DSSP: Diffusion State Space Policy with Full-History Encoding

Zhiyuan Guan[1] **Jianshu Hu**[1] **Han Fang**[1] **Yunpeng Jiang**[1]
Yize Huang[2] **Shujia Li**[1] **Xiao Li**[2] **Yutong Ban**[1]*

[1]Global College, Shanghai Jiao Tong University
[2]School of Mechanical Engineering, Shanghai Jiao Tong University

Abstract

Diffusion-based imitation learning has shown strong promise for robot manipulation. However, most existing policies condition only on the current observation or a short window of recent observations, limiting their ability to resolve history-dependent ambiguities in long-horizon tasks. To address this, we introduce DSSP, a history-conditioned Diffusion State Space Policy that enables efficient, full-history conditioning for robot manipulation. Leveraging the continuous sequence modeling properties of State Space Models (SSMs), our history encoder effectively compresses the entire observation stream into a compact context representation. To ensure this context preserves critical information regarding future state evolution, the encoder is optimized with a dynamics-aware auxiliary training objective. This high-level context representation is then seamlessly fused with recent state observations to form a hierarchical conditioning mechanism for action generation. Furthermore, to maintain architectural consistency and minimize GPU memory overhead, we also instantiate the diffusion backbone itself using an SSM. Extensive experiments across simulation benchmarks and real-world manipulation tasks show that DSSP achieves state-of-the-art performance with a significantly smaller model size, demonstrating superior efficiency of the hierarchical conditioning in capturing crucial information as the history length increases.

1 Introduction

Deploying robots in complex, unstructured environments requires policies capable of reasoning over multi-step tasks and high-dimensional sensory inputs. Imitation learning [33, 64, 66] has emerged as a central paradigm for acquiring such skills from expert demonstrations. Among recent imitation learning approaches, diffusion policies [8] have shown strong potential for modeling complex and multi-modal action distributions. Recent works further improve diffusion-based policies through expressive denoising backbones [58], multi-modal conditioning representations [63, 38], and improved action generation formulations [65, 47]. However, most policies still rely on short-context conditioning, predicting future actions from the current observation or a short observation window.

This short-context conditioning becomes a critical bottleneck in long-horizon manipulation. In multi-stage tasks, visually similar observations may correspond to different task progress, such as whether an object has already been moved, whether a container has been filled, or which object was previously placed in a buffer location. The desired actions therefore depend not only on local visual details, but also on sparse historical events that reveal task progress and previous interactions. Without access to such history, policies may fail to disambiguate temporally aliased states, leading them to ignore completed subgoals or undo previous actions.

*Corresponding author: yban@sjtu.edu.cn.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

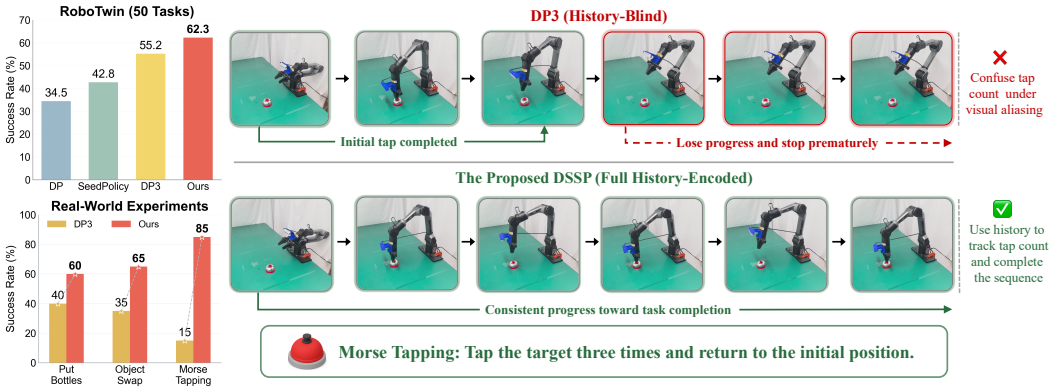


Figure 1: The proposed **DSSP** leverages full-history context to resolve visual aliasing when history-blind baselines lose track of task progress, enabling consistent execution in long-horizon tasks. DSSP achieves superior success rates across both simulation tasks and real-world experiments.

Existing methods have attempted to address the limitations of short-context policies by adding temporal context. A direct solution extends the observation horizon or feeds longer observation histories into the policy, but this increases memory and inference cost and may introduce redundant visual inputs or spurious correlations [55, 52, 67, 40]. Attention-based and multi-frame VLA methods provide more flexible access to past observations, yet remain costly because they explicitly process additional context as the horizon grows [34, 23, 30]. To reduce this overhead, recent works therefore explore more compact memory forms, including keyframes [40, 54], visual traces [67], point tracks [5], or recurrent latent states [68]. However, compactness does not ensure task relevance: compressed memories may discard progress-critical events or focus on nuisance visual changes. These limitations motivate a streaming history mechanism that efficiently compresses long-horizon context while preserving events predictive of future state.

To fulfill these requirements, State Space Models (SSMs) offer a well-suited architecture for history encoding. Their recurrent formulation maintains a compact hidden state that is updated online as new observations arrive, enabling linear-time aggregation of long histories without re-encoding the entire past. Moreover, modern selective SSMs such as Mamba [14] make the state update input-dependent, allowing the model to adaptively decide what to preserve or discard. This content-dependent memory update is particularly suitable for long-horizon manipulation, where visually similar observations may correspond to different task stages and require different actions. Building on this insight, we propose DSSP, a full-history encoding diffusion state-space policy for long-horizon manipulation. Following the use of SSMs in diffusion-based robot policies [3, 24, 61], DSSP instantiates the action denoiser with an SSM backbone, but conditions it on a compact online memory compressed from the full multi-modal observation stream together with immediate state information. A dynamics-aware auxiliary objective encourages this memory to preserve cues predictive of future states. We further decouple task conditioning from diffusion-step modulation by injecting observation-derived conditions as a prefix and timestep information through Adaptive Layer Normalization (AdaLN) [42].

Our contributions are summarized as follows:

- **Hierarchically conditioned state space policy.** We design a policy framework instantiated with a state space model and a hierarchical conditioning mechanism. Specifically, learned context representations and immediate state representations are fused via prefix conditioning, while the diffusion timestep is decoupled and injected independently through AdaLN.

- **Full-history context learning.** We propose a causal state-space history encoder that maintains a compact context representation by recurrently integrating incoming multi-modal observations. Together with a dynamics-aware auxiliary objective, the encoder summarizes the full observation history while retaining task-relevant events predictive of future states.

- **Comprehensive evaluation.** We evaluate DSSP on extensive simulated and real-world long-horizon manipulation tasks, showing improved success rates, effective history summarization, perturbation robustness, and efficient inference with increasing history length.

2 Related Work

Imitation Learning for Robot Manipulation. Imitation learning enables policies to acquire skills from expert demonstrations. While early behavior cloning methods directly predict actions from observations, recent approaches improve closed-loop consistency by modeling temporally extended or structured actions, such as action chunks [66] and multimodal action representations [46, 32]. Diffusion-based policies further generate continuous action trajectories via conditional denoising [8], with 3D Diffusion Policy extending this framework to point-cloud observations [63]. Recent variants improve diffusion policies in perception [38], efficiency [11, 47], and generation quality [58, 39]. Our method follows this paradigm, using conditional denoising for action generation.

