hole.

PickCube: The robot arm picks up a cube and places it on a target location.

StackCube: The robot arm stacks one cube over the other.

During testing, we vary the background and lighting conditions in all tasks to generate different visual appearances. 50 episodes are tested for each setting. Details of the environmental variations can be found in Appendix C.2.

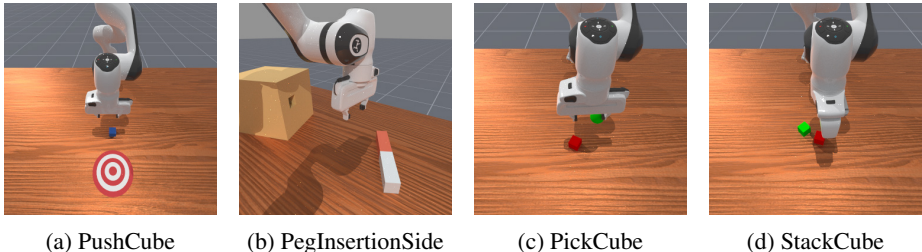


Figure 5: **SAPIEN 3 environments**. The figure showcases the visual appearance and task configuration of each setting.

MimicGen is a system for generating large diverse datasets from a small number of human demonstrations. We utilize the benchmark to evaluate the performance of our approach and conduct experiments on a variety of tasks: **MugCleanup**, **Threading**, **ThreePieceAssembly** and **Coffee**. MimicGen offers numerous task variations, each characterized by different initial distributions. We adopt the default initial distribution (D0) for both training data and test environments. For evaluation, we employ a single image as the input to the policy, using 500 samples out of a total of 1000 demos for training. We alter the table texture and object appearances to create different test environments. 300 episodes are tested for each setting of all the tasks. Detailed descriptions of the environmental variations can be found in Appendix C.3.

5.2 BASELINES

We compare *Stem-OB* with several data augmentation methods aimed at improving the generalizability of visual imitation learning. The baselines include **SRM** (Huang et al., 2022), which modifies images by adding random noise in the frequency domain, **Mix-Spectrum** (Mix) (Lee & Hwang), which enhances the agent’s focus on semantic content by blending original images with randomly selected reference images from the same dataset in the frequency domain, and **original images** (Org) without any modification. Additionally, as highlighted in (Yuan et al., 2024b), **Random Overlay** (RO) improves generalization in real-world experiments by blending original images with random real-world photos in the image domain. Therefore, we include an additional study using Random Overlay in our real-world experiments.

5.3 EXPERIMENT RESULTS

5.3.1 REAL-WORLD EXPERIMENTS

In this section, we report the average success rate across all predefined test positions for each task. As shown in Tab. 2, *Stem-OB* consistently outperforms the baseline models in all settings, demonstrating superior generalization capability. Under the training conditions, all methods achieve relatively high success rates, with our method performing slightly better than the baselines. However, in generalization testing, baseline methods exhibit significant performance drop, while our approach maintains a high success rate, showcasing the superior adaptability of *Stem-OB* to complex and noisy real-world visual disturbances.

Table 2: **Evaluation of real-world experiments.** Train.: evaluations in the same settings as the training dataset. Gen.: evaluations under different visual perturbations for generalizability analysis. All: evaluations including both Train. and Gen. The tasks are *C2P*, *D2B*, *OD*, and *ToF*. We report the mean and standard deviation of the success rate (%) over 6 settings for each task, and the best results are highlighted in **bold**.

Task \	<i>C2P</i>			<i>D2B</i>			<i>OD</i>			<i>ToF</i>		
	Train.	Gen.	All	Train.	Gen.	All	Train.	Gen.	All	Train.	Gen.	All
<i>Stem-OB</i>	89.0	93.3 \pm 6.1	92.6 \pm 5.7	78.0	44.4 \pm 22.2	50.0 \pm 24.1	100.0	100.0 \pm 0	100.0 \pm 0	100.0	88.9 \pm 19.2	90.7 \pm 17.8
Org	89.0	77.8 \pm 26.1	79.6 \pm 23.7	56.0	13.3 \pm 14.4	20.4 \pm 21.6	89.0	13.3 \pm 18.3	25.9 \pm 34.9	100.0	22.2 \pm 19.2	35.2 \pm 36.1
SRM	56.0	73.3 \pm 9.9	70.4 \pm 11.5	44.0	8.9 \pm 9.3	14.8 \pm 16.7	89.0	15.6 \pm 21.7	27.8 \pm 35.7	89.0	31.1 \pm 30.8	40.7 \pm 36.3
Mix	89.0	55.6 \pm 13.6	61.1 \pm 18.3	44.0	22.2 \pm 17.6	25.9 \pm 18.1	33.0	35.6 \pm 5.0	35.2 \pm 4.5	100.0	68.9 \pm 30.8	74.1 \pm 30.4
RO	56.0	53.3 \pm 9.3	53.7 \pm 8.4	44.0	15.6 \pm 12.7	20.4 \pm 16.4	89.0	88.9 \pm 7.9	88.9 \pm 7.0	78.0	82.2 \pm 20.2	81.5 \pm 18.1

Table 3: **Evaluations in simulated environments.** We compare the performance of *Stem-OB*, Org, SRM, and Mix in simulated environments. The left side of the table shows the results on the SAPIEN 3 environment, while the right side is about the MimicGen environment. The results are reported as the mean and standard deviation of the success rate(%) over their own various test settings. The performance of *Stem-OB* is highlighted in orange for better visibility. The best performance of each task is highlighted in **bold**.

Task \	SAPIEN 3				MimicGen			
	<i>PushCube</i>	<i>PegInsertionSide</i>	<i>PickCube</i>	<i>StackCube</i>	<i>MugCleanup</i>	<i>Threading</i>	<i>ThreePieceAssembly</i>	<i>Coffee</i>
<i>Stem-OB</i>	99.1 \pm 1.8	0.4 \pm 0.8	25.1 \pm 20.9	27.2 \pm 31.3	19.5 \pm 12.5	16.4 \pm 12.8	13.1 \pm 6.9	50.5 \pm 19.6
Org	61.8 \pm 33.1	0.0 \pm 0.0	8.6 \pm 18.2	4.3 \pm 19.2	16.4 \pm 12.0	18.1 \pm 12.8	14.8 \pm 8.4	43.3 \pm 17.3
SRM	62.5 \pm 30.5	0.0 \pm 0.0	8.4 \pm 16.1	0.7 \pm 3.4	15.4 \pm 0.08	25.8 \pm 11.7	13.6 \pm 7.6	39.4 \pm 12.0
Mix	97.2 \pm 3.35	0.6 \pm 0.1	12.9 \pm 16.3	8.2 \pm 18.0	22.3 \pm 11.8	18.9 \pm 8.5	14.2 \pm 6.6	41.8 \pm 9.9

The experiment results demonstrate that previous visual augmentation methods, such as SRM and Mix-Spectrum, struggle to generalize in real-world scenarios, which could be attribute to the complexity of real-world environments. Real images contain more redundant information, complicating the frequency domain and potentially leading to the failure of these augmentation methods. The light disturbance introduced in our experiments is a typical example, where wide-range but low-intensity noise is added to the images. Our approach effectively handles real-world noise by extracting high-level structures from the appearances, resulting in better generalization. Even in challenging scenarios like D2B, where baseline methods mostly fail, our method maintains a high success rate.

