

Article

SCE-DP: System-Context-Conditioned Diffusion Policy Under Simulated Visual–Control Deployment Shifts

Xiaoyu Xiong [†], Guangtie Zhang [†], Beiyu Xue and Hui Li ^{*}

School of Innovation and Entrepreneurship, University of Science and Technology Beijing, Beijing 100083, China; u202342061@xs.ustb.edu.cn (X.X.); u202340049@xs.ustb.edu.cn (G.Z.); u202340102@xs.ustb.edu.cn (B.X.)

^{*} Correspondence: lihui1989@ustb.edu.cn

[†] These authors contributed equally to this work.

Featured Application

The proposed method provides a simulation benchmark and policy design for studying visuomotor robot manipulation under camera-placement, action-scaling, and control-delay shifts without explicit recalibration or test-time model-weight updates.

Abstract

Generalization under deployment shifts remains a major challenge for visuomotor robot manipulation, especially when changes in camera placement and control behavior are unknown to the policy. This paper introduces the System-Context-Conditioned Diffusion Policy (SCE-DP), a history-conditioned manipulation policy designed to adapt its latent context to simulated changes in camera extrinsics, action scale, and bounded control delay without explicit recalibration or test-time model-weight updates. SCE-DP encodes a short history of issued commands and their subsequent visual and proprioceptive responses into a latent system context, which conditions the diffusion-based action denoising process. Auxiliary system-parameter regression and one-step response prediction further encourage the context to capture deployment-relevant system properties and their behavioral consequences. We evaluate SCE-DP on five independently trained ManiSkill manipulation tasks across held-out in-range configurations, a withheld camera-delay composition, and mild extrapolation settings. SCE-DP improves the five-task macro-average over domain-randomized Diffusion Policy by 14.5 percentage points on held-out configurations and 15.7 points on the withheld camera-delay composition. Across four shifted evaluation suites, it achieves an average success rate of 64.3%, compared with 48.9% for the domain-randomized baseline, while preserving nominal performance. Command-only, response-only, previous-action, recovery-source, and matched explicit-parameter controls indicate that the gain is not explained solely by command statistics, redundant command input, or recovery-data composition. These results show in simulation that command–response history is an effective source of online latent system context under coupled visual and control shifts; real-robot generalization remains to be established.



Academic Editor: Roberto Zivieri

Received: 29 July 2026

Revised: 27 August 2026

Accepted: 4 September 2026

Published: 6 September 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: diffusion policy; robot manipulation; visuomotor imitation learning; online system identification; visual–control deployment shift; history-conditioned adaptation

1. Introduction

Visuomotor imitation learning allows robots to acquire contact-rich manipulation skills from demonstrations, reducing the need to handcraft task dynamics or control

objectives [1–4]. Diffusion Policy is a representative approach: it models action sequences through conditional denoising and executes them in a receding-horizon manner [5]. Its strong performance is usually evaluated under the same observation and control settings used for training. In simulated deployment-shift studies, however, the camera may be moved, action commands may be rescaled, or additional latency may be introduced by the control stack. Although the robot and task remain unchanged, these changes alter how a command appears in subsequent visual and proprioceptive observations.

A single observation is often insufficient to identify the active configuration because task state and system changes can produce similar evidence. Image changes may result from object motion or camera motion, while a small observed response may be caused by contact, reduced action scale, or delayed actuation. Domain randomization can improve robustness across such variations, but it does not reveal which configuration is active. Explicit calibration can resolve part of the ambiguity, but it requires parameters that may not be available after hardware or control changes.

Recent methods for handling viewpoint variation use calibration inputs, geometric action representations, 3D scene models, or view synthesis [6–10]. These methods mainly address changes in camera geometry. Our setting also includes action scale and control delay, both of which affect the relation between an issued command and the motion observed afterward. We therefore study camera pose, action scale, and control delay jointly rather than treating camera variation in isolation.

Recent interaction provides a direct source of information about the active system. Each issued command can be compared with the visual and proprioceptive changes that follow it. Online system identification and rapid adaptation use the same principle to infer hidden dynamics from recent state and action histories [11–13]. We apply this idea to diffusion policies that use RGB observations. Because the control delay is unknown, the next response cannot always be assigned to the latest command. We therefore represent each response together with a window of recent commands and use a temporal model to infer the relevant lag.

Based on this idea, we propose the System-Context-Conditioned Diffusion Policy (SCE-DP). A temporal Transformer encodes a sliding window of issued commands and subsequent visual and proprioceptive responses into a latent representation of the active system. This representation is combined with the current observation to condition the denoising network and is recomputed at every replanning step. During training, two auxiliary objectives predict the simulated system parameters and the next visual and proprioceptive response. During evaluation, the policy does not use the true system parameters or modify its model weights; only the latent context is updated from recent interaction.

Our contributions are as follows. First, we formulate a simulated visuomotor generalization setting that couples camera extrinsic shifts, arm-command scale changes, and bounded control delay. Second, we introduce a delay-uncertain command-response history representation that keeps multiple candidate issued commands instead of assuming a fixed one-step command-response pairing. Third, we condition a diffusion policy on a latent system context trained with both privileged simulated parameter supervision and behavioral response prediction. Fourth, we provide composition, extrapolation, shortcut, recovery-source, and explicit-parameter controls to separate the effect of command-response evidence from command statistics, task state, and recovery-data imbalance.

2. Related Work

2.1. Diffusion Policies for Robot Manipulation

Diffusion models learn complex, high-dimensional data distributions through iterative denoising [14–16]. In robot learning, they have been applied to trajectory planning and

behavior generation [17]. Diffusion Policy conditions the denoising process on visual observations, predicts action sequences, and executes them with receding-horizon control [5]. Other imitation learning methods model multimodal behavior through energy-based objectives, transformer architectures, or action chunking [1,3,4]. More recent methods introduce 3D scene representations or geometric constraints to improve spatial generalization [7,18]. Our work retains the action-generation framework of Diffusion Policy and studies changes in how commands are executed and observed after deployment.

2.2. Robustness to Camera Viewpoint Changes

Existing methods address camera variation by incorporating calibration information, 3D geometry, or view transformation. Calibration Diffusion Policy learns a camera representation from demonstrations collected under different viewpoints [6]. Three-Dimensional Diffuser Actor generates actions from a 3D scene representation and can generalize across viewpoints when depth or multi-view observations are available [7]. AnyCamVLA synthesizes an observation from a canonical viewpoint, while VistaBot combines camera pose estimation with view synthesis for closed-loop control [8,9]. CamVLA predicts actions in the camera frame and transforms them into the robot base frame using an estimated hand–eye transformation [10]. These methods estimate, encode, or normalize camera geometry from the current observation. They do not directly model how an issued command is rescaled or delayed before its effect becomes visible. SCE-DP therefore uses recent commands and their observed effects to handle camera changes together with action scaling and bounded control delay.

2.3. System Identification and History-Based Adaptation

History-based adaptation uses recent interaction to infer properties of the current environment or system. UP-OSI combines a universal policy with an online module that estimates dynamics parameters from recent states and actions [11]. Rapid Motor Adaptation (RMA) learns a history encoder from privileged information and uses it to adapt without updating the policy parameters during deployment [12,13]. These approaches motivate our use of recent interaction but they are usually formulated around proprioceptive dynamics or explicit physical parameters. Our setting also changes the visual observation mapping through camera extrinsics and changes the command–response timing through control delay. We therefore apply history-based adaptation to visual imitation learning, condition the diffusion denoising process directly, and give the history encoder a window of recent commands instead of a fixed pairing between one command and the next observation.

2.4. Domain Randomization and Data Aggregation

Domain randomization trains policies under sampled environment parameters to improve robustness to modeling errors and deployment variation [19–21]. Its performance depends on the training distribution, and the policy is not explicitly informed of the configuration active in a given episode. Control delay presents an additional challenge because sensing and actuation latency can reduce transfer performance even when other dynamics are randomized [22]. We therefore compare SCE-DP with a Diffusion Policy trained under the same randomized configurations.

Behavioral cloning also suffers from a mismatch between expert trajectories used for training and the states reached by the learned policy during deployment [2,23]. To reduce this mismatch, we collect one fixed recovery dataset from policy rollouts and label the visited states with an expert that has access to the deployment parameters. The same recovery data are used for all randomized comparison methods.

3. Problem Formulation

We consider manipulation environments that share the same robot embodiment, task definition, and scene structure but differ in their camera and control settings. At control step t , the policy observes

$$o_t = (I_t, q_t), \quad (1)$$

where I_t is a monocular RGB image and q_t contains the proprioceptive state available at deployment. The policy predicts a sequence of normalized commands,

$$A_t = [a_t, a_{t+1}, \dots, a_{t+T_p-1}], \quad (2)$$

and sends the first T_a commands to the controller before replanning. We denote the command issued at step t by a_t^{issued} .

We assume that the deployment configuration is fixed within an episode. The deployment configuration is represented by

$$\zeta = [\Delta \mathbf{p}_{\text{cam}}, \Delta \mathbf{r}_{\text{cam}}, s_a, d_a], \quad (3)$$

where $\Delta \mathbf{p}_{\text{cam}} \in \mathbb{R}^3$ and $\Delta \mathbf{r}_{\text{cam}} \in \mathbb{R}^3$ are offsets in camera position and roll–pitch–yaw orientation, s_a scales the arm command, and d_a is an integer control delay. The arm command applied by the controller is

$$a_t^{\text{applied}} = \text{clip}(s_a a_{t-d_a}^{\text{issued}}, -1, 1), \quad (4)$$

where clipping is applied elementwise. Gripper commands are delayed by d_a but are not scaled. The delay is unknown to the policy but is assumed to lie within a predefined represented range $0 \leq d_a \leq d_{\text{max}}$, with $d_{\text{max}} = 4$ in the main model. Configurations beyond this represented delay window are evaluated only as stress tests and are not part of the main generalization claim. The configuration ζ is not provided as a policy input during evaluation.

The goal is to maintain task performance across configurations, including those not seen during training. During evaluation, the policy can use past observations and issued commands, but it does not observe ζ or the applied commands and receives no separate calibration trajectory. Model parameters remain fixed; accordingly, online latent-context adaptation denotes updating only a latent representation of the current system from recent interaction history.

4. Method

4.1. Overview

SCE-DP extends a conditional Diffusion Policy with an encoder that uses recent interaction history. At each replanning step, the last H completed transitions are converted into record tokens and processed by a temporal Transformer. The resulting latent representation, denoted by $z_{\text{sys}}(t)$, captures information about the current camera and control configuration. In parallel, the current visual and proprioceptive observations are encoded as $h_{\text{obs}}(t)$. The two representations are combined to form the condition c_t for the diffusion model.

We use $H = 16$, selected on the validation set as reported in Table A5. During training, two auxiliary heads encourage $z_{\text{sys}}(t)$ to encode the system parameters and their effects on the observed response. These heads are not used for action generation at deployment.

The overall architecture of SCE-DP is illustrated in Figure 1.

