NOISE-GATED LEARNING FROM PRIOR EXPERIENCE FOR GENERATIVE CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning generative policies via behavior cloning demands high-quality, task-specific demonstrations that can be time-consuming to collect. Prior experience, including policy rollouts and data collected on other tasks, is more readily available and contains useful information such as environment dynamics, but can harm task performance if naively imitated. We introduce **Noise-Gated Diffuser (NGD)**, a generative control framework that selectively draws supervision from prior experience beyond limited expert demonstrations. NGD jointly models actions and a compact future-state latent with independent noise levels. These define a two-dimensional noise plane that connects behavior imitation and dynamics modeling through a continuum of learning objectives. We use noise thresholds to select which regions receive supervision from each data source, with a learned embedding making the source identity explicit to the model. Across simulated and real-robot manipulation tasks with limited expert demonstrations, NGD consistently benefits from prior experience, with intermediate noise thresholds yielding the strongest average performance. On the real-robot tasks, NGD achieves nearly four times the average success rate of expert-only Diffusion Policy. Moreover, with expert demonstrations held fixed, the performance of NGD increases linearly with the amount of prior experience, highlighting its promise for scaling generative control with diverse trajectory collections.

1 INTRODUCTION

Generative control policies (Chi et al., 2023) have enabled robots to learn increasingly complex behaviors from demonstrations. However, their performance on the task of interest remains constrained by how well the expert data cover relevant states and behaviors, and collecting sufficiently diverse demonstrations can be costly and time-consuming. Meanwhile, a robot often already has access to a broader range of experience, such as exploratory interactions (Lynch et al., 2019), suboptimal policy rollouts, or demonstrations collected for other tasks (Wei et al., 2026). We refer to these trajectories collectively as *prior experience*. Such experience may not accomplish the target task, but can contain locally useful behavior patterns and record evidence of how robot actions affect the environment. As a result, naively pooling it with expert demonstrations for imitation can introduce behaviors that undermine task performance, while using expert demonstrations alone discards potentially reusable structure. It remains challenging to effectively utilize prior experience to improve generative policy learning beyond limited expert demonstrations.

Addressing this challenge requires distinguishing what each data source can reliably teach the policy. Expert demonstrations provide task-specific behavior supervision, while trajectories from either source record state transitions that support general forward- and inverse-dynamics learning. A prior trajectory can thus remain useful even when its actions are unsuitable for target-task imitation. Recent advances in jointly modeling actions and future states provide a unified framework that integrates behavior imitation with dynamics modeling (Zhu et al., 2025). Within a diffusion or flow-matching formulation, independently perturbing actions and future states at varying noise levels generalizes these objectives into a continuous range of prediction problems. We therefore investigate how to harmonize diverse learning objectives across data sources within this unified framework to improve generative control.

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.

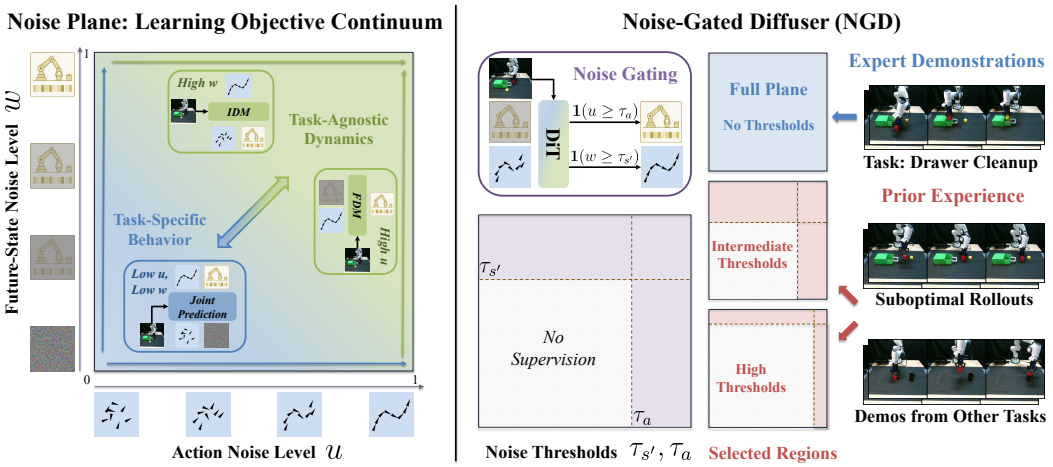


Figure 1: **Noise-Gated Diffuser.** Left: Perturbing actions and future states with independent noise levels defines a two-dimensional plane spanning a continuum of learning objectives. This continuum connects learning task-agnostic dynamics with task-specific behavior by varying noise levels of the two variables. Right: Expert demonstrations supervise the entire plane. Noise gating restricts supervision from prior experience to regions that emphasize useful transition structure while limiting the influence of potentially suboptimal behavior. NGD enables effective use of both data sources.

We introduce **Noise-Gated Diffuser (NGD)**, a lightweight flow-matching model (Lipman et al., 2023) that jointly generates actions and a compact latent representation of the resulting future state while selectively drawing supervision from prior experience through noise gating. Following prior work (Bao et al., 2023; Chen et al., 2024; Zhu et al., 2025), we corrupt the action chunk and future-state latent with independently sampled noise levels, defining a two-dimensional noise plane that spans a continuum of learning objectives (as shown in Figure 1, left). To control which learning objectives receive supervision, we employ *noise gating*, a mechanism that uses thresholds on the two noise levels to select regions of this plane. To make effective use of both data sources, expert demonstrations supervise the entire noise plane, while prior experience contributes within regions selected to extract useful structure and limit the influence of behavior unsuitable for target-task imitation. Additionally, we make the data source explicit to the policy through a learned embedding that identifies whether each trajectory chunk comes from expert demonstrations or prior experience. The selected learning objectives are applied under each source condition within a shared backbone, allowing knowledge learned from prior experience to inform expert-conditioned action generation.

Noise gating provides continuous control over which learning objectives are supervised by prior experience. Its thresholds determine the minimum information one variable must retain to serve as a conditioning input for predicting the other. Higher thresholds require cleaner conditioning inputs, concentrating prior supervision near standard forward- and inverse-dynamics objectives. Lowering the thresholds partially corrupts the conditions, requiring the model to infer action–outcome relations from incomplete information. This could encourage the learning of control-relevant representations and allow the learned structure to inform generation before either variable is fully resolved. When both variables are highly corrupted, however, prediction depends more heavily on the prior behavior distribution, including patterns that would hinder target-task completion if imitated. Intermediate thresholds therefore offer a promising route to more effective use of prior experience, extending supervision beyond clean-conditioned dynamics prediction to partially corrupted inputs while restricting direct modeling of the prior behavior distribution.

We evaluate NGD in simulated and real-robot environments with limited expert demonstrations and a larger collection of prior experience. While naively pooling the two data sources generally reduces performance in simulation, NGD achieves the highest success rate on every evaluated task. Across the two real-robot tasks, NGD substantially outperforms the expert-only baselines, achieving nearly four times the average success rate of expert-only Diffusion Policy. Intermediate noise thresholds yield the strongest average performance across all three benchmarks, showing the benefit of extending prior supervision beyond clean-conditioned dynamics while withholding it from regions that depend heavily on the prior behavior distribution. Furthermore, with expert demonstrations held

fixed, the performance of NGD increases approximately linearly with the amount of prior experience over the evaluated range, highlighting the potential of our approach to scale generative control with large, heterogeneous trajectory collections. Code and models will be released.

