JDSG: JITTER DECREASING SHORTCUT GUIDANCE FOR DIFFUSION POLICY IN PRECISION-SENSITIVE MANIPULATION TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative behavior cloning methods such as Diffusion Policy can produce intra-chunk trajectory jitter that may compromise contact stability in precise manipulation. We first show that post-hoc trajectory-preserving jitter-minimization improves task success, suggesting that intra-chunk trajectory jitter is a contributing factor to policy failure. Our analysis indicates that effective smoothing must balance jitter reduction with preservation of the generated trajectory. However, post-hoc smoothing modifies only the completed action chunk, leaving no opportunity for iterative denoising steps to refine the trajectory. Incorporating a smoothing objective into diffusion sampling offers an alternative, but directly differentiating through the policy network introduces additional inference cost. We introduce Jitter Decreasing Shortcut Guidance (JDSG), an inference-time method that applies trajectory-preserving jitter-minimization during diffusion sampling without policy-network backpropagation. At each denoising step, JDSG adjusts the intermediate clean action estimate using a quadratic objective that penalizes third-order temporal differences and deviations from the unmodified estimate. This adjustment requires only a small linear-system solve, after which sampling proceeds using the policy's predicted noise. On the evaluated simulation tasks, JDSG achieves higher average success rates than both the base Diffusion Policy and post-hoc jitter-minimizing optimization, including under the tested visual distribution shift. Real-world experiments further show improvements over the base policy. These findings support sampling-time, trajectory-preserving jitter-minimization as a useful approach to improving precise manipulation.

1 INTRODUCTION

Imitation learning enables robots to acquire behaviors from expert demonstrations rather than manually specified reward functions or control rules (Argall et al., 2009). Behavior cloning, a widely used imitation-learning approach, formulates this problem as supervised learning from observation–action pairs and trains a policy to predict expert actions (Osa et al., 2018; Pomerleau, 1988). However, conventional behavior cloning often produces a single point estimate and can struggle when the same observation admits multiple feasible actions or trajectories (Florence et al., 2022).

Generative behavior cloning addresses this limitation by modeling the conditional action distribution rather than a single action prediction (Shafiullah et al., 2022). Diffusion Policy (Chi et al., 2025) instantiates generative behavior cloning with a conditional denoising diffusion process over action chunks. By modeling complex action distributions and generating temporally coherent action sequences (Zhang et al., 2025) through iterative denoising (Pan et al., 2026), Diffusion Policy has demonstrated strong performance across diverse robot manipulation tasks. This paradigm also underlies many vision–language–action (VLA) models, which learn to map visual observations and language instructions to robot actions from large-scale robot demonstration datasets (Kim et al., 2024; Intelligence et al., 2025; Bjorck et al., 2025).

However, precision-sensitive manipulation remains challenging. Tasks involving sustained contact, narrow geometric tolerances, or delicate object interactions can fail even when a policy generates an action chunk whose overall motion appears appropriate for the task. Recent efforts have therefore

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

sought to understand why learned policies fail in contact-rich manipulation and how these failures can be mitigated (Parés-Morlans et al., 2026; Malhotra et al., 2025; Zhang et al., 2026). In this work, we focus on *intra-chunk trajectory jitter*: small, high-frequency position fluctuations within an action chunk that can destabilize contact during execution. We investigate whether reducing these fluctuations while preserving the intended motion can improve task success.

We first investigate whether post-hoc trajectory smoothing of generated action chunks can improve precise-manipulation performance. Specifically, we evaluate Gaussian filtering, Savitzky–Golay filtering (Savitzky & Golay, 1964), moving-average filtering, and trajectory-preserving jitter-minimization based on third-order temporal differences, motivated by minimum-jerk motion objectives (Flash & Hogan, 1985). On the evaluated state-based Tool Hang task, trajectory-preserving jitter-minimization yields the largest success-rate improvement over the original policy and is the only evaluated method whose reported 95% confidence interval for the improvement excludes zero; the full results are detailed in Section 4.2. This finding supports intra-chunk jitter as a contributing factor to failure and motivates studying how to reduce it effectively.

We next examine why trajectory-preserving jitter-minimization is effective. Reducing jitter alone may be insufficient, because smoothing can also distort task-relevant motion. In our comparison, trajectory-preserving jitter-minimization achieves the strongest jitter reduction while inducing less deviation from the original trajectory than Savitzky–Golay and moving-average filtering. This is because its objective explicitly balances a third-order finite-difference penalty against a penalty on trajectory changes. Together with the success-rate improvement, this result motivates trajectory-preserving jitter-minimization as a suitable objective for improving precise performance.