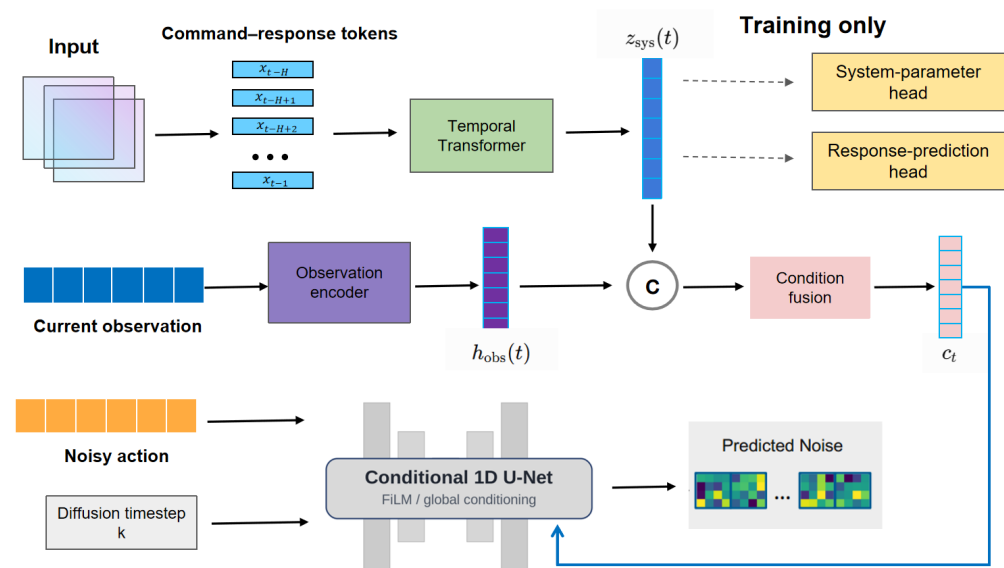


Figure 1. Overview of SCE-DP. A temporal Transformer encodes recent transition records into $z_{\text{sys}}(t)$, while the current observation is encoded as $h_{\text{obs}}(t)$. Their fused representation c_t conditions the Conditional 1D U-Net during denoising. The system parameter and response prediction heads are used only during training. Colors visually distinguish the observation, history-context, fusion, action, and auxiliary-learning components. Solid arrows denote the policy-conditioning and action-generation paths, whereas gray dashed arrows connect the auxiliary heads used only during training. The ellipses denote omitted intermediate history tokens or action-sequence elements for compact visualization and do not represent missing model components.

4.2. Diffusion Policy

Following DDPM and Diffusion Policy [5,14], let A_t^0 denote an expert action sequence, k a diffusion step, and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. The forward diffusion process produces

$$A_t^k = \sqrt{\bar{\alpha}_k} A_t^0 + \sqrt{1 - \bar{\alpha}_k} \epsilon. \quad (5)$$

A conditional 1D U-Net predicts the added noise:

$$\hat{\epsilon} = \epsilon_{\theta}(A_t^k, k, c_t). \quad (6)$$

The diffusion loss is

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{A_t^0, k, \epsilon} \left[\left\| \epsilon - \epsilon_{\theta}(A_t^k, k, c_t) \right\|_2^2 \right]. \quad (7)$$

We use an observation horizon $T_o = 2$, a prediction horizon $T_p = 16$, and an execution horizon $T_a = 4$ for all tasks. SCE-DP retains the action representation and diffusion objective of the baseline and changes how the condition c_t is constructed.

4.3. History Representation

When the control delay is unknown, the observed change between steps i and $i + 1$ cannot always be assigned to the most recent command. It may instead result from an earlier command $a_{i-d_a}^{\text{issued}}$. We therefore represent each completed transition together with a window of recent commands.

Let $d_{\text{max}} = 4$ be the maximum delay represented by the model. For a completed transition at index i , we define

$$\tau_i = (I_i, q_i, a_{i-d_{\text{max}}:i}^{\text{issued}}, I_{i+1}, q_{i+1}), \quad (8)$$

where $a_{i-d_{\max}:i}^{\text{issued}}$ contains the $d_{\max} + 1$ commands from $i - d_{\max}$ through i .
Let

$$f_i = F_{\text{rgb}}(I_i), \quad \Delta f_i = f_{i+1} - f_i, \quad \Delta q_i = q_{i+1} - q_i. \quad (9)$$

The token for transition i is

$$x_i = \text{MLP}_{\text{trans}} \left(\left[f_i, q_i, \Delta f_i, \Delta q_i, a_{i-d_{\max}:i}^{\text{issued}}, m_i^{\text{rec}}, m_{i-d_{\max}:i}^{\text{act}} \right] \right), \quad (10)$$

where m_i^{rec} indicates whether the transition record is valid and $m_{i-d_{\max}:i}^{\text{act}}$ identifies valid entries in the command window.

A history containing H transition records spans $H + 1$ observations and $H + d_{\max}$ issued commands. With $H = 16$ and $d_{\max} = 4$, the encoder receives information from 17 observations and 20 commands. At the beginning of an episode, unavailable entries are padded with zeros and excluded through the masks. Records are constructed only from completed transitions and do not cross episode boundaries.

4.4. System Context and Policy Conditioning

A four-layer Transformer encoder processes the sequence of transition tokens [24]. A learned [CLS] token is prepended to the sequence, and its projected output defines the 128-dimensional system representation:

$$z_{\text{sys}}(t) = E_{\phi}(x_{t-H:t-1}, m_{t-H:t-1}^{\text{rec}}) \in \mathbb{R}^{128}. \quad (11)$$

When no completed transition is available, the [CLS] token provides a learned initial representation. The remaining Transformer dimensions are reported in Appendix A.

The current observation is encoded by the visual and proprioceptive encoders of the Diffusion Policy:

$$h_t^{\text{obs}} = \text{Concat} \left(F_{\text{rgb}}(I_{t-T_0+1:t}), F_q(q_{t-T_0+1:t}) \right). \quad (12)$$

We combine the observation representation with the inferred system representation:

$$c_t = \text{MLP}_{\text{cond}} \left(\left[h_t^{\text{obs}}, z_{\text{sys}}(t) \right] \right). \quad (13)$$

The condition c_t is supplied through the FiLM conditioning interface of each residual block in the 1D U-Net [5,25,26].

4.5. Auxiliary Training Objectives

4.5.1. System Parameter Prediction

During simulation training, the camera and control parameters are available as supervision. We define the normalized target

$$\tilde{\xi} = \left[\mathcal{S}_p(\Delta \mathbf{p}_{\text{cam}}), \mathcal{S}_r(\Delta \mathbf{r}_{\text{cam}}), \mathcal{S}_s(s_a), \frac{d_a}{d_{\max}} \right], \quad (14)$$

where $\mathcal{S}_p$, $\mathcal{S}_r$, and $\mathcal{S}_s$ standardize the camera translation, camera rotation, and action scale using the mean and standard deviation of the training configurations. The delay is normalized only by $d_{\max}$.

A prediction head maps the system representation to an estimate of the normalized parameters:

$$\hat{\xi}_{\text{norm}} = P_{\omega}(z_{\text{sys}}(t)). \quad (15)$$

The corresponding loss is

$$\mathcal{L}_{\text{sys}} = \mathcal{L}_{\text{Huber}}(\tilde{\xi}_{\text{norm}}, \tilde{\xi}). \quad (16)$$

Delay is regressed as a continuous scalar rather than treated as a classification target. The normalized parameter predictions are used only for this auxiliary objective; the policy is conditioned directly on $z_{\text{sys}}(t)$.

4.5.2. Response Prediction

Parameter supervision does not directly describe how the current system responds to an issued command. We therefore train a second head to predict the proprioceptive change and visual feature at the next step:

$$(\widehat{\Delta q}_t, \hat{f}_{t+1}) = D_{\eta}(f_t, q_t, a_{t-d_{\text{max}}:t}^{\text{issued}}, m_{t-d_{\text{max}}:t}^{\text{act}}, z_{\text{sys}}(t)). \quad (17)$$

The target for the proprioceptive prediction is

$$\Delta q_t = q_{t+1} - q_t. \quad (18)$$

For the visual target, we maintain an exponential moving average copy of the RGB encoder:

$$\bar{\psi} \leftarrow \mu \bar{\psi} + (1 - \mu) \psi, \quad \mu = 0.99, \quad (19)$$

where ψ and $\bar{\psi}$ denote the parameters of the trainable and target RGB encoders, respectively. The target visual feature is

$$\bar{f}_{t+1} = \bar{F}_{\text{rgb}}(I_{t+1}). \quad (20)$$

The response prediction loss is

$$\mathcal{L}_{\text{dyn}} = \left\| \widehat{\Delta q}_t - (q_{t+1} - q_t) \right\|_1 + \beta_f \left\| \hat{f}_{t+1} - \text{sg}(\bar{f}_{t+1}) \right\|_2^2, \quad \beta_f = 0.1, \quad (21)$$

where $\text{sg}(\cdot)$ stops gradients through the target feature. The target encoder is updated only through the exponential moving average.

4.6. Training and Inference

The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\text{diff}} + \lambda_{\text{sys}} \mathcal{L}_{\text{sys}} + \lambda_{\text{dyn}} \mathcal{L}_{\text{dyn}}, \quad \lambda_{\text{sys}} = \lambda_{\text{dyn}} = 0.1. \quad (22)$$

Training begins from a Diffusion Policy checkpoint trained with the deployment randomization described in Section 5. We first pretrain the history encoder and auxiliary heads and then jointly optimize the history encoder, observation encoders, conditioning module, and diffusion network.

After the initial joint training stage, we continue training with the fixed recovery dataset described in Section 5.3, sampling expert and recovery data at a ratio of 3:1.

At the start of each episode, the history buffer is empty. At a replanning step, the completed transition records are encoded into $z_{\text{sys}}(t)$, which is combined with the current observation through Equation (13). The diffusion model then generates an action sequence, and the first T_a commands are sent to the controller. Their observed responses are added to the history buffer before the next replanning step. Model parameters remain fixed during deployment.

Full optimization settings and training schedules are provided in Appendix A.

5. Benchmark and Experimental Protocol

5.1. Tasks and Observations

We evaluate five tabletop manipulation tasks from ManiSkill [27]: PickCube-v1, PushCube-v1, StackCube-v1, PullCube-v1, and PegInsertionSide-v1. Together, they cover grasping, planar pushing, multi-stage placement, reverse pushing, and precision insertion. A separate policy is trained for each task using a Franka Panda arm with a parallel gripper.

Each policy receives a 128×128 monocular RGB image from a wrist-mounted eye-in-hand camera and the robot proprioception available during evaluation. Ground-truth object poses are not included. The main protocol includes the previous issued action in proprioception. Because this input duplicates command information available to the history encoder, we also retrain and evaluate SCE-DP after removing it. For a given task, all methods in the main comparison use the same controller, action space, observation inputs, and episode horizon.