2 Related Works

Joint State-Action Generative Modeling. Jointly predicting future states and actions can improve robotic decision-making by integrating dynamics modeling. Janner et al. (2022), Guo et al. (2024), and Zhu et al. (2025) jointly model future observations and robot actions through diffusion. Other approaches use a shared video-action representation with separate decoders (Li et al., 2025; 2026), augment policy learning with future-representation alignment (Zheng et al., 2025; Zhao et al., 2026), or condition action generation on features learned through video prediction (Ma et al., 2026; Zhou et al., 2026). World Action Models (Cheang et al., 2024; Cen et al., 2025; Won et al., 2026; Yuan et al., 2026; Wang et al., 2026; Ye et al., 2026) combine predictive modeling with action prediction to support multitask learning and generalization. Unlike prior works that generate the future in pixel or dense feature space, NGD jointly generates actions and a compact representation of the future observation. This formulation enables dynamics modeling within a lightweight flow-matching model, while providing the interface for studying noise-gated supervision from prior experience.

Learning from Prior Experience. Previous works leverage various forms of prior experience to improve imitation learning policies. Approaches using suboptimal target-task demonstrations select or weight prior experience by quality (Xu et al., 2022; Wu et al., 2025), while cross-task approaches retrieve similar behaviors (Du et al., 2023; Nasiriany et al., 2023), sometimes with auxiliary optical-flow supervision (Lin et al., 2025). $\pi_{0.7}$ conditions on detailed metadata to learn from heterogeneous data, including demonstrations across tasks and embodiments, autonomous rollouts, and non-robot data (Physical Intelligence et al., 2026). Other works initialize policies from multi-task, cross-embodiment demonstrations (Bousmalis et al., 2024) or play data (Dinyari et al., 2020), and incorporate autonomously collected experience via goal-conditioned imitation learning. Closest to our setting, Ambient Diffusion Policy (Wei et al., 2026) also uses suboptimal data, to supervise selected low- and/or high-noise intervals of an *action-only* diffusion model. NGD jointly models states and actions with independent noise levels, using thresholds over the resulting 2D noise plane to extract useful transition structure from prior experience without directly imitating its suboptimal behavior.

Role of Noise in Diffusion Models. Noise levels provide a means of controlling the learning signal in generative models. Ambient Diffusion (Daras et al., 2024; 2025a;b) adapts training objectives and data usage across noise levels to learn from noisy and mixed-quality data. Other approaches reweight losses or adjust timestep sampling to emphasize perceptually informative noise regimes and balance optimization across noise levels (Choi et al., 2022; Hang et al., 2023; Esser et al., 2024; Hang et al., 2025). Chen et al. (2024) and Huang et al. (2026) use varying noise levels to regulate the information available for prediction. Balaji et al. (2022) instead allocates model capacity across noise regimes by training specialized denoisers for distinct intervals. In NGD, noise thresholds act as state and action loss gates during training. Each threshold specifies the minimum information that one modality must retain for prior experience to supervise prediction of the other.

3 Method

We consider learning visuomotor control for a target task from limited expert demonstrations $\mathcal{D}_e$ and a broader collection of prior trajectories $\mathcal{D}_p$. The latter may include rollouts from non-expert policies, demonstrations collected for other tasks, or a mixture of both. We do not assume target-task quality annotations for prior trajectories, as such annotations can be difficult to obtain for large, heterogeneous datasets. We aim to use prior experience to improve target-task performance beyond what limited expert demonstrations alone support, without indiscriminately imitating prior behavior.

We introduce **Noise-Gated Diffuser (NGD)**, a generative control framework that jointly predicts actions and a future state representation while selectively drawing supervision from prior experience through noise gating. In the following, we first present a lightweight design for joint state-action modeling with compact future-state latents and independent noise levels, which defines a continuum of learning objectives. Building on this formulation, we then describe how noise gating selects which learning objectives along this continuum receive supervision from prior trajectories.

3.1 JOINT STATE-ACTION MODELING WITH COMPACT LATENT FUTURE

Prior trajectories can provide supervision about how actions affect the environment even when their behavior is unsuitable for target-task imitation. To use this information alongside expert action supervision, we extend the policy's prediction target to include a future state. In contrast to approaches that generate future video sequences (Li et al., 2026; Ye et al., 2026; Zhu et al., 2025), our model generates an action chunk $a_{t:t+H-1}$ together with a *compact latent representation* of the resulting future observation, $\psi(o_{t+H})$, conditioned on the current observation $\phi(o_t)$. Here, ϕ and ψ are visual encoders that map the high-dimensional observation to compact state features. Proprioception is encoded and concatenated to the state feature when available.

For brevity, let $A = a_{t:t+H-1}$, $S = \phi(o_t)$, and $S' = \psi(o_{t+H})$. Following prior work (Zhu et al., 2025), we independently sample flow timesteps $u, w \sim \mathcal{U}(0, 1)$ for the action chunk and future-state latent, respectively, and construct perturbed inputs through linear interpolation:

$$A^{(u)} = (1-u)\epsilon_a + uA, \qquad S'^{(w)} = (1-w)\epsilon_s + wS', \tag{1}$$

where ϵ_a and ϵ_s are standard Gaussian noise. Under this convention, a flow timestep of 0 corresponds to pure noise, while 1 corresponds to clean data. Larger flow timesteps retain more information from the clean target and correspond to lower corruption levels. Conditioned on the current state S, the model takes both perturbed variables $A^{(u)}, S'^{(w)}$ and their respective timesteps u, w as inputs and predicts flow velocities $V_\theta(S, A^{(u)}, S'^{(w)}, u, w) = [V_\theta^A, V_\theta^{S'}]$. We train the model with the following flow-matching objective:

$$\begin{gathered}\mathcal{L}_A(\theta) = \left\| V_\theta^A - (A - \epsilon_a) \right\|^2, \qquad \mathcal{L}_{S'}(\theta) = \left\| V_\theta^{S'} - (S' - \epsilon_s) \right\|^2, \\ \mathcal{L}(\theta) = \mathbb{E}_{\substack{(S,A,S')\sim\mathcal{D} \\ \epsilon,u,w}} [\mathcal{L}_A(\theta) + \mathcal{L}_{S'}(\theta)] \,.\end{gathered} \tag{2}$$

Model Architecture. Representing the future state as a compact latent allows us to build on an existing diffusion or flow-matching policy backbone with minimal architectural changes. Specifically, we extend a lightweight Diffusion Transformer (DiT) policy (TRI LBM Team, 2025) with one future-state token alongside the action tokens. Shared DiT blocks process the combined sequence, with the current observation representation and the corresponding modality timestep incorporated through AdaLN-Zero. Modality-specific MLP heads predict the action and future-state velocities. For the current observation encoder ϕ, we use a randomly initialized ResNet-18, which is trained end-to-end with the policy. The future observation encoder ψ can be instantiated with either an off-the-shelf pretrained visual encoder or a task-specific encoder trained from scratch. To generate future-state latents through flow matching, we follow Zheng et al. (2026) and set the hidden dimension of the DiT blocks equal to the dimension of the future-state latent. Because the latent representation is compact, this choice keeps the backbone width and computational cost modest.

3.2 SELECTIVE SUPERVISION VIA NOISE GATING

Noise corruption in Equation 1 can be interpreted as a form of partial masking. At training, independently varying the action and future-state noise levels defines a two-dimensional noise plane (Figure 1), where each point represents a prediction problem conditioned on a different amount of information. The plane ranges from imitating recorded actions and predicting their induced future outcomes when both u and w are low, to learning chunk-level forward dynamics (high u) and inverse dynamics (high w). Expert demonstrations provide reliable supervision for both task-specific behavior learning and dynamics modeling. Prior trajectories offer abundant transition data for learning task-agnostic dynamics, but their recorded behavior is often suboptimal for direct imitation. Effectively using prior experience requires selecting objectives that extract useful structure while limiting the influence of undesirable behavior.