Despite these benefits, post-hoc smoothing is applied only after diffusion sampling has completed. The modified trajectory therefore cannot benefit from iterative observation-conditioned denoising updates. Incorporating the jitter-reduction objective into sampling offers a way to repeatedly combine trajectory smoothing with the learned policy's predictions. Loss-based guidance provides a framework for such incorporation (Song et al., 2023), but directly differentiating an objective on a predicted clean action through the policy network requires an inference-time backward pass. This additional computation is undesirable for latency-sensitive robot control. Manifold Preserving Guided Diffusion (MPGD) offers a clean-estimate shortcut that avoids backpropagation through the diffusion model (He et al., 2024), providing a basis for applying guidance without this computational requirement.

Building on this shortcut formulation, we introduce **Jitter Decreasing Shortcut Guidance (JDSG)**, an inference-time method that steers diffusion sampling toward less jittery action trajectories. At each reverse-diffusion step, JDSG directly adjusts the position component of the intermediate clean action estimate using the trajectory-preserving jitter-minimization objective. The quadratic objective admits a closed-form solution, implemented through a small linear-system solve over the action horizon. JDSG then combines the adjusted clean estimate with the policy's predicted noise in the DDIM update. This formulation retains the learned policy as a generative prior while requiring only one policy-network evaluation per denoising step and no policy-network backpropagation.

We evaluate JDSG on precision-sensitive manipulation tasks in simulation, under a controlled visual distribution shift, and in real-world robot experiments. Across the evaluated simulation tasks, JDSG achieves higher average success rates than both the base Diffusion Policy and post-hoc jitter-minimizing optimization. This average advantage persists under the tested visual shift. On the real-world Peg-in-Hole and Tool Hang tasks, JDSG also improves success rates over the base policy.

Our contributions are as follows:

- We show that post-hoc trajectory-preserving jitter-minimization improves success on a precision-sensitive manipulation task, providing evidence that intra-chunk trajectory jitter contributes to policy failure.

- We provide empirical evidence supporting a trajectory-preserving jitter-minimization objective: trajectory-preserving jitter-minimization substantially reduces jitter while inducing less trajectory change than the evaluated Savitzky–Golay and moving-average filters.

- We propose JDSG, which integrates this objective into diffusion sampling through a clean-estimate shortcut, and demonstrate improved average simulation performance and real-world gains over the base policy without policy-network backpropagation.

2 Related Work

Generative behavior cloning for contact-rich manipulation. Generative behavior cloning methods, including Diffusion Policy, model potentially multimodal action distributions to generate robot actions (Chi et al., 2025). Despite their strong performance, contact-rich and precision-sensitive manipulation remains challenging because small action deviations can destabilize contact. Recent work has addressed this challenge primarily at training time. For example, Parés-Morlans et al. (2026) identifies insufficient training at low-noise timesteps as a source of failure in contact-rich tasks and improves performance by emphasizing these timesteps during training. Contractive Diffusion Policies mitigate score-matching and numerical solver errors that accumulate during reverse diffusion by encouraging contractive sampling dynamics (Soleimani Abyaneh et al., 2026). They add a score-Jacobian-based regularizer during training so that nearby reverse-diffusion trajectories are pulled together, reducing sensitivity to perturbations and unwanted action variance. In contrast, JDSG refines actions at inference time without modifying policy training.

Several inference-time methods improve generated actions through post-optimization or action selection. LiPo performs lightweight post-optimization for smoother actions in dynamic manipulation and focuses on discontinuities between successive action chunks (Son & Park, 2025). TacDiffusion reduces jitter in the force domain for tactile manipulation (Wu et al., 2025). KDPE uses kernel density estimation to select high-density actions and thereby rule out low-fidelity actions, but requires multiple parallel policy inferences (Rosasco et al., 2025). JDSG instead targets *intra-chunk* trajectory jitter in precision-sensitive manipulation and uses a single diffusion sampling trajectory.

Loss-based guidance for diffusion models. Loss-based guidance steers diffusion generation toward desired properties by optimizing an objective during sampling. Existing loss-based guidance methods either use exact guidance, which can require multiple denoising and backward passes, or approximate the guidance using an intermediate clean estimate while still differentiating through the diffusion model (Song et al., 2023). Beyond their computational cost, updates performed directly on noisy samples can move samples away from the noisy-data manifold on which the diffusion model was trained.

