Extended Abstract

Motivation. Dexterous manipulation policies trained in simulation face the core challenge of transferring across the sim-to-real gap. State-of-the-art systems such as SimToolReal (Kedia et al., 2026) sidestep the *visual* portion of this gap by relying on runtime 6D object-pose tracking, but this perception stack is brittle, accounting for 43.7% of their real-world failures. We ask whether an end-to-end visuomotor diffusion policy, operating directly on RGB images from a scene camera and proprioceptive state, can be trained purely in non-photorealistic simulation and still transfer across the visual domain gap, and whether the design of its training data can make this transfer more robust.

Method. We learn the manipulation policy through behavior cloning with Diffusion Policy (Chi et al., 2023), an end-to-end visuomotor diffusion policy that observes a two-step history of third-person RGB images and 29-D proprioceptive state and outputs short chunks of 29-D joint actions for a 22-DoF hand on a 7-DoF arm. To address covariate shift, we adopt PDP's noisy-state, clean-action scheme (Truong et al., 2024) where during data collection, we perturb the expert's executed actions with temporally-correlated Ornstein-Uhlenbeck noise (with per-group magnitudes for the arm and fingers), driving rollouts into noisy states paired with clean corrective labels. We compare two placements of this noise: 1) *uniform per-step noise* across the full trajectory, and 2) our *phase-gated closure* scheme, which detects the grasp-closure window via hand-object proximity and zeroes the accumulated noise state there, preserving clean supervision during the key contact formation phase.

Implementation. All training data is collected in IsaacGym with no visual domain randomization. We train CNN-backbone and Transformer-backbone diffusion policies with receding horizon control (horizon 16; replanning every 4 or 8 steps) on 1) clean, 2) uniform-noise, and 3) phase-gated datasets. We evaluate zero-shot in two closed-loop environments sharing identical physics but different camera renderes: 1) IsaacGym's native renderer (in-distribution visuals) and 2) Blender Cycles with matched camera calibration (photorealistic visual shift, serving as a real-world proxy). We report pick-place-release success on 9 in-distribution and 3 held-out objects over 32 episodes per object.

Results. Clean-data policies succeed in IsaacGym but lose most of their success under photorealistic rendering, confirming a substantial visual domain shift the diffusion policy must overcome. Phase-gated closure noising improves photorealistic transfer over its step-matched clean baseline in every setting, and our best policy (CNN, 4-step replanning) raises held-out photorealistic success from 22.9% to 34.4% and in-distribution photorealistic success from 6.2% to 21.2% while also improving IsaacGym performance. However, uniform noise transfers inconsistently, helping at 8-step replanning but underperforming the clean baseline with 4-step replanning. Finally, more frequent replanning also improves transfer, and Transformer backbones achieve the best IsaacGym success rates yet collapse to 0% for in-distribution objects in the photorealistic environment.

Discussion. The results demonstrate that the noisy-state, clean-action recipe from Truong et al. (2024) cannot be applied uniformly across a dexterous manipulation trajectory and that different contact phases require different noising strategies. While perturbations during approach and transport yield useful corrective labels, perturbations during grasp formation can corrupt the corrective labels. Consistently, our qualitative evaluation of failed photorealistic rollouts reveals that most failures occur during grasp formation. Moreover, photorealistic transfer is also strongly object-dependent (on our best policy, large-handled spatulas reach 48.4% for photorealistic OOD while the flat eraser fails entirely). Finally, successful rollouts exhibit re-grasp recovery behavior after objects slip out of the hand, notably without any runtime pose estimation.

Conclusion. To robustify diffusion policies for dexterous manipulation across the visual sim-to-sim gap, we find that *where* noise is applied during dataset collection matters just as much as whether it is applied. Injecting noise using our phase-gated closure method and training a CNN-based diffusion policy with receding horizon control leads to our best result, where held-out photorealistic success increases from 22.9% to 34.4% over the clean dataset baseline. Future work for training viusomotor diffusion policies for dexterous manipulation include tuning noise parameters, adding visual perturbations during data collection, and evaluating on real hardware.

