D-CODA: Diffusion for Coordinated Dual-Arm Data Augmentation

I-Chun Arthur Liu Jason Chen Gaurav S. Sukhatme* Daniel Seita
Department of Computer Science, University of Southern California

Abstract: Learning bimanual manipulation is challenging due to its high dimensionality and tight coordination required between two arms. Eye-in-hand imitation learning, which uses wrist-mounted cameras, simplifies perception by focusing on task-relevant views. However, collecting diverse demonstrations remains costly, motivating the need for scalable data augmentation. While prior work has explored visual augmentation in single-arm settings, extending these approaches to bimanual manipulation requires generating viewpoint-consistent observations across both arms and producing corresponding action labels that are both valid and feasible. In this work, we propose Diffusion for COordinated Dual-arm Data Augmentation (D-CODA), a method for offline data augmentation tailored to eye-in-hand bimanual imitation learning that trains a diffusion model to synthesize novel, viewpoint-consistent wrist-camera images for both arms while simultaneously generating joint-space action labels. It employs constrained optimization to ensure that augmented states involving gripper-to-object contacts adhere to constraints suitable for bimanual coordination. We evaluate D-CODA on 5 simulated and 3 real-world tasks. Our results across 2250 simulation trials and 300 real-world trials demonstrate that it outperforms baselines and ablations, showing its potential for scalable data augmentation in eye-in-hand bimanual manipulation. Our project website is at: https://dcodaaug.github.io/D-CODA/.

Keywords: Data augmentation, bimanual manipulation, diffusion models

1 Introduction

Bimanual robotic manipulation is often necessary for diverse real-world tasks [1]. Recently, researchers have shown the merits of wrist cameras in visual-based robot learning for manipulation [2, 3, 4], as they help simplify certain aspects of the visual scene and focus on task-relevant objects. However, a fundamental challenge remains: learning-based systems still require large amounts of data for effective generalization, and collecting additional data across different viewpoints and states is both costly and labor-intensive.

One way to address this issue is with data augmentation. This is a widely used technique in computer vision [5, 6] and visual reinforcement learning [7, 8] to broaden the training data and facilitate generalization. In robotics, prior work has explored ways to automatically generate and synthesize novel image views while preserving action labels [9, 10, 11, 12], although such efforts have been limited to single-arm settings. Bimanual manipulation introduces additional challenges, including higher degrees of freedom (DOFs), enforcing consistency across the two generated wrist-camera views, and ensuring that augmented actions remain valid for coordinated manipulation. A complementary approach to increase data coverage is Dataset Aggregation (DAgger) [13], which leverages a supervisor to provide corrective labels. However, this method incurs additional online environment interactions and assumes a supervisor is available, which is not always feasible.

*GSS holds concurrent appointments as a Professor at USC and as an Amazon Scholar. This paper describes work performed at USC and is not associated with Amazon.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

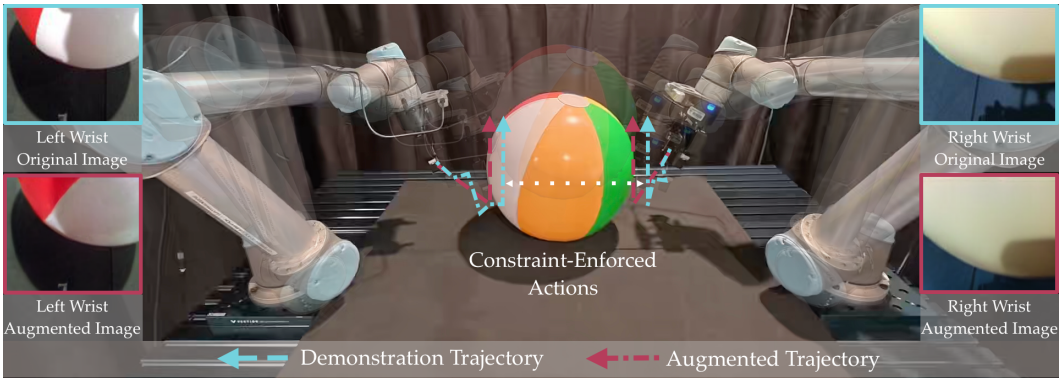


Figure 1: Overview of D-CODA for a coordinated bimanual lifting task with two UR5 arms. D-CODA is a method for offline data augmentation in bimanual eye-in-hand imitation learning. Given a pair of wrist camera images from demonstrations and sampled pose perturbations, a diffusion model generates novel and viewpoint-consistent wrist images for both arms (i.e., consistent object shape, color, position, and orientation). We use an optimization procedure to generate constraint-enforced actions to ensure augmented states are appropriate for bimanual coordination. This enables scalable data augmentation of diverse training data.

In this paper, we propose **D**iffusion for **CO**ordinated **D**ual-arm Data **A**ugmentation (**D-CODA**), a diffusion-based data augmentation framework tailored for eye-in-hand bimanual imitation learning. D-CODA synthesizes novel wrist-camera views along with consistent action labels to generate additional training data for bimanual manipulation policies *offline*, without the need for a simulator or the recreation of experimental setups. We design a diffusion model that takes two reference wrist images and camera pose perturbations as input and synthesizes novel viewpoint-consistent wrist-camera images for both arms. We leverage a Large Vision Model, SAM2 [14], to decompose any bimanual manipulation task into contactless (free-space) and contact-rich states. For contactless states, we uniformly sample random camera pose perturbations, while for contact-rich states, we employ constrained optimization to ensure that the perturbations satisfy coordination constraints required for bimanual manipulation. See Figure 1 for an overview.

Our contributions are as follows: (i) A novel method for bimanual manipulation that leverages diffusion models to generate diverse and consistent wrist camera images. (ii) A perception-based pipeline that decomposes any bimanual manipulation task into contactless and contact-rich states. For contact-rich states, we introduce a camera perturbation sampling procedure that generates constraint-enforced action labels. (iii) Experiments in 5 simulation and 3 real-world tasks that demonstrate the effectiveness of D-CODA over alternative baseline methods and ablations.

2 Related Work

Bimanual Manipulation. Bimanual manipulation [1] is essential for a wide range of real-world tasks that are difficult to perform with one arm, such as folding fabrics [15, 16, 17, 18, 19, 20], inserting objects into deformable bags [21, 22, 23], and handling food [24, 25]. These tasks often require tight coordination between the arms, either through simultaneous motions or an acting-stabilizing division of roles [25, 26] where one arm stabilizes parts of an item (e.g., holding food) to enable the other arm to act (e.g., cutting the food). While we mainly test our method for bimanual tasks where arms move simultaneously, our approach is not task-specific.

Some prior work on bimanual manipulation formalizes coordination via learned primitives [27] or constraint-based representations [28], but may suffer from generalization in unseen test-time scenarios. More general learning-based methods have emerged to address these limitations. Some rely on deep reinforcement learning (RL) [29, 30], which can be useful for simulation-based training of policies to control high-DOF hands [31, 32, 33, 34] or humanoids [35]. However, deep RL alone is generally difficult and brittle for bimanual manipulation [36]; therefore, researchers have explored imitation learning [37, 38, 39, 40, 41, 42, 43]. In a landmark paper, Zhao et al. [44] showed the benefit of predicting sequences of actions to learn fine-grained bimanual manipulation from

demonstrations. Data scaling [45, 46, 47] and improved robot hardware [48, 49, 50, 51, 52] have enabled great improvement and generalization in bimanual manipulation. Despite such progress, significant room remains to achieve human-level generalization, and methods still struggle when facing novel viewpoints or out-of-distribution states [12]. Our focus is complementary and is a general data augmentation approach compatible with diverse eye-in-hand imitation systems.

Data Augmentation in Robotics. Data augmentation is a widely used strategy to improve generalization in supervised learning systems such as behavioral cloning [53]. These methods suffer from compounding execution errors at test time, where small prediction errors lead to out-of-distribution states that result in larger errors [13]. Data augmentation techniques in robotics can be roughly divided into *environment-level* augmentation and *trajectory-level* augmentation. Environment-level methods aim to expand visual diversity or semantic richness of training data. These include automatic environment generation using LLMs [54, 55, 56, 57] and controllable visual and scene augmentation [58, 59, 60]. Some works also synthesize image-keypoint pairs [61] or hand-object interactions [62]. These techniques are complementary, as we study imitation learning from existing offline RGB trajectory data. More closely related works include RoVi-Aug [11] and VISTA [12], which use diffusion models to generate novel viewpoints but lack action label supervision. In contrast, D-CODA generates viewpoint-consistent images and corresponding joint-space action labels.