Manifold Preserving Guided Diffusion (MPGD) is primarily motivated by mitigating this manifold-deviation problem (He et al., 2024). It performs guidance on an intermediate clean estimate and then maps the modified estimate back into the diffusion trajectory, encouraging guidance that remains consistent with the data-manifold structure. Its clean-estimate shortcut also provides an important computational advantage: gradients are computed with respect to the clean estimate, avoiding backpropagation through the diffusion model. JDSG draws on this shortcut formulation to directly adjust intermediate clean action estimates using a trajectory-preserving jitter-minimization objective. Thus, JDSG reduces intra-chunk trajectory jitter during sampling without policy-network backpropagation.

Action coherence guidance. Action Coherence Guidance (ACG) improves action coherence in flow-based vision-language-action models and considers Gaussian filtering as a post-hoc smoothing baseline (Park et al., 2025). Whereas ACG does not investigate whether post-hoc smoothing improves success in precision-sensitive manipulation, we show that post-hoc trajectory-preserving jitter-minimization improves performance by reducing intra-chunk jitter while preserving the global trajectory trend. This result identifies intra-chunk trajectory jitter as one contributing factor to failure in precision-sensitive tasks. Methodologically, ACG is designed only for Transformer-based policies and uses classifier-free guidance, which requires conditional and unconditional model evaluations at each denoising step. JDSG operates on a UNet-based diffusion policy and uses a loss-based shortcut formulation requiring a single model evaluation per denoising step with negligible additional computation.

3 METHODOLOGY

3.1 PRELIMINARIES

3.1.1 DIFFUSION POLICY

We consider a Diffusion Policy that generates a normalized action chunk $\mathbf{A}_0 \in \mathbb{R}^{H\times D}$ conditioned on the current observation $\mathbf{o}$, where H is the action horizon and D is the action dimension (Chi et al., 2025). We denote the noisy action chunk at reverse diffusion timestep t by $\mathbf{A}_t$, and let $\bar{\alpha}_t$ be the cumulative noise schedule.

With the ϵ-prediction parameterization, the denoiser predicts $\boldsymbol{\epsilon}_\theta(\mathbf{A}_t, t, \mathbf{o})$ and yields the intermediate clean action estimate

$$\widehat{\mathbf{A}}_0^{(t)} = \frac{\mathbf{A}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta(\mathbf{A}_t, t, \mathbf{o})}{\sqrt{\bar{\alpha}_t}}. \tag{1}$$

For deterministic DDIM sampling, the next reverse-diffusion state is obtained from the clean estimate and predicted noise as

$$\mathbf{A}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\,\widehat{\mathbf{A}}_0^{(t)} + \sqrt{1-\bar{\alpha}_{t-1}}\,\boldsymbol{\epsilon}_\theta(\mathbf{A}_t, t, \mathbf{o}). \tag{2}$$

All smoothing and guidance operations in this work are applied in the normalized action space before the final action chunk is unnormalized for robot execution.

3.1.2 POST-HOC INTRA-CHUNK SMOOTHING

We first study post-hoc smoothing methods that operate on the final generated action chunk $\mathbf{A}_0$. Since contact stability is primarily affected by translational motion, we write $\mathbf{A}_0 = [\mathbf{P}_0, \mathbf{U}_0]$, where $\mathbf{P}_0 \in \mathbb{R}^{H\times 3}$ contains the Cartesian position coordinates and $\mathbf{U}_0$ contains the remaining action dimensions. Each method modifies only $\mathbf{P}_0$ and leaves $\mathbf{U}_0$ unchanged.

Moving-average filtering. We first consider moving-average filtering, which replaces each position with the average of positions in a local temporal window of radius m:

$$\widetilde{\mathbf{P}}_0[h] = \sum_{r=-m}^{m} k_r\,\mathbf{P}_0[h-r], \tag{3}$$

where h indexes the action horizon. For moving-average filtering, $k_r = 1/(2m+1)$ for all $r \in -m, \ldots, m$. Boundary values are replicated when $h - r$ falls outside the action chunk.

Gaussian filtering. Gaussian filtering also applies Eq. 3, but uses a normalized Gaussian kernel:

$$k_r^{\mathrm{G}} = \frac{\exp\!\left(-r^2/(2\sigma^2)\right)}{\sum_{q=-m}^{m}\exp(-q^2/(2\sigma^2))}, \qquad m = \lceil 3\sigma \rceil, \tag{4}$$

where σ controls the smoothing strength. Positions closer to timestep h receive larger weights than more distant positions.