Interestingly, DP without any image modification (Org) outperforms the baselines in some tasks, such as OD and ToF, suggesting that DP has inherent generalization capabilities. However, this generalizability is inconsistent across tasks. For instance, in C2P, we observe that object appearance (cup and plate) had little impact on DP’s performance, while the tablecloth pattern significantly affects it. In D2B, the duck’s appearance is critical, whereas the table cloth variation is more influential. In contrast, our method exhibits consistent generalization across diverse scenarios.

5.4 SIMULATION EXPERIMENTS

Sapien 3 The evaluation results on Sapien 3 are presented on the left side of Tab. 3. The complexity of the tasks poses challenges for DP in generalizing across various conditions. However, our method achieves a higher success rate than the baseline methods on most of the tasks. In *PushCube*, both Mix-Spectrum and our method perform well, but our approach is more robust and reach nearly 100% success. These results demonstrate that in more photorealistic simulation environments, our method generalizes more effectively across diverse tasks and conditions.

MimicGen. The evaluation results on the MimicGen benchmark are presented on the right side of Tab. 3. At first glance, the results seem unanticipated, as results seem unanticipated, as *Stem-OB* does not perform best in most settings. This can be attributed to the fact that is can be attributed to the fact that MimicGen environments are less photorealistic, with nearly texture-free and low-resolutions, limiting DP from fully leveraging images. Consequently, diffusion inversion processing complicates the observations, limiting DP from fully leveraging the advantages of *Stem-OB*. It is important to emphasize that our method is primarily designed for real-world scenarios, where diverse noise is inevitable, in contrast to the controlled Gen environment.

Table 4: **Ablation study on diffusion steps.** On the left side of the table, we increase the inversion step with fixed total number of steps, to intensify the effect of the diffusion process. Additionally, we compare the performance of the model with fixed ratio of steps, where the intensity of inversion is approximately the same. The results are reported as mean and standard deviation over 21 kinds of settings. The best performance is highlighted in orange and the second best is highlighted in pink.

Task \ Settings	Fixed Total Number of Steps (50)					Fixed Ratio of Steps (30%)			
	0/50	5/50	10/50	15/50	20/50	25/50	9/30	15/50	30/100
PushCube	61.8±33.1	97.1±4.0	91.1±18.9	99.1±1.8	98.5±3.2	96.3±3.8	98.1±3.8	99.1±1.8	98.3±2.9
PickCube	8.6±18.2	12.0±20.0	28.0±26.3	25.1±20.9	12.7±14.0	8.29±6.2	23.7±24.4	25.1±20.9	25.6±22.3
StackCube	4.3±19.2	20.7±29.6	27.2±31.3	19.6±24.9	11.6±12.8	2.0±3.6	19.5±20.5	19.6±24.9	20.2±27.0

5.5 ABLATION

In this section, we test with several design choices of *Stem-OB*, providing a better understanding of their impact on the final performance. We consider the following choices: the number of inversion steps with fixed total steps, the number of total steps with the constant proportion of inversion steps to the total steps, and the selection of inversion methods (DDPM or DDIM inversion). The experiments are conducted on SAPIEN 3. We choose *PushCube*, *PickCube* and *StackCube*, since they have a more significant performance variance in the main experiments.

Number of Inversion Steps. We compare the performance of *Stem-OB* with varying numbers of inversion steps, fixing the total steps to 50 and adjusting the inversion steps from 5 to 25 in increments of 5. The results, shown in Tab. 4, indicate that performance generally increases up to 15 inversion steps before declining. This can be attributed to insufficient removal of low-level appearance features with too few inversion steps, while excessive steps eliminate high-level structural information, hindering task performance. Optimal performance is observed around 10 to 15 inversion steps, depending on the complexity of the tasks.

Number of Total Steps. With the proportion of inversion steps to total steps fixed, we varied the total number of steps to 30, 50, and 100. The results, shown on the right side of Tab. 4, indicate that performance remains consistent regardless of the total number of steps. This suggests that the proportion of inversion steps is more critical to performance than the total number of steps.

Diffusion Inversion Methods. We compare the performance of two inversion methods: DDPM inversion and DDIM inversion, with results presented in Tab. 5. For DDIM inversion, we use 5/50 inversion steps instead of 15/50 due to differences in noise scheduling compared to DDPM inversion. The results show that DDPM inversion outperforms DDIM inversion on the benchmark datasets, supporting our choice of DDPM inversion as the primary diffusion inversion method.

Task	DDPM	DDIM
PushCube	99.1±1.8	81.3±15.7
PickCube	25.1±20.9	5.4±12.0

Table 5: **Ablation study on different inversion methods.** We compare the performance of DDPM and DDIM on the tasks of *PushCube* and *PickCube*. The results are reported as the success rate (%) of the task. The best performance is highlighted in bold.

6 CONCLUSION

In this work, we propose *Stem-OB*, a straightforward preprocessing method for visual IL. By inverting the observation in the diffusion latent space for certain steps, we effectively converge different observation variations to the node it stems from, Making it invariant to unseen low-level appearance changes in the observation space. Though we only test our method on the diffusion policy method, our method is general and compatible with any visual IL baselines in theory, and the plug-and-play nature of our method makes it easy to integrate. We plan to test our method with other visual IL baselines in simulation and real tasks in the future.

Reproducibility: The main algorithm of our method is simple as applying the open-sourced DDPM inversion method on the dataset before training. We’ve provided the code for our method in the supplementary material.