We propose *noise gating*, a mechanism that provides fine-grained control over objectives selected for optimization on each data source. Our key insight is that varying the corruption level of the conditioning input continuously interpolates between objectives for task-specific behavior modeling and task-agnostic dynamics prediction. As shown in Figure 1, noise gating uses thresholds on the action and future-state flow timesteps to select which prediction losses receive supervision. Each threshold specifies the minimum flow timestep a conditioning modality must reach for the prediction

loss of the other modality to be active. Let $\tau_{s'}, \tau_a$ denote the thresholds for the future-state latent and action chunk, respectively. The action velocity loss is active only when $w \geq \tau_{s'}$, while the future-state velocity loss is active only when $u \geq \tau_a$. The corresponding loss gates are

$$m_A(w; \tau_{s'}) = \mathbb{1}[w \geq \tau_{s'}], \qquad m_{S'}(u; \tau_a) = \mathbb{1}[u \geq \tau_a]. \tag{3}$$

When both thresholds are near 1, each loss receives supervision only when its conditioning modality is nearly clean, concentrating training near standard forward- and inverse-dynamics objectives. Lowering the threshold introduces partially corrupted conditioning inputs, requiring the model to extract action-outcome relations from incomplete information. As the conditioning inputs become less informative, the admitted objectives depend increasingly on the recorded behavior distribution. Noise gating thus provides continuous control over the range of learning objectives used for training.

To make effective use of both data sources, we let expert demonstrations supervise the full noise plane, while restricting supervision from prior experience to regions where useful information can be learned with minimal risk of imitating its suboptimal behavior. Let $y \in \{e, p\}$ denote the data source (i.e., $\mathcal{D}_e$ or $\mathcal{D}_p$), we can define the loss gates on each source as

$$m_A^{(y)} = \mathbb{1}\left[w \geq \tau_{s'}^{(y)}\right], \qquad m_{S'}^{(y)} = \mathbb{1}\left[u \geq \tau_a^{(y)}\right]. \tag{4}$$

We set $\tau_{s'}^{(e)} = \tau_a^{(e)} = 0$ for expert demonstrations and use tunable thresholds $\tau_{s'}^{(p)} = \tau_{s'}$ and $\tau_a^{(p)} = \tau_a$ for prior experience. In practice, we sample flow timesteps within the region selected by the thresholds to better utilize training samples from the prior dataset. Additionally, we make the data source explicit to NGD through a learned embedding, which identifies whether a trajectory chunk comes from expert demonstrations or prior data, and serves as extra conditioning information on velocity prediction $V_\theta(S, A^{(u)}, S'^{(w)}, u, w, y)$. We then apply the selected learning objectives under the corresponding source condition. The noise-gated training loss across both data sources is:

$$\mathcal{L}^{\text{NG}}(\theta) = \mathbb{E}_{\substack{(S,A,S',y)\sim\{\mathcal{D}_e,\mathcal{D}_p\} \\ \epsilon,u,w}} \left[m_A^{(y)} \mathcal{L}_A(\theta) + m_{S'}^{(y)} \mathcal{L}_{S'}(\theta) \right]. \tag{5}$$

Model Inference. Independent noise levels support different denoising orders for actions and future-state latent. During inference, we initialize both modalities from independent Gaussian noise by default and jointly denoise them through a synchronized schedule, conditioned on the expert source embedding. We then execute the generated actions in the environment.

Selecting τ. When selecting τ for prior experience, high thresholds usually provide a conservative starting point by learning clean-conditioned dynamics. Intermediate thresholds broaden this supervision to partially corrupted inputs, which may encourage control-relevant representations and allow learned action-outcome structure to inform the generation process before either modality is fully resolved. Lowering the thresholds further may admit supervision that models behavior unsuitable for the target task. We examine this trade-off empirically in the experiments.

4 Experiments

We investigate whether and how NGD enables more effective use of prior experience beyond limited expert demonstrations. Through extensive experiments, we answer the following questions: (1) How does the noise threshold affect which learning objectives receive supervision from prior experience and the resulting task performance? (2) How does noise gating compare with alternative ways of using prior experience? (3) How does NGD scale with the amount of available prior experience?

4.1 Experimental Setup

Simulated Environment. We perform evaluations over two simulated suites of robotic manipulation tasks, Robomimic (Mandlekar et al., 2021) and DexMimicGen (Jiang et al., 2025). Specifically, we include 4 single-arm manipulation tasks from Robomimic and 3 bi-manual manipulation tasks from DexMimicGen. Following the evaluation protocol from prior work (Chi et al., 2023; Ankile et al., 2025), we report the average success rate of the last 10 checkpoints, aggregated over 3 seeds. For each checkpoint, we perform evaluation on 48 different initial environment configurations for Robomimic and 20 for DexMimicGen.

Real-Robot Environment. We evaluate two tabletop manipulation tasks with a single Franka Emika Arm: (1) CoffeePod & Gear, where the robot needs to sequentially pick up a coffee pod and a gear that are randomly positioned on the table, and place them into a target blue bowl; (2) Drawer Cleanup, where the robot needs to open the drawer and place two randomly positioned tabletop objects inside. We report the success rate aggregated over 20 different initial object configurations.

Prior Data Preparation. For each task, we use a fixed limited expert dataset and a larger collection of prior trajectories throughout the experiments. In simulation, prior experience consists of unfiltered suboptimal rollouts collected by non-expert policies, with at least 7.5 times as many prior trajectories as expert demonstrations. For each real-robot task, prior experience combines suboptimal policy rollouts with demonstrations collected for other tasks and contains approximately 5 times as many trajectories as the expert dataset. We provide the exact composition and trajectory counts of each dataset in Appendix A.

Implementation Details. We instantiate NGD with an 8-layer diffusion transformer and optimize it with AdamW at a learning rate of 1e-4. As discussed in Sec. 3, we match the hidden dimension of DiT blocks with the dimension of the compact future latent. For simulated experiments, we use an exponential moving average of ϕ as the visual encoder to produce a 256-dimensional target for the future-state latent (i.e., $\psi = \phi_{\text{EMA}}$). For real-robot experiments, we encode the future observation with DINOv2 and use a 384-dimensional CLS feature for the future-state latent. Inspired by Li & He (2026), we adopt x-pred parameterization for the future-state latent and v-pred for the action chunk, which are both optimized with v-loss. At inference, we use a synchronized schedule to jointly denoise actions and the future state. Following Chi et al. (2023), we generate a chunk of 10 actions and execute the first 8 before re-planning. For real-robot experiments, we predict 24 actions per chunk and execute the first 16 before re-planning. We provide further details in Appendix B.

Baselines. When only limited expert demonstrations are provided, we evaluate: (1) Diffusion Policy (DP) (Chi et al., 2023); (2) Joint State–Action Model, which jointly predicts action chunks and a compact future-state latent. When prior experience is included, we compare NGD against baselines that do not require quality labels for the prior trajectories: (3) DP (Naive Pooling), which performs behavior cloning on the pooled expert and prior data; (4) DP (Source Conditioning), which augments the policy with a learned embedding indicating the source of each sample; (5) DP (Source Conditioning w/ CFG), which adds classifier-free guidance (CFG) to (4); (6) Joint State-Action Modeling (Naive Pooling); and (7) DWBC (Xu et al., 2022), which uses a discriminator trained to distinguish expert demonstrations from unlabeled prior samples to reweight the behavior-cloning objective. For all methods trained on both data sources, each training batch contains equal proportions of expert and prior samples. All methods follow the same evaluation protocol described above. For brevity, we use "NP" to denote Naive Pooling and "SC" to denote Source Conditioning.