Savitzky–Golay filtering. Savitzky–Golay filtering similarly uses Eq. 3, with coefficients obtained by fitting a degree-p polynomial to each local temporal window. Let $\mathbf{V} \in \mathbb{R}^{(2m+1)\times(p+1)}$ be the Vandermonde matrix whose row associated with offset $r \in -m, \ldots, m$ is $[1, r, \ldots, r^p]$. The kernel coefficients are

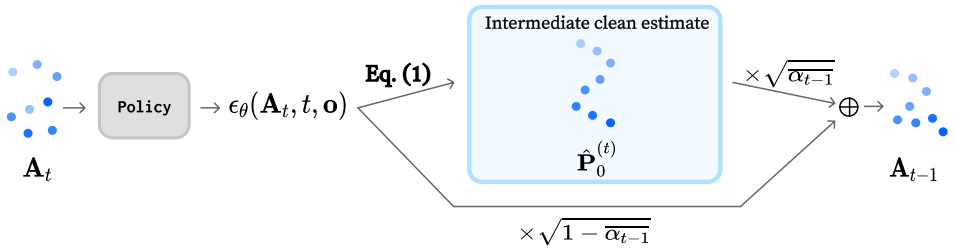
$$\mathbf{k}^{\mathrm{SG}} = \mathbf{e}_0^\top \left(\mathbf{V}^\top \mathbf{V}\right)^{-1} \mathbf{V}^\top, \tag{5}$$

where $\mathbf{e}_0$ selects the constant coefficient of the fitted polynomial. This filter attenuates high-frequency fluctuations while preserving low-order trajectory trends.

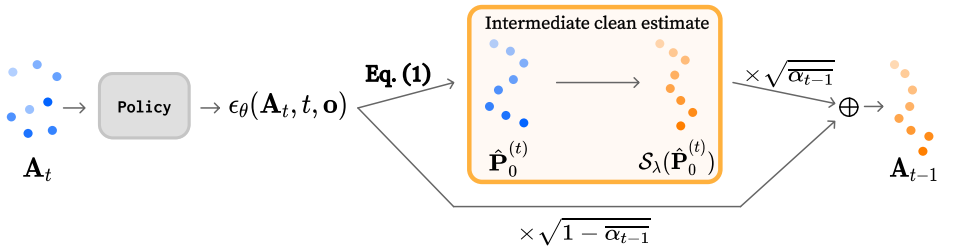
3.2 TRAJECTORY-PRESERVING JITTER-MINIMIZATION.

We use a third-order finite-difference operator $\mathbf{D}_3 \in \mathbb{R}^{(H-3)\times H}$ to measure high-frequency trajectory variation:

$$(\mathbf{D}_3\mathbf{P})_{h,:} = -\mathbf{P}_{h,:} + 3\mathbf{P}_{h+1,:} - 3\mathbf{P}_{h+2,:} + \mathbf{P}_{h+3,:}. \tag{6}$$



(a) DDIM update



(b) JDSG update

Figure 1: Comparison of the DDIM sampling update and JDSG. DDIM obtains an intermediate clean action estimate from the denoiser and combines it with the predicted noise to update the next diffusion state. JDSG applies the trajectory-preserving jitter-minimization operator $\mathcal{S}_\lambda$ to the position component of the intermediate clean estimate before the same DDIM update.

The corresponding jitter measure is $\|\mathbf{D}_3\mathbf{P}\|_F^2$. Given a reference trajectory $\mathbf{P}$, jitter-minimizing trajectory optimization solves

$$\mathcal{S}_\lambda(\mathbf{P}) = \arg\min_{\mathbf{Z}} \frac{1}{2}\|\mathbf{Z}-\mathbf{P}\|_F^2 + \frac{\lambda}{2}\|\mathbf{D}_3\mathbf{Z}\|_F^2, \tag{7}$$

where the first term preserves the generated trajectory and the second term penalizes intra-chunk jitter. This quadratic objective has the closed-form solution

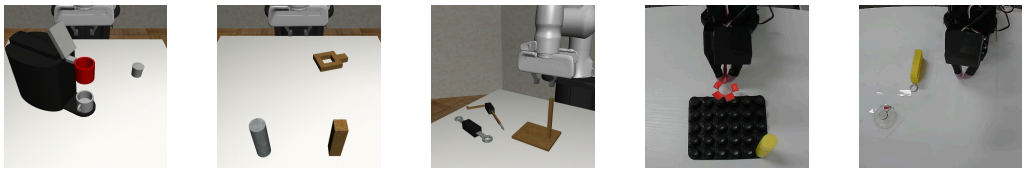
$$\mathcal{S}_\lambda(\mathbf{P}) = \left(\mathbf{I}_H + \lambda\mathbf{D}_3^\top\mathbf{D}_3\right)^{-1}\mathbf{P}. \tag{8}$$

In practice, we solve the linear system in Eq. 8 directly rather than explicitly forming the inverse. Post-hoc jitter-minimizing optimization applies this operator once after sampling:

$$\widetilde{\mathbf{A}}_0 = \left[\mathcal{S}_\lambda(\mathbf{P}_0),\ \mathbf{U}_0\right]. \tag{9}$$