Long-Horizon and History-Aware Policy Learning. Most existing robot policies condition action prediction on the current observation or a short window of recent observations, which is often sufficient for short-horizon or near-Markovian tasks. In long-horizon manipulation, resolving temporal ambiguity requires observation history. However, naively stacking past observations often degrades performance due to redundancy and causal confusion [18, 55, 56, 49]. Recent methods therefore explore more selective uses of history, such as regularizing long-context policies with past-token prediction [52], selecting task-relevant keyframes from history [40], using demonstration trajectories as in-context prompts [57], compressing long observation histories into an evolving latent state [15, 68], leveraging history sampling and progress prediction for repetitive manipulation [54], or incorporating previous action trajectories for stateful diffusion-based control [36]. In contrast, our method learns a compact, dynamics-aware history context that preserves future-relevant information for effective long-horizon policy learning.

State Space Models for Robot Policies. State space models have recently shown strong potential for sequence modeling by maintaining an evolving latent state over time. Mamba [14, 9, 31] further improves this paradigm with selective state updates and hardware-aware parallel scan, enabling efficient long-sequence modeling. Inspired by these properties, recent robotic learning methods have adopted Mamba as a backbone for imitation learning and diffusion-based action generation [41, 24, 3, 25], or as a recurrent encoder for long observation histories in temporally ambiguous tasks [53, 68]. These findings demonstrate the effectiveness of Mamba for modeling robot trajectories and temporal dependencies. In this paper, we introduce a unified Mamba-based policy that integrates dynamics-aware history encoding with diffusion-based action generation.

3 Preliminaries

Problem Formulation. We formulate long-horizon 3D manipulation as a Partially Observable Markov Decision Process (POMDP), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \Omega, \mathcal{O})$. At any time step t, the agent receives an observation $o_t \in \Omega$ generated from the unobserved true state $s_t \in \mathcal{S}$ via the observation function $\mathcal{O}(o_t \mid s_t) = P(o_t \mid s_t)$. The environment evolves according to transition dynamics $\mathcal{T}(s_{t+1} \mid s_t, a_t) = P(s_{t+1} \mid s_t, a_t)$ given action $a_t \in \mathcal{A}$. Since o_t is insufficient to infer s_t, we define the interaction history as $h_t = (o_0, a_0, \ldots, a_{t-1}, o_t) \in \mathcal{H}$, where $\mathcal{H}$ denotes the full history space. Our goal is to learn a history-dependent policy $\pi_\theta(a_t \mid h_t)$ imitating an expert policy π_E with expert trajectories $\mathcal{D}_E = \{\zeta_1, \ldots, \zeta_N\}$, where each trajectory is a sequence $\zeta = (o_0, a_0, \ldots, o_T)$. Due to the page limit, we include a detailed introduction to diffusion policy and SSM in Appendix A.

Observation Aliasing. A non-injective observation function alone does not imply that history is useful for imitating the expert: the aliased states may require the same action, or the expert may use information that is absent from the recorded history. We therefore use the following decision-relevant notion. In the definition, capital letters denote random variables under the expert trajectory distribution, and O_t is a component (and hence a measurable function) of H_t.

Definition 1 (History-resolvable observation aliasing)**.** *History-resolvable observation aliasing occurs when the conditional expert-action law given the available history is not almost surely deter-*

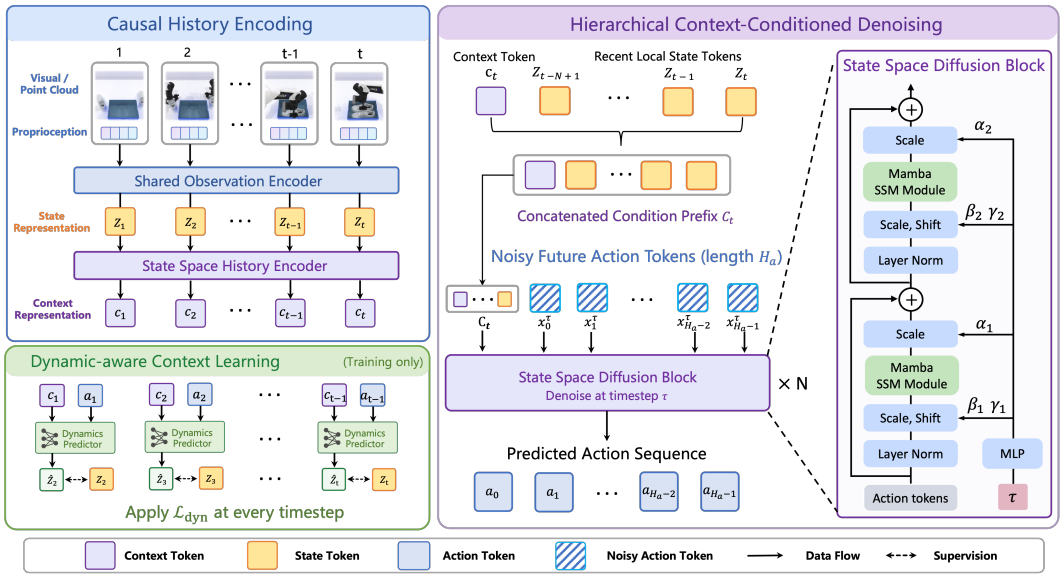


Figure 2: **Overview of DSSP.** DSSP summarizes past multi-modal observations into a compact context token using a state-space history encoder. A dynamics-aware auxiliary loss encourages this token to retain historical information predictive of future state evolution. The learned context token is then combined with recent state tokens as a hierarchical prefix condition for a state-space diffusion denoiser to generate future actions.

mined by the current observation:

$$P_E(A_t \in \cdot \mid H_t) \neq P_E(A_t \in \cdot \mid O_t) \quad \textit{on a set of histories of positive probability.} \tag{1}$$

Equivalently,

$$I_E(A_t; H_t \mid O_t) := \mathbb{E}[\mathrm{KL}(P_E(A_t \in \cdot \mid H_t) \,\|\, P_E(A_t \in \cdot \mid O_t))] > 0. \tag{2}$$

This definition concerns the information actually available to the learned policy. If the expert uses an unrecorded variable that cannot be predicted from H_t beyond O_t, then (1) does not hold and no strict benefit from history is claimed. Conversely, when it does hold, a history-conditioned diffusion policy can represent the distinct conditional action distributions rather than their observation-conditioned mixture; appendix E makes this statement precise at the level of optimal population diffusion loss.

4 Method

To effectively leverage history information within a robot manipulation policy, we propose diffusion state space policy (DSSP), a full-history conditioned diffusion policy (illustrated in Figure 2). Our approach instantiates the diffusion model with an SSM backbone and a dual-level conditioning mechanism that integrates high-level context with low-level state representations. The context representation serves as a compact encoding of historical multi-modal observations. To ensure this representation captures temporal dependencies, we introduce an auxiliary dynamics-aware loss focused on future state prediction. Section 4.1 introduces long-horizon context learning, including causal history encoding and dynamics-aware context learning; and Section 4.2 formalizes the hierarchical conditioning mechanism and diffusion-based action generation policy.

4.1 Long-Horizon Context Learning

Long-horizon manipulation requires historical context to resolve task-level ambiguities; otherwise, policies may experience perceptual aliasing, such as getting trapped in loops during repetitive wiping tasks. Because existing conditioning approaches like observation stacking or keyframe extraction often suffer from computational redundancy or overlook subtle causal events, we introduce a learnable history encoder that compresses the full history into a compact, task-relevant context representation.

To this end, our design is guided by two key principles. First, we employ a causal SSM to efficiently process the streaming observation history and extract a temporally integrated memory representation. Second, to ensure that this representation preserves historical cues useful for future decisions, we shape the latent space with a dynamics-aware auxiliary objective.