SimToolReal-RGB: Visuomotor Diffusion Policies for Dexterous Manipulation

Cayden Gu
Department of Computer Science
Stanford University
caydengu@stanford.edu

Karen Vo
Department of Computer Science
Stanford University
karenvo@stanford.edu

Christine Zhang
Department of Computer Science
Stanford University
chrzhang@stanford.edu

Abstract

Dexterous manipulation policies trained in simulation can struggle to transfer to the real world due to the sim-to-real gap, and current state-of-the-art methods sidestep the visual portion of this gap by relying on runtime 6D pose tracking that is brittle to occlusion, symmetry, and low contrast. We instead train a vision-conditioned, end-to-end visuomotor diffusion policy that operates directly on RGB and proprioceptive state, removing this pose-estimation dependency. Trained purely on simulated pick-place-release demonstrations from an RL expert, the policy is evaluated zero-shot in a photorealistic simulation that serves as a real-world proxy. To improve robustness and generalization, we inject noise into the expert's executed actions during data collection, comparing uniform per-step noise against a phase-gated scheme that suppresses perturbations during the unrecoverable grasp-closure phase while noising the approach and transport phases. Across CNN and Transformer backbones and varying replanning frequencies, we find that clean-data policies succeed in the training simulator but lose most of their success under photorealistic rendering, confirming a substantial domain shift. Our phase-gated closure noising method yields the strongest transfer, increasing held-out photorealistic success from 22.9% to 34.4%, while uniform noise helps far less. Transformer backbones achieve the best training-simulator scores yet collapse under photorealistic evaluation, indicating that optimizing simulator performance alone is misleading for robust transfer.

1 Introduction

Learning dexterous manipulation policies that operate robustly in the real world remains a central challenge in robotics. Simulation offers an attractive path forward: it provides cheap, safe, and massively parallel data collection, enabling policies to be trained on far more experience than could ever be gathered on physical hardware. However, policies trained in simulation often fail to transfer to reality due to the *sim-to-real gap*, which describes discrepancies in dynamics, contact physics, and visual appearance between simulated and real environments. Closing this gap without sacrificing the scalability of collecting simulation data is therefore a key bottleneck for deploying learned manipulation policies in the real world.

Current state-of-the-art sim-to-real approaches for dexterous tool manipulation bypass the visual sim-to-real gap through utilizing runtime 6D object pose tracking at deployment, which can be brittle

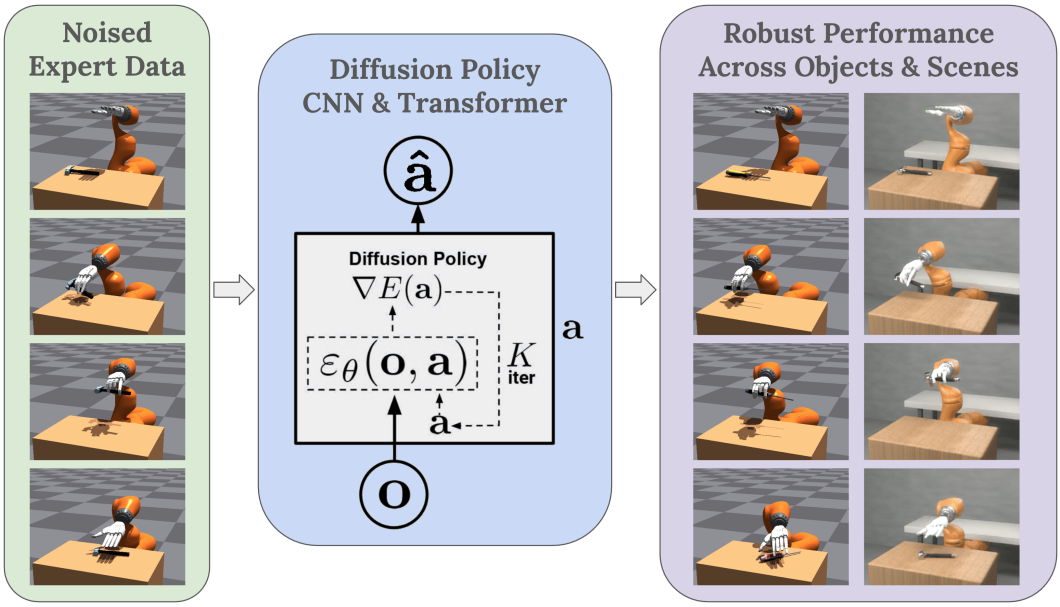


Figure 1: **Method Overview.** We noise pick-place-release data collected by an expert RL policy, train diffusion models on this data using CNN-based and Transformer-based architectures, and evaluate performance in the training environment and in a photorealistic simulation.

to occlusion, rotational symmetry, and low visual contrast (Kedia et al., 2026). On the other hand, the natural alternative of collecting real-world demonstrations covering every tool, task, and visual condition is impractical at scale.

In this project, we explore the sim-to-real gap through training a diffusion policy purely on non-photorealistic simulation data and investigating whether it can transfer zero-shot to a photorealistic simulation environment.[1] Specifically, we train a diffusion policy to be able to pick up, move, place, and release various tools, generalizing to unseen objects and environments at test time. To achieve robust generalization to the photorealistic simulation environment, we investigate different methods of adding noise to training data, as argued by Truong et al. (2024), to help with imitation learning robustness.