Threshold Parameterization. Although our formulation supports separate gating thresholds on the future-state and action noise levels, we use a shared threshold throughout the main experiments, setting $\tau_{s'} = \tau_a = \tau$. With this choice, a single value controls how clean either conditioning variable must be for prior experience to supervise prediction of the other. It also reduces threshold selection to a one-dimensional sweep with a consistent tuning budget across tasks. In preliminary experiments, we do not observe asymmetric thresholds yield consistent improvements over the shared-threshold setting. We report these results in Appendix C.

4.2 Effect of the Noise Threshold

We first examine how the shared noise threshold τ affects task performance by determining which learning objectives receive supervision from prior experience. Increasing τ restricts supervision to regions where the conditioning variable retains more information, allowing prior trajectories to provide more reliable supervision for learning the transition structure relating actions to future states while reducing the influence of their suboptimal behavior. When τ is too high (e.g., close to 1), however, prior supervision is applied only near clean-conditioned dynamics objectives, potentially discarding useful structure that can be learned from partially corrupted inputs. Conversely, lowering τ admits a broader range of learning objectives. At $\tau = 0$, all objectives across the full noise plane receive supervision from prior experience, including those that depend strongly on the recorded prior behavior distribution and can encourage imitation of suboptimal behavior. Intermediate thresholds may therefore balance the reliability and breadth of supervision drawn from prior experience.

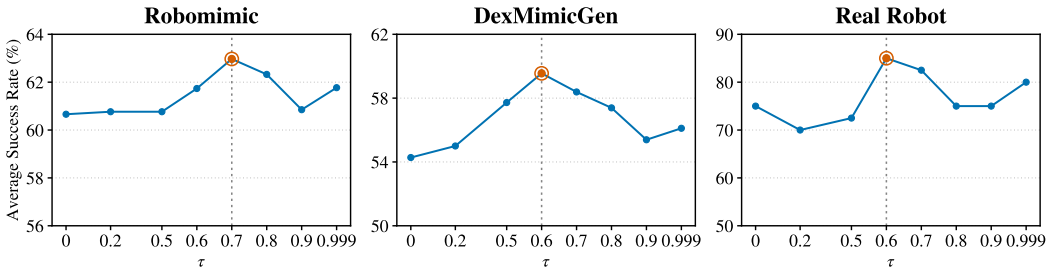


Figure 2: **Effect of the Noise Threshold.** We vary the shared threshold $\tau_{s'} = \tau_a = \tau$ over three evaluation benchmarks, and report the success rate averaged over all tasks within each benchmark. All three benchmarks achieve their best performance at an intermediate threshold.

Figure 2 illustrates this trade-off by reporting the average success rate across all tasks within each benchmark as τ varies. All three benchmarks achieve their strongest performance at an intermediate threshold. Specifically, $\tau = 0.7$ performs best on Robomimic, whereas $\tau = 0.6$ achieves the highest success rate on DexMimicGen and real-robot tasks. Both overly high and overly low thresholds lead to degraded performance. For both Robomimic and DexMimicGen, $\tau = 0$ yields the lowest average success rate. These results suggest that extending prior supervision beyond clean-conditioned dynamics is beneficial, while restricting supervision from regions that depend heavily on the prior behavior distribution remains important.

4.3 Noise Gating versus Alternative Uses of Prior Experience

Table 1 compares noise gating with alternative ways of using prior experience across the simulated and real-robot environments. We use τ_b to denote the best benchmark-level threshold from the sweep in Figure 2: 0.7 for Robomimic and 0.6 for DexMimicGen and the real-robot tasks. NGD with τ_b achieves the highest success rate across simulated and real-robot environments, demonstrating the effectiveness of selectively assigning prior supervision across the noise plane.

Prior rollouts often mix actions that advance the task with actions that lead to failure, making them unreliable targets for indiscriminate imitation. On the simulated tasks, naive pooling generally reduces performance for both DP and the Joint State–Action Model, with particularly large degradations on Transport and Tool Hang. On these same tasks, NGD with τ_b remains robust and outperforms all compared baselines. This highlights the ability of NGD to learn useful structure from prior trajectories while limiting supervision associated with suboptimal behavior. DWBC records near-zero success on Transport and Tool Hang, indicating that discriminator-based reweighting is not helpful when prior trajectories exhibit severely low behavior quality. When few expert demonstrations are available, prior rollouts can broaden the range of states observed during training and provide partially useful motions. In real-robot experiments, naive pooling improves DP on both tasks, but still degrades Joint State–Action Model performance on CoffeePod & Gear. NGD with τ_b performs best on both real-robot tasks, with a substantial improvement on Drawer Cleanup compared to the expert-only baseline. We provide qualitative results on real-robot experiments in Appendix D.

Source conditioning offers a simple way to distinguish expert and prior samples rather than treating them as a single behavior distribution. Combined with DP, it outperforms naive pooling, yet does not reliably surpass expert-only baselines, as indicated by the results on Transport and Coffee. Thus, separating the two data sources does not necessarily extract useful supervision beyond expert-only training, while adding CFG to source conditioning reduces performance on most tasks. Noise gating provides finer control over which learning objectives receive prior supervision, offering a more reliable way to retain useful structure while limiting the influence of suboptimal prior behavior.

4.4 Scaling with Prior Experience

We next examine how NGD scales with the amount of available prior experience. Figure 3 varies the fraction of the prior dataset used for training while holding the limited expert dataset fixed. Across all four tasks, NGD benefits from additional prior trajectories and reaches its highest success rate when trained with the full prior dataset. At every non-zero data fraction, NGD matches or exceeds source-conditioned DP. With full prior experience available, NGD substantially outperforms the baseline on real-robot tasks. These results show that NGD scales effectively with more prior experience in

Success Rate (%)	Expert Data Only		Expert Data & Prior Experience						
	DP	Joint State-Action Model	DWBC	DP			Joint State-Action Model		
				NP	SC	SC w/ CFG	NP	NGD w/ $\tau = 0$	NGD w/ τ_b
Can	70.4 ±3.3	73.5 ±2.3	68.0 ±2.4	69.4 ±2.4	75.6 ±1.3	75.1 ±1.3	70.8 ±1.7	75.9 ±1.1	**77.7** ±1.4
Square	49.0 ±1.1	50.6 ±3.9	23.5 ±5.4	44.7 ±1.4	51.3 ±2.4	52.5 ±2.2	45.6 ±3.8	**56.5** ±1.2	**56.7** ±2.7
Transport	71.7 ±1.6	70.8 ±6.1	1.3 ±0.8	40.6 ±1.8	71.3 ±1.2	70.6 ±1.5	42.6 ±0.8	70.6 ±2.3	**75.1** ±2.5
Tool Hang	29.9 ±2.9	32.0 ±2.3	0.0 ±0.0	16.3 ±0.6	38.5 ±1.9	36.8 ±1.5	17.4 ±0.2	39.7 ±3.6	**42.4** ±3.5
Coffee	48.7 ±3.5	48.0 ±0.9	38.7 ±4.3	46.5 ±3.1	48.2 ±5.8	47.0 ±4.8	49.0 ±2.3	44.2 ±0.8	**50.8** ±3.4
Box Cleanup	29.8 ±5.5	29.7 ±7.5	22.3 ±5.3	29.3 ±2.5	36.3 ±6.9	34.7 ±6.5	28.5 ±1.7	36.3 ±3.7	**43.5** ±6.8
Can Sorting	72.5 ±6.6	76.0 ±4.3	51.0 ±4.8	71.7 ±3.2	75.5 ±2.3	73.8 ±1.5	73.7 ±2.4	82.3 ±2.5	**84.3** ±1.3
CoffeePod & Gear	25.0	55.0	–	40.0	65.0	55.0	45.0	75.0	**85.0**
Drawer Cleanup	20.0	25.0	–	45.0	65.0	50.0	50.0	75.0	**85.0**