Causal History Encoding. We first introduce how to build the context representation for history encoding. Let o_t denote the multi-modal observation at timestep t, comprising visual inputs and robot proprioceptive states. We project this raw observation into a state representation z_t:

$$z_t = E_{\text{obs}}(o_t), \tag{3}$$

where E_{obs} contains parallel visual and proprioceptive encoders. To capture long-horizon temporal dependencies, a causal history encoder G processes the sequence of step-wise state representations $z_{1:t}$ into the temporally-integrated context representation $\tilde{z}_{1:t}$:

$$\tilde{z}_{1:t} = G_\psi(z_{1:t}). \tag{4}$$

A critical design choice is the architecture of the history encoder G, which must produce a compact yet effective history representation. This design must satisfy two primary criteria: (1) maintaining a scalable computation when processing extended temporal horizons, and (2) distilling a representation that selectively retains salient events rather than passively memorizing the entire observation stream.

To meet these requirements, we instantiate the history backbone using a State-Space Model (SSM) and define the context representation c_t as the final output token of the encoded sequence:

$$c_t = \tilde{z}_t. \tag{5}$$

SSMs support streaming histories with linear-time complexity, and we use Mamba as the history encoder for input-dependent selective updates. This allows the encoder to filter redundant observations while preserving sparse task-relevant events, such as object-state transitions, contact changes, or subgoal completion. The resulting context c_t serves as a compressed memory for action generation.

Dynamics-Aware Context Learning. Compressing the observation history into c_t does not by itself guarantee that the representation preserves historical information relevant to future decisions. For long-horizon manipulation, the context representation should encode not only past observations, but also history-dependent cues that are predictive of future state evolution. To encourage this property, we introduce a dynamics-aware auxiliary objective applied to the context representation at each timestep. Given the context representation c_t and the executed action a_t at time t, we train a lightweight dynamics predictor g_ϕ to predict the next state representation:

$$z_{t+1} = E_{\text{obs}}(o_{t+1}), \qquad \hat{z}_{t+1} = g_\phi(c_t, a_t). \tag{6}$$

We supervise this prediction with a cosine similarity loss:

$$\mathcal{L}_{\text{dyn}}(\psi, \phi) = \mathbb{E}_{\zeta \sim \mathcal{D}_E, t \sim [0, T-1]} \left[1 - \cos\left(g_\phi(c_t, a_t), \text{sg}(z_{t+1}) \right) \right], \tag{7}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operation, and the expectation is taken over expert trajectories $\zeta \sim \mathcal{D}_E$ and trajectory timesteps t. This objective encourages the context representation c_t to retain action-relevant historical information by requiring it to support prediction of the next state under the executed action. The learned context representation therefore provides a more informative conditioning signal for downstream action generation. Additional runtime diagnostics further demonstrate that the learned dynamics predictor captures action-dependent state transitions in Appendix D.4.

4.2 Hierarchical Context-Conditioned Denoising

Given the learned history representation, the remaining question is how to inject it into a policy. Long-horizon manipulation requires both long-term task-progress information and recent local observations: the former resolves visual ambiguity across task stages, while the latter provides fine-grained control cues. Therefore, we propose a diffusion policy utilizing a hierarchical conditioning mechanism that integrates the context representation with immediate state observations. We organize these signals as a causal prefix to the noisy action sequence, allowing long-term context to contextualize recent observations before being propagated to the action tokens during denoising. For

efficiency, we instantiate the diffusion backbone with a compact SSM, yielding a lightweight policy while maintaining a unified SSM-based architecture for both history encoding and action denoising.

Hierarchical Prefix Conditioning. To preserve long-term progress information and local control cues, we condition the denoising model on a hierarchical prefix. Let $x_0 = \mathbf{a}_{t:t+H_a-1}$ denote the clean future action trajectory, and let x_τ denote its noisy version at diffusion step τ. We construct the condition sequence as

$$C_t = [c_t, z_{t-N+1}, \ldots, z_t], \tag{8}$$

where c_t is the context representation produced by the history encoder, $z_{t-N+1:t}$ are the most recent state tokens, and N is the local observation window size. We prepend C_t to the noisy action sequence as a prefix condition. The resulting sequence is processed by the SSM denoising backbone:

$$\hat{x}_0 = f_\theta(x_\tau, \tau, C_t). \tag{9}$$

In this design, c_t captures long-horizon progress and z_t retains local geometric and proprioceptive details. Thus, the policy leverages historical context without losing manipulation precision.

Causal Action Denoising. The hierarchical prefix formulation casts action denoising as a causal prefix-conditioned sequence modeling. Ordering the sequence as $[c_t, z_{t-N+1}, \ldots, z_t, x_\tau]$ allows long-term context to contextualize recent states before propagating information to noisy action tokens. We instantiate f_θ with a Mamba backbone, whose recurrent selective state updates naturally match this left-to-right conditioning flow while providing linear scaling for iterative diffusion sampling.

Timestep-Decoupled Action Denoising. To provide a stable conditioning signal throughout iterative denoising, we decouple timestep modulation from prefix conditioning. Since the diffusion timestep τ describes the noise level of the action trajectory, we inject the timestep embedding through AdaLN only into the action tokens while keeping the prefix condition unchanged. This keeps C_t as a stable representation of history and local observations throughout the denoising process. Meanwhile, the actions remain aware of the current noise level and can progressively refine the predicted action sequence. This design separates task conditioning from diffusion-step modulation.

Training. The policy is optimized with two objectives: the diffusion reconstruction loss for action generation and the dynamics-aware auxiliary loss for context learning. For an action window starting at time t, we construct the hierarchical condition C_t causally from observations up to time t and train the denoising model to predict the clean action trajectory:

$$\mathcal{L}_{\text{diff}}(\theta, \psi) = \mathbb{E}_{\zeta \sim \mathcal{D}_E, t, \tau, \epsilon}\left[\| f_\theta(x_\tau, \tau, C_t) - x_0 \|_2^2\right], \tag{10}$$

where the expectation is taken over expert trajectories $\zeta \sim \mathcal{D}_E$, trajectory timesteps $t \sim \mathcal{U}(0, T - H_a)$, diffusion steps $\tau \sim \mathcal{U}(1, L)$, and Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. Accordingly, the overall objective is:

$$\mathcal{L}(\theta, \psi, \phi) = \mathcal{L}_{\text{diff}}(\theta, \psi) + \lambda \mathcal{L}_{\text{dyn}}(\psi, \phi) \tag{11}$$

where λ balances action denoising and context representation learning.

4.3 Theoretical Analysis

Here we provide a theoretical analysis of the benefits of history conditioning for imitation learning in partially observable environments. Our goal is to characterize how the integration of temporal context mitigates the information loss inherent in POMDPs. To analyze the impact of information sets on performance, we denote $\mathcal{L}_{\text{diff}}(\pi; X_t)$ as the expected diffusion loss for a policy π conditioned on variable X_t. With the formalization of POMDP and observation aliasing in section 3, and the imitation objective defined in eq. (10), we present two core propositions.