2 Related Work

Sim-to-real for dexterous tool manipulation. Our work primarily builds on SimToolReal, which demonstrates that a single simulation-trained RL policy can generalize to real tools by training on procedurally generated tool primitives and conditioning on a sequence of goal poses that the tools should follow (Kedia et al., 2026). However, this representation intentionally bypasses the visual sim-to-real gap by relying on estimated 6D tool pose and grasp-region geometry, which are obtained through FoundationPose and SAM 3D, instead of raw RGB frames. While effective, these runtime dependencies are the leading source of real-world failure, as object pose estimation accounts for 43.7% of failed rollouts. Our project addresses this limitation by replacing the live perception stack with an end-to-end visuomotor diffusion policy (Chi et al., 2023) that operates directly on RGB, bypassing the FoundationPose-dependent failure mode.

End-to-end vision-based dexterous manipulation. DextrAH-RGB (Singh et al., 2025) is the closest precedent for the zero-shot sim-to-real transfer of an end-to-end RGB-based dexterous grasping policy. However, their approach distills a state-based RL teacher into a deterministic stereo-RGB student via supervised learning, producing a unimodal policy. We instead train a vision-conditioned diffusion policy (Chi et al., 2023) which models the full multimodal distribution over expert actions and is better suited to tasks with multiple equally-valid solutions.

[1] Due to hardware constraints, we were unable to test on a real robot within the project timeframe. We use a realistic simulation environment as a proxy for real world testing.

Robust behavior cloning under distribution shift. A key paper for our implementation is Physics-Based Character Animation via Diffusion Policy (Truong et al., 2024), which addresses the covariate shift problem of naive BC by collecting data with action noise injected during expert rollouts while storing the expert's clean action at each visited noisy state. This scheme of collecting noisy-state clean-action data improves out-of-distribution recovery, which we hypothesize adds the necessary robustification to enable sim-to-sim and sim-to-real transfer of vision-conditioned diffusion policies for tool manipulation.

3 Method

We learn the manipulation policy through behavior cloning with Diffusion Policy from Chi et al. (2023), replacing the live perception stack from SimToolReal with an end-to-end visuomotor diffusion policy that operates directly on RGB and proprioceptive state. The policy observes a 2-step history (consisting of the current and previous RGB images and 29-D proprioceptive states) and outputs a short chunk of 29-D joint actions. Training fits a noise-prediction network ϵ_θ to recover Gaussian noise injected into expert action chunks (Section 3.1) conditioned on the observation history. At inference time, the policy samples an action chunk by starting from Gaussian noise and denoising it over 100 DDPM steps. Figure 1 summarizes the full pipeline.

The diffusion policy offers several advantages that make it well suited to our setting. First, it is multimodal by design: because it models the full distribution over expert actions rather than a single deterministic mapping, it naturally handles tasks that admit several equally valid solutions, such as grasping a handle at different points. Second, training is stable and reward-free. Unlike reinforcement learning, the policy is trained through supervised behavior cloning on expert demonstrations, which avoids the difficulty of reward engineering and the instability associated with online exploration. Finally, we robustify the policy through noise injection (Section 3.3), adopting the noisy-state, clean-action scheme of PDP (Truong et al., 2024). By perturbing the expert's executed actions during data collection, we drive rollouts into noised states paired with clean corrective labels, improving out-of-distribution recovery and helping bridge sim-to-sim transfer.

3.1 Data Generation

All data is generated using the pretrained SimToolReal policy (Kedia et al., 2026) in a non-photorealistic IsaacGym simulator with no visual domain randomization. The expert is a privileged RL policy controlling a 22-DoF Sharpa five-fingered hand mounted on a 7-DoF KUKA iiwa 14 arm. Each stored transition contains the RGB scene image (rendered at 256×192), 29-D proprioceptive state, and 29-D expert action.