Trajectory-level augmentation methods [63, 64, 65, 66] synthesize new robot states, transitions, and/or actions. MimicGen [64], SkillMimicGen [65], and DexMimicGen [66] generate full demonstration trajectories, but rely on access to simulation or environment interaction during data generation, whereas D-CODA operates offline. Other works augment state-based inputs [67, 68] which limits applicability to vision-based learning. Zhou et al. [10] use NeRF [69] to augment visual input for corrective imitation but assume static scenes. Among the most closely related approaches is Diffusion Meets DAgger (DMD) [9], which augments single-arm eye-in-hand images with action labels using a diffusion model [70]. D-CODA builds on this foundation by demonstrating how to extend it to bimanual setups through a unified framework that synthesizes left and right wrist-camera views and employs constrained optimization to generate action labels suitable for bimanual manipulation.

3 Problem Statement and Preliminaries

We assume a bimanual robot with a left arm l and right arm r. Throughout the following sections, mathematical notations with superscripts l and r denote the left and right arms, respectively. We study vision-based eye-in-hand imitation learning, which trains a policy π_θ parameterized by θ that learns from demonstration data with wrist camera images. To indicate the source arm for each wrist camera image, we use the I^l and I^r notation, though we may suppress the superscripts if the distinction is not necessary. To represent images at time t in a demonstration, we use I^l_t and I^r_t. All images are in $\mathbb{R}^{H\times W\times 3}$ with matching height H and width W values. These form the policy input, which produces actions $\mathbf{a}_t = \pi_\theta((I^l_t, I^r_t))$. Here, $\mathbf{a}_t = (\mathbf{a}^l_t, \mathbf{a}^r_t)$, where $\mathbf{a}^l_t$ and $\mathbf{a}^r_t$ are target joint positions for the respective arms. To train π_θ, imitation learning uses a dataset of expert demonstrations $\mathcal{D} = \{\tau_1, \ldots, \tau_M\}$. Each τ_i is a sequence of wrist-camera images observations and actions: $\tau_i = (I^l_1, I^r_1, \mathbf{a}^l_1, \mathbf{a}^r_1, \ldots, I^l_T, I^r_T, \mathbf{a}^l_T, \mathbf{a}^r_T)$ for a demonstration with T time steps.

Synthesizing Novel Bimanual Images and Actions: Our method synthesizes novel eye-in-hand viewpoint images while automatically deriving suitable actions to make the robot return to in-distribution data. Based on [9], we formalize this problem as learning a function f_ψ that creates an eye-in-hand image conditioned on a current image and a pose perturbation Δp. In this case, let $\Delta p = {}_aT_b$ represent the pose transformation between two cameras a and b, where a is the source and b is the target. To represent images from these cameras for both arms, we suppress t and instead use the following notation: $\{I^l_a, I^r_a, I^l_b, I^r_b\}$. However, if notation requires specifying a camera $\{a, b\}$ as well as time t, both camera and time are included in the subscript (e.g., $I^l_{b,t}$), with the camera listed first, then the timestep. Given the source images I^l_a and I^r_a and pose transformations Δp^l and Δp^r as input, f_ψ must synthesize novel and consistent images $\tilde{I}^l_b$ and $\tilde{I}^r_b$ for the two cameras, matching the targets I^l_b and I^r_b. Additionally, we use Δp to compute perturbed actions $\tilde{\mathbf{a}}_t = (\tilde{\mathbf{a}}^l_t, \tilde{\mathbf{a}}^r_t)$. Finally, an augmented dataset of novel viewpoints with corresponding action labels, $\tilde{\mathcal{D}}$, is generated.

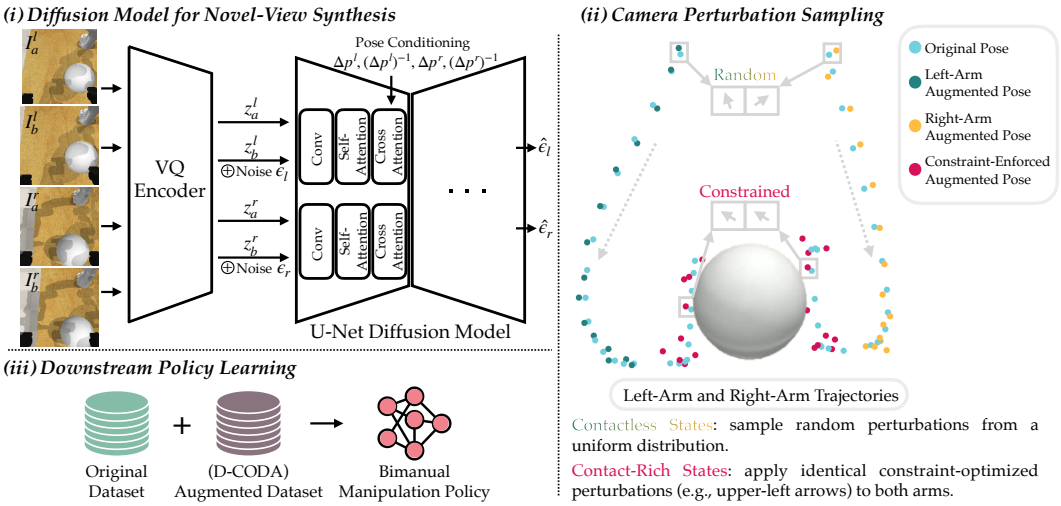


Figure 2: **Overview of D-CODA. (i):** The diffusion model is an iterative denoiser that learns to map source wrist-camera images I_a^l and I_a^r to target wrist-camera images I_b^l and I_b^r, conditioned on pose transformations Δp^l and Δp^r, using the original dataset (i.e., the dataset to be augmented). **(ii):** We use SAM2 [14] to decompose a bimanual manipulation task into contactless and contact-rich states. We uniformly sample random camera pose perturbations for contactless states (green and yellow dots). For contact-rich states (maroon dots), we use constrained optimization to sample perturbations that satisfy a set of constraints suitable for coordinated manipulation. We then employ the trained diffusion model to synthesize novel views based on the original dataset, using its images and corresponding sampled perturbations. This generates an augmented dataset. **(iii):** We combine the original and augmented datasets to train a bimanual manipulation policy.

4 Method: D-CODA

We introduce D-CODA, a diffusion-based framework for data augmentation of eye-in-hand bimanual imitation learning, which synthesizes novel wrist-camera views with action labels (see Figure 2).

4.1 Diffusion Model for Novel-View Synthesis

We modify the conditional diffusion model proposed by Zhang et al. [9] to synthesize novel wrist camera views for the two arms. The diffusion model, denoted as ϵ_ϕ, is an iterative denoiser. It is conditioned on the source images I_a^l and I_a^r and the pose transformations Δp^l and Δp^r. The diffusion targets are I_b^l and I_b^r. Both source and target images are passed through a VQ-GAN autoencoder V [71, 72] to allow denoising on the latent representations $\{z_a^l,\ z_{b,t}^l,\ z_a^r,\ z_{b,t}^r\}$, which correspond to the source and target images of both robot arms. This enables the diffusion process to operate in the latent space of the autoencoder rather than the high-dimensional pixel space. The model is trained to predict $\hat{\epsilon}_l$ and $\hat{\epsilon}_r$, which correspond to the noise terms ϵ_l and ϵ_r that were added to the latent vectors of the noise targets $z_{b,t}^l$ and $z_{b,t}^r$. Thus, the training objective is to minimize $\mathcal{L}$:

$$\mathcal{L} = \|\epsilon_l - \hat{\epsilon}_l\|_2^2 + \|\epsilon_r - \hat{\epsilon}_r\|_2^2 \quad \text{where} \quad \{\hat{\epsilon}_l, \hat{\epsilon}_r\} = \epsilon_\phi\Big(z_{b,t}^l, V(I_a^l), \Delta p^l,\ z_{b,t}^r, V(I_a^r), \Delta p^r, t\Big) \quad (1)$$

and where $z_{b,0}^l = V(I_b^l)$ and $z_{b,0}^r = V(I_b^r)$. The diffusion model architecture [70] is based on U-Net [73], which consists of convolution, cross-attention, and self-attention layers. To condition the model on pose transformations, we inject $\Delta p^l, (\Delta p^l)^{-1}, \Delta p^r, (\Delta p^r)^{-1}$ into the cross-attention layers. This improves the feature representations by incorporating relative camera pose information between the source and target views [70]. During training, we randomly sample images $\{I_a^l, I_b^l, I_a^r, I_b^r\}$ from a robot trajectory to construct the input $(I_a^l, I_b^l, \Delta p^l, I_a^r, I_b^r, \Delta p^r)$ for the model, and we compute $\Delta p = {}_aT_b$ by taking the matrix product of the inverse of camera pose a and camera pose b. Given a dataset of expert demonstrations $\mathcal{D}$, we train the diffusion model on $\mathcal{D}$ for a fixed number of iterations. We then use the trained model and sampled camera perturbations (subsection 4.2) to synthesize novel wrist camera views based on the same dataset.