Proposition 4.1. *The minimum achievable diffusion-based imitation loss for a history-conditioned policy is always less than or equal to that of a reactive policy. That is,*

$$\min_\theta \mathcal{L}_{\text{diff}}(\pi_\theta; h_t) \leq \min_\theta \mathcal{L}_{\text{diff}}(\pi_\theta; o_t). \tag{12}$$

Proposition 4.2. *In the presence of observation aliasing, history conditioning strictly reduces the minimum achievable imitation loss compared to a reactive policy:*

$$\min_\theta \mathcal{L}_{\text{diff}}(\pi_\theta; h_t) < \min_\theta \mathcal{L}_{\text{diff}}(\pi_\theta; o_t), \quad \textit{whenever} \quad I_E(a_t; h_t \mid o_t) > 0. \tag{13}$$

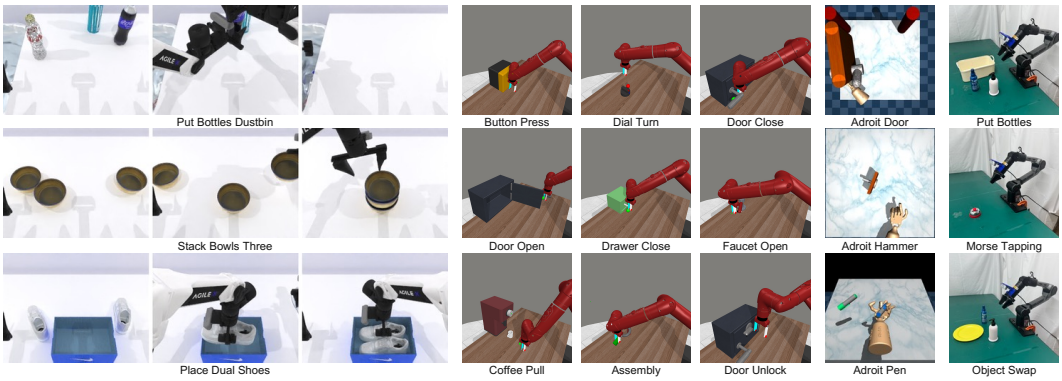


Figure 3: **Overview of Experimental Environments.** The figure summarizes representative environments from both simulation and real-world experiments. In each row, the left three panels visualize the RoboTwin tasks, the middle three columns present representative MetaWorld tasks, the next column shows Adroit tasks, and the rightmost column shows our real-world tasks.

Table 1: **Main Results on RoboTwin 2.0.** Tasks are grouped into Short, Mid, and Long horizons. "Obs." denotes the observation modality, including RGB and point-cloud observations. DSSP achieves the best overall performance with the smallest model size.

Method	Obs.	Success Rate by Horizon ↑				Params ↓
		Short (18)	Mid (15)	Long (17)	Average	
ACT (RSS'23)	RGB	29.94	26.40	32.47	29.74	80.0M
DP (RSS'23)	RGB	31.28	26.00	26.41	28.04	96.8M
π_0(arXiv'24)	RGB	44.56	44.07	50.47	46.42	3.3B
RDT (ICLR'25)	RGB	36.72	32.47	33.94	34.50	1.2B
SeedPolicy (arXiv'26)	RGB	43.39	36.53	47.59	42.76	147.3M
DP3 (RSS'24)	P.C.	59.83	52.53	52.76	55.24	264.4M
FlowPolicy (AAAI'25)	P.C.	54.89	37.40	29.59	41.04	264.4M
MambaPolicy (IROS'25)	P.C.	61.67	56.27	58.88	59.10	47.9M
Diff-Control (IROS'24)	P.C.	64.00	54.67	58.82	59.44	264.4M
MTIL (RA-L'25)	P.C.	**66.22**	54.73	56.18	59.36	60.5M
CycleManip (CVPR'26)	P.C.	57.78	49.27	54.47	54.10	279.2M
DSSP (Ours)	P.C.	64.78	**57.33**	**64.06**	**62.30**	**44.3M**

This condition implies that history resolves state ambiguity by capturing mutual information that is inaccessible to a reactive policy.

Proposition 4.1 establishes that history conditioning never degrades the theoretical performance limit of the policy, while proposition 4.2 proves that history conditioning strictly improves performance in the presence of observation aliasing. These results demonstrate that history acts as a sufficient statistic for the belief state, allowing the policy to disambiguate latent environmental configurations through the capture of action-relevant mutual information. We summarize the high-level logic here and defer the complete mathematical derivations to the Appendix E.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate DSSP on diverse simulation benchmarks and real-world memory-dependent manipulation tasks, as shown in Figure 3. For simulation, we use 87 tasks across three benchmarks: 50 bimanual tasks from RoboTwin 2.0 [6], 34 single-arm tabletop tasks from MetaWorld [62], and 3 dexterous in-hand tasks from Adroit [45]. To analyze performance across task horizons, we group the 50 RoboTwin tasks into short-, mid-, and long-horizon categories based on average episode length. Detailed grouping criteria, task lists, and subset selection are provided in Appendix C.3.

Table 2: **Results on Adroit and MetaWorld.** A comprehensive comparison across 37 tasks using 3 random seeds. Numbers in parentheses indicate the number of tasks in each benchmark.

Method	Adroit (3)	MetaWorld				Average
		Easy (21)	Medium (4)	Hard (4)	Very Hard (5)	
DP (RSS'23)	21.0±7.7	50.7±6.1	11.0±2.5	5.25±2.5	22.0±5.0	35.2±5.3
Adaflow (NeurIPS'24)	30.0±7.7	49.4±6.8	12.0±5.0	5.75±4.0	24.0±4.8	35.6±6.1
CP (arXiv'24)	29.7±6.7	69.3±4.2	21.2±6.0	17.5±3.9	30.0±4.9	50.1±4.7
DP3 (RSS'24)	67.3±5.0	87.3±2.2	44.5±8.7	32.7±7.7	39.4±9.0	68.7±4.7
FlowPolicy (AAAI'25)	71.0±2.3	84.8±2.2	58.2±7.9	40.2±4.5	52.2±5.0	71.6±3.5
MP1 (AAAI'26)	**75.7±2.3**	88.2±1.1	**68.0±3.1**	**58.1±5.0**	67.2±2.7	78.9±2.1
DSSP (ours)	73.0±2.9	**90.5±2.1**	67.4±4.1	54.6±2.4	**71.3±3.8**	**80.1±2.6**

For real-world evaluation, we use an AgileX robotic platform equipped with a fixed Intel RealSense L515 camera for point-cloud observations. We design three tabletop tasks that require long-horizon execution or progress tracking: (i) *Put Bottles*, where the robot sequentially places two bottles into a basket; (ii) *Object Swap*, where the robot swaps two objects through an intermediate buffer slot; and (iii) *Morse Tapping*, where the robot taps a target three times before returning to the initial position.

Baselines. We compare DSSP with representative imitation-learning baselines on each benchmark. DP3 [63] is the most direct baseline, as it uses the same 3D point-cloud observation and action interfaces as DSSP, while DP [8] serves as a standard diffusion-policy baseline. On RoboTwin, we include ACT [66], RDT [35], and π_0 [1] from the official benchmark evaluation, along with Seed-Policy [15], FlowPolicy [65], and MambaPolicy [3]. We further consider temporally informed baselines: Diff-Control [36], which models action statefulness using prior action trajectories; MTIL [68], which encodes full trajectory history with Mamba; and CycleManip [54], which leverages history sampling and progress prediction for repetitive manipulation tasks. On MetaWorld and Adroit, we compare with recent policy-learning methods, including AdaFlow [20], CP [43], and MP1 [47].

Evaluation Metric and Protocol. We use task success rate as the primary metric. For RoboTwin, we evaluate 100 test episodes under the in-distribution *demo clean* setting, defining success as completion within the maximum horizon. For MetaWorld and Adroit, we evaluate 20 episodes every 200 training epochs and report the average of the top five success rates. For real-world experiments, each method undergoes 20 trials per task with randomized initial configurations based on task-specific criteria.

Implementation Details. All models are trained with AdamW on NVIDIA RTX 4090 GPUs. We use 50 demonstrations per task for RoboTwin and real-world experiments, and 10 demonstrations per task for MetaWorld and Adroit. DSSP employs trajectory-wise batching for causal history encoding. Training schedules, architectural details, and hyperparameters are specified in Appendix C.