The primary behaviors and evaluation focus of the five tasks are summarized in Table 1.

Table 1. Evaluation tasks. Each task is trained independently and receives equal weight in the macro-average success rate.

Task	Primary Behavior	Evaluation Focus
PickCube-v1	Grasp and lift	Visuomotor alignment
PushCube-v1	Planar pushing	Sustained contact
StackCube-v1	Grasp, transport, and place	Multi-stage manipulation
PullCube-v1	Reverse pushing	Direction-sensitive control
PegInsertionSide-v1	Precision insertion	Spatial and timing accuracy

The simulator, controller, camera, timing, and action-space settings are reported in Appendix A.

5.2. Deployment Settings

At the beginning of each episode, a wrapper sets the camera translation, camera rotation, arm-command scale, and control delay. These values remain fixed until the episode ends. Camera offsets are applied to the mounted camera at reset, and the delay queue is initialized with zero commands. The wrapper applies Equation (4) before sending commands to the controller.

Desired, issued, and applied commands are stored in the data-generation logs. The policy and history encoder use only issued commands. We also verify that the workspace remains visible under every camera setting.

Camera translation is defined in the nominal camera coordinate frame: $+x$ points to the right in the image, $+y$ points downward, and $+z$ points forward from the camera. Rotation is represented by extrinsic roll–pitch–yaw offsets in xyz order.

The training/validation and mild-extrapolation ranges are summarized in Table 2.

Table 2. Ranges used to construct the training, validation, and extrapolation configurations.

Factor	Training/Validation Range	Mild Extrapolation Range
Camera translation	Each axis in $[-5, 5]$ cm	Selected-axis magnitude in $(5, 10]$ cm
Camera rotation	Each axis in $[-10^\circ, 10^\circ]$	Selected-axis magnitude in $(10^\circ, 15^\circ]$
Arm-command scale	$[0.8, 1.2]$	$[0.7, 0.8) \cup (1.2, 1.3]$
Control delay	0–2 steps	3–4 steps

For extrapolation settings, selected axes are assigned a sign sampled uniformly from $\{-1, +1\}$ and an absolute magnitude sampled from the corresponding extrapolation interval. The remaining axes stay within the training range. All configurations are sampled before training and stored in a fixed catalog. The training, validation, held-out, composition, and extrapolation splits use non-overlapping configuration identifiers.

5.3. Demonstrations and Recovery Data

Demonstrations are generated by an expert with access to the camera and control configuration. To compensate for arm-command scaling, the desired command is converted as

$$a_t^{\text{scale}} = \text{clip}\left(\frac{a_t^{\text{des}}}{s_a}, -1, 1\right). \quad (23)$$

The command sequence is advanced by d_a steps to account for the control delay. Gripper commands are advanced by the same amount but are not divided by s_a .

Each converted trajectory is replayed in the corresponding environment. Initial no-op commands, terminal holds, and delay-queue flushing ensure that the complete delayed sequence is executed. A trajectory is added to the expert dataset only if it satisfies the task success condition after the queue has been flushed.

For each training configuration, we target ten successful demonstrations and require at least eight. The data manifest records the number of attempts, accepted trajectories, command-clipping ratio, configuration identifier, and random seeds. Configurations with an acceptance rate below 50% or an arm-command clipping ratio above 10% are inspected and regenerated when necessary.

After the initial joint-training stage, warm-start DP + DR and SCE-DP policies separately collect trajectories under the training configurations. The configuration-aware expert labels recoverable visited states with future command sequences. Data collected by both policies are merged into a fixed recovery dataset, capped at 25% of the number of expert transitions, and shared by DP + DR, DP + History, SysID + Params, Oracle, and SCE-DP. States without a reliable action label may be used for $\mathcal{L}_{\text{sys}}$ and $\mathcal{L}_{\text{dyn}}$, but not for $\mathcal{L}_{\text{diff}}$.

Because this construction could favor SCE-DP through exposure to states visited by an earlier SCE-DP collector, we additionally audit the source composition of the recovery set and train recovery-source controls. The merged action-labeled recovery set is nearly balanced between the two collectors, as shown in Table 3. We further evaluate SCE-DP trained with only the DP + DR-collected recovery subset and report this control in Section 6. It tests whether the final improvement is caused by access to SCE-DP's own visited-state distribution.

Table 3. Source composition of the recovery dataset used by the main methods. Counts denote action-labeled recovery transitions; states without action labels are counted only in the auxiliary-only column.

Task	DP + DR Collector	SCE-DP Collector	Auxiliary Only	SCE Share
PickCube	4105	4117	2056	50.1%
PushCube	4728	4733	2365	50.0%
StackCube	5927	5954	2970	50.1%
PullCube	5050	5054	2526	50.0%
PegInsertionSide	6638	6659	3330	50.1%

Dataset sizes, acceptance rates, clipping statistics, and recovery-data counts are reported in Appendix A.

5.4. Configuration Splits

All configurations are generated before training and assigned unique identifiers. The training and validation splits contain camera-only, scale-only, delay-only, camera + scale, and scale + delay configurations. Camera + delay and camera + scale + delay combinations are omitted from both splits.

The composition suite contains in-range camera + delay configurations with unit action scale. It therefore tests a combination of factors that is absent from training and validation. A separate three-factor suite evaluates simultaneous changes in camera pose, action scale, and control delay as an additional stress test.

The configuration suites used for training, model selection, and evaluation are summarized in Table 4.

Table 4. Configuration suites used for training, model selection, and evaluation.

Suite	No.	Purpose
Nominal	1	Reference configuration
Train	60	Expert data collection and recovery-data generation
Validation	12	Hyperparameter and checkpoint selection (4 camera, 4 scale, 2 delay, 1 camera + scale, 1 scale + delay)
Held-out in-range	8	New configurations from factor combinations seen during training
Composition	8	Camera + delay configurations omitted from training and validation
Extrapolation, single	8	Mild extrapolation of one factor
Extrapolation, combined	8	Mild extrapolation of two factors
Three-factor stress	8	Simultaneous camera, scale, and delay changes

Configuration identifiers, rollout seeds, object initializations, and trajectory identifiers are kept disjoint where required by the split.

5.5. Baselines and Ablations

We compare SCE-DP with the following methods:

DP-Nominal is trained only under the nominal configuration. **DP + DR** is trained using the same randomized camera and control configurations as SCE-DP. **DP + History** adds the same history encoder as SCE-DP but omits $\mathcal{L}_{\text{sys}}$ and $\mathcal{L}_{\text{dyn}}$.

SysID + Params uses the same history encoder and parameter targets as SCE-DP. It predicts the normalized configuration and supplies only these predicted parameters to the policy. **SysID + Params + Dyn** uses the same parameter and response-prediction supervision as SCE-DP, but the policy still receives only the predicted explicit parameters. This matched control isolates the effect of the latent context from the effect of the auxiliary response-prediction loss. **Oracle** uses the same conditioning interface but receives the true normalized configuration. The comparison between SysID + Params + Dyn and SCE-DP tests whether the latent representation contains behaviorally useful information beyond the explicit camera, scale, and delay parameters.

The **target fine-tuning** reference receives ten demonstrations from the target configuration and is optimized for 5000 updates with a learning rate of 10^{-5} . Only the observation-conditioning MLP and Conditional 1D U-Net are updated. Its demonstrations and evaluation seeds are disjoint.

We ablate the auxiliary objectives, history length, context assignment, context updating, temporal encoding, and command–response alignment.

DP + DR, DP + History, SysID + Params, SysID + Params + Dyn, Oracle, and SCE-DP use the same demonstrations, merged recovery dataset, number of training updates, diffusion scheduler, and checkpoint-selection procedure.

Published viewpoint-robust manipulation methods are not included as direct baselines because their observation, calibration, action, or backbone requirements differ from the single-RGB, unknown-configuration setting evaluated here. Table 5 states the mismatch for each cited method rather than treating prior work as one homogeneous category. These methods remain complementary to SCE-DP, especially when calibration, depth, view synthesis, or a pretrained VLA backbone is available.

Table 5. Input-assumption mismatch for published viewpoint-robust methods. The table explains why these methods are discussed but not used as direct baselines under our single-RGB, unknown-camera, action-scale, and delay setting.

Published Method	Calibration/Geometry	Visual Input	Control-Shift Model	Reason Not Directly Comparable
Calibration Diffusion Policy [6]	Camera calibration parameters supplied to the policy	RGB plus calibration history	No	Assumes calibration input that is deliberately unavailable in our unknown-camera protocol
3D Diffuser Actor [7]	3D scene coordinates	RGB-D from one or more views	No	Requires sensed depth and predicts 3D pose trajectories rather than our normalized command interface
AnyCamVLA [8]	Target-view normalization through novel-view synthesis	RGB and a pretrained VLA policy	No	Adapts a VLA by synthesizing its training view; it does not model action-scale or command latency
VistaBot [9]	4D geometry estimation and view synthesis	RGB/video with ACT or π_0	No	Adds geometric/video generative modules and a different policy backbone, without delayed or rescaled execution
CamVLA [10]	Predicts a 6-DoF hand-eye transform	Single RGB plus language	No	Uses a VLA and camera-frame action representation; it does not address delayed command execution or action scaling

5.6. Metrics and Statistical Analysis

Task success rate is the primary metric. Secondary measures include task progress, degradation from nominal performance, recovery of the gap to Oracle, parameter-prediction error, delay Spearman correlation, response-prediction error, context stability, replanning latency, peak GPU memory, and parameter count.

Each method is trained with three random seeds. Each checkpoint is evaluated using exactly 50 matched rollout seeds for each test configuration. We report mean success rates with 95% bootstrap confidence intervals and use the same rollout seeds for paired method comparisons.

All confidence intervals are computed from 10,000 hierarchical bootstrap resamples using percentile intervals. For suite-level results, the resampling hierarchy proceeds from task to training seed, configuration, and matched rollout seed. Comparisons between SCE-DP and DP + DR, SysID + Params, or SysID + Params + Dyn use the same resampled indices, and the statistic is their paired difference in success rate. Training-seed labels define aligned experimental blocks across methods because the corresponding runs use the same data split and evaluation conditions; independently trained policies remain the principal independent units in the seed-level sensitivity analysis. Let $\hat{\Delta}$ denote the observed paired difference, and let Δ_b^* denote the paired difference obtained in bootstrap replicate b . To approximate the null hypothesis of zero difference, we center each bootstrap replicate as

$$\Delta_b^{*(0)} = \Delta_b^* - \hat{\Delta}. \quad (24)$$

With $B = 10,000$ resamples, the two-sided bootstrap p value is

$$p = \frac{1 + \sum_{b=1}^B \mathbf{1}\left(\left|\Delta_b^{*(0)}\right| \geq \left|\hat{\Delta}\right|\right)}{B + 1}, \quad (25)$$

where $\mathbf{1}(\cdot)$ is the indicator function. Holm correction is applied across the four shifted-suite comparisons within each baseline family. Line plots show 95% confidence bands computed with the same resampling procedure.