3.3 Trajectory-preserving jitter-minimization as a Guidance Objective

We choose trajectory-preserving jitter-minimization as the guidance objective for JDSG for three reasons. First, in our post-hoc smoothing study, it is the only evaluated smoothing method that improves task success over the original policy; we discuss this result in detail in Section 4.2. Second, as shown in Eq. 7, trajectory-preserving jitter-minimization reduces jitter while preserving the global motion trend of the generated trajectory. The first term penalizes deviations from the original generated position sequence, whereas the second term, based on third-order finite differences, suppresses high-frequency fluctuations. We provide empirical evidence for this favorable jitter–trajectory preservation trade-off in Section 4.3. Third, as shown in Eq. 8, the resulting quadratic objective admits a closed-form solution. Thus, it requires only a linear-system solve over the action horizon and introduces little additional computational overhead. This efficiency makes trajectory-preserving jitter-minimization suitable for application at every reverse-diffusion step as an inference-time guidance objective.



Coffee | Square | Tool Hang | Peg-in-Hole (real) | Tool Hang (real)

Figure 2: Manipulation environments used in our experiments. We evaluate simulation policies on Coffee, Square, and Tool Hang, and evaluate real-world policies on Peg-in-Hole and Tool Hang.

3.4 Jitter Decreasing Shortcut Guidance

Post-hoc smoothing modifies an action only after the diffusion process has ended. JDSG instead incorporates jitter-minimizing optimization objective within each reverse-diffusion step. At timestep t, we first obtain the clean estimate $\widehat{\mathbf{A}}_0^{(t)} = [\widehat{\mathbf{P}}_0^{(t)}, \widehat{\mathbf{U}}_0^{(t)}]$ using Eq. 1. JDSG then replaces its position component with the closed-form optimized estimate:

$$\widetilde{\mathbf{A}}_0^{(t)} = \left[\mathcal{S}_\lambda\left(\widehat{\mathbf{P}}_0^{(t)}\right), \widehat{\mathbf{U}}_0^{(t)}\right]. \tag{10}$$

Finally, we retain the denoiser's predicted noise and use the optimized clean estimate in the DDIM update:

$$\mathbf{A}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\,\widetilde{\mathbf{A}}_0^{(t)} + \sqrt{1-\bar{\alpha}_{t-1}}\,\boldsymbol{\epsilon}_\theta(\mathbf{A}_t, t, \mathbf{o}). \tag{11}$$

This shortcut formulation guides the diffusion trajectory through its intermediate clean action estimates while retaining the diffusion policy as the generative prior. Crucially, as shown in Eq. 11, it does not need to differentiate through $\boldsymbol{\epsilon}_\theta(\mathbf{A}_t, t, \mathbf{o})$. Consequently, JDSG performs sampling-time jitter reduction without a policy-network backward pass.

4 Experiments

Our experiments answer four questions: (Q1) Can post-hoc smoothing improve performance in precision-sensitive manipulation tasks? (Q2) Why is trajectory-preserving jitter-minimization (TPJM) a suitable objective? (Q3) Does JDSG improve upon Diffusion Policy (DP) and post-hoc TPJM in simulation, including under visual distribution shifts? (Q4) Does JDSG transfer to real-world manipulation?

4.1 Experimental Setup

Post-hoc smoothing study (Q1–Q2). We use the state-based Tool Hang task from RoboMimic(Mandlekar et al., 2021). Policies use DDIM sampling with 100 denoising steps, end-effector control, an action horizon of 16, and an execution horizon of 8. For each method and seed, we evaluate 1,000 episodes; reported results are averaged over three random seeds. We compare the original DP with DP augmented by each post-hoc smoothing method: Gaussian filtering, Savitzky–Golay filtering, moving-average filtering, and TPJM.

Simulation evaluation of JDSG (Q3). We evaluate image-based policies on Coffee from MimicGen(Mandlekar et al., 2023) and Square and Tool Hang from RoboMimic. All methods use the same DDIM sampler, action horizon, and execution horizon as above. Each result averages 100 trials over three seeds. To assess robustness under visual distribution shifts, we remove the original texture from selected task-relevant objects and recolor them using the average original texture color with its lightness reduced by 10% in the HSL color space.

Real-world evaluation of JDSG (Q4). We evaluate DP and JDSG on Peg-in-Hole and Tool Hang tasks using an SO-101 robot arm. All methods use the same DDIM sampler, action horizon, and execution horizon as above. Each result averages 100 trials over three seeds. Inference is performed on an NVIDIA RTX 5080 Laptop GPU.