5.2 Simulation Experiments

Main Results. As shown in Table 1, DSSP achieves the highest average success rate of 62.30% on RoboTwin 2.0, outperforming the strongest baseline, Diff-Control, by 4.8% relatively and DP3 by 12.8%. On long-horizon tasks, DSSP achieves 64.06% success, representing relative improvements of 8.8% over MambaPolicy and 21.4% over DP3, demonstrating the benefits of full-history conditioning. Moreover, DSSP achieves these results with the smallest model size of 44.3M parameters. On Adroit and MetaWorld, DSSP also achieves the highest overall average success rate among the compared methods, as shown in Table 2.

Analysis of History Encoding. To better understand how DSSP uses historical information, we conduct controlled studies on the six-task long-horizon subset defined in Appendix C.3. We analyze three aspects: (1) how temporal backbone and history length affect performance, (2) how efficiently the history encoder scales to full-history conditioning, and (3) whether the policy truly relies on context representation when recent observations are corrupted. Together, these studies show that the gains of DSSP come from effective long-history utilization, while maintaining practical efficiency.

We first compare the impact of history length on different encoder backbones (Table 3). While the Transformer encoder performs best with a short window ($T_h = 10$) and stagnates with longer

Table 3: Comparison of temporal backbones across history lengths. We report average success over six long-horizon tasks and history-prefix encoding cost in p95 latency and peak GPU memory.

Encoder Backbone	History Length (T_h)	Success Rate (%) ↑	Encoding Cost ↓
Transformer	10	68.00	1.43 ms / 176.5 MB
	20	61.83	1.43 ms / 176.7 MB
	Full history	66.00	3.61 ms / 586.2 MB
Mamba	10	53.00	1.87 ms / 181.1 MB
	20	60.00	1.90 ms / 181.5 MB
	Full history (Ours)	**71.33**	**1.97 ms / 238.5 MB**

histories, the state-space encoder scales effectively with increasing context. Mamba achieves its peak success rate of 71.33% using full history, which is an 8.1% relative improvement over the full-history Transformer. These results confirm that the recurrent formulation of Mamba is better suited for aggregating varying length observations into a compact context representation.

Beyond improving success rates, DSSP also scales more efficiently to long histories. Full-history Transformer encoding costs 3.61 ms and a larger peak GPU memory (586.2 MB), whereas our full-history Mamba encoder requires only 1.97 ms and 238.5 MB, reducing latency by 45.4% and peak memory by 59.3%. This advantage stems from the linear-time state-space formulation, which aggregates history through recurrent state updates instead of pairwise attention over all historical tokens. Meanwhile, during streaming inference, DSSP only maintains a compact hidden-state cache, making the per-step encoding overhead nearly independent of accumulated history length.

Table 4: Robustness comparison under Gaussian perturbations to recent state representations. We report the six tasks' average success rate (%) under different perturbation scales (σ) during inference.

Method	**Noise Scale (σ)**			
	0.00 (Clean)	**0.05**	**0.10**	**0.15**
DP3 (Baseline)	58.67	43.00	15.83	3.17
DSSP (w/o full history)	60.00	41.67	23.33	11.33
DSSP (Full history)	**71.33**	**52.33**	**37.00**	**20.83**

We further test whether the policy uses historical context when recent observations are unreliable by perturbing the most recent three state tokens during inference: $z_i^{\text{pert}} = z_i + \sigma\epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \mathbf{I})$. As shown in Table 4, DP3 and our short-window ($T_h = 10$) variant degrade sharply as the perturbation scale increases, while DSSP with full history maintains substantially higher success rates. These results indicate that the learned context token incorporates earlier information, preventing the policy from degenerating into merely relying on recent observations for action generation.

To further assess the memory capabilities of DSSP, we evaluate it on memory-dependent manipulation tasks from RMBench [7], with additional results provided in Appendix D.2.

Table 5: Ablation on six history-sensitive RoboTwin tasks under the *demo clean* setting. Hist., TD, Recent, Dyn., and Trans. denote the history encoder, timestep-decoupled denoising, recent-state conditioning, dynamics-aware loss, and Transformer denoising backbone, respectively.

Metric	DP3	w/o Hist.	w/o TD	w/o Recent	w/o Dyn.	w/ Trans.	Full
Success Rate (%)	58.67	64.33	66.67	68.17	69.50	68.50	**71.33**
Relative Improvement (%)	–	+9.65	+13.64	+16.21	+18.46	+16.75	**+21.56**

Ablation Study. We ablate the key components of DSSP on the six-task history-sensitive subset defined in Section 5.1. As shown in Table 5, the full model achieves the best success rate of 71.33%, a 21.56% relative improvement over DP3. Removing the history encoder leads to the largest drop, highlighting the importance of long-horizon context. Replacing the Mamba action denoising backbone with a Transformer-based one remains stronger than DP3 but underperforms the full model, showing that Mamba further improves temporal action generation. Without timestep-decoupled

conditioning, performance drops to 66.67%, validating the effectiveness of our decoupled conditioning design. Finally, removing recent-state conditioning or the dynamics-aware loss also degrades performance, confirming the importance of local grounding and predictive context learning.

5.3 Real-World Experiments

We evaluate DSSP on three real-world manipulation tasks that require long-horizon memory or progress tracking, as shown in Table 6. DSSP substantially outperforms DP3 across all tasks, increasing the average success rate from 30% to 70% (a 133.3% relative improvement). The gains are particularly significant on Morse Tapping, where DSSP improves the success rate from 15% to 85%. These results demonstrate the effectiveness of full-history context for real-world memory-dependent manipulation, where the decision making often depends on previous interactions rather than the current observation alone. To better understand where the gains come from, we provide a failure-mode analysis for each real-world task in Appendix D.5.

Table 6: Real-world long-horizon task (avg. steps) evaluation.

Method	Put Bottles (666)	Object Swap (713)	Morse Tapping (366)
FlowPolicy (AAAI'25)	40%	30%	10%
MP1 (AAAI'26)	45%	30%	25%
DP3 (RSS'24)	40%	35%	15%
DSSP (ours)	**60%**	**65%**	**85%**

6 Conclusion

In this paper, we introduce DSSP, an efficient full-history conditioned diffusion state-space policy for long-horizon robot manipulation. Our results show that compactly encoding the full observation history improves temporal disambiguation and task-progress tracking in history-sensitive tasks. The proposed state-space history encoder, dynamics-aware objective, and hierarchical timestep-decoupled conditioning jointly integrate long-horizon context with recent observations for action generation.

7 Limitations

DSSP is primarily designed to improve temporal disambiguation and task-progress tracking through full-history conditioning. It does not directly eliminate low-level perception or control failures, such as inaccurate target localization, grasping, or placement, which remain a source of real-world failure. Moreover, our real-world evaluation is limited to three tabletop tasks with a fixed-camera point-cloud setup. Extending DSSP to more diverse embodiments, viewpoints, deformable objects, and dynamic environments remains an important direction for future work.

Acknowledgments and Disclosure of Funding

This work has been supported by the National Natural Science Foundation of China (No. 62503322), the AI for Science Seed Program of Shanghai Jiao Tong University (Project No. 2025AI4SQY06), the Shanghai Jiao Tong University Xiaomi Scholar Fund, and the Shanghai Municipal Special Program for Basic Research on General AI Foundation Models (Grant No. 2025SHZDZX026D18).

References

[1] Junjie Bai, Yu-Wei Chao, Qizhi Chen, Jinwei Gu, Moo Jin Kim, Zhaoshuo Li, Xuan Li, Tsung-Yi Lin, Ming-Yu Liu, Nic Ma, Kaichun Mo, Delin Qu, Shangkun Sun, Hongchi Xia, Fangyin Wei, and Xiaohui Zeng. Openpi Comet: Competition Solution For 2025 BEHAVIOR Challenge, December 2025. URL `http://arxiv.org/abs/2512.10071`. arXiv:2512.10071 [cs].