For each method, task, and suite, success is first averaged over rollout seeds within each configuration and then over configurations. The result is subsequently averaged over the three training seeds. The overall score is the unweighted mean across the five tasks.

To summarize performance across the four shifted suites, we define

$$S_{\text{AvgShift}} = \frac{1}{4} \left(S_{\text{held}} + S_{\text{comp}} + S_{\text{extra-single}} + S_{\text{extra-combined}} \right). \quad (26)$$

Each suite therefore contributes equally to S_{AvgShift} , regardless of its number of configurations.

Before the main evaluation, we set the following success criteria: SCE-DP must improve the held-out macro-average by at least eight percentage points over DP + DR, improve at least four of the five tasks, remain within three points of DP-Nominal on the nominal suite, achieve a statistically significant improvement on the composition suite, and add less than 20% to the total replanning latency.

5.7. Hardware and Efficiency Measurement

Training and evaluation use one NVIDIA L40 GPU with 48 GB memory, an x86_64 server CPU with 32 logical cores, and 128 GB RAM. The software environment consists of Ubuntu 22.04, Python 3.10, PyTorch 2.4.1, CUDA 12.1, cuDNN 9.1, and ManiSkill v3.0.1. Each training run uses one GPU.

Inference latency is measured with batch size 1, FP16 autocast for neural-network inference, and FP32 environment state. We perform 100 warm-up replanning calls followed by 1000 measured calls, with CUDA synchronization immediately before and after each timed region.

The reported neural-policy latency includes RGB and proprioceptive encoding, record-token construction, Transformer encoding, condition fusion, and diffusion denoising. It excludes environment simulation, rendering, host-to-device image transfer, logging, and video encoding. Peak GPU memory is measured after warm-up using PyTorch's peak-memory counter.

Training time, memory use, and data-generation cost are reported in Appendix A. Section 6 reports the additional deployment cost introduced by the history encoder.

6. Results

6.1. Validation and Diagnostic Checks

The validation sweep over $H \in \{1, 4, 8, 16\}$ selects $H = 16$, which gives the highest validation Avg Shift while satisfying the latency constraint. Full results are provided in Appendix B. On the validation set, the parameter prediction errors are 0.92 cm for camera translation, 1.9° for camera rotation, 0.031 for action scale, and 0.34 steps for control delay. The Spearman correlation for delay is 0.89. The response-prediction error reported here is the dimensionless visual-feature mean-squared error after encoder-feature normalization; it is not the full mixed loss in Equation (21). Replacing the inferred context with one from another configuration of the same task increases this error from 0.034 to 0.081 and reduces policy success from 73.2% to 54.7%.

The full-history estimates on the held-out catalog in Table A10 are slightly lower than the corresponding validation errors (for example, 0.84 versus 0.92 cm for camera translation). This does not indicate training on the held-out configurations: validation and held-out use disjoint fixed catalogs drawn from the same in-range bounds, and the catalogs are not ordered by identification difficulty. Their finite configuration mix and task macro-averaging can therefore reverse the expected numerical ordering. We interpret the small difference as catalog-level variation and base the generalization claim on policy success over the separately frozen held-out suite, not on a presumption that held-out parameter error must exceed validation error.

6.2. Main Results

On the frozen test suites, DP-Nominal achieves 86.5% under the nominal configuration. Its success rate falls to 63.1% ($n = 3000$), 68.4% ($n = 3000$), and 58.6% ($n = 1500$) on the camera-only, action-scale-only, and control-delay-only test subsets, respectively. For Pick, Push, Stack, Pull, and Peg, the corresponding task scores are 72.5%, 68.0%, 64.0%, 65.0%, and 46.0% under camera shifts; 78.0%, 74.0%, 68.0%, 73.0%, and 49.0% under action scaling; and 70.0%, 65.0%, 58.0%, 61.0%, and 39.0% under control delay. Camera and delay shifts reduce success substantially across all five tasks, whereas action scaling is less damaging but still degrades precision-heavy performance.

The macro-average success rates on the nominal and four shifted suites are summarized in Table 6.

Table 6. Macro-average success rate across five tasks (% , higher is better). The upward arrow (↑) indicates that higher values are better. Values are reported as mean with 95% hierarchical bootstrap confidence interval. For each shifted suite, $n = 6000$ rollout outcomes, corresponding to 5 tasks $\times$ 3 training seeds $\times$ 8 configurations $\times$ 50 matched rollout seeds. Avg Shift is defined in Equation (26). Bold indicates the best method among those that use neither oracle parameters nor target demonstrations.

Method	Nominal ↑ ($n = 750$)	Held-Out ↑ ($n = 6000$)	Comp. ↑ ($n = 6000$)	Extra-Single ↑ ($n = 6000$)	Extra-Combined ↑ ($n = 6000$)	Avg Shift ↑ ($n = 24,000$)
DP-Nominal	86.5 [77.2, 93.2]	46.3 [36.8, 54.1]	37.0 [27.6, 44.3]	34.7 [25.2, 42.6]	27.7 [21.9, 33.0]	36.4 [27.9, 43.5]
DP + DR	85.7 [76.3, 92.5]	60.0 [50.5, 68.8]	50.1 [41.3, 58.0]	47.7 [40.3, 54.3]	37.7 [31.0, 44.0]	48.9 [40.7, 56.3]
DP + History	86.8 [77.5, 93.5]	59.0 [48.8, 68.2]	48.4 [39.4, 55.6]	46.4 [38.4, 53.1]	37.5 [31.3, 43.4]	47.8 [39.5, 55.1]
SysID + Params	86.1 [76.5, 92.8]	70.1 [62.1, 77.0]	60.6 [52.4, 67.4]	56.2 [48.0, 63.1]	50.0 [41.6, 57.1]	59.2 [51.0, 66.1]
SysID + Params + Dyn	87.3 [75.3, 95.3]	72.5 [65.0, 78.6]	63.9 [56.6, 70.0]	58.9 [50.7, 65.5]	53.0 [44.7, 60.2]	62.1 [54.6, 68.2]
Oracle	89.5 [80.1, 95.9]	78.4 [70.3, 85.3]	69.7 [61.6, 76.2]	66.9 [58.2, 74.1]	57.0 [46.7, 65.1]	68.0 [59.2, 75.2]
Target fine-tune	87.6 [78.1, 94.3]	68.9 [61.2, 75.2]	60.5 [52.6, 67.4]	57.2 [46.5, 65.1]	49.1 [42.2, 55.0]	58.9 [50.6, 65.7]
SCE-DP	86.0 [76.7, 92.7]	74.5 [66.2, 81.5]	65.8 [56.6, 73.7]	61.0 [52.5, 68.0]	55.7 [47.7, 62.5]	64.3 [55.8, 71.4]

On the Held-out and Composition test suites, Oracle conditioning reaches 78.4% and 69.7%, respectively, compared with 46.3% and 37.0% for DP-Nominal. Relative to the nominal test score of 86.5%, Oracle conditioning recovers 79.9% of the performance gap on Held-out and 66.1% on Composition.

SCE-DP improves over DP + DR by 14.5 points on Held-out, 15.7 points on Composition, 13.3 points on single-factor extrapolation, and 18.0 points on combined extrapolation. Its Avg Shift is 64.3%, compared with 48.9% for DP + DR and 47.8% for DP + History. SysID + Params reaches 59.2%, while the matched SysID + Params + Dyn control reaches 62.1%. SCE-DP exceeds this matched control by 2.2 Avg Shift points, but the suite-level comparisons are not significant after Holm correction. On the nominal suite, SCE-DP scores 86.0%, compared with 86.5% for DP-Nominal.

Paired hierarchical bootstrap tests show significant improvements over DP + DR and SysID + Params on all four shifted suites after Holm correction. SCE-DP also has higher

mean success than SysID + Params + Dyn on every suite, but none of these four matched-control comparisons reaches significance after correction. We therefore do not claim statistically significant superiority over the matched explicit-parameter control. Confidence intervals and corrected p values are reported in Appendix C.

On the additional three-factor stress suite, SCE-DP achieves 57.4% success, compared with 41.0% for DP + DR. The full results are provided in Appendix C.

Figure 2 visualizes the macro-average performance differences across the four shifted suites.

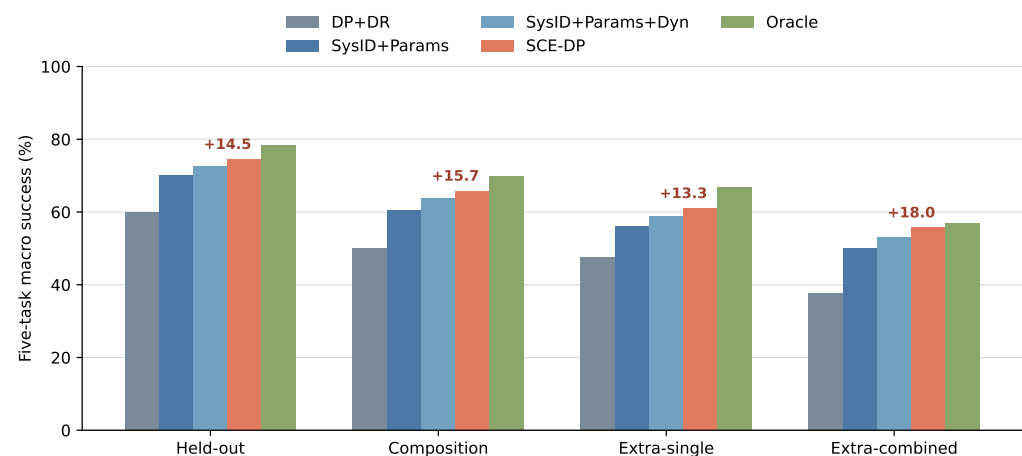


Figure 2. Macro-average success rates on the four shifted suites. Annotations report the improvement of SCE-DP over DP + DR in percentage points.

SCE-DP also outperforms DP + DR on every task in the Held-out and Composition suites. The Held-out gains range from 9.0 points on PickCube to 16.4 points on PegInsertion-Side. On Composition, they range from 15.1 points on PushCube to 16.8 points on PullCube.

6.3. Ablations and Diagnostics

These ablations modify only the fields supplied to the history-token encoder; the current policy-observation pathway, auxiliary-head targets, and all other model components remain unchanged. To make the information-source ablations in Table 7 unambiguous, all variants use the same mask-construction rule, history length $H = 16$, and $d_{\max} = 4$. For removed fields, the corresponding field masks are set to zero, and no values are imputed or replaced by learned quantities. The full command–response variant retains $(f_i, q_i, \Delta f_i, \Delta q_i)$ together with the recent issued-command window $a_{i-d_{\max}:i}^{\text{issued}}$. Command-only retains only that issued-command window and masks the four visual/proprioceptive fields; response-only retains the four visual/proprioceptive fields and masks the issued-command window; and response-difference retains only $(\Delta f_i, \Delta q_i)$, masking the absolute visual/proprioceptive fields and the issued-command window. Each variant was fully retrained independently under the same demonstrations, merged recovery data, DP + DR initialization protocol, model architecture, data augmentation, feature normalization, optimizer, loss weights, batch size, number of updates, update schedule, checkpoint-selection rule, and three training seeds; no variant reuses weights from another. The system-probe columns below are mean absolute errors from each variant’s learned auxiliary system-parameter regression head, evaluated on Held-out configurations (camera translation in cm, camera rotation in degrees, action scale, and delay in steps). They are not the post hoc linear-probe scores reported in Table 8.