3.2 The Noising Process

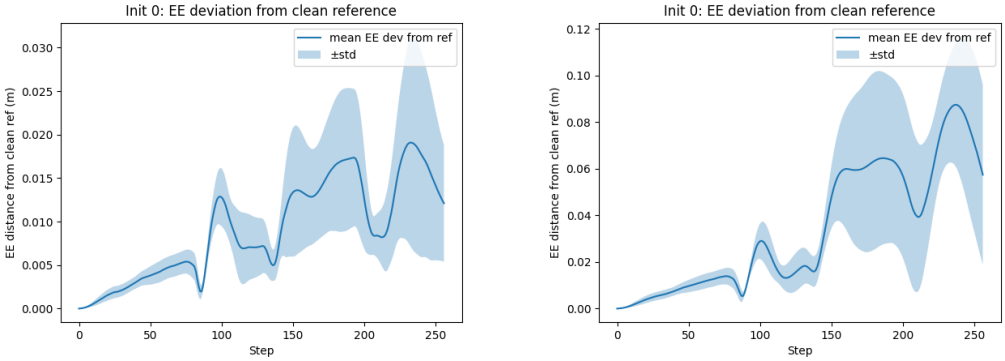
We perturb the expert's executed action with an Ornstein–Uhlenbeck process (Lillicrap et al., 2016), which produces smoothly varying noise that drifts over time while being pulled back toward zero, so perturbations are correlated across timesteps rather than independent. For each action group g (defined below), the noise state evolves as

$$\mathbf{n}_t \;\leftarrow\; \mathbf{n}_{t-1} + \theta\,(\mu - \mathbf{n}_{t-1})\,\Delta t \;+\; \sigma_g\sqrt{\Delta t}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I), \tag{1}$$

with $\theta = 0.15$, $\mu = 0$, $\Delta t = 1$. The executed action is the clean expert action plus the accumulated noise, clamped to the action range $\mathbf{a}_t^{\text{exec}} = \text{clip}(\mathbf{a}_t^{\text{clean}} + \mathbf{n}_t,\, -1,\, 1)$. Because $\mathbf{n}_t$ are temporally correlated, perturbations integrate over a rollout instead of averaging out, which matters for grasp closure.

Contact is far more sensitive to finger perturbation than to gross arm motion, so we use different noise magnitudes σ_g per action group. The values in Table 1 were selected empirically through a sweep, as we chose the largest perturbation magnitudes that still preserved reliable task completion while keeping off-nominal trajectories coherent. Figure 2 compares faithfulness to the clean trajectory for two noise levels. Across the sweep, we measure how far noisy rollouts drift from the clean reference over an episode, and report a representative lower and higher noise level to show the two regimes. At the lower scale (Fig. 2a), rollouts stay within $\sim$2 cm and form a tight band; at the higher scale (Fig. 2b) deviation grows to 6–9 cm and the band widens sharply after grasp (note the $\sim 4\times$ difference

in vertical scale in the graphs). In both, deviation is small during approach and accumulates through transport, motivating magnitudes large enough to widen the state distribution but not so large that rollouts diverge into incoherent trajectories.



(a) Lower noise scale.

(b) Higher noise scale.

Figure 2: End-effector deviation of noisy rollouts from the clean reference trajectory, from a fixed initialization (mean $\pm$ std over rollouts). Deviation is small during approach and accumulates through transport, as temporally correlated perturbations integrate over the rollout. The higher noise scale (right) produces substantially larger divergence; note the differing vertical axes.

Table 1: Per-group OU noise magnitudes σ_g (normalized action units).

	Arm		**Hand**				
	Base	Wrist	Thumb	Index	Middle	Ring	Pinky
σ_g	0.020	0.040	0.060	0.040	0.040	0.032	0.032

3.3 Noise-Injection Strategies

We compare two noise-injection strategies on our data, each augmenting the clean demonstrations with noisy-state, clean-action pairs to improve robustness and generalization to the photorealistic simulation environment. Figure 3 highlights the difference in the strategies across the pick-place-release trajectory.

Method 1: Uniform Per-Step Noise. At every timestep, we query the deterministic expert for its clean action, add temporally correlated Ornstein-Uhlenbeck (OU) noise to its clean action, execute the perturbed action in simulation, and store the clean action as the label. As in PDP, we store noisy-state, clean-action pairs, with noise applied across the entire pick-place-release trajectory. Each start configuration (a sampled initial object pose and arm configuration) is reused for up to 10 rollouts, each with an independent noise realization, so the policy observes many diverse recovery trajectories from the same starting state and learns the multi-modality of valid grasps. Only rollouts that complete the task are kept.

Method 2: Phase-gated Closure. We find that grasp closure is the least recoverable phase of the task. Once the fingers begin to close, contact dynamics make perturbations effectively irreversible, so a noised closure produces an action label inconsistent with a successful grasp.

This method applies OU action noise during approach and transport, but when the palm-object distance falls below 0.08 m, it suppresses the accumulated OU noise on the wrist and finger action groups for the closure window. We also suppress the noise for 5 padding steps surrounding this window to ensure the grip is not re-perturbed mid-grasp.

We note that since the OU process carries noise forward across timesteps, suppressing only the fresh increment would leave residual perturbation on the fingers. Instead, the gate zeroes the entire

accumulated noise state $\mathbf{n}_t$, ensuring the gated joints are truly clean during closure. The base of the arm is also excluded so the gross arm position stays perturbed and diverse.