REFERENCES

- 540
541
542 Manuel Brack, Felix Friedrich, Katharia Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian
543 Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models.
544 In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.
545 8861–8870, 2024.
- 546 Edoardo Cetin and Oya Celiktutan. Domain-robust visual imitation learning with mutual informa-
547 tion constraints, 2021. URL <https://arxiv.org/abs/2103.05079>.
- 548 Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric A. Cousineau, Benjamin Burchfiel, and Shuran
549 Song. Diffusion policy: Visuomotor policy learning via action diffusion. *ArXiv*, abs/2303.04137,
550 2023. URL <https://api.semanticscholar.org/CorpusID:257378658>.
- 551 Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ
552 Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching with-
553 out in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.
- 554 Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam
555 Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for
556 high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*,
557 2024.
- 558 Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel
559 Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual
560 inversion. *arXiv preprint arXiv:2208.01618*, 2022.
- 561 Jialu Gao, Kaizhe Hu, Guowei Xu, and Huazhe Xu. Can pre-trained text-to-image models generate
562 visual goals for reinforcement learning? *Advances in Neural Information Processing Systems*,
563 36, 2024.
- 564 Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise:
565 Real image inversion through iterative noising, 2024. URL <https://arxiv.org/abs/2403.14602>.
- 566 Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiaing Liu, Tongzhou Mu, Yihe Tang, Stone
567 Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao
568 Su. Maniskill2: A unified benchmark for generalizable manipulation skills. In *International
569 Conference on Learning Representations*, 2023.
- 570 Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or.
571 Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*,
572 2022.
- 573 Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning, 2016. URL <https://arxiv.org/abs/1606.03476>.
- 574 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint
575 arxiv:2006.11239*, 2020.
- 576 Shaul Hochstein and Merav Ahissar. View from the top: Hierarchies and reverse hierarchies in
577 the visual system. *Neuron*, 36(5):791–804, 2002. ISSN 0896-6273. doi: [https://doi.org/10.1016/S0896-6273\(02\)01091-7](https://doi.org/10.1016/S0896-6273(02)01091-7). URL <https://www.sciencedirect.com/science/article/pii/S0896627302010917>.
- 582 Yangru Huang, Peixi Peng, Yifan Zhao, Guangyao Chen, and Yonghong Tian. Spectrum random
583 masking for generalization in image-based reinforcement learning. *Advances in Neural Informa-
584 tion Processing Systems*, 35:20393–20406, 2022.
- 585 Yangru Huang, Peixi Peng, Yifan Zhao, Guangyao Chen, and Yonghong Tian. Spectrum random
586 masking for generalization in image-based reinforcement learning. *Advances in Neural Informa-
587 tion Processing Systems*, 35:20393–20406, 2022.
- 588 Ziqi Huang, Tianxing Wu, Yuming Jiang, Kelvin CK Chan, and Ziwei Liu. Reversion: Diffusion-
589 based relation inversion from images. *arXiv preprint arXiv:2303.13495*, 2023.
- 590 Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpn noise
591 space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer
592 Vision and Pattern Recognition*, pp. 12469–12478, 2024.
- 593

- 594 Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and
595 Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the*
596 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6007–6017, 2023.
- 597 Jeong Woon Lee and Hyoseok Hwang. Mix-spectrum for generalization in visual reinforcement
598 learning. Available at SSRN 4818827.
- 600 Jeong Woon Lee and Hyoseok Hwang. Mix-spectrum for generalization in visual reinforcement
601 learning. Available at SSRN 4818827, 2024.
- 602 Siyuan Li, Xun Wang, Rongchang Zuo, Kewu Sun, Lingfei Cui, Jishiyu Ding, Peng Liu, and Zhe
603 Ma. Robust visual imitation learning with inverse dynamics representations, 2023. URL <https://arxiv.org/abs/2310.14274>.
- 606 Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretoiyo Akinola, Yashraj Narang, Linxi Fan,
607 Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using
608 human demonstrations. In *7th Annual Conference on Robot Learning*, 2023.
- 609 Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon.
610 Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint*
611 *arXiv:2108.01073*, 2021.
- 612 Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast
613 image inversion for editing with text-guided diffusion models, 2023. URL <https://arxiv.org/abs/2305.16807>.
- 614 Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for
615 editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference*
616 *on Computer Vision and Pattern Recognition*, pp. 6038–6047, 2023.
- 617 Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *International*
618 *Conference on Machine Learning*, 2000. URL [https://api.semanticscholar.org/](https://api.semanticscholar.org/CorpusID:266239314)
619 [CorpusID:266239314](https://api.semanticscholar.org/CorpusID:266239314).
- 620 Zhihong Pan, Riccardo Gherardi, Xiufeng Xie, and Stephen Huang. Effective real image edit-
621 ing with accelerated iterative diffusion inversion, 2023. URL [https://arxiv.org/abs/](https://arxiv.org/abs/2309.04907)
622 [2309.04907](https://arxiv.org/abs/2309.04907).
- 623 Giuseppe Paolo, Jonas Gonzalez-Billandon, and Balázs Kégl. A call for embodied ai. *arXiv preprint*
624 *arXiv:2402.03824*, 2024.
- 625 Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-
626 resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Con-*
627 *ference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.
- 628 Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and
629 structured prediction to no-regret online learning, 2011. URL [https://arxiv.org/abs/](https://arxiv.org/abs/1011.0686)
630 [1011.0686](https://arxiv.org/abs/1011.0686).
- 631 Dvir Samuel, Barak Meiri, Nir Darshan, Shai Avidan, Gal Chechik, and Rami Ben-Ari. Regularized
632 newton raphson inversion for text-to-image diffusion models, 2024. URL [https://arxiv.org/abs/](https://arxiv.org/abs/2312.12540)
633 [2312.12540](https://arxiv.org/abs/2312.12540).
- 634 Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL
635 <https://arxiv.org/abs/2010.02502>.
- 642 Stone Tao, Fanbo Xiang, Arth Shukla, Chen Bao, Nan Xiao, Rui Chen, Tongzhou Mu, Tse-Kai
643 Chan, Xander Hinrichsen, Xiaodi Yuan, Xinsong Lin, Xuanlin Li, Yuan Gao, Yuzhe Qin, Zhiao
644 Huang, and Hao Su. Maniskill 3 (beta). 2024. GitHub repository.
- 645 Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for
646 text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Com-*
647 *puter Vision and Pattern Recognition*, pp. 1921–1930, 2023.

- 648 A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
649
- 650 Shenghua Wan, Yucen Wang, Minghao Shao, Ruying Chen, and De-Chuan Zhan. Semail: Eliminat-
651 ing distractors in visual imitation via separated models, 2023. URL [https://arxiv.org/
652 abs/2306.10695](https://arxiv.org/abs/2306.10695).
- 653 Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and An-
654 ima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv
655 preprint arXiv:2302.12422*, 2023.
- 656 Yanghao Wang and Long Chen. Improving diffusion-based data augmentation with inversion spher-
657 ical interpolation. *arXiv preprint arXiv:2408.16266*, 2024.
- 659 Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao
660 Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A
661 simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and
662 Pattern Recognition (CVPR)*, June 2020.
- 663 Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the generalization gap in imitation
664 learning for visual robotic manipulation. *2024 IEEE International Conference on Robotics and
665 Automation (ICRA)*, pp. 3153–3160, 2023. URL [https://api.semanticscholar.org/
666 CorpusID:259375845](https://api.semanticscholar.org/CorpusID:259375845).
- 668 Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh,
669 Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imag-
670 ined experience. *arXiv preprint arXiv:2302.11550*, 2023.
- 671 Zhecheng Yuan, Tianming Wei, Shuiqi Cheng, Gu Zhang, Yuanpei Chen, and Huazhe Xu. Learning
672 to manipulate anywhere: A visual generalizable framework for reinforcement learning, 2024a.
673 URL <https://arxiv.org/abs/2407.15815>.
- 674 Zhecheng Yuan, Sizhe Yang, Pu Hua, Can Chang, Kaizhe Hu, and Huazhe Xu. RL-vigen: A rein-
675 forcement learning benchmark for visual generalization. *Advances in Neural Information Pro-
676 cessing Systems*, 36, 2024b.
- 678 Zhongqi Yue, Jiankun Wang, Qianru Sun, Lei Ji, Eric I Chang, Hanwang Zhang, et al. Exploring
679 diffusion time-steps for unsupervised representation learning. *arXiv preprint arXiv:2401.11430*,
680 2024.
- 681 Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion
682 policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings
683 of Robotics: Science and Systems (RSS)*, 2024.
- 684 Haoran Zhang, Yinghong Tian, Liang Yuan, and Yue Lu. Invariant adversarial imitation learning
685 from visual inputs. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech
686 and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10096070.
- 688 Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empiri-
689 cal risk minimization, 2018. URL <https://arxiv.org/abs/1710.09412>.
- 690 Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual ma-
691 nipulation with low-cost hardware, 2023. URL <https://arxiv.org/abs/2304.13705>.
- 692 Ray Chen Zheng, Kaizhe Hu, Zhecheng Yuan, Boyuan Chen, and Huazhe Xu. Extraneousness-
693 aware imitation learning. In *2023 IEEE International Conference on Robotics and Automation
694 (ICRA)*, pp. 2973–2979, 2023. doi: 10.1109/ICRA48891.2023.10161521.
- 695
696
697
698
699
700
701