Table 7. Shortcut and information-source ablations for the history encoder. Values are macro-average success rates (%; higher is better) and auxiliary-head system-parameter errors on Held-out configurations. The four input variants and the retained/removed fields are defined in the preceding paragraph; these errors are not post hoc linear-probe results. Bold indicates the highest success rate or the lowest parameter-prediction error among the compared variants.

History Input	Held-Out	Comp.	Cam. cm	Rot. deg	Scale	Delay
Command-only	63.0 [54.1, 70.3]	52.8 [43.4, 60.6]	2.44	5.3	0.071	0.84
Response-only	66.2 [58.3, 73.0]	57.7 [49.3, 64.8]	1.52	3.4	0.049	0.58
Response-difference	68.9 [59.9, 77.0]	59.0 [50.2, 66.6]	1.37	3.1	0.044	0.50
Full command–response	74.5 [66.2, 81.5]	65.8 [56.6, 73.7]	0.87	1.9	0.030	0.32

The task-state and context-swap diagnostics are summarized in Table 8.

Table 8. Task-state and context-swap diagnostics on the Held-out suite. Matched swaps exchange contexts only between rollouts with the same task. Among eligible source rollouts from a different configuration, we select the nearest neighbor in normalized progress and the ℓ_2 distance between the recent action-history vectors; the strongest match also adds nearest-neighbor matching in normalized initial-state variables. Probe mean score is the average of R^2 for continuous targets, computed after target normalization, and balanced accuracy for binary contact.

Diagnostic	Policy Success	Response MSE	Interpretation
Unmatched different-configuration context	55.1%	0.081	Task state and configuration both differ
Matched progress and action history	58.7%	0.073	Matching task phase recovers part of the loss
Matched progress, initial state, and action history	60.1%	0.069	Stronger task-state matching remains insufficient
Correct inferred context	74.5%	0.034	Best response prediction and control
System-parameter probe, mean score		0.82	Camera, scale, and delay are strongly decodable from z_{sys}
Task-state probe, mean score		0.40	Task variables are also partially decodable, so the latent is not a pure system identifier

Command-only histories remain above DP + DR on some suites, indicating that configuration-aware expert conversion leaves weak statistical traces in the issued-command distribution. However, they remain 11.5 points below the full command–response model on Held-out and 13.0 points below it on Composition. Response-only and response-difference variants recover part of the gap. All histories in this control are generated by the deployed policy rather than copied from expert trajectories. These results support the narrower conclusion that command statistics alone are insufficient and that the strongest performance requires observed responses.

For the latent probes in Table 8, we freeze the trained SCE-DP encoder and fit lightweight linear probes from z_{sys} to held-out labels. The system-parameter probe predicts camera translation, camera rotation, action scale, and control delay. The task-state probe predicts end-effector pose, object position, normalized task progress, and binary contact state. Probe training uses the validation configurations, and the reported score is evaluated on the Held-out suite. For continuous targets, the score is the coefficient of determination R^2 after target normalization; for binary contact, it is balanced accuracy. The mean score averages these target-level scores over the five tasks and three training seeds.

Table 9 reports the controls for the previous issued action and recovery-source composition.

Table 9. Controls for previous issued action in proprioception and recovery-source composition. Values are macro-average success rates (% , higher is better) on the SCE-DP architecture. Bold indicates the main SCE-DP configuration used in the primary comparison.

Control	Held-Out	Comp.	Interpretation
Main SCE-DP, merged recovery	74.5 [66.2, 81.5]	65.8 [56.6, 73.7]	Includes previous issued action
Without previous action	72.4 [63.9, 80.3]	63.3 [54.9, 71.2]	Removes redundant command input
DP + DR-recovery only	70.9 [61.5, 78.8]	61.4 [52.6, 69.0]	Removes SCE-DP-collected states

Removing the previous issued action reduces SCE-DP by 2.1 points on Held-out and 2.5 points on Composition, while it remains 12.4 and 13.2 points above DP + DR. Training with only DP + DR-collected recovery data lowers success by 3.6 and 4.4 points, respectively, but retains gains of 10.9 and 11.3 points over DP + DR. The main improvement therefore does not depend on either redundant command input or recovery states collected by SCE-DP.

The auxiliary-objective and context-intervention ablations are reported in Table 10.

Table 10. Ablations on the Held-out and Composition suites. Values are macro-average success rates across five tasks, reported as mean with 95% hierarchical bootstrap confidence interval ($n = 6000$ roll-out outcomes per suite). All variants follow the same checkpoint-selection procedure and use matched rollout seeds.

Variant	Held-Out	Comp.	Change Tested
SCE-DP	74.5 [66.2, 81.5]	65.8 [56.6, 73.7]	Full model
Without $\mathcal{L}_{\text{sys}}$	70.6 [62.4, 77.7]	61.2 [52.0, 69.0]	Parameter prediction loss
Without $\mathcal{L}_{\text{dyn}}$	68.9 [60.5, 75.9]	59.4 [50.2, 67.1]	Response prediction loss
Shuffled context	57.3 [49.0, 64.3]	46.8 [37.4, 54.7]	Correct context assignment
Random context	56.1 [47.8, 63.1]	45.4 [36.2, 53.2]	Added conditioning input
Context from another configuration	56.0 [47.8, 63.0]	44.9 [35.6, 52.7]	Configuration specificity
Fixed initial context	64.8 [56.2, 71.9]	55.3 [46.2, 63.1]	Context updates during the episode

Removing $\mathcal{L}_{\text{sys}}$ reduces success by 3.9 points on Held-out and 4.6 points on Composition. Removing $\mathcal{L}_{\text{dyn}}$ produces larger drops of 5.6 and 6.4 points. The DP + History result in Table 6 shows that history conditioning alone does not improve over DP + DR; the gain appears when history conditioning is paired with structured parameter and response supervision. The variant without $\mathcal{L}_{\text{sys}}$ remains competitive, which indicates that privileged simulated parameter labels are useful but not strictly necessary for a deployable latent-context policy.

The three context perturbations reduce Held-out success by 17.2–18.5 points and Composition success by 19.0–20.9 points. Keeping the initial context fixed throughout the episode reduces success by 9.7 points on Held-out and 10.5 points on Composition.

Table 11 compares alternative history representations on suites with nonzero control delay.

Fixed one-step pairing reduces Composition success by 11.9 points relative to the proposed representation, while using only the latest command reduces it by 9.0 points. Predicting the delay and selecting one command performs better but remains 4.5 points below the recent-command window. Retaining multiple candidate commands is therefore more effective than committing to a single estimated alignment.

The delay-5 stress test beyond the represented command window is reported in Table 12.

Table 11. Ablation of the history representation on suites containing nonzero control delay. Values are macro-average success rates across five tasks, reported as mean with 95% hierarchical bootstrap confidence interval ($n = 6000$ rollout outcomes per suite). Bold indicates the best deployable history representation that does not use the true control delay; the final row uses the true delay during history construction and serves as an oracle-alignment upper bound.

History Representation	Comp.	Extra-Combined	Description
Fixed one-step pairing	53.9 [44.4, 61.8]	42.8 [35.0, 49.6]	Pairs each response with a_i
Most recent command only	56.8 [47.4, 64.7]	45.7 [38.0, 52.5]	Uses one command without representing delay uncertainty
No temporal encoding	58.2 [48.9, 66.0]	47.1 [39.4, 53.9]	Removes the timing information in the record sequence
Predict delay and align	61.3 [52.1, 69.1]	51.0 [43.2, 57.7]	Selects one command using the predicted delay
Recent-command window	65.8 [56.6, 73.7]	55.7 [47.7, 62.5]	Retains all candidate commands for the Transformer
Alignment with true delay	67.1 [58.2, 75.1]	57.5 [49.8, 64.4]	Uses the true value of d_a

Table 12. Delay stress test beyond the represented command window. The model uses $d_{\max} = 4$, whereas this suite fixes the delay to five control steps ($n = 6000$ per method). Brackets report 95% hierarchical-bootstrap confidence intervals.

Suite	DP + DR	SCE-DP	Interpretation
Delay-5, beyond $d_{\max}$	28.9 [21.7, 34.9]	44.3 [36.5, 50.5]	The causal command can fall outside the represented window

The delay-5 result clarifies the scope of the delay extrapolation claim. SCE-DP retains a 15.4-point mean advantage over DP + DR, but its absolute success falls to 44.3% once the causal command is no longer guaranteed to appear in the recent-command window. The main conclusion is therefore limited to unknown delay within the represented range, not unrestricted latency.

6.4. Efficiency and Failure Analysis

DP + DR requires 36.9 ms per neural replanning call, 4.82 GB of peak GPU memory, and 132.4 M parameters. SCE-DP requires 42.9 ms on average, with a standard deviation of 2.4 ms, median latency of 42.9 ms, and p95 latency of 46.9 ms. This corresponds to a 16.1% increase over DP + DR in neural-policy latency. Its peak memory use is 5.21 GB, and it contains 139.1 M parameters. These measurements exclude simulation, rendering, image transfer, communication, logging, and video encoding; they should therefore not be interpreted as end-to-end real-robot control latency.

The inference cost and model size of the main methods are summarized in Table 13.

Table 13. Inference cost and model size for the main baselines. Latency is neural-policy replanning time only and excludes rendering and communication. Oracle and Target fine-tune are included for completeness: Oracle uses the same policy-side parameter-conditioning interface as SysID + Params but receives the true configuration vector, whereas Target fine-tune uses the DP + DR deployment architecture after per-target weight updates.

Method	Mean $\pm$ SD ms	Median ms	p95 ms	Params M
DP + DR	36.9 $\pm$ 2.0	36.9	40.1	132.4
DP + History	41.3 $\pm$ 2.2	41.4	44.7	138.7
SysID + Params	42.1 $\pm$ 2.2	42.1	45.5	139.0
SysID + Params + Dyn	42.2 $\pm$ 2.2	42.2	45.8	139.0
Oracle	Same policy-side conditioning path as SysID + Params			132.7
SCE-DP	42.9 $\pm$ 2.4	42.9	46.9	139.1
Target fine-tune	Same deployment architecture as DP + DR			132.4

Figure 3a shows that most of the performance gain appears after four to eight completed transition records, although $H = 16$ achieves the highest final success rate. Performance is lowest before the policy has observed any completed transition.