4.2 Camera Perturbation Sampling

While the prior formulation enables image synthesis, it lacks constraint-enforced action sampling to ensure that sampled perturbations are valid. We introduce a novel camera pose sampling procedure for coordinated bimanual manipulation tasks. Given such a task, we decompose it into contactless and contact-rich states. To detect such contact, we use SAM2 [14] to extract segmentation masks of the grippers and track them throughout the robot trajectories. This approach enables accurate segmentation even when the grippers are closing or partially closed. If depth images from the wrist cameras are available, we use them in conjunction with the masks to detect contact events using z-score filtering, depth thresholding, and mask filling. If depth images are unavailable, we infer contact events by checking whether the grippers are fully visible within the wrist camera view using the Structural Similarity Index (SSIM) [74].

If gripper-to-object contact is absent, we uniformly sample a random direction for each arm within a predefined range of magnitudes $[m_{lb}, m_{ub}]$ and rotations $[r_{lb}, r_{ub}]$ to generate camera perturbations. However, when contact is detected, we formulate camera perturbation sampling as a constrained optimization problem. Our key insight is to apply identical perturbations to both arms during contact events, ensuring coordinated behavior. For this, we employ Dual Annealing [75], a global optimization algorithm capable of handling constraints, with early stopping. The decision variables are the translation coordinates c_{trans}, representing the transformation applied to the camera perturbations (normalized to [-1, 1]). The cost function penalizes perturbations that are too small, and end-effector poses that are either too close to the table or too close to the other end-effector. Additionally, we use an inverse kinematics solver based on the Levenberg-Marquardt (LM) method to check the feasibility of the perturbed end-effector poses. We define the overall optimization problem as:

$$\underset{c_{\text{trans}}}{\text{minimize}} \quad \text{Cost}(c_{\text{trans}}) \quad \text{subject to} \quad \begin{cases} c_{\text{trans}} \in [-1, 1]^3 \quad \text{and} \quad c_{\text{trans}} \geq m_{\text{lb}} \\ \text{ProximityToTable}(c_{\text{trans}}) \geq d_{\text{table}} \\ \text{ProximityToOtherEEF}(c_{\text{trans}}) \geq d_{\text{eff}} \\ \text{IKSolver}(c_{\text{trans}}) = \text{valid} \end{cases}$$

We construct the transformation matrix for the camera perturbation T using the lowest-cost c_{trans} and the identity rotation matrix. In short, this sampling strategy aims to identify a subset of feasible perturbations that better supports coordinated bimanual manipulation tasks.

4.3 Action Labeling and Dataset Construction

Given the dataset $\mathcal{D}$ and its corresponding sampled camera perturbations, we use the trained diffusion model to synthesize novel wrist camera views for both arms. To generate perturbed end-effector poses, we perform matrix multiplications involving the camera perturbation transformation T, the original camera pose C, and the end-effector pose E: $C \cdot T \cdot (C)^{-1} \cdot E$. Since our eye-in-hand imitation learning algorithm operates in joint space, we use the LM inverse kinematics solver to compute the perturbed target joint positions $\tilde{\mathbf{a}}_t$ ($\tilde{\mathbf{a}}_t^l$ and $\tilde{\mathbf{a}}_t^r$). If the resulting configuration is invalid, we discard the augmentation for that state and retain the original state information. Otherwise, we replace the original state with the augmented (out-of-distribution) state every k timesteps, which helps mitigate the issue of compounding errors in behavior cloning policies. The non-augmented action labels and corresponding states remain in-distribution, which guides the behavior cloning policies to complete the tasks. This will result in an augmented dataset of novel views $\tilde{\mathcal{D}}$, and π_θ is trained on $\mathcal{D} = \mathcal{D} \cup \tilde{\mathcal{D}}$.

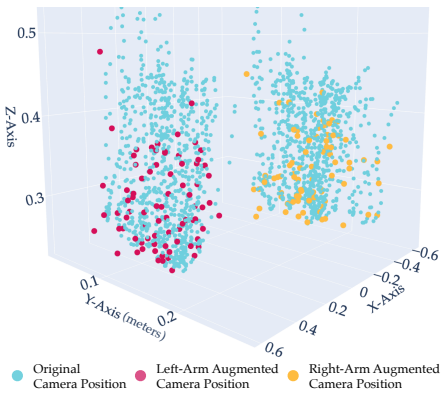


Figure 3: Isometric view of original and augmented camera positions for the real-world `Lift Ball` task. The augmented camera positions (maroon and yellow dots) provide broader coverage of state-space regions not occupied by the original camera positions (blue dots).

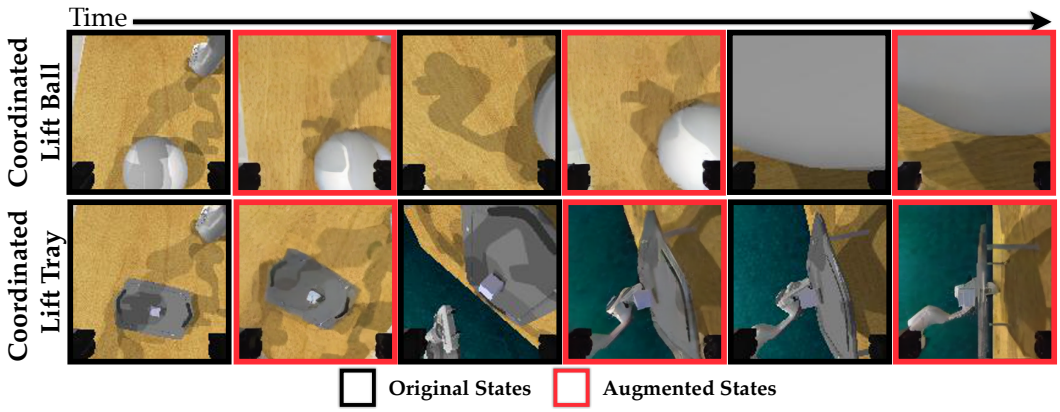


Figure 4: Examples of the original and synthesized wrist-camera images using D-CODA on `Coordinated Lift Ball` and `Coordinated Lift Tray` tasks in simulation. The first black column of images are the original states where the following column of red images are the augmented (perturbed original) states. All original and augmented state pairs are at the same timestep and each task is from the same episode. See Appendix G for more examples.

See Figure 3 for a visualization of the original and augmented camera positions in a combined dataset, and Figure 4 for examples of the synthesized images.

5 Experiments and Results

Simulation. We adopt five bimanual tasks from PerAct2 [76], which is built on top of RLBench [77], a popular robot manipulation benchmark. To improve the performance of the ACT baseline [44], we simplify certain tasks by reducing the axes of variation (e.g., shrinking the workspace), as mentioned below. We use the following five bimanual tasks (see the Appendix for more details):

- **`Coordinated Lift Ball`**: a ball is randomly spawned in a workspace of 0.65×0.91 m, same as PerAct2. A success is when the ball is lifted to a height above 0.95 m.
- **`Coordinated Lift Tray Easy`**: a tray with an item placed on top is randomly spawned in a workspace of 0.46×0.64 m, 70% of PerAct2's original workspace, and the tray does not rotate. A success is when the tray and the item are lifted to a height above 1.2 m.
- **`Coordinated Push Box Easy`**: a large box and a target area are randomly spawned in a workspace of 0.59×0.82 m, 90% of PerAct2's original workspace, and the box may be randomly rotated by up to 4 degrees. A success is when the box reaches the target area.
- **`Dual Push Buttons`**: three buttons with different colors are randomly spawned in a workspace of 0.65×0.91 m, as in PerAct2. A success is when two specified buttons are pressed simultaneously.
- **`Bimanual Straighten Rope`**: a long rope and target areas are randomly spawned in a workspace of 0.65×0.91 m, same as PerAct2. A success is when both ends of a rope are in their target areas.

Real-World. We use three coordinated tasks: `Lift Ball`, `Lift Drawer`, and `Push Block`, as coordinated dual-arm data augmentation is our primary focus. We use two CB2 UR5 6-DOF robot arms in a bimanual setup in a 0.97×0.79 m workspace, with a front camera and two wrist-mounted cameras. Each arm has a Robotiq 2F-85 parallel-jaw gripper and an Intel RealSense D415 RGB-D wrist camera. An experienced roboticist uses GELLO [78] to teleoperate the robots and collects $\sim$32 demonstrations per task. For evaluation, we perform 20 rollouts per task. In `Lift Ball`, a 0.35 m diameter ball is placed in randomized positions within a 0.64×0.20 m region. A success is when the ball is lifted to a height above 0.25 m. In `Lift Drawer`, a $0.29 \times 0.29 \times 0.15$ m square drawer is placed randomly within a 0.48×0.38 m region and randomly rotated up to 25°. A success is when the drawer is lifted to a height above 0.22 m. In `Push Block`, a $0.07 \times 0.35 \times 0.12$ m foam block is randomly placed within a 0.97×0.43 m region and rotated up to 13°. A success is when the block is pushed past the front of the workspace.