Table 1: Post-hoc smoothing results on the state-based Tool Hang task. Numbers in parentheses are gains over DP. The 95% CIs are for success-rate gains (percentage points); jitter and trajectory changes are measured relative to DP.

Method	Success rate (%)	95% CI (pp)	Jitter change	Trajectory change
DP	60.83	Reference	Reference	Reference
DP + Gaussian	62.50 (+1.67)	[-0.27, +3.60]	-0.26%	0.003
DP + Savitzky–Golay	62.23 (+1.40)	[-0.70, +3.50]	-87.66%	2.880
DP + Moving Average	62.30 (+1.47)	[-0.67, +3.60]	-89.90%	2.964
DP + TPJM	**63.20 (+2.37)**	**[+0.27, +4.47]**	-97.09%	1.898

Table 2: Simulation success rates (%) on precision-sensitive manipulation tasks. Numbers in parentheses are gains over DP in the corresponding setting.

Setting	Method	Coffee	Square	Tool Hang	Avg. gain
Clean	DP	85.33	77.00	55.67	Reference
	DP + TPJM	87.00 (+1.67)	**81.00 (+4.00)**	59.00 (+3.33)	+3.00
	DP + JDSG	**88.00 (+2.67)**	80.33 (+3.33)	**62.67 (+7.00)**	**+4.33**
Visual shift	DP	88.33	75.00	56.67	Reference
	DP + TPJM	**91.00 (+2.67)**	78.67 (+3.67)	56.67 (+0.00)	+2.11
	DP + JDSG	90.67 (+2.33)	**80.00 (+5.00)**	**58.33 (+1.66)**	**+3.33**

4.2 Can Post-hoc Smoothing Improve Precision-sensitive Manipulation?

We first investigate whether post-hoc smoothing can improve performance in precision-sensitive manipulation. We evaluate Gaussian filtering, Savitzky–Golay filtering, moving-average filtering, and TPJM. For each method, we sweep its smoothing hyperparameter and report the configuration that achieves the highest task success rate.

As shown in Table 1, TPJM achieves the largest improvement over the original DP, increasing the success rate by 2.37 percentage points. To assess whether the observed improvements are statistically reliable, we compute 95% confidence intervals (CIs) for the success-rate improvement relative to DP. An improvement is considered statistically significant when its CI excludes zero. Among the evaluated post-hoc smoothing methods, only TPJM yields a statistically significant improvement, with a 95% CI of $[+0.27, +4.47]$ percentage points.

The existence of a post-hoc smoothing method that reliably improves task performance suggests that intra-chunk trajectory jitter is a contributing source of failure in precision-sensitive manipulation. However, the results also show that reducing jitter alone is insufficient: the choice of smoothing objective is important. We investigate why TPJM is particularly suitable in Section 4.3.

4.3 Why Is TPJM a Suitable Objective?

Table 1 also answers Q2 by comparing the trade-off between jitter reduction and deviation from the original generated trajectory. Gaussian filtering produces almost no trajectory change (0.003), but it also provides negligible jitter reduction (-0.26%). In contrast, Savitzky–Golay and moving-average filtering substantially reduce jitter (-87.66% and -89.90%, respectively), but induce large trajectory changes of 2.880 and 2.964.

TPJM achieves the strongest jitter reduction (-97.09%) while inducing substantially less trajectory change (1.898) than the two strongly smoothing methods. Importantly, this favorable trade-off coincides with the highest task success rate. These results provide empirical evidence that an effective smoothing objective should jointly suppress intra-chunk jitter and preserve the task-relevant global trajectory trend. As shown in Eq. 7, TPJM directly encodes this trade-off through its trajectory-preservation term and third-order jitter penalty, making it a suitable objective for precision-sensitive manipulation.

Table 3: Real-world success rates (%). Numbers in parentheses are gains over DP.

Method	Peg-in-Hole	Tool Hang	Avg. gain
DP	61.00	52.33	Reference
DP + JDSG	**66.33 (+5.33)**	**63.00 (+10.66)**	**+8.00**

4.4 Does JDSG Improve Performance in Simulation?

Table 2 answers Q3. Under clean visual observations, JDSG improves the average success rate by 4.33 percentage points over DP, exceeding the 2.33-point gain obtained by post-hoc TPJM. JDSG achieves the best performance on Coffee and Tool Hang, while post-hoc TPJM performs best on Square. Thus, although the strongest method can vary by task, JDSG provides the best average performance across the three tasks.