A APPENDIX

A.1 DERIVATION OF DDPM INVERSION ATTRIBUTE LOSS

The derivation of DDPM inversion is straightforward by simply letting $\sigma = \sqrt{1 - \bar{\alpha}_t}$ in Eq. (7).

A.2 DERIVATION OF DDIM INVERSION ATTRIBUTE LOSS

We first rewrite Eq. (5) as a linear combination of \mathbf{x}_0 and a noise variable. To do so, we expand the recursive equation in Eq. (5). Note that we assume that $\epsilon_\theta(\mathbf{x}_{t-1}, t, \mathcal{C})$ is a Gaussian distribution with mean $\boldsymbol{\mu}_t$ and use ϵ_t for simplicity in the following derivation.

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}} \mathbf{x}_{t-1} + \left(\sqrt{\frac{1}{\bar{\alpha}_t} - 1} - \sqrt{\frac{1}{\bar{\alpha}_{t-1}} - 1} \right) \epsilon_t \\
&= \sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}} \left(\sqrt{\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_{t-2}}} \mathbf{x}_{t-2} + \left(\sqrt{\frac{1}{\bar{\alpha}_{t-1}} - 1} - \sqrt{\frac{1}{\bar{\alpha}_{t-2}} - 1} \right) \epsilon_{t-1} \right) + \left(\sqrt{\frac{1}{\bar{\alpha}_t} - 1} - \sqrt{\frac{1}{\bar{\alpha}_{t-1}} - 1} \right) \epsilon_t \\
&= \sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-2}}} \left(\sqrt{\frac{\bar{\alpha}_{t-2}}{\bar{\alpha}_{t-3}}} \mathbf{x}_{t-3} + \left(\sqrt{\frac{1}{\bar{\alpha}_{t-2}} - 1} - \sqrt{\frac{1}{\bar{\alpha}_{t-3}} - 1} \right) \epsilon_{t-2} \right) \\
&\quad + \sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}} \left(\sqrt{\frac{1}{\bar{\alpha}_{t-1}} - 1} - \sqrt{\frac{1}{\bar{\alpha}_{t-2}} - 1} \right) \epsilon_{t-1} + \left(\sqrt{\frac{1}{\bar{\alpha}_t} - 1} - \sqrt{\frac{1}{\bar{\alpha}_{t-1}} - 1} \right) \epsilon_t \\
&\dots \\
&= \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sum_{i=1}^t \sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_i}} \left(\sqrt{\frac{1}{\bar{\alpha}_i} - 1} - \sqrt{\frac{1}{\bar{\alpha}_{i-1}} - 1} \right) \epsilon_i \\
&= \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sum_{i=1}^t \sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_i}} \left(\sqrt{\frac{1}{\bar{\alpha}_i} - 1} - \sqrt{\frac{1}{\bar{\alpha}_{i-1}} - 1} \right) \boldsymbol{\mu}_t + \sqrt{\sum_{i=1}^t \frac{\bar{\alpha}_t}{\bar{\alpha}_i} \left(\sqrt{\frac{1}{\bar{\alpha}_i} - 1} - \sqrt{\frac{1}{\bar{\alpha}_{i-1}} - 1} \right)^2} \boldsymbol{\epsilon}
\end{aligned} \tag{12}$$

where the last equality holds considering the property of summing gaussian distributions, i.e., $\mathcal{N}(\boldsymbol{\mu}_1, \sigma_1^2 \mathbf{I}) + \mathcal{N}(\boldsymbol{\mu}_2, \sigma_2^2 \mathbf{I}) \sim \mathcal{N}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2, (\sigma_1^2 + \sigma_2^2) \mathbf{I})$. And $\sum_{i=1}^t \sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_i}} \left(\sqrt{\frac{1}{\bar{\alpha}_i} - 1} - \sqrt{\frac{1}{\bar{\alpha}_{i-1}} - 1} \right) \boldsymbol{\mu}_t$ represents the mean shift resulting from the mean bias of model ϵ_θ . For two similar images, the biases are approximately equal and thus cancel out when substituted into Eq. (7). Consequently, we can approximately derive the attribute loss for DDIM inversion.

$$\begin{aligned}
loss_{DDIM}(\mathbf{x}_0, \mathbf{y}_0, t) &\approx \frac{1}{2} \left[1 - \operatorname{erf} \left(\frac{\|\sqrt{\bar{\alpha}_t}(\mathbf{y}_0 - \mathbf{x}_0)\|}{2\sqrt{2\sum_{i=1}^t \frac{\bar{\alpha}_t}{\bar{\alpha}_i} \left(\sqrt{\frac{1}{\bar{\alpha}_i} - 1} - \sqrt{\frac{1}{\bar{\alpha}_{i-1}} - 1} \right)^2}} \right) \right] \\
&= \frac{1}{2} \left[1 - \operatorname{erf} \left(\frac{\|(\mathbf{y}_0 - \mathbf{x}_0)\|}{2\sqrt{2\sum_{i=1}^t \frac{1}{\bar{\alpha}_i} \left(\sqrt{\frac{1}{\bar{\alpha}_i} - 1} - \sqrt{\frac{1}{\bar{\alpha}_{i-1}} - 1} \right)^2}} \right) \right]
\end{aligned} \tag{13}$$

B TRAINING DETAILS

The hyperparameters for Diffusion Policy across all experiments are listed in Tab. 6. We use DDPM inversion in all the basic experiments, with the specific inversion steps for each experiment detailed in Tab. 7.

Hyperparameter	value	Hyperparameter	value
epoch_every_n_steps	100	ddim_num_train_timesteps	100
seq_length	15	ddim_num_inference_timesteps	10
frame_stack	2	ddim_beta_schedule	squaredcos_cap_v2
batch_size	128	ddim_clip_sample	TRUE
num_epochs	1500	ddim_set_alpha_to_one	TRUE
learning_rate_initial	0.0001	ddim_steps_offset	0
learning_rate_decay_factor	0.1	ddim_prediction_type	epsilon
regularization	L2/0.0	VisualCore_feature_dimension	64
observation_horizon	2	VisualCore_backbone_class	ResNet18Conv
action_horizon	8	VisualCore_pool_class	SpatialSoftmax
prediction_horizon	16	VisualCore_pool_num_kp	32
unet_diffusion_embed_dim	256	VisualCore_pool_temperature	FALSE
unet_down_dims	[256, 512, 1024]	VisualCore_pool_noise_std	1
unet_kernel_size	4	obs_randomizer_class	CropRandomizer
unet_n_groups	8	num_crops	1
ema_power	0.75		

Table 6: **Hyperparameters used for Diffusion Policy.** We use the same hyperparameters of diffusion policy for all the experiments.