Method	# of Cameras	Coordinated Lift Ball	Coordinated Lift Tray	Coordinated Push Box	Dual Push Buttons	Bimanual Straighten Rope
Fine-tuned VISTA	3	61.3	2.6	**76.0**	1.3	26.7
D-CODA (ours)	3	**77.3**	34.7	58.7	34.7	**48.0**
Bimanual DMD	2	50.7	13.3	32.0	48.0	13.0
ACT (more data)	2	48.0	26.7	29.3	49.3	26.7
ACT (w/o augment.)	2	56.0	37.3	36.0	46.7	18.7
D-CODA (ours)	2	73.3	**44.0**	56.0	**53.3**	30.7

Table 1: Results from simulation experiments comparing D-CODA against four baselines (see Section 5.1). The success rate results are the average evaluation over three seeds. The ACT policy is used across all methods.

5.1 Baselines

In simulation, we compare D-CODA against strong baselines: **Fine-tuned VISTA** [12] and **Bimanual DMD**. All methods generate an augmented dataset, and we train **Action Chunking with Transformers (ACT)** [44], a state-of-the-art imitation learning method for bimanual manipulation, on both the augmented and original data to evaluate task performance. For all baselines, we adopt PerAct2's ACT implementation with fine-tuned action chunk sizes. VISTA leverages a diffusion-based novel view synthesis model, ZeroNVS [79], to augment third-person viewpoints from a single third-person view. We fine-tune VISTA on each task's training data using 10 randomly sampled overhead camera viewpoints drawn from a quarter-circle arc distribution. Following the training strategy of the best-performing VISTA variant [12], we train ACT on both the augmented overhead camera images and the original wrist-camera images. The Bimanual DMD baseline uses one DMD model per arm to synthesize wrist-camera images and employs the same k, interval at which original states are replaced, and random seed as D-CODA to generate perturbed actions and augmented states. The ACT (w/o augment.) baseline is trained only on the original dataset, serving as a reference for ACT performance without data augmentation. The ACT (more data) is trained on the original dataset along with 100 additional demonstrations without data augmentation, serving as an upper bound for ACT performance with more expert data.

5.2 Experiment Protocol and Evaluation

In simulation, we use the same training, validation, and testing data with the same environment seeds across all methods to ensure a fair comparison. Demonstrations are generated using a waypoint-based motion planner in RLBench [77]. We train the ACT policy for all methods using 100 episodes of training data along with their corresponding augmented data, saving a checkpoint every 2,000 iterations up to a total of 260,000 iterations. All checkpoints are validated using the same 25 episodes of validation data. Based on validation performance, the best-performing checkpoint is then evaluated on 25 unseen test data. In real-world experiments, we use the last checkpoint for each method and attempt to use the same starting configurations (e.g., object spawn locations and rotations).

5.3 Simulation Results

Table 1 reports the test success rates of different methods in simulation. D-CODA outperforms the baselines on 4 out of 5 tasks, including non-coordinated tasks such as `Dual Push Buttons` and `Bimanual Straighten Rope`. However, its performance is lower than VISTA on `Coordinated Push Box` because wrist-camera views offer poor visibility of the scene (i.e., the position of the box relative to the target area). As a result, augmenting wrist-camera views does not significantly improve the ACT baseline performance, although we still achieve a 20% improvement. Qualitatively, all methods can fail due to imprecise grasping, pushing, or placing of objects. Baseline methods, particularly VISTA, struggle with tasks that require a low tolerance for error, such as grasping tray handles or pushing small buttons. Overall, D-CODA makes fewer errors in these scenarios. We also observe that both Bimanual DMD and D-CODA, which generate out-of-distribution states, learn to recover from failures. For instance, when the grippers slide off the box during the `Coordinated Push Box` task, they recover by repositioning the grippers and continuing to push

the box to complete the task. Another interesting observation is that all methods using an overhead camera have worse performance by the downstream ACT policy on high-precision tasks (e.g., `Coordinated Lift Tray` and `Dual Push Buttons`) compared to ACT without using the overhead camera. Therefore, we suspect this limitation is from the design of the downstream ACT policy rather than the data augmentation methods. Further, we found that learning from more data, ACT (more data), does not always improve policy performance, and data augmentation shows potential for improving performance. See the Appendix for details of our ablation study.

5.4 Real-World Results

Real-world results are shown in Table 2 and example rollouts in Figure 6. The top three rows use ACT as the downstream manipulation policy, while the bottom two use π_0-FAST [80], a vision-language-action model. π_0-FAST is fine-tuned using Low-Rank Adaptation (LoRA) with the Gemma-2B-LoRA variant for 150,000 training steps, provided in [80]. Fine-tuned VISTA uses all three cameras, following its best-performing variant, whereas the other methods use only the wrist cameras, except in `Push Block`, where we found that all methods benefit from third-person views. D-CODA outperforms baselines on all three tasks based on evaluations over 20 trials.

Method	Lift Ball	Lift Drawer	Push Block
Fine-tuned VISTA	12 / 20	0 / 20	**20 / 20**
ACT (w/o augment.)	15 / 20	7 / 20	15 / 20
D-CODA (ours)	**17 / 20**	**14 / 20**	**20 / 20**
π_0-FAST (w/o augment.)	2 / 20	**1 / 20**	**20 / 20**
D-CODA (ours)	**12 / 20**	**1 / 20**	**20 / 20**

Table 2: Real-world experiment results comparing D-CODA with baselines, with 20 trials per method and task combination.

We observe that in `Lift Ball`, when using ACT, the robot arms freeze less frequently with D-CODA compared to baselines when the arms are in contact and lifting the ball. When using π_0-FAST, the baseline frequently misses the ball by moving the arms over it, or squeezes the ball so tightly that it triggers a force limit error on the robot. In contrast, D-CODA more reliably completes the task by positioning the arms beneath the ball to lift it, a strategy not seen in the baseline. However, most failures of D-CODA are due to large action values generated from the policy, causing the arms to deviate from the intended trajectory. We suspect that the large actions may result from the discontinuous nature of action tokens, as the augmented states are out-of-distribution, perturbed original states, which could inadvertently cause the policy to learn to output actions that suddenly deviate from the trajectory. This issue does not appear in ACT and might be mitigated by adopting a smoother action token representation. In `Push Block`, the robot arms using D-CODA get stuck less often when the block is positioned farther from the grippers, compared to the ACT baseline. In `Lift Drawer`, our method reaches the sides of the drawer more frequently than the baselines, an intermediate subgoal necessary to complete the task. Fine-tuned VISTA performs very poorly on this task, similar to its performance in `Coordinated Lift Tray Easy`. In 9 out of 20 trials, VISTA successfully reached the sides of the drawer, but the policy failed to close the grippers. These results suggest that VISTA struggles with tasks requiring precise manipulation, as shown in both simulation and real-world experiments. We suspect that this limitation arises from the use of augmentations for third-person views, which may adversely affect policy learning when wrist-camera views are more critical for task success. In other words, the policy may prioritize learning invariant features from third-person views rather than focusing on task-relevant features in the wrist-camera views. Overall, both π_0-FAST and ACT demonstrate improved performance with our data augmentation; however, with limited training data, ACT appears to exhibit greater reliability.

6 Conclusion

In this paper, we study data augmentation for bimanual manipulation, focusing specifically on eye-in-hand bimanual imitation learning. Our method, D-CODA, uses a diffusion model to generate diverse and consistent wrist camera images while enforcing and generating appropriate action labels using constrained optimization. By augmenting data, we obtain improved imitation learning performance across a range of diverse bimanual tasks. We hope our work inspires future exploration of data augmentation methods for bimanual manipulation.

7 Limitations

While promising, D-CODA has limitations that suggest opportunities for future work. First, our method is limited to augmenting wrist view images and is not intended for third-person view augmentation. Augmenting third-person views while modifying the action labels is nontrivial, as it requires the augmented views to reflect the change in movements of the robot arms implied by the augmented action labels. Another limitation is that our method relies on the distribution of novel camera poses being "sufficiently similar" to those in training, and would likely suffer with substantially different camera poses. Finally, although using D-CODA improves downstream policy performance by reducing the number of failures, it does not completely eliminate them.