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu,

Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

[3] Jiahang Cao, Qiang Zhang, Jingkai Sun, Jiaxu Wang, Hao Cheng, Yulin Li, Jun Ma, Kun Wu, Zhiyuan Xu, Yecheng Shao, Wen Zhao, Gang Han, Yijie Guo, and Renjing Xu. Mamba Policy: Towards Efficient 3D Diffusion Policy with Hybrid Selective State Models, June 2025. URL http://arxiv.org/abs/2409.07163. arXiv:2409.07163 [cs].

[4] Ying Chai, Litao Deng, Ruizhi Shao, Jiajun Zhang, Kangchen Lv, Liangjun Xing, Xiang Li, Hongwen Zhang, and Yebin Liu. Gaf: Gaussian action field as a 4d representation for dynamic world modeling in robotic manipulation, 2025. URL https://arxiv.org/abs/2506.14135.

[5] Jingjing Chen, Hongjie Fang, Chenxi Wang, Shiquan Wang, and Cewu Lu. History-Aware Visuomotor Policy Learning via Point Tracking, March 2026. URL http://arxiv.org/abs/2509.17141. arXiv:2509.17141 [cs].

[6] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A Scalable Data Generator and Benchmark with Strong Domain Randomization for Robust Bimanual Robotic Manipulation, August 2025. URL http://arxiv.org/abs/2506.18088. arXiv:2506.18088 [cs].

[7] Tianxing Chen, Yuran Wang, Mingleyang Li, Yan Qin, Hao Shi, Zixuan Li, Yifan Hu, Yingsheng Zhang, Kaixuan Wang, Yue Chen, et al. Rmbench: Memory-dependent robotic manipulation benchmark with insights into policy design. *arXiv preprint arXiv:2603.01229*, 2026.

[8] 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, March 2024. URL http://arxiv.org/abs/2303.04137. arXiv:2303.04137 [cs].

[9] Tri Dao and Albert Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality, May 2024. URL http://arxiv.org/abs/2405.21060. arXiv:2405.21060 [cs].

[10] Karthik Dharmarajan, Wenlong Huang, Jiajun Wu, Li Fei-Fei, and Ruohan Zhang. Dream2flow: Bridging video generation and open-world manipulation with 3d object flow, 2025. URL https://arxiv.org/abs/2512.24766.

[11] Han Fang, Yize Huang, Yuheng Zhao, Paul Weng, Xiao Li, and Yutong Ban. Omp: One-step meanflow policy with directional alignment, 2026. URL https://arxiv.org/abs/2512.19347.

[12] Xiao Fu, Xintao Wang, Xian Liu, Jianhong Bai, Runsen Xu, Pengfei Wan, Di Zhang, and Dahua Lin. Learning video generation for robotic manipulation with collaborative trajectory control, 2026. URL https://arxiv.org/abs/2506.01943.

[13] Dechen Gao, Boqi Zhao, Andrew Lee, Ian Chuang, Hanchu Zhou, Hang Wang, Zhe Zhao, Junshan Zhang, and Iman Soltani. Vita: Vision-to-action flow matching policy, 2026. URL https://arxiv.org/abs/2507.13231.

[14] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, May 2024. URL http://arxiv.org/abs/2312.00752. arXiv:2312.00752 [cs].

[15] Youqiang Gui, Yuxuan Zhou, Shen Cheng, Xinyang Yuan, Haoqiang Fan, Peng Cheng, and Shuaicheng Liu. SeedPolicy: Horizon Scaling via Self-Evolving Diffusion Policy for Robot Manipulation, March 2026. URL http://arxiv.org/abs/2603.05117. arXiv:2603.05117 [cs].

[16] Jun Guo, Xiaojian Ma, Yikai Wang, Min Yang, Huaping Liu, and Qing Li. Flowdreamer: A rgb-d world model with flow-based motion representations for robot manipulation, 2025. URL https://arxiv.org/abs/2505.10075.

[17] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation, 2026. URL https://arxiv.org/abs/2510.10125.

[18] Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal Confusion in Imitation Learning, November 2019. URL http://arxiv.org/abs/1905.11979. arXiv:1905.11979 [cs].

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

[20] Xixi Hu, Bo Liu, Xingchao Liu, and Qiang Liu. AdaFlow: Imitation Learning with Variance-Adaptive Flow-Based Policies, November 2024. URL http://arxiv.org/abs/2402.04292. arXiv:2402.04292 [cs].

[21] Helong Huang, Min Cen, Kai Tan, Xingyue Quan, Guowei Huang, and Hong Zhang. Graphcot-vla: A 3d spatial-aware reasoning vision-language-action model for robotic manipulation with ambiguous instructions, 2025. URL https://arxiv.org/abs/2508.07650.

[22] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $_{0.5}$: a Vision-Language-Action Model with Open-World Generalization, April 2025. URL http://arxiv.org/abs/2504.16054. arXiv:2504.16054 [cs].

[23] Huiwon Jang, Sihyun Yu, Heeseung Kwon, Hojin Jeon, Younggyo Seo, and Jinwoo Shin. ContextVLA: Vision-Language-Action Model with Amortized Multi-Frame Context, October 2025. URL http://arxiv.org/abs/2510.04246. arXiv:2510.04246 [cs].

[24] Xiaogang Jia, Qian Wang, Atalay Donat, Bowen Xing, Ge Li, Hongyi Zhou, Onur Celik, Denis Blessing, Rudolf Lioutikov, and Gerhard Neumann. MaIL: Improving Imitation Learning with Mamba, November 2024. URL http://arxiv.org/abs/2406.08234. arXiv:2406.08234 [cs].

[25] Xiaogang Jia, Atalay Donat, Xi Huang, Xuan Zhao, Denis Blessing, Hongyi Zhou, Han A. Wang, Hanyi Zhang, Qian Wang, Rudolf Lioutikov, and Gerhard Neumann. X-IL: Exploring the Design Space of Imitation Learning Policies, February 2025. URL http://arxiv.org/abs/2502.12330. arXiv:2502.12330 [cs].

[26] Yuhua Jiang, Shuang Cheng, Yan Ding, Feifei Gao, and Biqing Qi. Asyncvla: Asynchronous flow matching for vision-language-action models, 2025. URL https://arxiv.org/abs/2511.14148.

[27] Zhennan Jiang, Kai Liu, Yuxin Qin, Shuai Tian, Yupeng Zheng, Mingcai Zhou, Chao Yu, Haoran Li, and Dongbin Zhao. World4rl: Diffusion world models for policy refinement with reinforcement learning for robotic manipulation, 2026. URL https://arxiv.org/abs/2509.19080.

[28] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model, September 2024. URL http://arxiv.org/abs/2406.09246. arXiv:2406.09246 [cs].

[29] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. URL https://arxiv.org/abs/2502.19645.

[30] Myungkyu Koo, Daewon Choi, Taeyoung Kim, Kyungmin Lee, Changyeon Kim, Younggyo Seo, and Jinwoo Shin. HAMLET: Switch your Vision-Language-Action Model into a History-Aware Policy, April 2026. URL http://arxiv.org/abs/2510.00695. arXiv:2510.00695 [cs].

[31] Aakash Lahoti, Kevin Y. Li, Berlin Chen, Caitlin Wang, Aviv Bick, J. Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved Sequence Modeling using State Space Principles, March 2026. URL http://arxiv.org/abs/2603.15569. arXiv:2603.15569 [cs].