Task	Step	Task	Step	Task	Step
Cup2Plate	15/50	PushCube	15/50	MugCleanup	10/50
Duck2Bowl	15/50	PegInsertionSide	15/50	Threading	5/50
OpeningDrawer	15/50	PickCube	15/50	ThreePieceAssembly	5/50
Turn on Faucet	15/50	StackCube	10/50	Coffee	5/50

Table 7: **Task and Inversion Step used for training the Diffusion Policy.** We all use DDPM as the inversion method.

C EXPERIMENT DETAILS

C.1 REALWORLD DETAILS

Fig. 6 shows the camera views of the real-world environments. Each task is represented by two rows, corresponding to different camera angles. The columns represent task settings, with the first being the default. As lighting conditions vary across tasks, images are grouped in pairs: the first under warm light and the second under cold light. Object appearances also differ between pairs

C.2 SAPIEN ENVIRONMENTS SETTINGS

Each task includes 21 distinct testing environments, as shown in Fig. 7. Images are grouped in sets of three, all sharing the same tablecloth texture but under different lighting conditions: white, yellow, and red light. Across tasks, seven unique tablecloth textures are used.

C.3 MIMICGEN ENVIRONMENTS SETTINGS

Fig. 9 displays the four MimicGen environments used in the experiments, and each row in Fig. 8 represents a task. In *MugCleanup*, the robot arm opens a drawer, retrieves a mug, and places it back. *Threading* involves precise insertion of a stick into a hole. *ThreePieceAssembly* requires collecting and assembling three components in a specific order. In *Coffee*, the robot opens the lid of the coffee machine and places a cup inside. Eight different table textures are used across all tasks (first eight columns). Additionally, *MugCleanup*, *Threading*, and *ThreePieceAssembly* employ alternating object texture, as indicated in the last column, creating distinct testing environments.

810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863

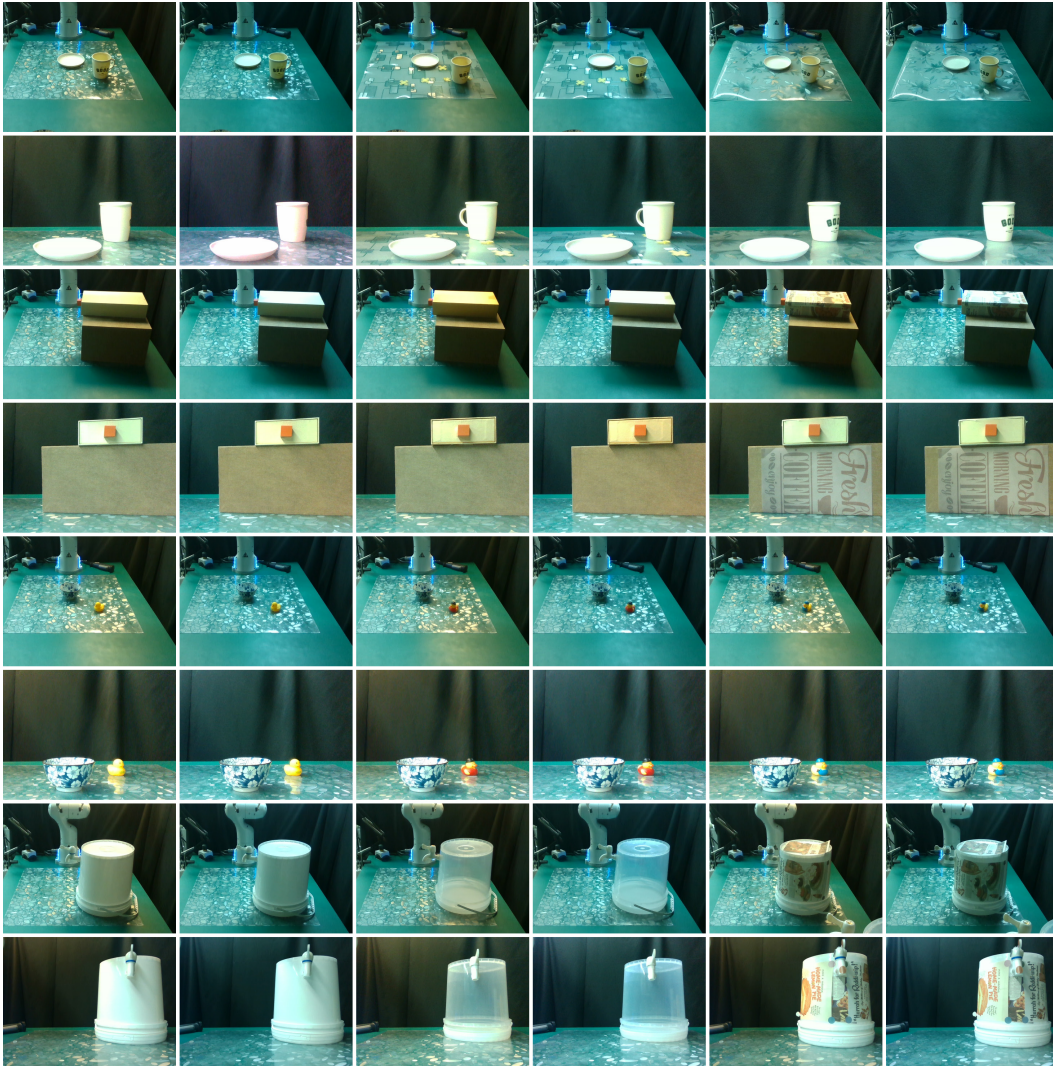


Figure 6: **Realworld Camera Views.** Each task is represented by two rows, with each representing a different camera view angle. For each task, we perform experiments on three instances with different object appearances, and two light conditions, as displayed in the columns. The first column represents the training setup.

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917



Figure 7: **Sapien Testing Settings.** Each task consists of 21 distinct settings, arranged in two rows. The images are grouped in threes, with each group sharing the same tablecloth texture but differing in lighting conditions: white, yellow, and red light. A total of seven unique tablecloth textures are used.