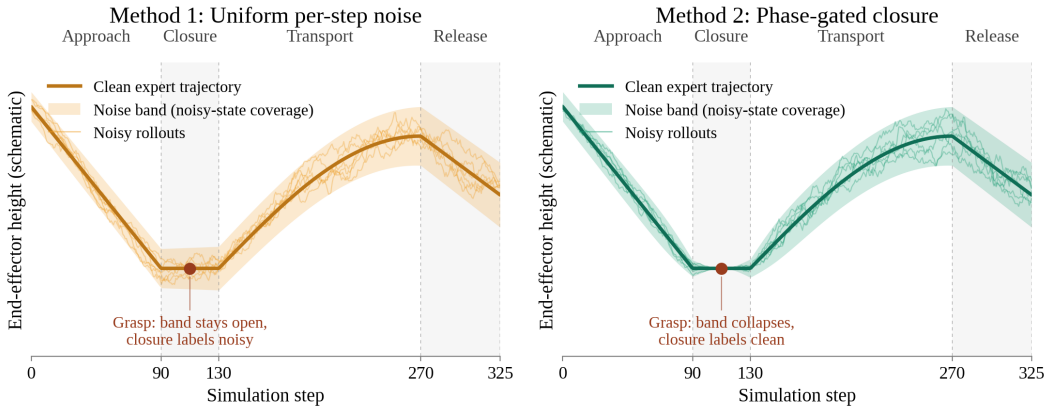


Figure 3: **Noise-injection strategies.** Left: uniform per-step noise keeps the band open through grasp closure. Right: phase-gating collapses the band at closure.

4 Experimental Setup

We train and evaluate on a pick-place-release tool manipulation task in which a dexterous hand must grasp a tool from a tabletop and place it near a goal on the other side of the table. A rollout is considered successful only if the object is lifted, transported a minimum distance, placed within a target tolerance, and released stably.

Following Chi et al. (2023), we evaluate two backbones for the diffusion policy:

- **CNN-based diffusion policy**: robust out-of-the-box and generally requires little hyperparameter tuning on real-world tasks.
- **Time-series diffusion Transformer**: requires more tuning but can achieve stronger performance when task complexity and the rate of action change are high.

Both architectures use a horizon length of 16 with **receding-horizon control**, replanning within the horizon. For each architecture we train two variants: one that **replans every 8 steps** and one that **replans every 4 steps**. Other than these hyperparameter changes, we largely use the default hyperparameters from the original Diffusion Policy repository.

For each backbone and replanning frequency, we first train a diffusion policy on a clean pick-place-release dataset with no noise injected as a baseline. We then train a separate policy on each of the two noised datasets to test the hypothesis that injecting noise into the training data improves robustness and generalization. Each model is trained to 30 epochs, and we run evaluations on the checkpoints with the lowest validation losses[2] in two closed-loop environments that share the same physics but differ in the renderer used for policy observations:

- **IsaacGym:** we first use IsaacGym's native, non-photorealistic renderer to measure the policy's performance under in-distribution visual observations.
- **Photorealistic:** our second environment keeps the same IsaacGym physics but replaces the renderer with Blender Cycles, evaluating the policy under photorealistic visual-domain shift. We first build a base Blender scene containing the environment objects, lighting, and a camera with intrinsics and extrinsics matching IsaacGym's. Then, at each render step, we read the world-frame poses of the robot links and object from IsaacGym, copy them into the base scene, render a photorealistic image, and feed that image into the diffusion policy.

[2]CNN-based architectures generally plateau in validation loss as early as 15 epochs while Transformer-based models exhibit continually decreasing loss until 28-30 epochs.

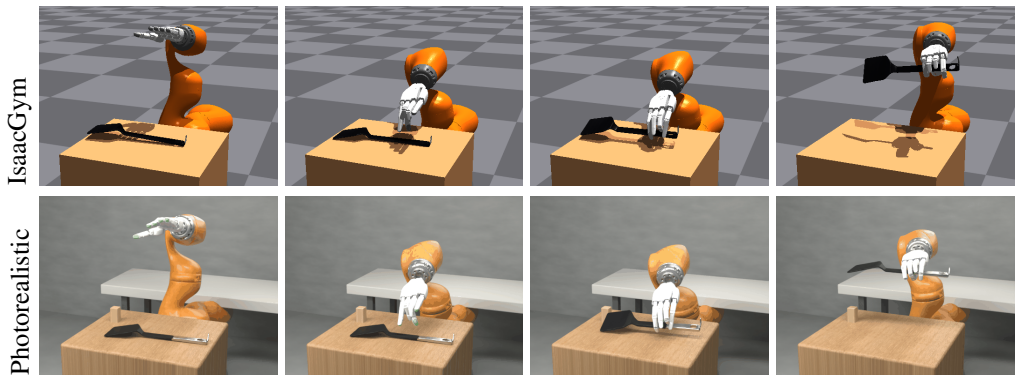


Figure 4: **The same rollout rendered in both evaluation environments.** A pickup of the held-out OOD flat spatula shown in IsaacGym (top) and the photorealistic Blender environment (bottom). The two environments share identical physics (NVIDIA PhysX) and differ only in the renderer.