JDSG also remains effective under visual distribution shifts. When task-relevant objects are recolored, JDSG improves the average success rate by 3.33 percentage points over DP, compared with a 2.11-point gain from post-hoc TPJM. It achieves the highest success rates on Square and Tool Hang under this shift. These results suggest that applying trajectory-preserving jitter-minimization objective during diffusion sampling is more effective than modifying only the final generated action chunk, and that the benefit persists when visual observations differ from the training distribution.

4.5 Does JDSG Transfer to Real-world Manipulation?

Table 3 answers Q4. JDSG improves success on both real-world tasks, increasing the success rate from 61.00% to 66.33% on Peg-in-Hole and from 52.33% to 63.00% on Tool Hang. These correspond to gains of 5.33 and 10.66 percentage points, respectively, with an average gain of 8.00 points. These results indicate that the benefit of sampling-time jitter reduction transfers beyond simulation to real-world precision-sensitive manipulation.

5 Conclusion

We investigated intra-chunk trajectory jitter as a contributing source of failure in precision-sensitive manipulation with Diffusion Policy. Our post-hoc smoothing study showed that trajectory-preserving jitter-minimization (TPJM) is the only evaluated method that yields a statistically significant success-rate improvement over the base policy. The analysis further showed that TPJM provides a favorable trade-off between suppressing high-frequency fluctuations and preserving the generated trajectory's global motion trend.

Building on these observations, we introduced Jitter Decreasing Shortcut Guidance (JDSG), which incorporates TPJM into each reverse-diffusion step. JDSG's clean-estimate shortcut retains the diffusion policy as a generative prior while avoiding policy-network backpropagation at inference time. Across the evaluated simulation tasks, JDSG achieved higher average success rates than both the base Diffusion Policy and post-hoc TPJM under clean observations and the tested visual distribution shift. JDSG also improved real-world performance on both Peg-in-Hole and Tool Hang.

Limitations. JDSG currently requires selecting the TPJM regularization strength through a hyperparameter sweep. The best setting can differ across tasks, so applying the method to a new task may require additional task-specific evaluation. Moreover, JDSG refines only Cartesian position trajectories and therefore does not directly address failures associated with other action dimensions, such as orientation, gripper control, or contact forces.

Future work. A promising direction is to use a world model to roll out candidate action trajectories produced with multiple TPJM hyperparameters and select the action associated with the predicted best outcome. Such model-based selection could replace offline task-specific sweeps with adaptive, observation-dependent hyperparameter selection at inference time. Further extensions could jointly guide additional action dimensions, incorporate contact-aware objectives or tactile feedback, and adapt the jitter-reduction strength across diffusion timesteps and stages of a manipulation task.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing and to draft portions of the manuscript. We did not use generative AI tools to generate or analyze experimental data, execute experiments, or evaluate experimental results. All AI-assisted text, including technical descriptions, claims, and citations, was reviewed, revised, and verified by the authors. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe the JDSG algorithm, its trajectory-preserving jitter-minimization objective, and its closed-form solution in Section 3.2. We report the evaluated tasks, baselines, evaluation protocol, random-seed setting, and quantitative results in Section 4.1. For post-hoc smoothing and JDSG, we specify the optimized action component and the relevant smoothing or regularization hyperparameters in Appendix. Upon publication, we will release the implementation, evaluation configurations, trained checkpoints, and scripts used to reproduce the reported simulation and real-world experiments. These details are intended to enable reproduction of the method and facilitate comparison with future inference-time improvements to diffusion-based robot policies.

REFERENCES

Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *The journal of Neuroscience*, 5(7):1688–1703, 1985.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on robot learning*, pp. 158–168. PMLR, 2022.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, WeiHsiang Liao, Yuki Mitsufuji, Zico Kolter, Ruslan Salakhutdinov, et al. Manifold preserving guided diffusion. In *International Conference on Learning Representations*, volume 2024, pp. 44819–44850, 2024.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Rhea Malhotra, Yuejiang Liu, and Chelsea Finn. Self-guided action diffusion. *arXiv preprint arXiv:2508.12189*, 2025.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2): 1–179, 2018.

Chaoyi Pan, Giridharan Anantharaman, Nai-Chieh Huang, Claire Jin, Daniel Pfrommer, Chenyang Yuan, Frank Permenter, Guannan Qu, Nicholas Boffi, Guanya Shi, et al. Much ado about noising: Dispelling the myths of generative robotic control. In *International Conference on Learning Representations*, volume 2026, pp. 90575–90614, 2026.

Carlota Parés-Morlans, Nils Kuhn, Isabel Liu, Alberta Longhini, and Jeannette Bohg. Demystifying when and why vlas fail in contact-rich tasks and how to fix them. *arXiv preprint arXiv:2608.01402*, 2026.