[32] Seungjae Lee, Yibin Wang, Haritheja Etukuru, H. Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior Generation with Latent Actions, June 2024. URL http://arxiv.org/abs/2403.03181. arXiv:2403.03181 [cs].

[33] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-End Training of Deep Visuomotor Policies, April 2016. URL http://arxiv.org/abs/1504.00702. arXiv:1504.00702 [cs].

[34] Hao Li, Shuai Yang, Yilun Chen, Xinyi Chen, Xiaoda Yang, Yang Tian, Hanqing Wang, Tai Wang, Dahua Lin, Feng Zhao, and Jiangmiao Pang. CronusVLA: Towards Efficient and Robust Manipulation via Multi-Frame Vision-Language-Action Modeling, October 2025. URL http://arxiv.org/abs/2506.19816. arXiv:2506.19816 [cs].

[35] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a Diffusion Foundation Model for Bimanual Manipulation, March 2025. URL http://arxiv.org/abs/2410.07864. arXiv:2410.07864 [cs].

[36] Xiao Liu, Yifan Zhou, Fabian Weigend, Shubham Sonawani, Shuhei Ikemoto, and Heni Ben Amor. Diff-control: A stateful diffusion-based policy for imitation learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7453–7460. IEEE, 2024.

[37] Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. Gwm: Towards scalable gaussian world models for robotic manipulation, 2025. URL https://arxiv.org/abs/2508.17600.

[38] Yiyang Lu, Yufeng Tian, Zhecheng Yuan, Xianbang Wang, Pu Hua, Zhengrong Xue, and Huazhe Xu. H$^{\mathbf{3}}$DP: Triply-Hierarchical Diffusion Policy for Visuomotor Learning, May 2025. URL http://arxiv.org/abs/2505.07819. arXiv:2505.07819 [cs] version: 1.

[39] Jiahua Ma, Yiran Qin, Yixiong Li, Xuanqi Liao, Yulan Guo, and Ruimao Zhang. CDP: Towards Robust Autoregressive Visuomotor Policy Learning via Causal Diffusion, August 2025. URL http://arxiv.org/abs/2506.14769. arXiv:2506.14769 [cs].

[40] Max Sobol Mark, Jacky Liang, Maria Attarian, Chuyuan Fu, Debidatta Dwibedi, Dhruv Shah, and Aviral Kumar. BPP: Long-Context Robot Imitation Learning by Focusing on Key History Frames, February 2026. URL http://arxiv.org/abs/2602.15010. arXiv:2602.15010 [cs].

[41] Nayoung Oh, Jaehyeong Jang, Moonkyeong Jung, and Daehyung Park. Dispo: Diffusion-ssm based policy learning for coarse-to-fine action discretization, 2026. URL https://arxiv.org/abs/2409.14719.

[42] William Peebles and Saining Xie. Scalable Diffusion Models with Transformers, March 2023. URL http://arxiv.org/abs/2212.09748. arXiv:2212.09748 [cs].

[43] Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency Policy: Accelerated Visuomotor Policies via Consistency Distillation, June 2024. URL http://arxiv.org/abs/2405.07503. arXiv:2405.07503 [cs].

[44] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model, 2025. URL https://arxiv.org/abs/2501.15830.

[45] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations, June 2018. URL http://arxiv.org/abs/1709.10087. arXiv:1709.10087 [cs].

[46] Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior Transformers: Cloning k modes with one stone, October 2022. URL http://arxiv.org/abs/2206.11251. arXiv:2206.11251 [cs].

[47] Juyi Sheng, Ziyi Wang, Peiming Li, and Mengyuan Liu. MP1: MeanFlow Tames Policy Learning in 1-step for Robotic Manipulation, October 2025. URL http://arxiv.org/abs/2507.10543. arXiv:2507.10543 [cs].

[48] Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation, 2026. URL https://arxiv.org/abs/2508.19236.

[49] Gokul Swamy, Sanjiban Choudhury, J. Andrew Bagnell, and Zhiwei Steven Wu. Sequence Model Imitation Learning with Unobserved Contexts, January 2023. URL http://arxiv.org/abs/2208.02225. arXiv:2208.02225 [cs].

[50] Jiaming Tang, Yufei Sun, Yilong Zhao, Shang Yang, Yujun Lin, Zhuoyang Zhang, James Hou, Yao Lu, Zhijian Liu, and Song Han. Vlash: Real-time vlas via future-state-aware asynchronous inference, 2025. URL https://arxiv.org/abs/2512.01031.

[51] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy, 2024. URL https://arxiv.org/abs/2405.12213.

[52] Marcel Torne, Andy Tang, Yuejiang Liu, and Chelsea Finn. Learning Long-Context Diffusion Policies via Past-Token Prediction, May 2025. URL http://arxiv.org/abs/2505.09561. arXiv:2505.09561 [cs].

[53] Toshiaki Tsuji. Mamba as a Motion Encoder for Robotic Imitation Learning. *IEEE Access*, 13: 69941–69949, 2025. ISSN 2169-3536. doi: 10.1109/ACCESS.2025.3561283. URL https://ieeexplore.ieee.org/document/10966860/.

[54] Yi-Lin Wei, Haoran Liao, Yuhao Lin, Pengyue Wang, Zhizhao Liang, Guiliang Liu, and Wei-Shi Zheng. Cyclemanip: Enabling cyclic task manipulation via effective historical perception and understanding. *arXiv preprint arXiv:2512.01022*, 2025.

[55] Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting Copycat Agents in Behavioral Cloning from Observation Histories, October 2020. URL http://arxiv.org/abs/2010.14876. arXiv:2010.14876 [cs].

[56] Chuan Wen, Jierui Lin, Jianing Qian, Yang Gao, and Dinesh Jayaraman. Keyframe-Focused Visual Imitation Learning, June 2021. URL http://arxiv.org/abs/2106.06452. arXiv:2106.06452 [cs].

[57] Junlin Xie, Xu Luo, Hao Wu, Ji Zhang, Youguang Xing, Lianli Gao, and Jingkuan Song. In-context adaptation for generalizable imitation learning. In *CoRL 2025 Workshop RemembeRL*.

[58] Ge Yan, Jiyue Zhu, Yuquan Deng, Shiqi Yang, Ri-Zhao Qiu, Xuxin Cheng, Marius Memmel, Ranjay Krishna, Ankit Goyal, Xiaolong Wang, and Dieter Fox. ManiFlow: A General Robot Manipulation Policy via Consistency Flow Training, September 2025. URL http://arxiv.org/abs/2509.01819. arXiv:2509.01819 [cs].

[59] Ganlin Yang, Zhangzheng Tu, Yuqiang Yang, Sitong Mao, Junyi Dong, Tianxing Chen, Jiaqi Peng, Jing Xiong, Jiafei Cao, Jifeng Dai, Wengang Zhou, Yao Mu, and Tai Wang. Eventvla: Event-driven visual evidence memory for long-horizon vision-language-action policies, 2026. URL https://arxiv.org/abs/2606.20092.

[60] Tenny Yin, Zhiting Mei, Zhonghe Zheng, Miyu Yamane, David Wang, Jade Sceats, Samuel M. Bateman, Lihan Zha, Apurva Badithela, Ola Shorinwa, and Anirudha Majumdar. Playworld: Learning robot world models from autonomous play, 2026. URL https://arxiv.org/abs/2603.09030.

[61] Youngju Yoo, Jiaheng Hu, Yifeng Zhu, Bo Liu, Qiang Liu, Roberto Martín-Martín, and Peter Stone. RoboSSM: Scalable In-context Imitation Learning via State-Space Models, September 2025. URL http://arxiv.org/abs/2509.19658. arXiv:2509.19658 [cs].