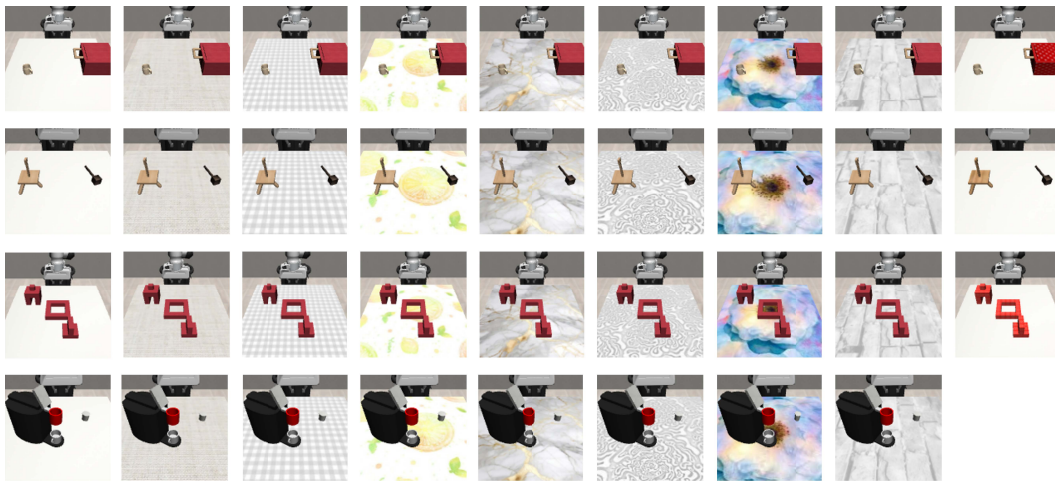
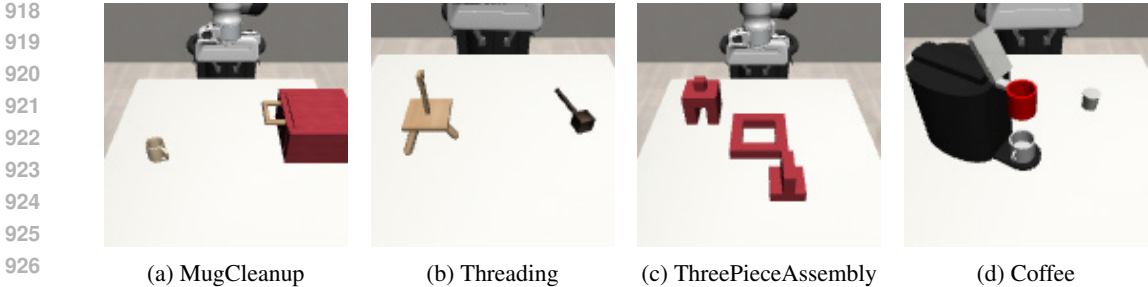
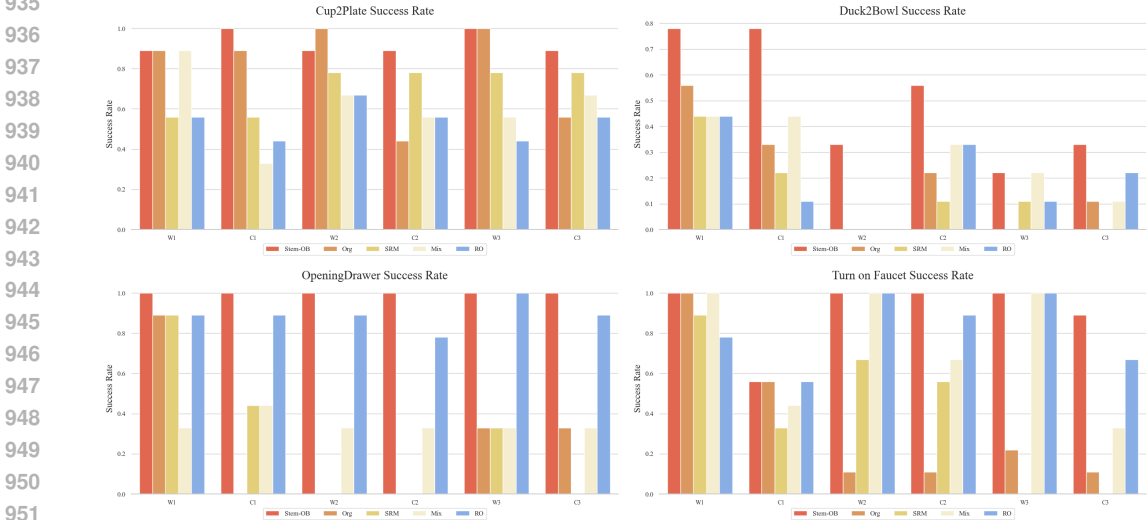


Figure 8: **MimicGen Testing Settings.** Each row represents a specific task, with eight different tablecloth textures used across the first eight columns. Additionally, *MugCleanup*, *Threading*, and *ThreePiece-Assembly* employ alternate object textures in the last column.



928 **Figure 9: MimicGen environments.** These tasks are based on MimicGen benchmark. (a) The agent
929 must open the drawer, pick up the mug, and place it into the drawer. (b) This task requires the agent
930 to thread a string through a hole. (c) The agent must assemble three pieces together. (d) The agent
931 is required to open the lid, and then place the coffee cup inside it.

932
933
934 **C.4 DETAILS OF EXPERIMENT RESULTS**



952 **Figure 10: Realworld Success Rate.** This figure presents the success rates for real-world tasks.
953 The groups are labeled as W1, C1 through W4, C4. W1 to W4 are conducted under the same warm
954 lighting conditions with different object appearances. C1 to C4 are performed under identical cold
955 lighting conditions, with varying object appearances. The order of the groups is consistent with that
956 shown in Fig. 6.

957
958
959 Fig. 10, Fig. 11, and Fig. 12 illustrate the success rates of *Stem-OB* and baseline methods in real-
960 world, SAPIEN 3, and MimicGen environments, respectively. These three figures display the per-
961 formance of *Stem-OB* under each setting.

962 **C.5 DETAILS OF ILLUSTRATIVE EXPERIMENTS**

963 *Image Distance Calculation:* We use the 2-norm to compute the image distance $D(\mathbf{x}, \mathbf{y})$ in the latent
964 space, where a smaller distance indicates greater similarity between the two images.

965 *Intra-Category Distance Calculation:* the intra-category distance is calculated as the mean of the
966 pairwise distances between images within the same category \mathcal{I}

967
968
969
970
971

$$D_{intra} = \frac{2}{N(N-1)} \sum_{i=1}^N \sum_{j=i+1}^N D(\mathbf{x}_i, \mathbf{x}_j) \tag{14}$$

where N is the number of images within \mathcal{I} and \mathbf{x}_i is the i -th image.