8 Acknowledgments

We thank our colleagues from the Robotic Embedded Systems Laboratory (RESL) and the Sensing, Learning, and Understanding for Robotic Manipulation (SLURM) lab at the University of Southern California for their fruitful discussions and helpful writing feedback.

References

[1] F. Krebs and T. Asfour. A Bimanual Manipulation Taxonomy. In *IEEE Robotics and Automation Letters (RA-L)*, 2022.

[2] K. Hsu, M. J. Kim, R. Rafailov, J. Wu, and C. Finn. Vision-Based Manipulators Need to Also See from Their Hands. In *International Conference on Learning Representations (ICLR)*, 2022.

[3] M. J. Kim, J. Wu, and C. Finn. Giving Robots a Hand: Learning Generalizable Manipulation with Eye-in-Hand Human Video Demonstrations. *arXiv preprint arXiv:2307.05959*, 2023.

[4] S. Young, D. Gandhi, S. Tulsiani, A. Gupta, P. Abbeel, and L. Pinto. Visual Imitation Made Easy. In *Conference on Robot Learning (CoRL)*, 2020.

[5] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR)*, pages 958–963. IEEE, 2003.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NeurIPS)*, 2012.

[7] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas. Reinforcement learning with augmented data. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[8] I. Kostrikov, D. Yarats, and R. Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations (ICLR)*, 2021.

[9] X. Zhang, M. Chang, P. Kumar, and S. Gupta. Diffusion Meets DAgger: Supercharging Eye-in-hand Imitation Learning. In *Robotics: Science and Systems (RSS)*, 2024.

[10] A. Zhou, M. J. Kim, L. Wang, P. Florence, and C. Finn. NeRF in the Palm of Your Hand: Corrective Augmentation for Robotics via Novel-View Synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[11] L. Y. Chen, C. Xu, K. Dharmarajan, M. Z. Irshad, R. Cheng, K. Keutzer, M. Tomizuka, Q. Vuong, and K. Goldberg. RoVi-Aug: Robot and Viewpoint Augmentation for Cross-Embodiment Robot Learning. In *Conference on Robot Learning (CoRL)*, 2024.

[12] S. Tian, B. Wulfe, K. Sargent, K. Liu, S. Zakharov, V. Guizilini, and J. Wu. View-Invariant Policy Learning via Zero-Shot Novel View Synthesis. In *Conference on Robot Learning (CoRL)*, 2024.

[13] S. Ross, G. J. Gordon, and J. A. Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[14] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[15] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg. SpeedFolding: Learning Efficient Bimanual Folding of Garments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[16] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel. Cloth Grasp Point Detection Based on Multiple-View Geometric Cues with Application to Robotic Towel Folding. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[17] A. Canberk, C. Chi, H. Ha, B. Burchfiel, E. Cousineau, S. Feng, and S. Song. Cloth funnels: Canonicalized-alignment for multi-purpose garment manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[18] A. Colomé and C. Torras. Dimensionality reduction for dynamic movement primitives and application to bimanual manipulation of clothes. In *IEEE Transactions on Robotics*, 2018.

[19] G. Salhotra, I.-C. A. Liu, and G. Sukhatme. Learning robot manipulation from cross-morphology demonstration. In *Conference on Robot Learning (CoRL)*, 2023.

[20] T. Weng, S. Bajracharya, Y. Wang, K. Agrawal, and D. Held. Fabricflownet: Bimanual cloth manipulation with a flow-based policy. In *Conference on Robot Learning (CoRL)*, 2021.

[21] L. Y. Chen, B. Shi, D. Seita, R. Cheng, T. Kollar, D. Held, K. Goldberg, K. Goldberg, K. Goldberg, K. Goldberg, K. Goldberg, and K. Goldberg. AutoBag: Learning to Open Plastic Bags and Insert Objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[22] L. Y. Chen, B. Shi, R. Lin, D. Seita, A. Ahmad, R. Cheng, T. Kollar, D. Held, and K. Goldberg. Bagging by Learning to Singulate Layers Using Interactive Perception. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[23] A. Bahety, S. Jain, H. Ha, N. Hager, B. Burchfiel, E. Cousineau, S. Feng, and S. Song. Bag All You Need: Learning a Generalizable Bagging Strategy for Heterogeneous Objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[24] J. Grannen, Y. Wu, S. Belkhale, and D. Sadigh. Learning Bimanual Scooping Policies for Food Acquisition. In *Conference on Robot Learning (CoRL)*, 2022.

[25] J. Grannen, Y. Wu, B. Vu, and D. Sadigh. Stabilize to act: Learning to coordinate for bimanual manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[26] I.-C. A. Liu, S. He, D. Seita, and G. Sukhatme. VoxAct-B: Voxel-Based Acting and Stabilizing Policy for Bimanual Manipulation. In *Conference on Robot Learning (CoRL)*, 2024.

[27] A. Batinica, B. Nemec, A. Ude, M. Rakovic, and A. Gams. Compliant movement primitives in a bimanual setting. In *IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017.

[28] L. Ureche and A. Billard. Constraints extraction from asymmetrical bimanual tasks and their use in coordinated behavior. *Robotics and Autonomous Systems*, 103:222–235, 2018.

[29] R. Chitnis, S. Tulsiani, S. Gupta, and A. Gupta. Efficient bimanual manipulation using learned task schemas. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[30] Y. Li, C. Pan, H. Xu, X. Wang, and Y. Wu. Efficient Bimanual Handover and Rearrangement via Symmetry-Aware Actor-Critic Learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[31] Y. Chen, Y. Yang, T. Wu, S. Wang, X. Feng, J. Jiang, S. M. McAleer, H. Dong, Z. Lu, and S.-C. Zhu. Towards Human-Level Bimanual Dexterous Manipulation with Reinforcement Learning. In *Neural Information Processing Systems (NeurIPS)*, 2022.

[32] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik. Learning Visuotactile Skills with Two Multifingered Hands. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[33] T. Lin, Z.-H. Yin, H. Qi, P. Abbeel, and J. Malik. Twisting Lids Off with Two Hands. In *Conference on Robot Learning (CoRL)*, 2024.

[34] K. Zakka, P. Wu, L. Smith, N. Gileadi, T. Howell, X. B. Peng, S. Singh, Y. Tassa, P. Florence, A. Zeng, and P. Abbeel. Robopianist: Dexterous piano playing with deep reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2023.

[35] C. Sferrazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel. HumanoidBench: Simulated Humanoid Benchmark for Whole-Body Locomotion and Manipulation. In *Robotics: Science and Systems (RSS)*, 2024.

[36] N. Chernyadev, N. Backshall, X. Ma, Y. Lu, Y. Seo, and S. James. BiGym: A Demo-Driven Mobile Bi-Manual Manipulation Benchmark. In *Conference on Robot Learning (CoRL)*, 2024.

[37] G. Franzese, L. d. S. Rosa, T. Verburg, L. Peternel, and J. Kober. Interactive imitation learning of bimanual movement primitives. *IEEE/ASME Transactions on Mechatronics*, 28(1):1–13, 2023.

[38] F. Xie, A. Chowdhury, M. C. De Paolis Kaluza, L. Zhao, L. L. Wong, and R. Yu. Deep Imitation Learning for Bimanual Robotic Manipulation. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[39] A. Bahety, P. Mandikal, B. Abbatematteo, and R. Martín-Martín. ScrewMimic: Bimanual Imitation from Human Videos with Screw Space Projection. In *Robotics: Science and Systems (RSS)*, 2024.

[40] L. X. Shi, A. Sharma, T. Z. Zhao, and C. Finn. Waypoint-based imitation learning for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[41] B. Zhou, H. Yuan, Y. Fu, and Z. Lu. Learning Diverse Bimanual Dexterous Manipulation Skills from Human Demonstrations. *arXiv preprint arXiv:2410.02477*, 2024.

[42] H. Zhou, R. Wang, Y. Tai, Y. Deng, G. Liu, and K. Jia. You Only Teach Once: Learn One-Shot Bimanual Robotic Manipulation from Video Demonstrations. *arXiv preprint arXiv:2501.14208*, 2025.

[43] G. Lu, T. Yu, H. Deng, S. S. Chen, Y. Tang, and Z. Wang. AnyBimanual: Transferring Unimanual Policy for General Bimanual Manipulation. *arXiv preprint arXiv:2412.06779*, 2024.

[44] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Robotics: Science and Systems (RSS)*, 2023.

[45] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. RDT-1B: a Diffusion Foundation Model for Bimanual Manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[46] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.

[47] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models. *arXiv preprint arXiv:2501.09747*, 2025.

[48] A. . Team. Aloha 2: An enhanced low-cost hardware for bimanual teleoperation, 2024. URL https://aloha-2.github.io/.