Table 1: **Simulated and Real-Robot Evaluation.** We perform extensive evaluation across simulated and real-robot environments. For simulated tasks, we report the average performance of the last 10 checkpoints, aggregated over 3 seeds. For real-robot tasks, we report the success rate of 20 trials with different initial object configurations. "NP" and "SC" denote naive pooling and source conditioning, respectively. τ_b is the best benchmark-level threshold identified in Figure 2.

both simulated and real-robot environments, highlighting its potential for scaling generative control with large-scale heterogeneous datasets.

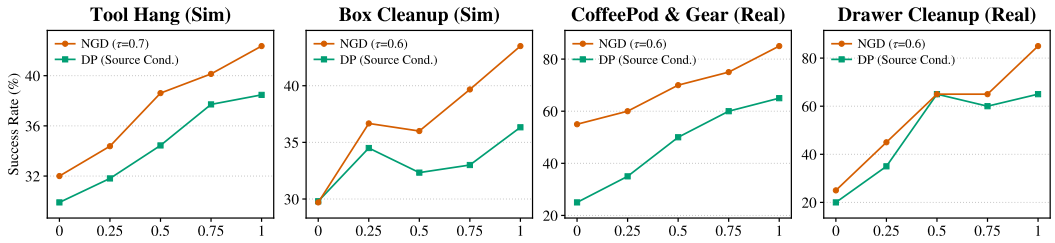


Figure 3: **Impact of Prior Data Scale.** We compare NGD with DP (Source Conditioning) given different amounts of prior experience. A fraction of 0 corresponds to expert-only training, while 1 uses the full prior dataset. We observe that NGD effectively scales with prior experience and performs best with the full prior dataset.

We further ask whether the improvement from noise gating can be similarly achieved by using filtered prior trajectories for direct imitation. In addition to the limited expert demonstrations, we compare NGD, trained with the full unfiltered prior dataset, against DP and the Joint State-Action Model trained with increasing amounts of prior rollouts filtered by oracle trajectory-level success labels. Figure 4 shows the performance of the filtered baselines as the total training set size increases, expressed as N times the size of the provided expert demonstrations. Here, $N = 1$ denotes the expert-only setting. On Box Cleanup, both filtered baselines reach the performance of NGD when the total training set grows to twice the size of the expert dataset. On Square, the filtered Joint State-Action Model becomes comparable to NGD only after the training set exceeds four times the expert-set size, while filtered DP remains below NGD even when all successful trajectories are included. Consistent with the observation from prior work (Sun & Song, 2026), we observe that training on additional filtered trajectories does not consistently lead to improvement on Transport and Tool Hang. Since trajectory-level success does not guarantee that every action within a rollout is suitable for imitation, suboptimal behavior can still present even in the successful rollouts. As a result, NGD outperforms both filtered baselines with all filtered rollouts, highlighting the robustness of noise gating to suboptimal behavior from prior experience.

4.5 Additional Ablations

Composition of Real-Robot Prior Experience. We study whether NGD can benefit from different sources of prior experience in the real-robot tasks. As shown in Table 2, adding either suboptimal policy rollouts or demonstrations collected for other tasks improves over expert-only training on both tasks. Other-task demonstrations yield higher success rates than suboptimal rollouts, while combining the two sources achieves the strongest performance. Although demonstrations from other tasks pursue different overall task goals, they provide high-quality examples of reusable action prim-

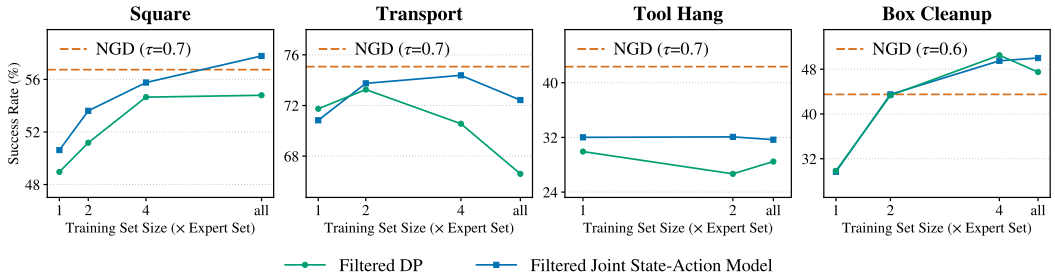


Figure 4: **Comparison with Filtered Baselines.** Solid curves show DP and the Joint State–Action Model trained by directly imitating the limited expert demonstrations together with increasing amounts of filtered prior rollouts. Dashed lines show NGD trained with the limited expert demonstrations and the full unfiltered prior dataset. The horizontal axis denotes the total number of training trajectories, including expert and filtered prior trajectories, as a multiple of the expert dataset size.

Type of Prior Experience	CoffeePod & Gear	Drawer Cleanup
Expert Only	55.0%	25.0%
w/ Suboptimal Rollouts	65.0%	60.0%
w/ Other Task Demos	75.0%	70.0%
w/ Both	85.0%	85.0%

Table 2: **Ablation on Prior Experience Composition.** NGD benefits from either source of prior experience on both real-robot tasks and achieves the strongest performance when the two sources are combined.

Task	ϕ_{EMA}	DINOv2 CLS
Tool Hang (Sim)	**39.7** ± 3.6	27.4 ± 1.1
Can Sorting (Sim)	**82.3** ± 2.5	74.5 ± 1.3
Drawer Cleanup (Real)	70.0	**75.0**

Table 3: **Ablation on Choice of Future Encoder ψ.** We find that using the task-specific encoder ϕ_{EMA} achieves higher success rates on simulated tasks, while DINOv2 enables better performance for real-robot experiments.

itives, such as precise object picking, in states underrepresented by the target-task expert data. These action primitives are valuable because accurate object interaction is essential to task success. The gains from combining both sources suggest that they provide complementary supervision and that NGD can effectively learn from heterogeneous trajectory collections.

Choice of Future-State Encoder. We compare two choices for the future-state encoder ψ in NGD under $\tau = 0$: an exponential moving average of the task-specific observation encoder, ϕ_{EMA}, and the pretrained DINOv2 CLS representation. As shown in Table 3, ϕ_{EMA} performs better on both simulated tasks, whereas DINOv2 performs better in the real-robot environment. This contrast may reflect DINOv2's large-scale pretraining on real-world images, which provides transferable features for real-robot observations but introduces a domain gap for rendered simulation inputs. These results motivate our use of ϕ_{EMA} in simulation and DINOv2 in the real-robot experiments.