At each simulation step, the policy receives the two-step history of rendered RGB images along with 29-D proprioceptive states and predicts an action chunk, and IsaacGym then advances under NVIDIA PhysX. Figure 4 shows a successful pickup of an object rendered in both environments, illustrating the visual gap the visuomotor diffusion policy must bridge.

To further evaluate how well the policy generalizes, we split the 12 objects released by Kedia et al. (2026) (consisting of six tool categories with two instances each) into 9 in-distribution (ID) training objects and 3 held-out out-of-distribution (OOD) objects, and report pick-place-release success separately on each set. The nine training objects are the claw hammer, the sharpie and staples markers, the flat and handle erasers, the blue and red brushes, and the long and short screwdrivers. We form the OOD set by randomly holding out one instance from one category and both instances from another to test two types of object generalization: 1) a new instance of a category seen in training (the mallet hammer) and 2) objects from an entirely unseen category (the flat and spoon spatulas). We run evaluations for 32 episodes per object in each setting and average the number of successes over total attempts to obtain the ID and OOD success rates.

5 Results

Tables 2 and 3 report pick-place-release success for CNN-based and Transformer-based diffusion policies respectively across the IsaacGym and photorealistic environments. We find that three factors shape the transfer to the realistic simulation: 1) where noise is injected, 2) the replanning frequency, and 3) the policy backbone. Overall, our results support the central hypothesis that appropriately placed noise injection improves policy generalization to new objects and visual observations. On the CNN backbone, phase-gated closure noise injection improves photorealistic transfer over the step-matched clean baseline in every setting while uniform noise transfers inconsistently, helping at 8-step replanning but underperforming the clean baseline at 4-step. The benefits do not extend to the Transformer backbone, a finding that is consistent with the backbone behavior reported by Chi et al. (2023) and indicative of overfitting to the training renderer.

Noise placement is critical to photorealistic transfer. For the CNN-based diffusion policy, whether noise injection improves photorealistic transfer depends on *where* the noise is applied. Uniform noise with 4-step replanning achieves the highest IsaacGym OOD success rate, improving from 22.0% for the step-matched clean baseline to 71.9%. However, this improvement does not translate as strongly to the photorealistic environment, where realistic OOD success falls from 22.9% for the clean 4-step baseline to 17.7%. This suggests that uniformly perturbing the full trajectory can improve robustness under the training renderer while still degrading the precise contact behavior needed to overcome the visual domain shift.

The strongest overall method is phase-gated closure noise with 4-step replanning. Compared to the clean 4-step baseline, it improves IsaacGym ID success from 64.0% to 82.3%, IsaacGym OOD from 22.0% to 59.4%, realistic ID success from 6.2% to 21.2%, and realistic OOD success from 22.9% to

34.4%. Moreover, unlike uniform noise, phase-gated closure improves photorealistic success over its step-matched clean baseline at both replanning frequencies (17.7% vs. 5.2% at 8 steps). These results indicate that noise is most useful when applied during recoverable phases such as approach and transport, while grasp closure should remain comparatively clean.

Method	IsaacGym ID	IsaacGym OOD	Realistic ID	Realistic OOD
Clean data, 8 steps (Baseline)	61.6	26.7	1.4	5.2
Clean data, 4 steps (Baseline)	64.0	22.0	6.2	22.9
Uniform noise, 8 steps	63.2	56.3	3.8	9.4
Uniform noise, 4 steps	69.4	**71.9**	7.3	17.7
Phase-gated closure, 8 steps	74.2	55.2	5.6	17.7
Phase-gated closure, 4 steps	**82.3**	59.4	**21.2**	**34.4**

Table 2: **Success rate percentage** comparison between our methods for the **CNN-based diffusion policy**.

More frequent replanning improves transfer. Across the CNN configurations, replanning every 4 steps matches or outperforms replanning every 8 steps in nearly every setting, with the largest gains under photorealistic rendering especially for OOD objects (e.g., 34.4% vs. 17.7% for phase-gated closure). We attribute this to the visual-domain shift inducing larger observation errors, so shorter open-loop execution enables the policy to better correct misestimates before they compound into failed grasps.