[49] Z. Fu, T. Z. Zhao, and C. Finn. Mobile ALOHA: Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation. In *Conference on Robot Learning (CoRL)*, 2024.

[50] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid. ALOHA Unleashed: A Simple Recipe for Robot Dexterity. In *Conference on Robot Learning (CoRL)*, 2024.

[51] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu. DexCap: Scalable and Portable Mocap Data Collection System for Dexterous Manipulation. In *Robotics: Science and Systems (RSS)*, 2024.

[52] R. Ding, Y. Qin, J. Zhu, C. Jia, S. Yang, R. Yang, X. Qi, and X. Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[53] D. A. Pomerleau. Efficient Training of Artificial Neural Networks for Autonomous Navigation. *Neural Computation*, 3, 1991.

[54] L. Wang, Y. Ling, Z. Yuan, M. Shridhar, C. Bao, Y. Qin, B. Wang, H. Xu, and X. Wang. GenSim: Generating Robotic Simulation Tasks via Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2024.

[55] P. Hua, M. Liu, A. Macaluso, Y. Lin, W. Zhang, H. Xu, and L. Wang. GenSim2: Scaling Robot Data Generation with Multi-modal and Reasoning LLMs. In *Conference on Robot Learning (CoRL)*, 2024.

[56] P. Katara, Z. Xian, and K. Fragkiadaki. Gen2Sim: Scaling up Robot Learning in Simulation with Generative Models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[57] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan. RoboGen: Towards Unleashing Infinite Data for Automated Robot Learning via Generative Simulation. In *International Conference on Machine Learning (ICML)*, 2024.

[58] Z. Chen, Z. Mandi, H. Bharadhwaj, M. Sharma, S. Song, A. Gupta, and V. Kumar. Semantically Controllable Augmentations for Generalizable Robot Learning. In *International Journal of Robotics Research (IJRR)*, 2024.

[59] C. Yuan, S. Joshi, S. Zhu, H. Su, H. Zhao, and Y. Gao. RoboEngine: Plug-and-Play Robot Data Augmentation with Semantic Robot Segmentation and Background Generation. *arXiv preprint arXiv:2503.18738*, 2025.

[60] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar. RoboAgent: Generalization and Efficiency in Robot Manipulation via Semantic Augmentations and Action Chunking. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[61] G. Tang, S. Rajkumar, Y. Zhou, H. R. Walke, S. Levine, and K. Fang. KALIE: Fine-Tuning Vision-Language Models for Open-World Manipulation without Robot Data. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[62] Y. Ye, X. Li, A. Gupta, S. D. Mello, S. Birchfield, J. Song, S. Tulsiani, and S. Liu. Affordance Diffusion: Synthesizing Hand-Object Interactions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[63] M. Laskey, J. Lee, R. Fox, A. D. Dragan, and K. Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on Robot Learning (CoRL)*, 2017.

[64] 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. In *Conference on Robot Learning (CoRL)*, 2023.

[65] C. Garrett, A. Mandlekar, B. Wen, and D. Fox. SkillMimicGen: Automated Demonstration Generation for Efficient Skill Learning and Deployment. In *Conference on Robot Learning (CoRL)*, 2024.

[66] 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. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[67] P. Mitrano and D. Berenson. Data Augmentation for Manipulation. In *Robotics: Science and Systems (RSS)*, 2022.

[68] L. Ke, Y. Zhang, A. Deshpande, S. Srinivasa, and A. Gupta. CCIL: Continuity-based Data Augmentation for Corrective Imitation Learning. In *International Conference on Learning Representations (ICLR)*, 2024.

[69] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

[70] J. J. Yu, F. Forghani, K. G. Derpanis, and M. A. Brubaker. Long-Term Photometric Consistent Novel View Synthesis with Diffusion Models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[71] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[72] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[73] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[74] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi:10.1109/TIP.2003.819861.

[75] Y. Xiang, D. Sun, W. Fan, and X. Gong. Generalized simulated annealing algorithm and its application to the thomson model. *Physics Letters A*, 233(3):216–220, 1997. ISSN 0375-9601. doi:https://doi.org/10.1016/S0375-9601(97)00474-X. URL https://www.sciencedirect.com/science/article/pii/S037596019700474X.

[76] M. Grotz, M. Shridhar, T. Asfour, and D. Fox. PerAct2: Benchmarking and Learning for Robotic Bimanual Manipulation Tasks. *arXiv preprint arXiv:2407.00278*, 2024.

[77] S. James, Z. Ma, D. Rovick Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. In *IEEE Robotics and Automation Letters (RA-L)*, 2020.

[78] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel. GELLO: A General, Low-Cost, and Intuitive Teleoperation Framework for Robot Manipulators. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[79] K. Sargent, Z. Li, T. Shah, C. Herrmann, H.-X. Yu, Y. Zhang, E. R. Chan, D. Lagun, L. Fei-Fei, D. Sun, and J. Wu. ZeroNVS: Zero-Shot 360-Degree View Synthesis from a Single Image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[80] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

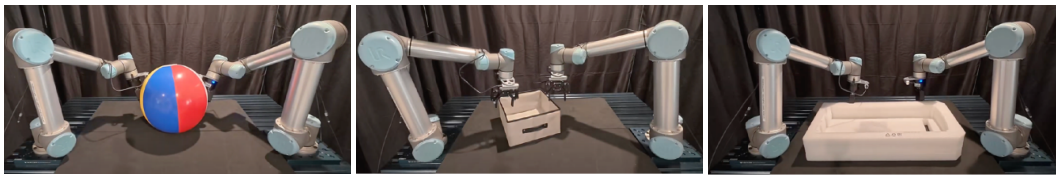
[81] A. Kuznetsova, H. Rom, N. Alldrin, J. R. R. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:2112.10752*, 2018.

[82] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018.

A Task Details

Coordinated Lift Ball | Coordinated Lift Tray Easy | Coordinated Push Box Easy | Dual Push Buttons | Bimanual Straighten Rope

Figure 5: Simulation environments for our bimanual manipulation tasks, adapted from PerAct2 [76].



Lift Ball | Lift Drawer | Push Block

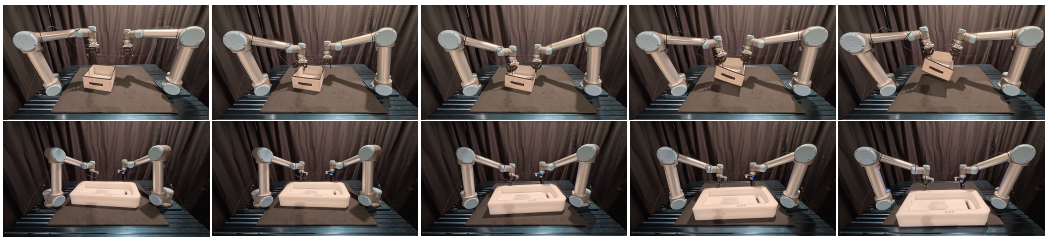


Figure 6: Top: Real-world bimanual manipulation tasks. Bottom: Example successful rollouts (`Lift Drawer` on top row; `Push Block` on bottom row) of D-CODA on a real-world bimanual setup with UR5s. See Section 5.4 for quantitative results.

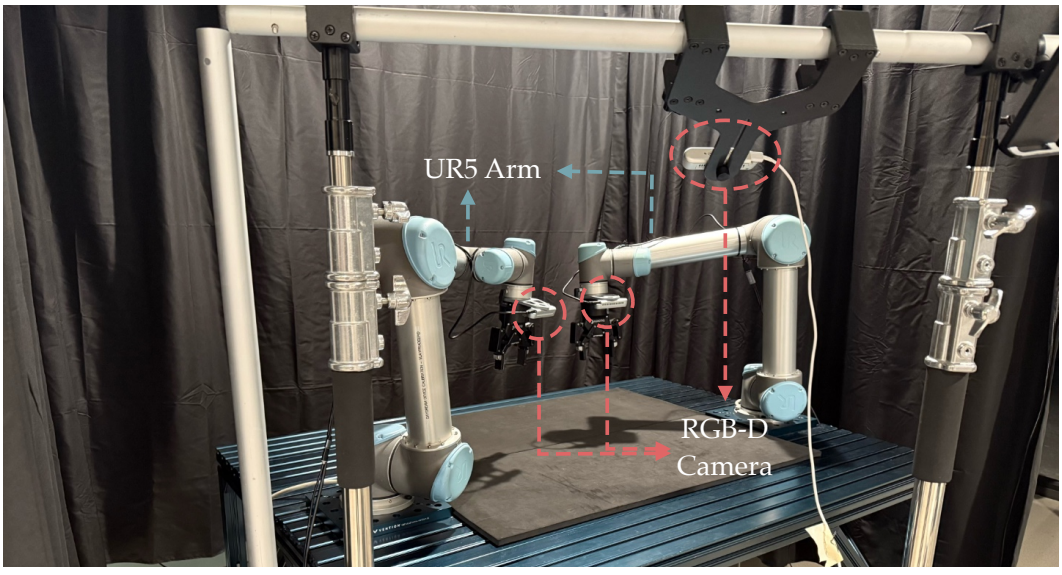


Figure 7: Real-world bimanual UR5 setup.