5 Conclusion

We have studied how heterogeneous prior experience can improve generative control when target-task expert demonstrations are limited. We introduced NGD, a framework that selectively draws supervision from prior experience beyond limited expert demonstrations. NGD jointly models action chunks and compact future-state latents with independent noise levels, exposing a continuum of learning objectives between behavior modeling and dynamics prediction. We use noise thresholds to selectively assign supervision from prior trajectories across this continuum, allowing us to extract useful transition structure while limiting the influence of suboptimal behavior. Across simulated and real-robot evaluations, NGD outperforms alternative methods in utilizing prior experience, with intermediate thresholds yielding the strongest average performance. In particular, NGD achieves substantial improvement over expert-only baselines on real-robot tasks. Furthermore, with expert demonstrations held fixed, we observe that NGD scales effectively as prior experience grows. These results show that selectively assigning supervision across learning objectives provides an effective way to scale generative control with large, heterogeneous trajectory collections.

AI USE STATEMENT

In this work, we used generative AI tools for coding assistance, finding related works and to polish writing in some parts of the paper. The authors reviewed and tested AI-generated code for correctness and independently verified and studied the suggested references. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. We do not foresee any ethical or societal risks from this work.

REPRODUCIBILITY STATEMENT

We are committed to enabling full reproducibility of our work. We have included comprehensive implementation and evaluation details throughout the paper and in the Appendix. We will release code and data to reproduce all of the experiments in this work to facilitate future research upon publication.

REFERENCES

Lars Ankile, Zhenyu Jiang, Rocky Duan, Guanya Shi, Pieter Abbeel, and Anusha Nagabandi. Residual off-policy rl for finetuning behavior cloning policies, 2025. URL https://arxiv.org/abs/2509.19301.

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *International Conference on Machine Learning*, 2023.

Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Manon Devin, Alex X. Lee, Maria Bauza Villalonga, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, Antoine Laurens, Claudio Fantacci, Valentin Dalibard, Martina Zambelli, Murilo Fernandes Martins, Rugile Pevceviciute, Michiel Blokzijl, Misha Denil, Nathan Batchelor, Thomas Lampe, Emilio Parisotto, Konrad Zolna, Scott Reed, Sergio Gómez Colmenarejo, Jonathan Scholz, Abbas Abdolmaleki, Oliver Groth, Jean-Baptiste Regli, Oleg Sushkov, Thomas Rothörl, Jose Enrique Chen, Yusuf Aytar, David Barker, Joy Ortiz, Martin Riedmiller, Jost Tobias Springenberg, Raia Hadsell, Francesco Nori, and Nicolas Heess. Robocat: A self-improving generalist agent for robotic manipulation. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=vsCpILiWHu.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

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. In *Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11462–11471. IEEE, 2022.

Giannis Daras, Alex Dimakis, and Constantinos Costis Daskalakis. Consistent diffusion meets tweedie: Training exact ambient diffusion models with noisy data. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10091–10108. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/daras24a.html.

Giannis Daras, Yeshwanth Cherapanamjeri, and Constantinos C Daskalakis. How much is a noisy image worth? data scaling laws for ambient diffusion. In *International Conference on Learning Representations*, volume 2025, pp. 28165–28193, 2025a.

Giannis Daras, Adrian Rodriguez-Munoz, Adam Klivans, Antonio Torralba, and Constantinos Costis Daskalakis. Ambient diffusion omni: Training good models with bad data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=MVYz4GmcUH.

Rostam Dinyari, Pierre Sermanet, and Corey Lynch. Learning to play by imitating humans. *arXiv preprint arXiv:2006.06874*, 2020.

Maximilian Du, Suraj Nair, Dorsa Sadigh, and Chelsea Finn. Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu (eds.), *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi: 10.15607/RSS.2023.XIX.011. URL https://doi.org/10.15607/RSS.2023.XIX.011.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7407–7417. IEEE, 2023.

Tiankai Hang, Shuyang Gu, Jianmin Bao, Fangyun Wei, Dong Chen, Xin Geng, and Baining Guo. Improved noise schedule for diffusion training. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4796–4806. IEEE, 2025.

Wen Huang, Haoran Sun, Yongjian Guo, Yunxuan Ma, Haoran Li, Jing Long, Zhouying Mo, Zhong Guan, Yucheng Guo, Shuai Di, et al. Noisegate: Learning per-latent timestep schedules as information gating in world action models. *arXiv preprint arXiv:2605.07794*, 2026.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control, 2026. URL https://arxiv.org/abs/2601.21998.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model, 2025. URL https://arxiv.org/abs/2503.00200.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise, 2026. URL https://arxiv.org/abs/2511.13720.

Li-Heng Lin, Yuchen Cui, Amber Xie, Tianyu Hua, and Dorsa Sadigh. Flowretrieval: Flow-guided data retrieval for few-shot imitation learning. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4084–4099. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/lin25a.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on Robot Learning*, 2019.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. Dit4dit: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026.

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. In *arXiv preprint arXiv:2108.03298*, 2021.

Soroush Nasiriany, Tian Gao, Ajay Mandlekar, and Yuke Zhu. Learning and retrieval from prior data for skill-based imitation learning. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2181–2204. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/nasiriany23a.html.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. Technical report, Physical Intelligence, 2026.

Zhanyi Sun and Shuran Song. Latent policy barrier: Learning robust visuomotor policies by staying in-distribution. *Advances in Neural Information Processing Systems*, 38:174280–174305, 2026.

TRI LBM Team. A careful examination of large behavior models for multitask dexterous manipulation, 2025. URL https://arxiv.org/abs/2507.05331.

Siyin Wang, Junhao Shi, Zhaoyang Fu, Xinzhe He, Feihong Liu, Chenchen Yang, Yikang Zhou, Zhaoye Fei, Jingjing Gong, Jinlan Fu, et al. World action models: The next frontier in embodied ai. *arXiv preprint arXiv:2605.12090*, 2026.

Adam Wei, Nicholas Pfaff, Thomas Cohn, Arif Kerem Dayı, Constantinos Daskalakis, Giannis Daras, and Russ Tedrake. Ambient diffusion policy: Imitation learning from suboptimal data in robotics. *arXiv preprint arXiv:2606.12365*, 2026.

John Won, Kyungmin Lee, Huiwon Jang, Dongyoung Kim, and Jinwoo Shin. Dual-stream diffusion for world-model augmented vision-language-action model. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=NWApJP7UKT.

Kun Wu, Ning Liu, Zhen Zhao, Di Qiu, Jinming Li, Zhengping Che, Zhiyuan Xu, and Jian Tang. Learning from imperfect demonstrations with self-supervision for robotic manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16899–16906, 2025. doi: 10.1109/ICRA55743.2025.11127918.

Haoran Xu, Xianyuan Zhan, Honglei Yin, and Huiling Qin. Discriminator-weighted offline imitation learning from suboptimal demonstrations. In *International Conference on Machine Learning*, pp. 24725–24742. PMLR, 2022.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026. URL https://arxiv.org/abs/2602.15922.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Han Zhao, Jingbo Wang, Wenxuan Song, Shuai Chen, Yang Liu, Yan Wang, Haoang Li, and Donglin Wang. Frappe: Infusing world modeling into generalist policies via multiple future representation alignment. *arXiv preprint arXiv:2602.17259*, 2026.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0u1LigJaab.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

Pengfei Zhou, Shengcong Chen, Di Chen, Jiaxu Wang, Rongjun Jin, Bingwen Zhu, Yike Pan, Songen Gu, Kuanning Wang, Shufeng Nan, et al. τ_0-wm: A unified video-action world model for robotic manipulation. *arXiv preprint arXiv:2606.01027*, 2026.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

A Prior Data Preparation

For each task, all methods are trained on the same limited expert dataset $\mathcal{D}_e$ and the same prior dataset $\mathcal{D}_p$. Table A1 provides a full list of expert and prior dataset sizes across benchmarks. The expert dataset is a fixed subset of each task's source demonstrations provided by the benchmark; the remaining demonstrations are held out and never used for training.