The Transformer backbone collapses under visual shift. The Transformer results show a different pattern. Transformer policies tend to perform well in IsaacGym evaluation, with the best OOD success reaching 84.4%, far exceeding the best of any CNN-backbone policies. However, transformer-based policies collapse under photorealistic evaluation, with **0.0% realistic ID success** for every configuration and at most 5.2% realistic OOD success. This suggests that without focused hyperparameter tuning, the Transformer backbone is far more sensitive to the visual domain shift than the CNN backbone despite achieving stronger or comparable IsaacGym scores for every corresponding noising strategy and re-planning frequency.

Method	IsaacGym ID	IsaacGym OOD	Realistic ID	Realistic OOD
Clean data, 8 steps (Baseline)	69.3	68.6	0.0	0.0
Clean data, 4 steps (Baseline)	73.2	59.4	0.0	0.0
Uniform noise, 8 steps	73.9	68.8	0.0	1.0
Uniform noise, 4 steps	77.4	67.7	0.0	2.1
Phase-gated closure, 8 steps	**79.2**	72.9	0.0	**5.2**
Phase-gated closure, 4 steps	76.0	**84.4**	0.0	0.0

Table 3: **Success rate percentage** comparison between our methods for the **time-series diffusion Transformer**.

Failure modes and grasp recovery. Qualitatively, we observe that the dominant failure mode in photorealistic simulation is the failure to establish the initial grasp. In many failed rollouts, the hand approaches with a small height or lateral offset error, closes around empty space, pushes the object away, or makes weak contact that cannot support the subsequent lift. However, in many successful photorealistic rollouts, the policy exhibits solid corrective behavior after imperfect first contact by re-centering the wrist, re-approaching the tool, and forming a second grasp before lifting and transporting it. Figure 5 shows a representative example in which the policy initially grasps the sharpie marker, loses it during lift (marker falls to table), then re-centers and re-grasps the marker before lifting and placing it at the goal. This suggests that noise injection can teach grasp-recovery behavior that is significantly weaker when training only on clean demonstrations. Notably, our end-to-end RGB policy performs this recovery without runtime object pose estimation.

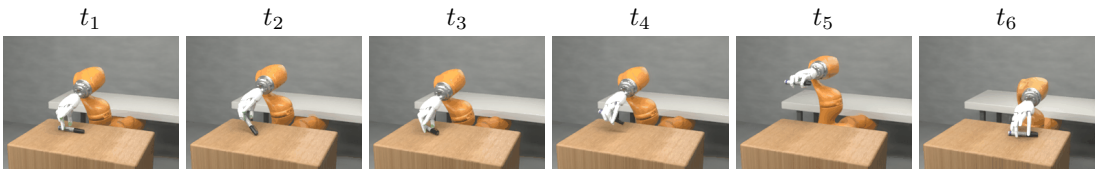


Figure 5: **Recovery from a mid-grasp slip in photorealistic simulation.** The policy reaches and forms an initial grasp (t_1) before the sharpie marker slips out of the fingers during initial lift (t_2). The policy then re-grasps the marker (t_3) and goes on to lift and place the marker at the goal ($t_4 - t_6$).

6 Discussion

Different contact phases require different noising strategies. Our results suggest that the noisy-state, clean-action recipe cannot be applied uniformly across a manipulation trajectory. The scheme assumes that, after a perturbation, the expert's action remains a meaningful corrective label (Truong et al., 2024). This assumption is plausible during approach and transport, where small deviations are likely to be corrected smoothly, but not during grasp closure, where small changes in wrist pose or finger configuration can lead to degenerative contact formation, leaving the stored labels inconsistent. The stronger photorealistic transfer of phase-gated closure noising over uniform noising indicates that contact-rich phases require reduced noise, phase-specific noise scales, or cleaner supervision. We do not claim that removing closure noise entirely is optimal, but treating all phases with the same perturbation schedule appears too crude for dexterous manipulation. A tuned, phase-dependent noise schedule may possibly perform better, but we leave such investigation to future work.

Photorealistic failures concentrate at grasp acquisition. The qualitative failure analysis (Sec. 5) shows that photorealistic rollouts overwhelmingly fail before transport begins—the hand approaches with a small offset, closes around empty space, or makes weak contact—rather than during placement or release. Contact setup is thus the bottleneck for visual transfer in our task, which is consistent with the phase-gated result: the phases where corrective data helps most are precisely the ones preceding and surrounding first contact.