972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

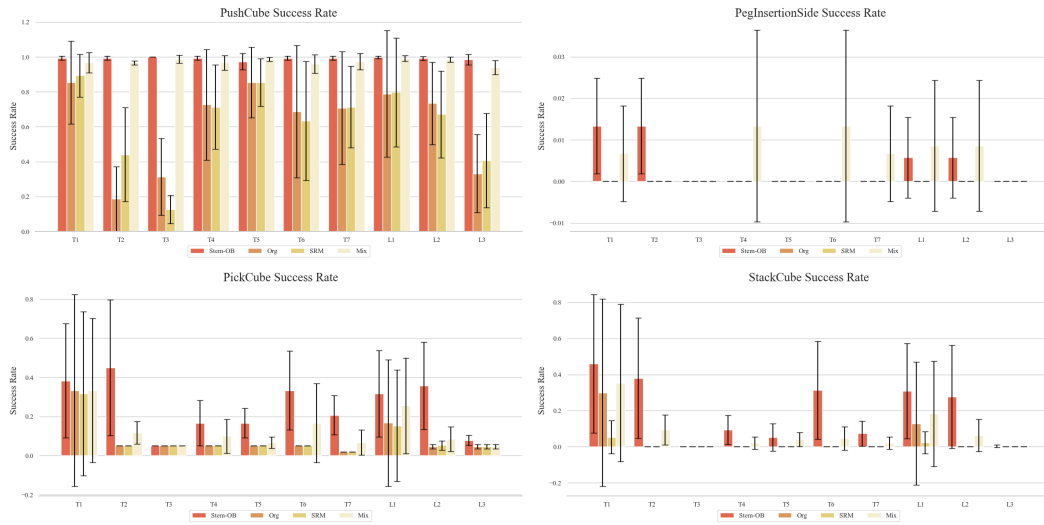


Figure 11: **Sapien Success Rate.** This figure presents the success rates for tasks within the SAPIEN 3 benchmark. The groups are labeled as T1 to T7 and L1 to L3. The settings are varied across 7 different tablecloth textures (T1 to T7) and 3 distinct lighting conditions (L1 to L3). For T1 to T7, the mean and standard deviation of the success rates are calculated over different lighting conditions with 3 lighting conditions combined with one texture. Conversely, for L1 to L3, the mean and standard deviation are computed over various tablecloth textures under a fixed lighting condition, with 7 textures evaluated for each light condition. The order of tablecloth textures and lighting conditions aligns with that presented in Fig. 7.

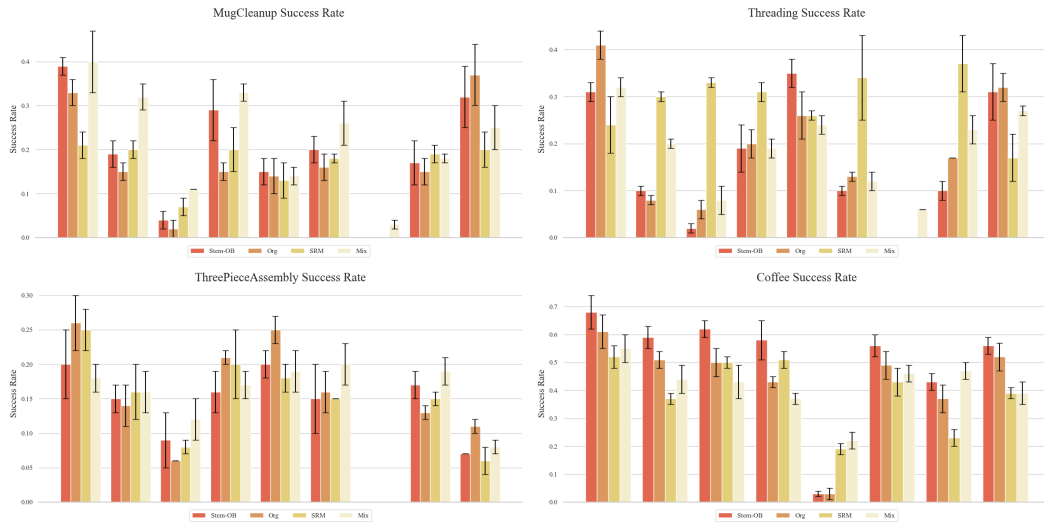


Figure 12: **MimicGen Success Rate.** This figure presents the success rates of *Stem-OB* and baselines across four distinct tasks. The mean and standard deviation are computed over 300 episodes. Each group of bars corresponds to one experimental setting. The bar order is consistent with the arrangement shown in Fig. 8.



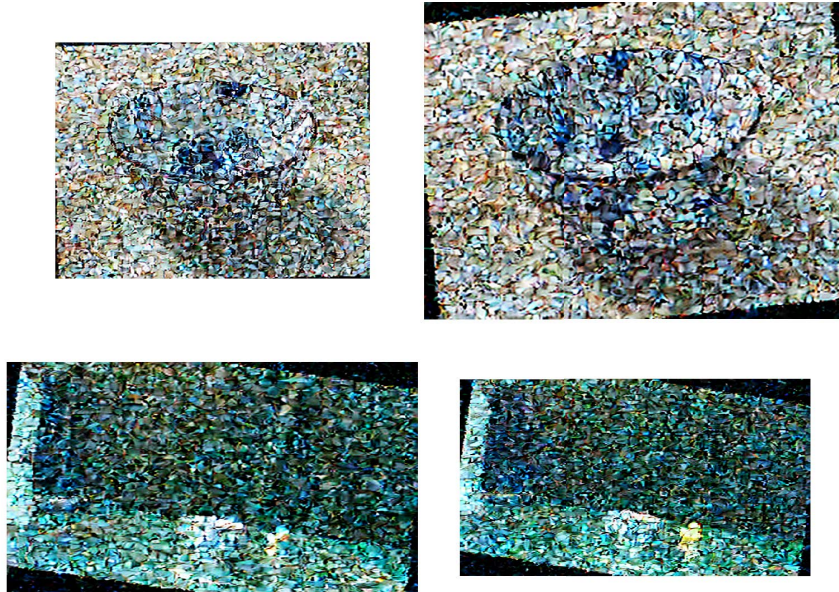
Figure 13: **Illustrative Experiment Objects.**

1026 *Cross-Category Distance Calculation:* the cross-category distance is calculated as the mean of the
 1027 pairwise distances between images from two different categories \mathcal{I}_1 and \mathcal{I}_2
 1028

$$1029 \quad D_{intra} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N D(\mathbf{x}_i, \mathbf{y}_j) \quad (15)$$

1030 where M and N are the size of \mathcal{I}_0 and \mathcal{I}_1 , respectively, $\mathbf{x}_i \in \mathcal{I}$ and $\mathbf{y}_i \in \mathcal{I}_2$.
 1031
 1032