Simulation. Prior experience consists of rollouts from a Diffusion Policy trained only on the limited expert dataset of the same task. While this policy trains, we periodically evaluate it and record every evaluation episode regardless of its outcome, until a fixed budget of 800 (Can and Square) or 600 (all other tasks) trajectories is reached. The prior data therefore covers policies at different stages of training and mixes successful and failed attempts; we apply no filtering or relabeling on the collected trajectories. For Robomimic, we additionally perturb the executed actions with Gaussian noise ($\sigma = 0.05$) to diversify the collected behavior.

Real Robot. Prior experience combines suboptimal policy rollouts on the target task with demonstrations collected for other tasks, and contains approximately five times as many trajectories as the expert dataset (Table A1). Similar to the simulated setting, we use a non-expert Diffusion Policy to collect rollouts from the target task, with randomly initialized object configurations. For each real-robot task, we additionally collect demonstrations of eight other tasks in the same workspace, using the same robot and camera setup as the target task. Each task uses a different subset of objects, drawing on objects from the target task as well as new ones, but no task reproduces the full target task. For CoffeePod & Gear, the tasks range from grasping the coffee pod or the gear alone, to placing one of them into the blue bowl at a different position than in the target task, to scenes that add a black mug. For Drawer Cleanup, they range from grasping a single object, to opening the drawer and placing a novel object (a green ball, a blue cube, or a black mug) inside, to manipulating the target task's red cube and yellow ball without the drawer.

			Prior Experience		
Suite	Task	Expert Demos	Suboptimal Rollouts	Other-Task Demos	Total
Robomimic	Can	20	800	–	800
	Square	50	800	–	800
	Transport	40	600	–	600
	Tool Hang	80	600	–	600
DexMimicGen	Coffee	51	600	–	600
	Box Cleanup	51	600	–	600
	Can Sorting	21	600	–	600
Real Robot	CoffeePod & Gear	25	61	69	130
	Drawer Cleanup	25	60	64	124

Table A1: Number of trajectories in the expert dataset $\mathcal{D}_e$ and the prior dataset $\mathcal{D}_p$, with prior experience broken down by type. In simulation, prior experience consists only of suboptimal rollouts; on the real robot, it additionally includes demonstrations collected for other tasks.

B Implementation Details

We list detailed hyperparameters for NGD in Table A2, and provide additional implementation details below:

Baselines. When applicable, we keep the observation encoder, optimizer, learning-rate schedule, batch composition, number of gradient steps, EMA, and evaluation protocol across all methods. To facilitate fair comparison, our DP implementation uses the same diffusion transformer as NGD with 8 blocks and a hidden dimension of 256, while predicting only the action chunk with flow matching. DP (SC) adds a learned source embedding to the conditioning and samples with the expert embedding. DP (SC w/ CFG) additionally drops out the expert label for 10% of expert samples during training and samples with guidance $v = (1+\omega)\, v_{\text{expert}} - \omega\, v_{\text{null}}$, $\omega = 1$. The Joint State–Action Model reuses the diffusion transformer backbone and jointly optimizes state and action losses; under Naive Pooling (NP), it treats prior trajectories as imitation targets indiscriminately and optimizes

both losses across the full noise plane. DWBC (Xu et al., 2022) follows the released implementation, including its two-layer, 256-unit tanh-Gaussian actor, its discriminator, and its policy-correction terms, with $\alpha = 7.5$ and $\eta = 0.5$. The discriminator is updated every 100 policy steps. Because the original method takes low-dimensional states, we give the actor and the discriminator separate copies of our observation encoder. DWBC predicts a single action per step, uses Adam with a constant learning rate of 1e-4 and a policy weight decay of 0.005, and trains for 300k steps with 64 expert and 64 prior samples per batch.

Hyperparameter	Simulation	Real Robot
DiT blocks	8	8
Attention heads	8	8
Hidden dimension	256	384
Timestep embedding dimension	128	128
Observation encoder ϕ	ResNet-18	ResNet-18
Future-state encoder ψ	ϕ_{EMA}	DINOv2-S (frozen)
Future-state latent dimension	256	384
Optimizer	AdamW	AdamW
Learning rate	10^{-4}	10^{-4}
Learning-rate schedule	cosine	cosine
Weight decay	10^{-5}	10^{-5}
Batch size	128	128
Gradient steps	300k	300k
EMA decay	0.999	0.999

Table A2: Hyperparameters of NGD.

Inference. For generative control models, we initialize the action chunk and the future-state latent from independent Gaussian noise and integrate both with the Euler method on a synchronized schedule ($u = w$ at every step), conditioned on the expert source embedding when available. For the Joint State–Action Model, we perform 25 integration steps in both simulated and real-robot experiments.

C Asymmetric τ Experiments.

Our formulation allows separate thresholds for the two conditioning modalities: $\tau_{s'}$ sets how clean the future-state latent must be before prior data supervises the action loss, and τ_a sets how clean the action chunk must be for prior data to supervise the future-state loss. In the main experiments, we use a shared threshold $\tau_{s'} = \tau_a = \tau$, and apply the benchmark-level threshold $\tau = 0.7$ for Robomimic and $\tau = 0.6$ for DexMimicGen. Here we test how decoupling thresholds affect the task performance.

Protocol. We run the asymmetric sweep on three tasks: Tool Hang, Box Cleanup, and Coffee. For each task, we start from the shared threshold and vary one threshold while holding the other at the shared value, setting the varied threshold to 0, 0.3, or 0.9. This yields two one-dimensional sweeps per task that intersect at the shared setting, for 6 asymmetric configurations per task and 18 in total. Setting a threshold to 0 removes the corresponding gate entirely, so that prior data supervises that loss over the full noise plane. All other settings, including the training and evaluation protocol, are identical to the main experiments, and we report the success rate averaged over the last 10 checkpoints and 3 seeds.

Results. Figure A1 shows the results. No asymmetric configuration consistently outperforms the shared threshold. On Box Cleanup, all six asymmetric configurations perform worse than the shared threshold (43.5%), by 4.3 to 11.7 points. On Tool Hang, all six stay within 2.1 points of the shared threshold (42.4%), which is smaller than the variation across seeds. On Coffee, only one configuration exceeds the shared threshold: removing the gate on the action loss ($\tau_{s'} = 0$) reaches 52.0% compared with 50.8%, a difference within the seed variation. The same configuration yields no gain on Tool Hang and lowers performance by 4.3 points on Box Cleanup. Removing the gate on the future-state loss ($\tau_a = 0$) lowers performance on all three tasks, by up to 10.3 points. Overall, decoupling the thresholds does not yield consistent improvements, while it doubles the number of

values to tune. We therefore use a single shared threshold throughout the main experiments, which reduces threshold selection to a one-dimensional sweep.

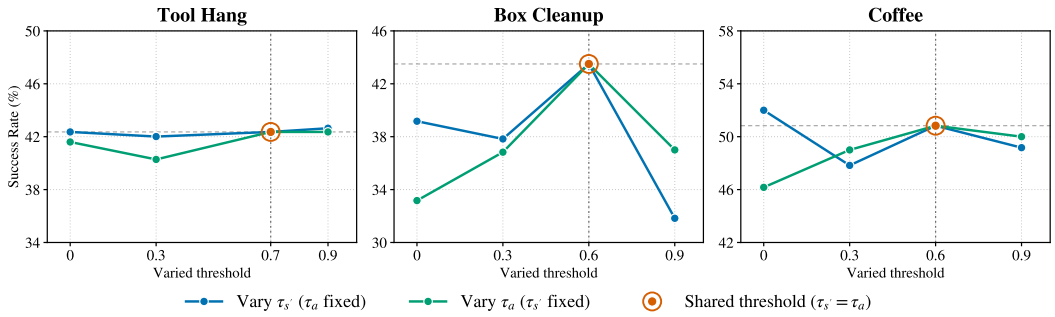


Figure A1: **Asymmetric threshold sweep.** Starting from the shared threshold (orange; $\tau = 0.7$ for Tool Hang and $\tau = 0.6$ for Box Cleanup and Coffee), we vary either $\tau_{s'}$ (blue) or τ_a (green) while holding the other at the shared value. The dashed horizontal line marks the success rate of the shared setting. Success rates are averaged over the last 10 checkpoints and 3 seeds.