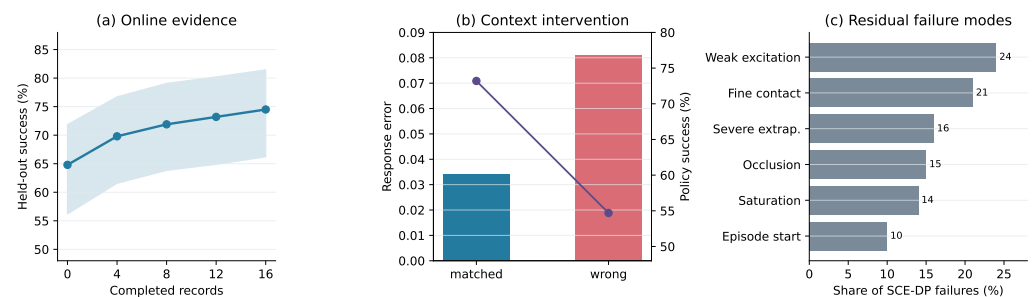


Figure 3. Adaptation and failure analysis. (a) Held-out success as completed transition records accumulate. (b) On the validation suite, response-prediction error and policy success with the inferred context and with a context from another configuration. (c) Distribution of failure categories on the shifted suites.

The failure taxonomy in Figure 3c and Table A12 is based on a stratified sample of 500 failed SCE-DP rollouts from the four shifted suites, balanced as evenly as possible across task and suite. One annotator assigned a single mutually exclusive category to each sample using rollout video, success flags, action-saturation logs, and context-probe errors. The largest categories are weak excitation and sensitivity during fine contact, accounting for 24% and 21% of failed rollouts, respectively. Other failures arise from severe extrapolation, occlusion, action clipping, and insufficient history near the beginning of an episode. Because this is a single-annotator taxonomy, it should be read as a diagnostic summary rather than a statistically adjudicated human-labeling study.

A paired qualitative rollout under the same Composition-suite initial condition is shown in Figure 4.

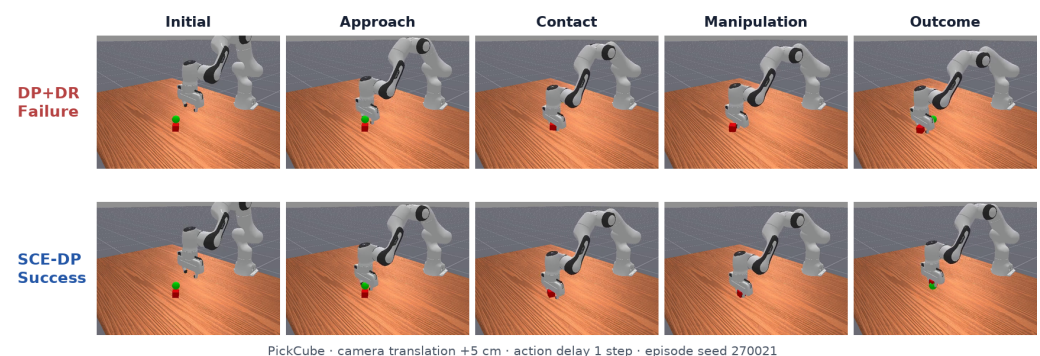


Figure 4. Paired PickCube rollout from the Composition suite (test-composition-000, +5 cm camera translation, one-step control delay, episode seed 270021). This illustrative pair was selected after aggregate analysis using the qualitative criterion that both methods share the same initial condition, DP + DR fails to complete the lift, and SCE-DP reaches the success condition. It was not used for model selection, statistical testing, or quantitative aggregation.

7. Discussion

The context intervention provides direct evidence that the history representation affects both response prediction and action generation. On the validation suite, replacing the inferred context with one from another configuration increases the normalized visual-feature response-prediction error from 0.034 to 0.081 and reduces policy success from 73.2% to 54.7%. On the Held-out suite, an unmatched context swap reduces success from 74.5% to 55.1%;

matching task, progress, initial state, and action history recovers part of the loss but reaches only 60.1%. Random and shuffled contexts produce similarly large drops, while removing either auxiliary objective leads to smaller but consistent losses. The command-only and no-previous-action controls show that issued commands carry some configuration information, but neither command statistics nor redundant command input explains the full gain.

The matched controls also clarify the role of the latent representation. The dynamics-supervised explicit-parameter control uses the same parameter and response-prediction supervision as SCE-DP but exposes only predicted explicit parameters to the policy. Its higher performance than SysID + Params indicates that response supervision improves the explicit-parameter baseline. SCE-DP has a further 2.2-point mean Avg Shift advantage, although none of the four suite-level differences is significant after Holm correction. The latent may retain image-space sensitivity under the current viewpoint, contact-conditioned effective response, saturation effects after command scaling, or uncertainty over delay alignment, but the present evidence does not isolate these components. Task-state probes also show that z_{sys} is not a pure system identifier. We therefore interpret it as a deployment-relevant behavioral context rather than a disentangled estimate of only physical system parameters.

Geometry-based methods handle viewpoint changes through calibration, 3D representations, or view normalization. SCE-DP addresses a different source of variation by modeling how issued commands appear in subsequent visual and proprioceptive observations. This allows the same representation to capture camera pose, action scale, and bounded control delay, which is particularly useful when visual and control changes occur together. Combining interaction history with explicit geometric representations is a natural direction for future work, especially for static external cameras where command-induced image motion and observability differ from the wrist-mounted setting studied here.

Calibration-dependent robotic control also appears outside camera-viewpoint manipulation. For example, eye-tracking control of a collaborative robot relies on an explicit calibration step that maps measured gaze to control targets [28]. Such systems illustrate the value of calibration when a reliable procedure and reference targets are available. Our simulated setting instead asks whether recent command–response evidence can provide a useful behavioral context when the deployment camera and control parameters are not supplied; it should not be read as evidence that learned context universally replaces safety-critical geometric or human–interface calibration.

8. Limitations

The experiments are conducted entirely in ManiSkill and do not establish transfer to physical robots. The reported latency is neural-policy latency, not end-to-end robot-system timing. Camera and control settings remain fixed within each episode, so the current method does not address gradual camera motion or time-varying latency. The main delay experiments assume a shared integer delay for arm and gripper channels and an unknown delay bounded by the predefined represented range $d_{\text{max}} = 4$. The delay-5 stress test shows clear degradation beyond this window. Stochastic delay and channel-heterogeneous delay are not evaluated and remain important extensions.

The evaluation covers camera extrinsics, action scale, and control delay for a wrist-mounted eye-in-hand camera, but not static external cameras, changes in camera intrinsics, lighting, appearance, mass, friction, state estimation, or robot embodiment. The system-parameter loss uses privileged simulated camera and control labels during training; deployment does not require these labels, but collecting equivalent supervision on physical systems may be costly. The main composition suite focuses on camera and delay, and broader combinations of deployment factors have not been evaluated systematically. Finally, each task uses an independently trained policy, leaving multi-task and cross-robot adaptation open.

9. Conclusions

We introduced SCE-DP, which uses recent commands and their visual and proprioceptive responses to infer a latent representation of the current simulated camera and control configuration. This representation conditions a Diffusion Policy and is updated during execution without changing the model parameters.

Across five ManiSkill tasks, SCE-DP achieves an Avg Shift success rate of 64.3%, compared with 48.9% for DP + DR, while preserving nominal performance. Shortcut, previous-action, recovery-source, matched explicit-parameter, and delay-5 controls narrow the interpretation of this gain: command–response history provides useful behavioral context, but superiority over the matched SysID + Params + Dyn control is not statistically established. The current evidence is limited to simulation with fixed within-episode configurations and bounded represented delay; physical-robot generalization remains an important next step.

Author Contributions: Conceptualization, X.X., G.Z. and H.L.; methodology, X.X. and G.Z.; software, X.X. and B.X.; validation, X.X., G.Z. and B.X.; formal analysis, X.X. and G.Z.; investigation, X.X., G.Z. and B.X.; resources, H.L.; data curation, X.X., G.Z. and B.X.; writing—original draft preparation, X.X., G.Z. and B.X.; writing—review and editing, X.X., G.Z. and H.L.; visualization, X.X. and G.Z.; supervision, H.L.; project administration, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DP	Diffusion Policy
DR	Domain Randomization
SCE	System-Context Encoder
SCE-DP	System-Context-Conditioned Diffusion Policy
UP-OSI	Universal Policy with Online System Identification
RMA	Rapid Motor Adaptation

Appendix A. Protocol and Implementation Details

Table A1. Frozen model and training configuration used for the main SCE-DP results. These values are selected on the validation suite only and are not retuned on any test suite.

Item	Setting
Visual encoder	ResNet-18 RGB backbone initialized from ImageNet weights; input 128×128 ; two observation frames; 512-D frame feature projected to 256-D per frame
Data augmentation	Random resized crop in $[0.90, 1.00]$, color jitter 0.10, brightness/contrast 0.10, random translation up to 4 pixels; the same spatial transform and color transform are applied to all frames in an observation/history window; disabled for evaluation
Proprioception encoder	Two-layer MLP, hidden size 256, SiLU activation; output 128-D
Diffusion network	Conditional 1D U-Net with channel widths (256, 512, 1024), kernel size 5, GroupNorm, FiLM global conditioning, sinusoidal diffusion-step embedding

Table A1. *Cont.*

Item	Setting
Diffusion process	DDPM training with 100 noise steps, squared-cosine beta schedule; DDIM sampler [16] with 16 denoising steps at evaluation
Horizons	Observation $T_o = 2$, prediction $T_p = 16$, execution $T_a = 4$, main system history $H = 16$, maximum represented delay $d_{\max} = 4$
System-context encoder	Four-layer Transformer, four heads, hidden size 256, feed-forward size 1024, dropout 0.1, learned [CLS], 128-D system latent
Optimizer and learning rate	AdamW; U-Net and observation encoders 1.0×10^{-4} ; SCE, parameter head, and response head 3.0×10^{-4} ; conditioning MLP 1.0×10^{-4}
Regularization and scheduler	Weight decay 1.0×10^{-6} ; 2000-step linear warm-up; cosine decay to 1.0×10^{-6} ; global gradient clipping at 1.0
Training stages	DP + DR pretraining 100 k updates; SCE context pretraining 20 k context-only updates; joint warm start 30 k; recovery aggregation one frozen round; final joint training 80 k; 210 k policy updates in total
Loss and feature normalization	System parameters are standardized using training-configuration statistics; RGB features are ℓ_2 -normalized before feature-difference and response-prediction losses; $\lambda_{\text{sys}} = \lambda_{\text{dyn}} = 0.1$ is selected on validation Avg Shift
Freezing policy	RGB backbone frozen during context pretraining; U-Net frozen during first 10 k joint updates; all online encoders unfrozen afterwards; EMA target RGB encoder with $\mu = 0.99$
Batching and precision	Batch size 256 expert/recovery sequences; recovery fraction 25% of diffusion minibatch; automatic mixed precision FP16 with FP32 loss accumulation
Checkpoint and seeds	Three training seeds {1, 2, 3}; checkpoint selected by validation Avg Shift; no test rollout is used for checkpoint selection

Table A2. Robot, controller, camera, and timing configuration. These simulator settings are fixed across all methods. The delay wrapper operates at the policy-control rate, so one control-delay step corresponds to 50 ms.