B Ablations

Ablations of D-CODA. In simulation, we test the following methods:

- **D-CODA with Replaced Encoders**: uses a VQGAN encoder trained on the Open Images [81] dataset from Latent Diffusion [72] instead of the RealEstate10K [82] dataset.
- **D-CODA w/o Constrained Optim.**: does not use constrained optimization to sample camera perturbations (i.e., random sampling).

Table 3 indicates that D-CODA performs best with constrained optimization and the original VQGAN encoder. D-CODA without constrained optimization fails more often than D-CODA during ball lifting, as expected, since the perturbations generated by the model are not constraint-enforced and are entirely random. See Section F for details on the importance of constraint-enforced action sampling. Additionally, the model with replaced encoders generates images with more artifacts, resulting in poorer policy performance.

Method	Coordinated Lift Ball
D-CODA with Replaced Encoders	53.3
D-CODA w/o Constrained Optim.	57.3
D-CODA (ours)	**73.3**

Table 3: Ablation experiment results in simulation.

C Generalization Experiment

We evaluate the diffusion model's generalization capability to unseen objects and tasks. For the zero-shot and few-shot experiments, we train the model on 100 demonstrations each from the following PerAct2 [76] tasks: `Coordinated Lift Tray`, `Pick Up Notebook`, `Pick Up Plate`, `Sweep Dust Pan`, and `Coordinated Push Box`. We then use the trained model to synthesize images for the `Coordinated Lift Ball` dataset. The following methods are tested:

Method	Coordinated Lift Ball
Zero-Shot	44.0
Few-Shot (10 demos)	60.0
Train from Scratch (100 demos)	**73.3**

Table 4: Generalization experiment results in simulation.

- **Zero-Shot**: uses the trained diffusion model to synthesize images without any fine-tuning.
- **Few-Shot (10 demos)**: fine-tunes the trained model for 3000 additional epochs using 10 demonstrations from the target `Coordinated Lift Ball` dataset, which is then use for image synthesis.
- **Train from Scratch (100 demos)**: Trains the diffusion model directly on 100 demonstrations from the `Coordinated Lift Ball` dataset, without using demonstrations from other tasks.

As shown in Table 4, the diffusion model performs best when trained directly on the target dataset (i.e., the dataset to be augmented). However, when data collection for the target task is costly, the model still achieves reasonable performance in the few-shot setting. Qualitatively, the images synthesized by the model trained from scratch contain the fewest artifacts, with image quality degrading as fewer target demonstrations are used during training.

D Additional Implementation Details

For training the diffusion model, we use the same VQ-GAN pre-trained checkpoint at 2000 epochs with frozen codebooks as DMD [9]. To randomly sample images $I_a^l, I_b^l, I_a^r, I_b^r$ from a robot trajectory to construct the input $(I_a^l, I_b^l, \Delta p^l, I_a^r, I_b^r, \Delta p^r)$, we sample from a range of $\{5, \ldots, 15\}$ for all simulation tasks, except for `Dual Push Buttons`, where we use $\{10, \ldots, 30\}$. For real-world experiments, we use a range of $\{1, \ldots, 3\}$ for all tasks. For example, if I_a^l, I_a^r are from timestep t, then I_b^l, I_b^r are sampled from a future timestep between $t+1$ and $t+3$ in real-world tasks. We use two NVIDIA 4060 Ti GPUs for both training the diffusion model and performing image synthesis.

For camera perturbation sampling, the translation magnitudes $[m_{lb}, m_{ub}]$ are set to 0.01 and 0.02 meters, respectively, for contactless and contact-rich states. For contactless states, the rotation bounds $[r_{lb}, r_{ub}]$ are set to –28.7 and 28.7 degrees, respectively. For k (i.e., the interval at which original states are replaced), we set $k = 6$ for `Coordinated Lift Ball`, `Coordinated Lift Tray Easy`, `Dual Push Buttons`, and all real-world tasks, and $k = 9$ for `Coordinated Push Box Easy` and `Bimanual Straighten Rope`. Figure 8 shows that our method is largely insensitive to the choice of k, which motivates our choice of 6 as the default value for most tasks.

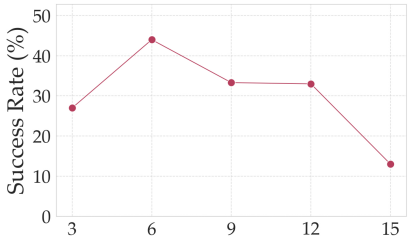


Figure 8: Effects of k on downstream ACT performance on `Coordinated Lift Tray Easy`.

Table 5 summarizes the ACT hyperparameters. While PerAct2 uses a default action chunk size of 10, we found it to yield suboptimal performance across most tasks. To address this, we tune the chunk size for all tasks except `Coordinated Lift Ball`, using a chunk size of 15 for `Coordinated Lift Tray Easy` and `Coordinated Push Box Easy`, and 60 for `Dual Push Buttons` and `Bimanual Straighten Rope`. We use a chunk size of 2 across all real-world tasks. In both simulation and real-world experiments, the RGB images have dimensions of 128×128. An NVIDIA 2080 Ti GPU is used to train the ACT policy.

Hyperparameter	Value
learning rate	1e-5
batch size	16
# encoder layers	4
# decoder layers	7
feedforward dimension	3200
hidden dimension	512
# heads	8
beta	100
dropout	0.1

Table 5: Hyperparameters of ACT

For real-world experiments, we use Intel RealSense D415 cameras to capture RGB images at a resolution of 640×480 pixels. These images are first zero-padded and then rescaled to 128×128. We use the python-urx library to control the robot arms and I/O programming to operate the Robotiq 2F-85 grippers.

E Additional Implementation Details for the Baselines

For fine-tuned VISTA, 10 overhead camera viewpoints are randomly sampled from a quarter-circle arc distribution and are used to train ZeroNVS with VISTA's default fine-tuning parameters. The ZeroNVS model is fine-tuned for 5,000 steps on four NVIDIA A40 GPUs. The resulting model is then used to synthesize overhead camera views for all timesteps in each episode. These synthesized images replace all the original overhead images and are used to train ACT.

F Lack of Constraint-Enforced Action Sampling

Figure 9 shows a visualization comparing constraint-enforced actions and random actions in the `Coordinated Lift Ball` task. At this timestep, the robot arms have reached the bottom of the ball and are about to lift it. If the next sampled actions are random (blue arrows), they may cause the ball to fall due to an increased distance between the end-effectors (black arrows). In contrast, if constraint-enforced actions are sampled (maroon arrows), the distance and orientations of the end-effectors are maintained, preserving the conditions necessary for the ball to remain stable atop the

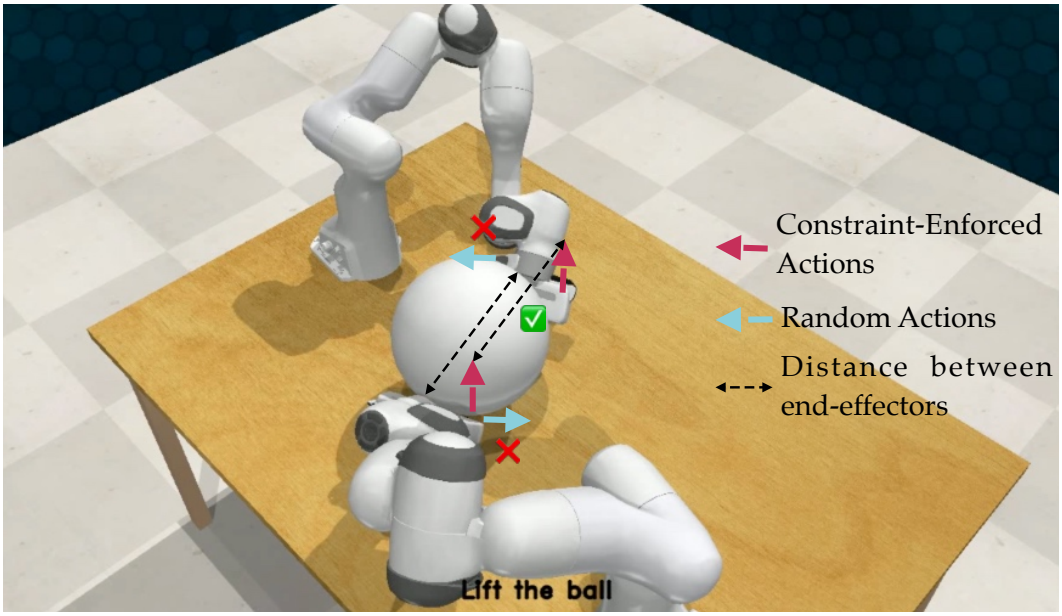


Figure 9: Visualization comparing constraint-enforced actions and random actions.

grippers. Thus, constraint-enforced actions are critical for achieving coordinated bimanual manipulation.