Minho Park, Kinam Kim, Junha Hyung, Hyojin Jang, Hoiyeong Jin, Jooyeol Yun, Hojoon Lee, and Jaegul Choo. Acg: Action coherence guidance for flow-based vision-language-action models. *arXiv preprint arXiv:2510.22201*, 2025.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Andrea Rosasco, Federico Ceola, Giulia Pasquale, and Lorenzo Natale. Kdpe: A kernel density estimation strategy for diffusion policy trajectory selection. *arXiv preprint arXiv:2508.10511*, 2025.

Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

Amin Soleimani Abyaneh, Charlotte Morissette, Mohamad H Danesh, Anas Houssaini, David Meger, Gregory Dudek, and Hsiu-Chin Lin. Contractive diffusion policies. In *International Conference on Learning Representations*, volume 2026, pp. 17569–17604, 2026.

Dongwoo Son and Suhan Park. Lipo: A lightweight post-optimization framework for smoothing action chunks generated by learned policies. *International Journal of Control, Automation and Systems*, 23(11):3284–3292, 2025.

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, pp. 32483–32498. PMLR, 2023.

Yansong Wu, Zongxie Chen, Fan Wu, Lingyun Chen, Liding Zhang, Zhenshan Bing, Abdalla Swikir, Sami Haddadin, and Alois Knoll. Tacdiffusion: Force-domain diffusion policy for precise tactile manipulation. In *2025 IEEE international conference on robotics and automation (ICRA)*, pp. 11831–11837. IEEE, 2025.

Thomas T Zhang, Daniel Pfrommer, Chaoyi Pan, Nikolai Matni, and Max Simchowitz. Action chunking and exploratory data collection yield exponential improvements in behavior cloning for continuous control. *arXiv preprint arXiv:2507.09061*, 2025.

Zhiyuan Zhang, Pokuang Zhou, Kaidi Zhang, Adeesh Desai, Temitope Amosa, Davood Soleymanzadeh, Jiuzhou Lei, Minghui Zheng, and Yu She. Contactworld: What matters in vision-tactile world models for contact-rich manipulation. *arXiv preprint arXiv:2606.13877*, 2026.

Table 4: Selected TPJM regularization weights λ for post-hoc TPJM and JDSG.

Method	Evaluation setting	Coffee	Square	Tool Hang
Post-hoc TPJM	Clean	10^{-6}	10^{-4}	$\{10^{-5}, 10^{-3}\}$
JDSG	Clean	$\{10^{-4}, 10^{-2}\}$	10^{-7}	10^{-3}
Post-hoc TPJM	Visual OOD	$\{10^{-1}, 10^{0}\}$	10^{-7}	—
JDSG	Visual OOD	10^{-5}	10^{-4}	10^{-10}

A Appendix

A.1 Smoothing and Regularization Hyperparameters

For the post-hoc smoothing study in Section 4.2, we sweep the hyperparameters of each smoothing method and select the setting with the highest average success rate. For Savitzky–Golay filtering, we evaluate window-length and polynomial-order pairs $(W, P) \in \{(5,2),(7,2),(7,3),(9,2),(9,3)\}$. For moving-average filtering, we evaluate window lengths $W \in \{3,5,7\}$. For Gaussian filtering, we evaluate Gaussian standard deviations $\sigma \in \{0.25, 0.5, 1.0, 1.5\}$. For trajectory-preserving jitter-minimization (TPJM), we sweep the regularization weight over $\lambda \in \{10^{0}, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. For the state-based Tool Hang study, we select $\lambda = 10^{-1}$.

We tune the TPJM regularization weight λ separately for post-hoc TPJM and JDSG, since the objective is applied to the final action chunk in post-hoc TPJM but to intermediate clean estimates at every denoising step in JDSG. Table 4 reports the selected task- and setting-specific values.

A.2 Evaluation Protocol

For the state-based Tool Hang study, we evaluate each setting over 1,000 episodes for each of three random seeds. For the image-based simulation experiments on Coffee, Square, and Tool Hang, we evaluate each setting over 100 episodes for each of three random seeds. We use the same 100-episode, three-seed protocol for the real-world Peg-in-Hole and Tool Hang experiments. Real-world experiments are conducted using an SO-101 robot arm with an NVIDIA RTX 5080 Laptop GPU.

A.3 Visual Perturbations

To evaluate robustness under visual distribution shifts, we create perturbed variants of the original simulation environments by modifying the appearance of selected objects. Specifically, we remove the original texture of each selected object and replace it with a uniform color equal to the average color of the original texture after reducing its lightness by 10% in the HSL color space. The task geometry, object poses, dynamics, and evaluation protocol remain unchanged.