Item	Setting
Robot	Franka Emika Panda, 7-DoF arm with two-finger parallel gripper
Controller	End-effector delta pose/position controller in the robot base frame; arm commands are normalized to $[-1, 1]$ before controller scaling
Action dimensions	Pick/Push/Stack/Pull use 4-D actions: $(\Delta x, \Delta y, \Delta z, \Delta g)$; PegInsertionSide uses 7-D actions: $(\Delta x, \Delta y, \Delta z, \Delta r_x, \Delta r_y, \Delta r_z, \Delta g)$
Action semantics	Translation components command bounded end-effector displacement per control step; rotation components command axis-angle delta; Δg opens or closes the gripper
Proprioception	Joint positions, joint velocities, end-effector pose, gripper width, and previous issued action; object poses are not included. A dedicated control removes the previous issued action
Control and simulation rate	Policy/control frequency 20 Hz; physics simulation 100 Hz with five physics substeps per policy step
Delay interpretation	$d_a = 1$ delays arm and gripper commands by one 20 Hz control interval, i.e., 50 ms; $d_a = 4$ corresponds to 200 ms
Episode horizon	PickCube 100 steps, PushCube 100 steps, PullCube 100 steps, StackCube 200 steps, PegInsertionSide 200 steps
Camera intrinsics	Single RGB camera, 128×128 resolution, square pixels, 90° horizontal field of view, principal point at image center
Nominal camera pose	Eye-in-hand camera mounted near the wrist; nominal translation (0.046, 0.000, 0.037) m and roll-pitch-yaw ($0^\circ, -35^\circ, 90^\circ$) relative to the end-effector frame

Table A3. Dataset scale and filtering audit used in the result tables.

Task	Expert Traj.	Attempted Traj.	Expert Trans.	Recovery Action	Recovery Auxiliary	Avg. Len.	Accept. Rate	Clip Ratio
PickCube	594	691	42.8 k	8.2 k	2.1 k	72.1	0.860	2.1%
PushCube	598	729	49.3 k	9.5 k	2.4 k	82.4	0.820	2.8%
StackCube	586	792	61.9 k	11.9 k	3.0 k	105.6	0.740	3.6%
PullCube	596	754	52.6 k	10.1 k	2.5 k	88.3	0.790	3.1%
PegInsertionSide	572	854	69.4 k	13.3 k	3.3 k	121.3	0.670	4.7%

Table A4. Training time and compute budget on one NVIDIA L40 GPU. Values are averaged over the five tasks and three seeds; intervals are across task–seed runs. Data generation time includes shift-aware expert replay and the single frozen recovery round. These budget numbers are reported to define comparable resource use across methods.

Method	Train Time/ Task-Seed	Peak Train Mem.	Data Gen./Task	Target Demos	Full 5-Task 3-Seed Train	Total GPU-Hours
DP + DR	3.8 ± 0.4 h	18.6 ± 1.1 GB	1.6 ± 0.3 h	0	57.0 h	65.0
DP + History	4.1 ± 0.4 h	19.8 ± 1.1 GB	1.6 ± 0.3 h	0	61.5 h	69.5
SysID + Params	4.4 ± 0.5 h	20.1 ± 1.2 GB	1.6 ± 0.3 h	0	66.0 h	74.0
SysID + Params + Dyn	4.6 ± 0.5 h	20.8 ± 1.3 GB	1.6 ± 0.3 h	0	69.0 h	77.0
Oracle	3.9 ± 0.4 h	18.9 ± 1.1 GB	1.6 ± 0.3 h	0	58.5 h	66.5
SCE-DP	5.2 ± 0.6 h	22.4 ± 1.4 GB	1.6 ± 0.3 h	0	78.0 h	86.0
Target fine-tune	8.0 ± 0.7 h	19.4 ± 1.2 GB	3.2 ± 0.4 h	1650	120.0 h	136.0

Appendix B. Validation Details

Table A5. Validation-only history-length selection for SCE-DP. Success is validation macro-average with 95% hierarchical-bootstrap confidence intervals ($n = 750$ nominal and $n = 9000$ shifted); latency is neural replanning latency measured by the protocol in Section 5. This table is the sole basis for selecting $H = 16$ before test evaluation. Bold indicates the selected history length and its corresponding best validation Avg Shift and delay MAE.

History Length	Val. Nominal	Val. Avg Shift	Delay MAE	Latency
$H = 1$	85.3 [76.5, 92.0]	57.8 [49.6, 64.8]	0.74 [0.66, 0.82] steps	39.1 ms
$H = 4$	85.9 [76.7, 92.5]	65.1 [56.9, 72.1]	0.51 [0.45, 0.57] steps	40.3 ms
$H = 8$	86.1 [77.1, 92.8]	71.9 [63.6, 78.9]	0.38 [0.33, 0.43] steps	41.2 ms
$H = 16$	85.6 [76.1, 92.4]	73.2 [64.8, 80.2]	0.34 [0.30, 0.38] steps	42.9 ms

Appendix C. Additional Results

Table A6. Paired statistical comparisons for SCE-DP. Differences are SCE-DP minus the comparison method in success-rate points. The hierarchical bootstrap resamples tasks, training seeds, configurations, and matched rollout seeds in that order. Raw p values are computed from the null-centered paired bootstrap distribution, and p_{Holm} values are Holm-corrected within each comparison family.

Comparison	Suite	Paired Diff.	95% CI	Raw p	p_{Holm}
SCE-DP vs. DP + DR	Held-out	+14.5	[+11.6, +16.7]	0.0001	0.0004
SCE-DP vs. DP + DR	Composition	+15.7	[+14.6, +16.8]	0.0001	0.0004
SCE-DP vs. DP + DR	Extra-single	+13.3	[+11.6, +14.8]	0.0001	0.0004
SCE-DP vs. DP + DR	Extra-combined	+18.0	[+16.6, +19.4]	0.0001	0.0004

Table A6. Cont.

Comparison	Suite	Paired Diff.	95% CI	Raw <i>p</i>	<i>p</i> _{Holm}
SCE-DP vs. SysID + Params	Held-out	+4.4	[+3.5, +5.3]	0.0001	0.0004
SCE-DP vs. SysID + Params	Composition	+5.2	[+3.9, +6.5]	0.0001	0.0004
SCE-DP vs. SysID + Params	Extra-single	+4.8	[+4.1, +5.5]	0.0001	0.0004
SCE-DP vs. SysID + Params	Extra-combined	+5.7	[+4.7, +6.9]	0.0001	0.0004
SCE-DP vs. SysID + Params + Dyn	Held-out	+2.0	[−0.4, +4.3]	0.1011	0.2748
SCE-DP vs. SysID + Params + Dyn	Composition	+1.9	[−1.3, +5.0]	0.2374	0.2748
SCE-DP vs. SysID + Params + Dyn	Extra-single	+2.1	[−0.4, +4.6]	0.0916	0.2748
SCE-DP vs. SysID + Params + Dyn	Extra-combined	+2.7	[−0.2, +5.4]	0.0607	0.2428

Table A7. Training-seed-level sensitivity analysis for Avg Shift. Values are the unweighted average of Held-out, Composition, Extra-single, and Extra-combined success rates for each independently trained seed.

Method	Seed 1	Seed 2	Seed 3	Mean ± SD
DP + DR	48.9	48.9	48.8	48.9 ± 0.1
SysID + Params	59.3	59.2	59.2	59.2 ± 0.1
SysID + Params + Dyn	63.7	61.4	61.1	62.1 ± 1.4
SCE-DP	64.3	64.2	64.2	64.3 ± 0.1

The seed-level analysis treats the three trained policies as the principal independent units. It is intended as a sensitivity check rather than a replacement for the matched hierarchical bootstrap, because three training seeds provide limited power for seed-only hypothesis testing.

Table A8. Three-factor camera + scale + delay stress test ($n = 1200$ per task and $n = 6000$ per macro-average). Macro values include the 95% hierarchical-bootstrap confidence interval. This suite is not used to support the main camera–delay composition claim because no three-factor configurations are observed during training or validation. Bold indicates the best result among methods that do not use oracle configuration parameters.

Method	Pick	Push	Stack	Pull	Peg	Macro
DP + DR	50.4	45.6	39.2	42.1	27.8	41.0 [33.8, 47.2]
SysID + Params	61.5	57.8	50.2	54.7	38.5	52.5 [45.1, 58.8]
SCE-DP	66.8	63.1	55.4	59.0	42.6	57.4 [49.5, 64.0]
Oracle	72.8	69.6	61.3	66.5	49.0	63.8 [56.0, 70.4]

Table A9. Per-task success rates (%; higher is better) for the two primary generalization suites. “Macro” is the unweighted mean of PickCube, PushCube, StackCube, PullCube, and PegInsertionSide and must match the corresponding entries in Table 6. For compactness, this table reports the main randomized and context-conditioned comparisons; full per-task results for DP-Nominal and target fine-tuning are provided in Appendix Table A13. Bold indicates the best result among methods that do not use oracle configuration parameters.

Method	Pick	Push	Stack	Pull	Peg	Macro
<i>Held-out in-range configurations</i>						
DP + DR	73.2	67.0	56.0	61.1	42.8	60.0
DP + History	72.5	67.6	53.5	60.8	40.7	59.0
SysID + Params	78.6	77.9	65.7	72.4	55.9	70.1
SysID + Params + Dyn	78.8	79.8	69.2	75.6	59.2	72.5
Oracle	87.4	85.1	74.4	81.8	63.3	78.4
SCE-DP	82.2	83.0	70.8	77.3	59.2	74.5

Table A9. *Cont.*

Method	Pick	Push	Stack	Pull	Peg	Macro
<i>Withheld camera + delay composition</i>						
DP + DR	62.7	55.1	49.3	49.8	33.6	50.1
DP + History	58.8	54.0	47.1	50.8	31.3	48.4
SysID + Params	71.2	64.8	59.7	61.5	45.8	60.6
SysID + Params + Dyn	72.8	68.7	62.6	64.9	50.7	63.9
Oracle	78.5	74.6	66.9	73.8	54.7	69.7
SCE-DP	78.3	70.2	65.2	66.6	48.8	65.8

Table A10. Online latent-context adaptation speed on held-out configurations at the natural replanning boundaries induced by $T_a = 4$. Success and parameter errors are reported as mean with 95% hierarchical-bootstrap confidence interval ($n = 6000$). The 0-record row uses the same learned initial context as the fixed-initial-context ablation in Table 10.