D Qualitative Results on Real Robot

We provide a qualitative comparison between DP, DP (SC), NGD with $\tau = 0$, and NGD with an intermediate threshold on the same initial configurations in Figure A2 and Figure A3. We observe that NGD with an intermediate threshold successfully completes the task, while other baselines either fail to approach the target object or execute imprecise grasps.

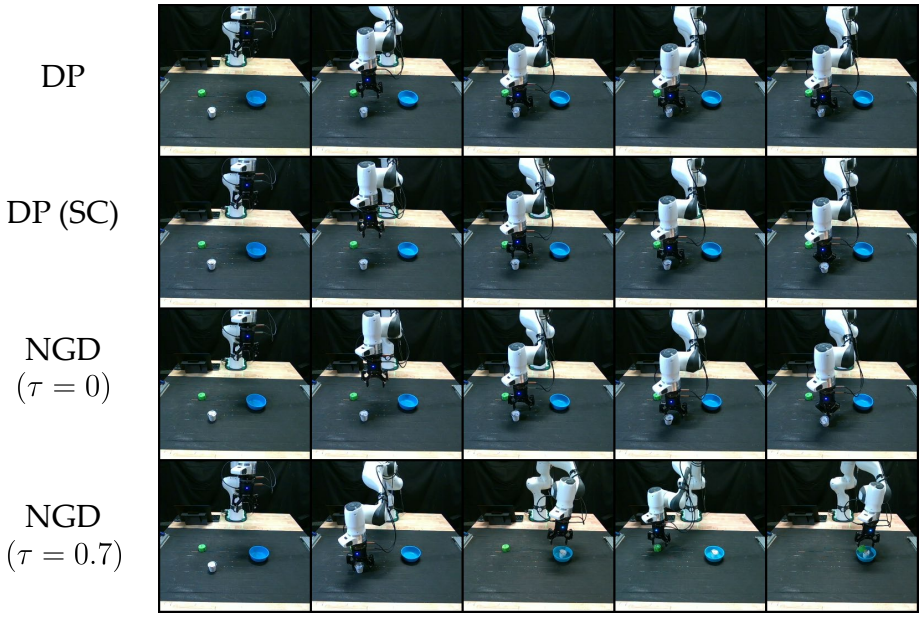


Figure A2: **Qualitative Results on CoffeePod & Gear**

E Contribution of Action and Future-State Supervision

Noise gating lets prior data supervise two prediction problems: predicting actions from a sufficiently clean future state, and predicting the future state from a sufficiently clean action chunk. We examine how much each contributes to the performance of NGD. To this end, we train two variants in which

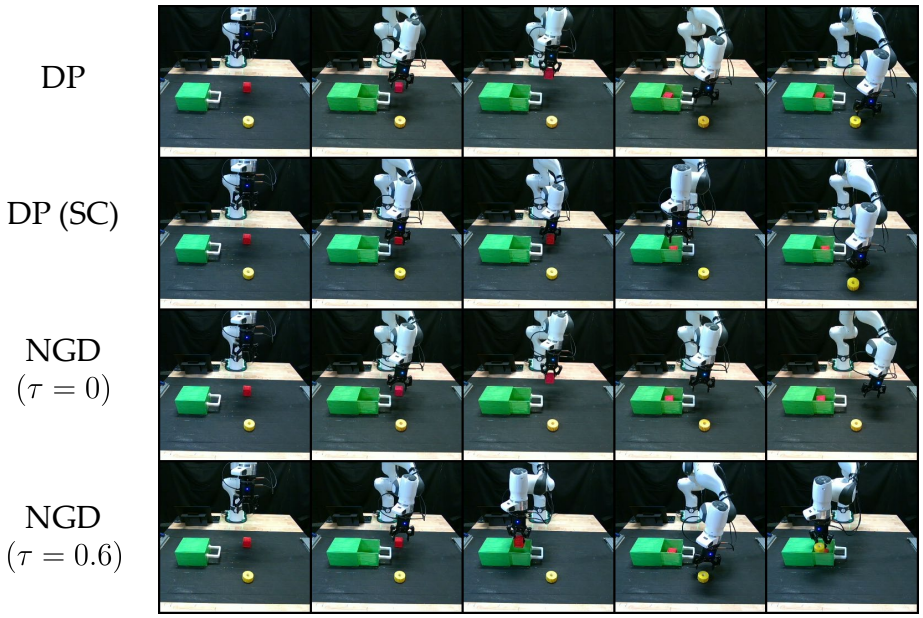


Figure A3: **Qualitative Results on Drawer Cleanup**

prior data supervises only one of the two losses within its gated region: *State Loss only*, where prior data supervises only the future-state loss, and *Action Loss only*, where prior data supervises only the action loss. In both variants, expert demonstrations still supervise both losses over the full noise plane, and all other settings, including the threshold, match NGD. We evaluate on Can, Square, Tool Hang, and Box Cleanup with 3 seeds each.

As shown in Figure A4, both variants perform worse than NGD on all four tasks. On average, NGD reaches 55.1%, compared with 51.5% for Action Loss only and 47.3% for State Loss only. Between the two variants, Action Loss only outperforms State Loss only, indicating that action supervision from prior data provides the larger share of the benefit. Neither form of supervision from prior data recovers the full performance of NGD on its own, which suggests that the two are complementary and that NGD benefits from drawing both action and future-state supervision from prior experience.

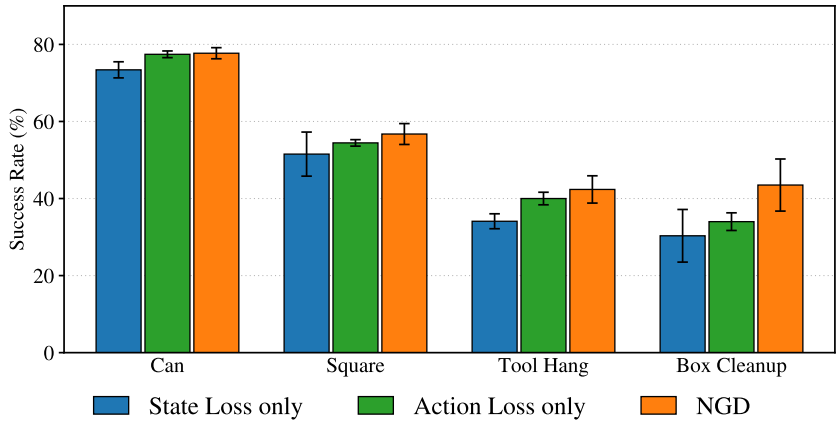


Figure A4: **Contribution of action and future-state supervision from prior data.** In *State Loss only* and *Action Loss only*, prior data supervises only the future-state or the action loss, respectively, while expert data supervises both losses in all variants. Bars show the mean success rate over 3 seeds, and error bars show the standard deviation.