G Examples of Synthesized Images

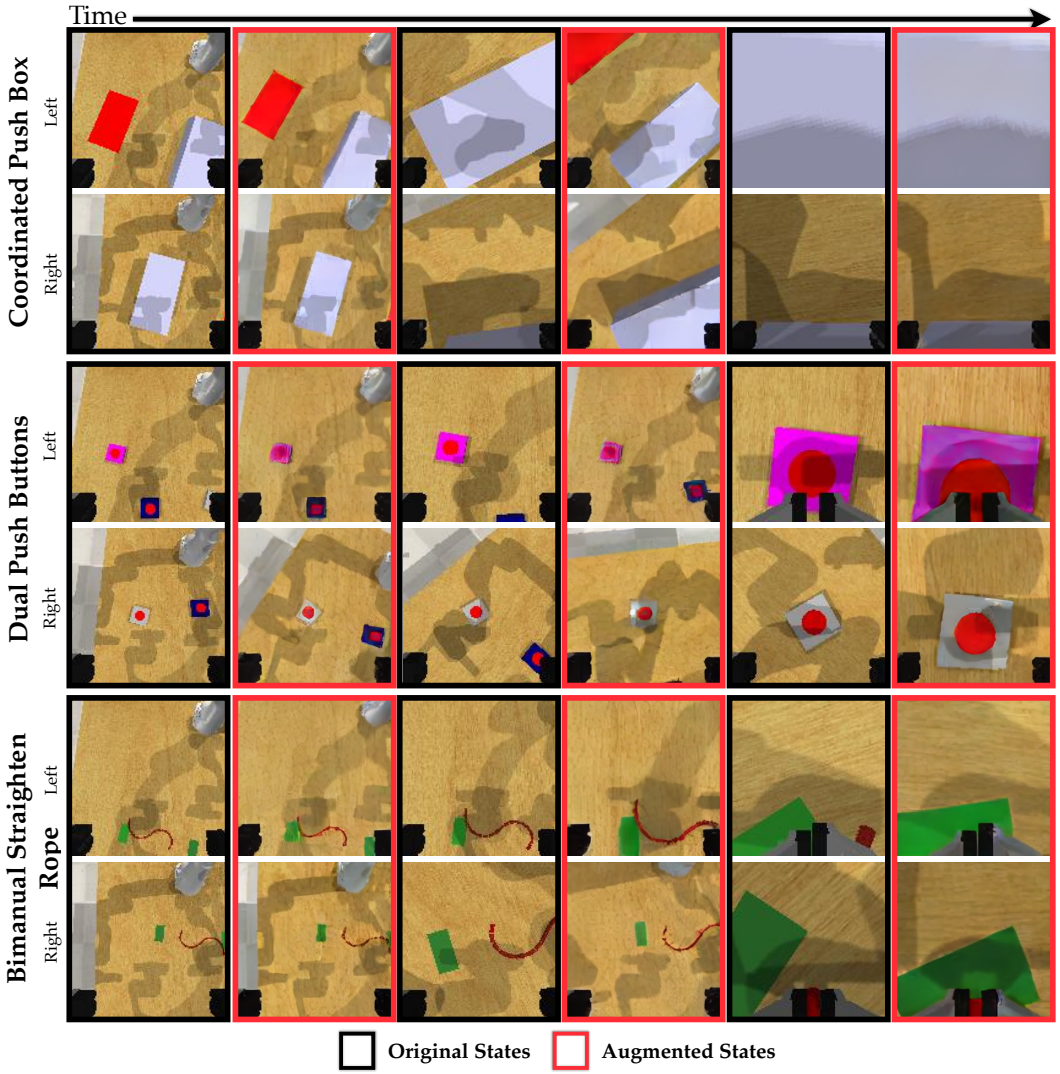


Figure 10: Examples of the original and synthesized wrist-camera images from both arms using D-CODA on `Coordinated Push Box`, `Dual Push Buttons`, and `Bimanual Straighten Rope` tasks in simulation. The first black column of images are the original states where the following column of red images are the augmented (perturbed original) states. All original and augmented state pairs are at the same timestep and each task is from the same episode.

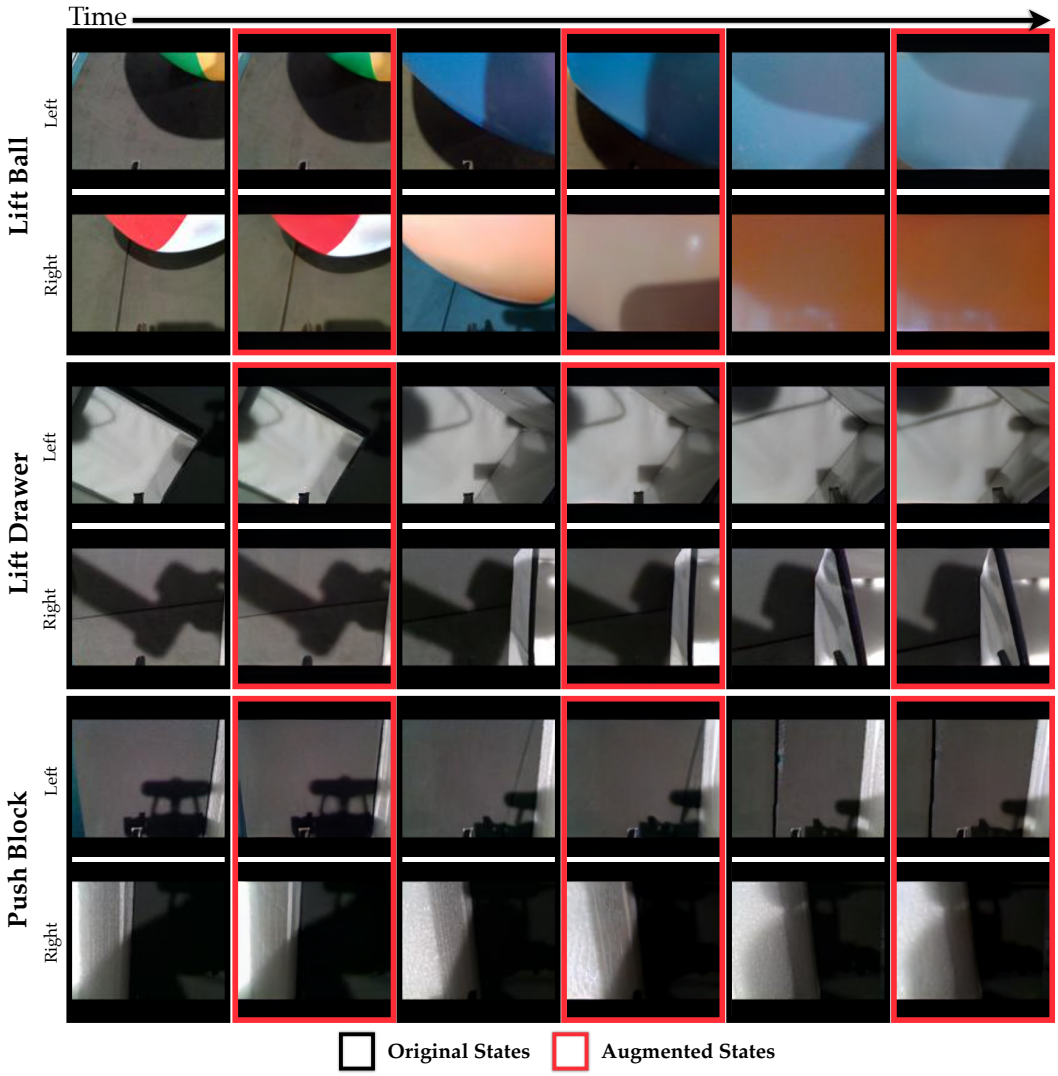


Figure 11: Examples of the original and synthesized wrist-camera images from both arms using D-CODA on the real-world `Lift Ball`, `Lift Drawer`, and `Push Block` tasks. The first black column of images are the original states where the following column of red images are the augmented (perturbed original) states. All original and augmented state pairs are at the same timestep and each task is from the same episode.