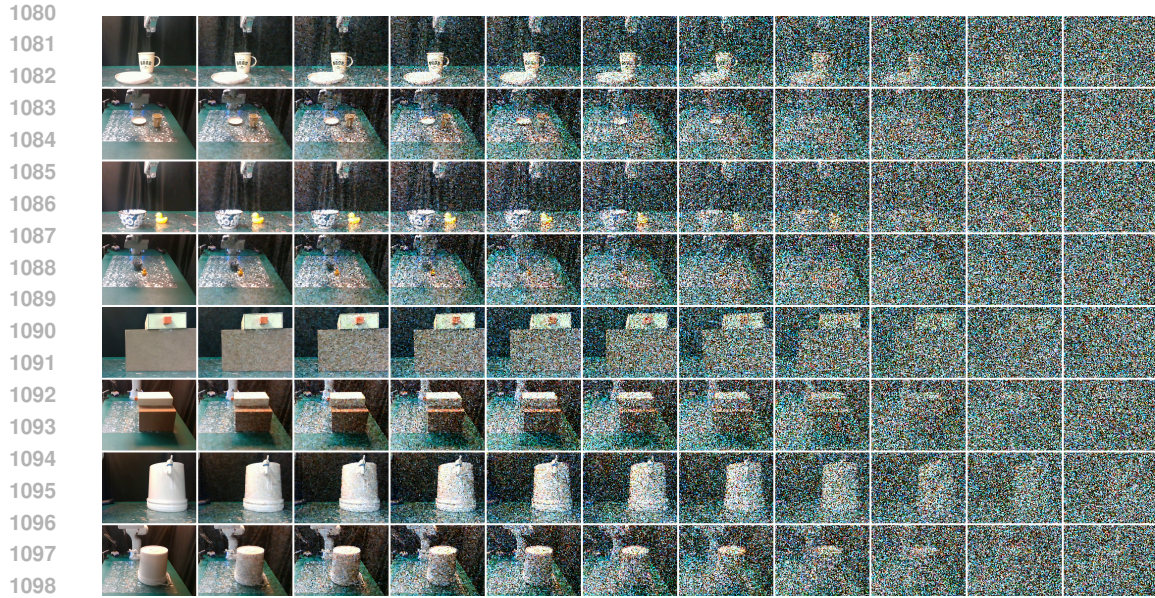
1033 C.6 DETAILS OF USER STUDY



1060 **Figure 14: Example of User Study Questions.** The upper two images represent one example from
 1061 the first section, and the lower two images represent one example from the second section. Each pair
 1062 is generated through a particularly selected inversion step. To prevent users from making judgments
 1063 based on image size or orientation, these images have been randomly rotated, cropped, and zoomed.
 1064

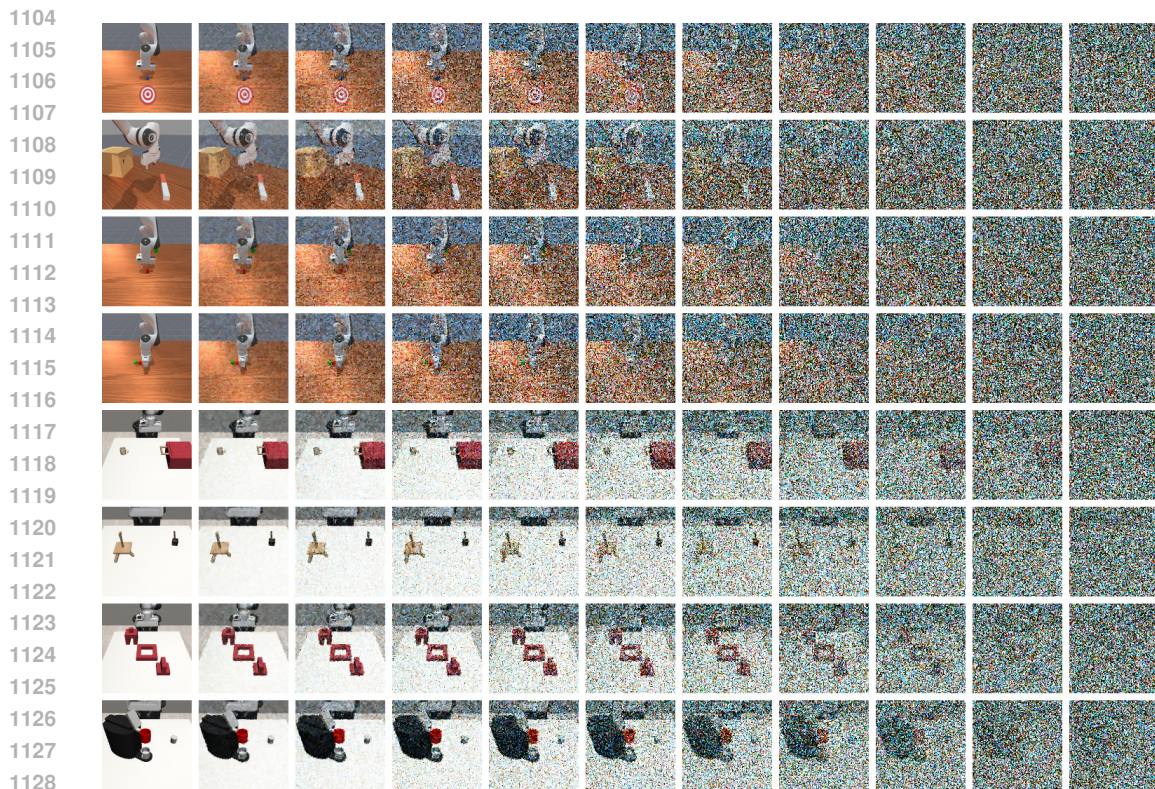
1065 We conducted a user study to evaluate the quality of the inversion results through a questionnaire-
 1066 based approach. In the first section, consisting of 42 questions, participants compare two images
 1067 per question and are asked to determine whether the images depict the same object, different objects
 1068 of the same category, or entirely different objects. Participants are informed that the images have
 1069 been randomly rotated, cropped, and zoomed, so image size or orientation should not influence their
 1070 judgment of object similarity. All images are real-world snapshots of common items like bowls and
 1071 faucets, with each pair inverted by a randomly selected inversion step.

1072 The second section, consisting of 56 questions, presented images from real-world tasks, as seen in
 1073 Fig. 6. In this section, participants evaluate whether the images represent the same task, different
 1074 scenes from the same task (with variations such as lighting or object appearance), or entirely
 1075 different tasks, where the target of the task changed (for example, a cup versus a drawer). Scene
 1076 variations could include changes in environmental conditions, such as lighting, or the appearance of
 1077 objects, like a change in a tablecloth or the texture of a drawer, without altering the underlying task
 1078 or function. To avoid bias from image size or angle, we also randomly cropped, rotated, and zoomed
 1079 the images. The example of images from the two sections is shown in Fig. 14. A total of 51 valid
 questionnaires were collected, with the results presented in Fig. 3.



1099
1100
1101
1102
1103

Figure 15: **Realworld Inversion Results.** Each task corresponds to two rows, with each representing a distinct camera view. The first column of each row is the original image, while the other columns are the DDPM inversion results. Beginning with the second column, the diffusion steps are increased from 5 to 50, with the incremental step being 5.



1129
1130
1131
1132
1133

Figure 16: **Simulation Inversion Results.** The first four rows correspond to the tasks in SAPIEN 3, while the last four rows correspond to the tasks in MimicGen. Each row represents a task, with the first column displaying the original image and the subsequent columns illustrating the DDPM inversion results. Starting from the second column, the diffusion steps are increased from 5 to 50, with the incremental step being 5.

1134 D VISUALIZATION

1135

1136 D.1 INVERSION VISUALIZATION

1137

1138 Fig. 15 and Fig. 16 display the inversion results for real-world and simulation environments, re-
1139 spectively. In the real-world environments, each task spans two rows, corresponding to the same
1140 camera view as in Fig. 6. In the simulation environments, each row represents a task, with the first
1141 four rows showing SAPIEN 3 environments and the last four showing MimicGen environments.
1142 The first column in each row is the original image, while subsequent columns present the inverted
1143 results. From the second column onward, the inversion results are generated by DDPM inversion,
1144 with the 50 total inversion steps and 5 incremental steps from 5 to 50.

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187