Available Completed Records	Success	Cam. MAE	Scale MAE	Delay MAE
0 records, learned prior	64.8 [56.2, 71.9]	3.21 [2.93, 3.49] cm	0.091 [0.081, 0.101]	1.18 [1.06, 1.30] steps
4 records	69.8 [61.6, 76.7]	1.41 [1.23, 1.59] cm	0.045 [0.039, 0.051]	0.52 [0.45, 0.59] steps
8 records	71.9 [63.8, 79.1]	1.02 [0.88, 1.16] cm	0.034 [0.029, 0.039]	0.39 [0.34, 0.44] steps
12 records	73.2 [64.9, 80.2]	0.91 [0.78, 1.04] cm	0.031 [0.027, 0.035]	0.35 [0.30, 0.40] steps
16 records	74.5 [66.2, 81.5]	0.84 [0.72, 0.96] cm	0.028 [0.024, 0.032]	0.31 [0.27, 0.35] steps

Table A11. Secondary metrics for SCE-DP on the four shifted suites. Task progress is the five-task macro-average normalized environment progress (% , higher is better). Degradation is nominal success (86.0%) minus shifted-suite success in percentage points (lower is better). Oracle-gap recovery is $(S_{\text{SCE}} - S_{\text{DP+DR}}) / (S_{\text{Oracle}} - S_{\text{DP+DR}})$ (% , higher is better). Context stability is the mean cosine similarity between consecutive system-context vectors over the final eight records (higher is better). Upward arrows ($\uparrow$) indicate that higher values are better, whereas the downward arrow ($\downarrow$) indicates that lower values are better. Brackets report 95% hierarchical-bootstrap confidence intervals.

Suite	Task Progress $\uparrow$	Degradation $\downarrow$	Oracle Gap Recovered $\uparrow$	Context Stability $\uparrow$
Held-out	84.6 [78.1, 90.2]	11.5	78.8%	0.94 [0.92, 0.96]
Composition	78.8 [71.4, 85.0]	20.2	80.1%	0.91 [0.88, 0.94]
Extra-single	75.9 [68.8, 82.0]	25.0	69.3%	0.89 [0.86, 0.92]
Extra-combined	69.8 [62.7, 76.1]	30.3	93.3%	0.85 [0.81, 0.89]

Table A12. SCE-DP failure-case taxonomy from failed shifted-suite rollouts. Categories are assigned by manual review of videos, success flags, action saturation logs, and context-probe errors.

Failure Mode	Count ($n = 500$)	Share	Typical Evidence
Weak excitation	120	24%	Limited response leaves scale or delay ambiguous
Fine contact sensitivity	105	21%	Near-contact attempts fail under small timing or pose errors
Occlusion or poor visual evidence	75	15%	The gripper or object occludes response evidence
Action saturation or clipping	70	14%	Compensation requires commands beyond controller limits
Severe extrapolation	80	16%	Shifts exceed the validation envelope and probe errors remain high
Episode-start information shortage	50	10%	Failure occurs before enough records are available

Appendix D. Additional Per-Task Results

Table A13. Full per-task success rates (% , higher is better) for DP-Nominal and target-configuration fine-tuning on the four shifted suites. “Macro” is the unweighted mean of the five task scores and must match the corresponding suite entry in Table 6.

Method	Suite	Pick	Push	Stack	Pull	Peg	Macro
DP-Nominal	Held-out	56.3	54.1	41.8	50.1	29.3	46.3
DP-Nominal	Composition	48.2	40.6	38.9	38.0	19.3	37.0
DP-Nominal	Extra-single	44.8	42.2	31.8	38.1	16.8	34.7
DP-Nominal	Extra-combined	34.8	31.8	23.9	30.8	17.3	27.7
Target fine-tune	Held-out	76.5	74.2	65.6	73.7	54.6	68.9
Target fine-tune	Composition	71.3	65.3	56.7	62.5	46.8	60.5
Target fine-tune	Extra-single	68.3	63.3	58.4	59.2	36.8	57.2
Target fine-tune	Extra-combined	57.8	52.3	46.1	52.8	36.5	49.1

References

1. Florence, P.; Lynch, C.; Zeng, A.; Ramirez, O.A.; Wahid, A.; Downs, L.; Wong, A.; Lee, J.; Mordatch, I.; Tompson, J. Implicit Behavioral Cloning. In *Proceedings of the 5th Conference on Robot Learning*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2022; Volume 164, pp. 158–168.
2. Mandlekar, A.; Xu, D.; Wong, J.; Nasiriany, S.; Wang, C.; Kulkarni, R.; Fei-Fei, L.; Savarese, S.; Zhu, Y.; Martín-Martín, R. What Matters in Learning from Offline Human Demonstrations for Robot Manipulation. In *Proceedings of the 5th Conference on Robot Learning*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2022; Volume 164, pp. 1678–1690.
3. Zhao, T.Z.; Kumar, V.; Levine, S.; Finn, C. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of the Robotics: Science and Systems*, Daegu, Republic of Korea, 10–14 July 2023. [\[CrossRef\]](#)
4. Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. RT-1: Robotics Transformer for Real-World Control at Scale. In *Proceedings of the Robotics: Science and Systems*, Daegu, Republic of Korea, 10–14 July 2023. [\[CrossRef\]](#)
5. Chi, C.; Feng, S.; Du, Y.; Xu, Z.; Cousineau, E.; Burchfiel, B.C.M.; Song, S. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of the Robotics: Science and Systems*, Daegu, Republic of Korea, 10–14 July 2023. [\[CrossRef\]](#)
6. Jin, T.; Song, W.; Zhou, T.; Zeng, J.; Jing, D.; Lu, Z. Free-View Robot Manipulation: Visuomotor Policy by Calibration Diffusion. *SSRN Electron. J.* **2026**. [\[CrossRef\]](#)
7. Ke, T.W.; Gkanatsios, N.; Fragkiadaki, K. 3D Diffuser Actor: Policy Diffusion with 3D Scene Representations. In *Proceedings of the 8th Conference on Robot Learning*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2025; Volume 270, pp. 1949–1974.
8. Heo, H.; Woo, S.; Kim, S.M.; Kim, J.; Lee, J.; Lee, Y.; Kim, Y.M. AnyCamVLA: Zero-Shot Camera Adaptation for Viewpoint Robust Vision-Language-Action Models. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, PA, USA, 27 September–1 October 2026; Accepted to IROS 2026; proceedings metadata not yet available. [\[CrossRef\]](#)
9. Gu, S.; Zheng, Y.; Li, W.; Zheng, Y.; Feng, Y.; Li, X.; Chen, Y.; Li, P.; Ding, W. VistaBot: View-Robust Robot Manipulation via Spatiotemporal-Aware View Synthesis. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vienna, Austria, 1–5 June 2026; Accepted to ICRA 2026; proceedings metadata not yet available. [\[CrossRef\]](#)
10. Li, W.; Jiang, X.; Qian, Q.; Zhao, D.; Lu, S.; Zhang, G.; Xu, R. From Fixed to Free Cameras: Calibration-Free View-Robust Vision-Language-Action Model. *arXiv* **2026**, arXiv:2607.05396. [\[CrossRef\]](#)
11. Yu, W.; Tan, J.; Liu, C.K.; Turk, G. Preparing for the Unknown: Learning a Universal Policy with Online System Identification. In *Proceedings of the Robotics: Science and Systems*, Cambridge, MA, USA, 12–16 July 2017. [\[CrossRef\]](#)
12. Kumar, A.; Fu, Z.; Pathak, D.; Malik, J. RMA: Rapid Motor Adaptation for Legged Robots. In *Proceedings of the Robotics: Science and Systems*, Virtual, 12–16 July 2021. [\[CrossRef\]](#)
13. Liang, Y.; Ellis, K.; Henriques, J. Rapid Motor Adaptation for Robotic Manipulator Arms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 16–22 June 2024; pp. 16404–16413. [\[CrossRef\]](#)
14. Ho, J.; Jain, A.N.; Abbeel, P. Denoising Diffusion Probabilistic Models. In *Proceedings of the Advances in Neural Information Processing Systems*, Virtual, 6–12 December 2020; Volume 33, pp. 6840–6851.
15. Song, Y.; Sohl-Dickstein, J.; Kingma, D.P.; Kumar, A.; Ermon, S.; Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. In *Proceedings of the International Conference on Learning Representations*, Virtual, 3–7 May 2021.
16. Song, J.; Meng, C.; Ermon, S. Denoising Diffusion Implicit Models. In *Proceedings of the International Conference on Learning Representations*, Virtual, 3–7 May 2021.

17. Janner, M.; Du, Y.; Tenenbaum, J.; Levine, S. Planning with Diffusion for Flexible Behavior Synthesis. In *Proceedings of the 39th International Conference on Machine Learning*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2022; Volume 162, pp. 9902–9915.
18. Tie, C.; Chen, Y.; Wu, R.; Dong, B.; Li, Z.; Gao, C.; Dong, H. ET-SEED: Efficient Trajectory-Level SE(3) Equivariant Diffusion Policy. In *Proceedings of the International Conference on Learning Representations*, Singapore, 24–28 April 2025.
19. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, BC, Canada, 24–28 September 2017; pp. 23–30. [\[CrossRef\]](#)
20. James, S.; Davison, A.J.; Johns, E. Transferring End-to-End Visuomotor Control from Simulation to Real World for a Multi-Stage Task. In *Proceedings of the 1st Annual Conference on Robot Learning*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2017; Volume 78, pp. 334–343.
21. Muratore, F.; Treede, F.; Gienger, M.; Peters, J. Domain Randomization for Simulation-Based Policy Optimization with Transferability Assessment. In *Proceedings of the 2nd Conference on Robot Learning*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2018; Volume 87, pp. 700–713.
22. Sandha, S.S.; Garcia, L.; Balaji, B.; Anwar, F.; Srivastava, M. Sim2Real Transfer for Deep Reinforcement Learning with Stochastic State Transition Delays. In *Proceedings of the 2020 Conference on Robot Learning*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2021; Volume 155, pp. 1066–1083.
23. Ross, S.; Gordon, G.; Bagnell, D. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2011; Volume 15, pp. 627–635.
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
25. Perez, E.; Strub, F.; de Vries, H.; Dumoulin, V.; Courville, A. FiLM: Visual Reasoning with a General Conditioning Layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, 2–7 February 2018; Volume 32. [\[CrossRef\]](#)
26. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2015; Volume 9351, pp. 234–241. [\[CrossRef\]](#)
27. Tao, S.; Xiang, F.; Shukla, A.; Qin, Y.; Hinrichsen, X.; Yuan, X.; Bao, C.; Lin, X.; Liu, Y.; Chan, T.K.; et al. Demonstrating GPU Parallelized Robot Simulation and Rendering for Generalizable Embodied AI with ManiSkill3. In *Proceedings of the Robotics: Science and Systems*, Los Angeles, CA, USA, 21–25 June 2025. [\[CrossRef\]](#)
28. Faura-Pujol, A.; Faundez-Zanuy, M.; Moral-Viñals, A.; López-Xarbau, J. Eye-Tracking Calibration to Control a Cobot. *Int. J. Comput. Methods Exp. Meas.* **2023**, *11*, 17–25. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.