Photorealistic transfer is strongly object-dependent. As shown in Table 4, the best photorealistic policy transfers unevenly across tool categories. Spatulas achieve the highest success rate (48.4%), followed by screwdrivers (35.9%), while erasers and markers remain difficult (6.2% and 12.5% respectively). This object dependence helps explain why OOD success can exceed ID success in our aggregate results, as the OOD split contains both spatulas while the ID split contains thin or flat objects such as markers and erasers. Qualitatively, the two objects on which the best policy never succeeds (flat eraser and staples marker) almost always fail at grasp acquisition: the hand frequently closes above the flat eraser, and the staples marker's thin cylindrical body rolls out of imprecise pinches.

Object Category	Photorealistic Success
Spatula	48.4%
Screwdriver	35.9%
Brush	23.4%
Hammer	20.3%
Marker	12.5%
Eraser	6.2%

Table 4: **Category-level photorealistic pick-place-release success for the best policy** (phase-gated closure noise, CNN backbone, 4-step replanning). Each category contains two objects with 32 rollouts per object.

Limitations. First, absolute photorealistic success is low across the board, with no policy achieving greater than 35% success on the pick-place-release task. In fact, every Transformer variant scores 0% realistic ID and less than 6% on realistic OOD, so we treat the photorealistic numbers as a relative

signal for comparing noising strategies rather than estimates of real-world performance. Moreover, our photorealistic evaluation uses a single Blender scene, camera, lighting setup, and renderer, making it only a rough proxy for sim-to-real transfer. True real-world deployment would introduce additional visual variation, sensor noise, and other contact dynamics not captured by our simulation pipeline.

7 Conclusion

We present visuomotor diffusion as a route to dexterous tool manipulation operating end-to-end on RGB and proprioceptive state. Training purely in simulation, we study action-noise injection during data collection as a means of robustifying zero-shot transfer to a photorealistic environment and compare uniform per-step noise against a phase-gated scheme.

We find that clean-data policies succeed in IsaacGym but lose most of their success under photorealistic rendering, isolating the failure as a purely visual domain shift since the dynamics are unchanged. Injecting noise using our phase-gated closure method, which adds temporally correlated (OU) noise during all parts of the trajectory except the grasping window, raises held-out photorealistic success from 22.9% to 34.4% for our best-performing model. Uniform noise, by contrast, transfers inconsistently, indicating that where noise is applied matters as much as whether it is applied, and that the grasp-closure window in particular benefits from clean supervision. Transformer backbones achieve the best training-simulator scores yet collapse under photorealistic evaluation, showing that optimizing simulator performance alone is a misleading objective for robust transfer.

Future work on training visuomotor diffusion policies for dexterous manipulation includes tuning joint-specific and phase-specific noise levels, exploring depth or 3D representations (Ze et al., 2024) as additional observations, extending to other critical skills for tool-use such as in-hand rotation (Kedia et al., 2026), and validating on real hardware after training with visual domain randomizations.

8 Team Contributions

- **Group Member 1:** Cayden worked on experimenting with noising methods, creating the photorealistic environment in Blender, and evaluating models in the environment.
- **Group Member 2:** Karen worked on experimenting with noising methods, creating the pick-place-release task, training CNN diffusion policies using those methods, and evaluation.
- **Group Member 3:** Christine worked on the initial setup of the data collection and training pipelines, training both CNN and transformer diffusion policies, and evaluation.

Changes from Proposal: Our project did not deviate much from our proposal, but one thing to note is that we focused more on testing different noising methods for the training data rather than tuning diffusion policy hyperparameters, as we believed that modifying the training data would be more impactful for robustness than hyperparameter tuning.

AI Tools Disclosure: We use AI coding tool assistance for supplementary infrastructure such as setting up data-loading boilerplate, monitoring GPU usage, and preparing the evaluations. All core methods are implemented by ourselves.

References

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. 2023. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*.

Kushal Kedia, Tyler Ga Wei Lum, Jeannette Bohg, and C. Karen Liu. 2026. SimToolReal: An Object-Centric Policy for Zero-Shot Dexterous Tool Manipulation. arXiv:2602.16863 [cs.RO] `https://arxiv.org/abs/2602.16863`

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*.

Ritvik Singh, Arthur Allshire, Ankur Handa, Nathan Ratliff, and Karl Van Wyk. 2025. DextrAH-RGB: Visuomotor Policies to Grasp Anything with Dexterous Hands. arXiv:2412.01791 [cs.RO] https://arxiv.org/abs/2412.01791

Takara Everest Truong, Michael Piseno, Zhaoming Xie, and Karen Liu. 2024. PDP: Physics-Based Character Animation via Diffusion Policy *(SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 86, 10 pages. doi:10.1145/3680528.3687683

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 2024. 3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations. *arXiv preprint arXiv:2403.03954* (2024).