[62] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning, June 2021. URL http://arxiv.org/abs/1910.10897. arXiv:1910.10897 [cs].

[63] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations, September 2024. URL http://arxiv.org/abs/2403.03954. arXiv:2403.03954 [cs].

[64] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Ayzaan Wahid, Vikas Sindhwani, and Johnny Lee. Transporter Networks: Rearranging the Visual World for Robotic Manipulation, January 2022. URL http://arxiv.org/abs/2010.14406. arXiv:2010.14406 [cs].

[65] Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 14754–14762, 2025.

[66] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware, April 2023. URL http://arxiv.org/abs/2304.13705. arXiv:2304.13705 [cs].

[67] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

[68] Yulin Zhou, Yuankai Lin, Fanzhe Peng, Jiahui Chen, Kaiji Huang, Hua Yang, and Zhouping Yin. MTIL: Encoding Full History With Mamba for Temporal Imitation Learning. *IEEE Robotics and Automation Letters*, 10(11):11761–11767, November 2025. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2025.3615520. URL https://ieeexplore.ieee.org/document/11184145/.

[69] Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation, 2025. URL https://arxiv.org/abs/2406.14540.

A Preliminaries

Diffusion Policy. Diffusion Policy [8] adapts Denoising Diffusion Probabilistic Models (DDPMs) [19] to action generation. The policy treats a future action sequence $a_{t:t+H_a-1}$ of horizon H_a as the clean sample x_0. We parameterize the model to predict the clean action sequence directly, conditioning the denoising process on the history representation h_t. During training, we optimize the reconstruction objective over L diffusion steps:

$$\mathcal{L}_{x_0} = \mathbb{E}_{x_0,\tau,\epsilon}\left[\|f_\theta(x_\tau, \tau, h_t) - x_0\|_2^2\right], \tag{14}$$

where $\tau \in \{1, \ldots, L\}$ is the uniform diffusion step and x_τ is the noise-corrupted action sequence.

State Space Models (SSMs). We employ Mamba as the sequence backbone for both history encoding and action diffusion. A standard discrete SSM updates a hidden state $s_t \in \mathbb{R}^N$ and outputs y_t via time-invariant parameters:

$$s_t = \bar{A} s_{t-1} + \bar{B} x_t, \qquad y_t = C s_t. \tag{15}$$

Mamba introduces a selective mechanism where parameters become input-dependent: $(\Delta_t, B_t, C_t) = f_\theta(x_t)$, with Δ_t serving as the discretization step size. The discretized parameters are:

$$\bar{A}_t = \exp(\Delta_t A), \qquad \bar{B}_t = (\Delta_t A)^{-1}\big(\exp(\Delta_t A) - I\big)\Delta_t B_t. \tag{16}$$

The recurrent update becomes $s_t = \bar{A}_t s_{t-1} + \bar{B}_t x_t$ and $y_t = C_t s_t$. This input-dependent selection enables the model to efficiently compress long observation histories and generate temporally coherent actions.

B Additional Related Works

B.1 Vision-Language-Action Models

Vision-language-action (VLA) models have recently become a central direction for scalable robot learning. RT-2 [2] transfers vision-language knowledge to robotic control by representing actions as language-like tokens. Octo [51] and OpenVLA [28] further introduce open-source generalist policies trained on large cross-embodiment robot datasets. Recent generative VLA models move beyond discrete action tokenization toward continuous action generation, including flow-matching-based policies such as π_0 [1] and large-scale diffusion policies such as RDT-1B [35].

Recent work has further improved VLA policies along several directions. OpenVLA-OFT [29] studies effective fine-tuning recipes with continuous actions and action chunking, while VITA-VLA [13] equips pretrained vision-language models with action-generation capability through action expert distillation. Other works improve inference efficiency and temporal consistency through asynchronous generation, coarse-to-fine action generation, or action coherence guidance [26, 50]. In parallel, SpatialVLA [44] and GraphCoT-VLA [21] enhance spatial reasoning and embodiment-aware manipulation.

B.2 World Models for Robotics

World models provide a complementary direction for robot learning by predicting future environment states for simulation, planning, or policy improvement. In robotic manipulation, action-conditioned video generation has been widely used to model robot-object dynamics. IRASim [69] learns an interactive real-robot action simulator conditioned on observations and robot actions. RoboMaster [12] improves trajectory-controlled robotic video generation by modeling robot-object interactions, while Ctrl-World [17] studies controllable multi-view world modeling for evaluating and improving generalist robot policies.

Recent works further introduce structured spatial representations for world modeling. FlowDreamer [16] uses RGB-D observations and 3D scene flow for action-conditioned future prediction. GAF [4] and GWM [37] represent dynamic manipulation scenes with Gaussian-based formulations, enabling future scene prediction and action refinement. Dream2Flow [10] connects video generation with robot control by extracting 3D object flow from generated videos. World4RL [27] and PlayWorld [60] further use learned world models for policy evaluation, reinforcement learning, or policy refinement.

C Implementation Details

C.1 Trajectory-Wise Training

Since DSSP conditions action generation on causal history context, we use trajectory-wise batch construction during training. At each optimization step, we randomly load one complete demonstration trajectory and compute its state representations and causal context representations following

Equations (3) and (4). We then sample B valid action-window start indices from this trajectory, where B is the training batch size.

For each sampled start time t_i, we construct the hierarchical condition using Equation (8), with the context representations and recent state representations available up to t_i. The corresponding diffusion target is the future action chunk starting from t_i. This ensures that every training sample is conditioned only on observations before its prediction time, without accessing future observations.

For fair comparison with window-based baselines, we keep the same effective batch size B and comparable optimization budget across methods. Thus, trajectory-wise training does not introduce additional demonstrations or a larger denoising batch; it only changes how causal history conditions are constructed for DSSP. The training objectives follow Equation (11).

Table 7: Hyperparameters used for RoboTwin simulation experiments.

Hyperparameter	Value
Recent Observation Horizon (N)	3
Horizon (H)	8
Action Execution Horizon (H_a)	6
Expert Demonstrations per Task	50
Optimizer	AdamW
Betas (β_1, β_2)	$(0.95, 0.999)$
Learning Rate	1.0×10^{-4}
Weight Decay	1.0×10^{-6}
Diffusion Training Steps	100
Inference Steps	10
Learning Rate Scheduler	Cosine
Warmup Steps	500
Prediction Type	Sample prediction
Dynamics Loss Weight (λ)	0.05
Dynamics Loss Type	Cosine distance
Mamba Backbone Layers	8
Mamba History Encoder Layers	2
Hidden Dimension	512
SSM State Dimension	64

Table 8: Hyperparameters used for Adroit simulation experiments.

Hyperparameter	Value
Recent Observation Horizon (N)	2
Horizon (H)	4
Action Execution Horizon (H_a)	3
Expert Demonstrations per Task	10
Optimizer	AdamW
Betas (β_1, β_2)	$(0.95, 0.999)$
Learning Rate	1.0×10^{-4}
Weight Decay	1.0×10^{-6}
Epoch	3000
Learning Rate Scheduler	Cosine
Warmup Steps	500
Diffusion Training Steps	100
Inference Steps	10
Prediction Type	Sample prediction
Dynamics Loss Weight (λ)	0.05
Dynamics Loss Type	Cosine distance
Mamba Backbone Layers	8
Mamba History Encoder Layers	2
Hidden Dimension	512
SSM State Dimension	64
Evaluation Episodes	20
Maximum Evaluation Steps	300

Table 9: Hyperparameters used for MetaWorld simulation experiments.

Hyperparameter	Value
Recent Observation Horizon (N)	2
Horizon (H)	4
Action Execution Horizon (H_a)	3
Expert Demonstrations per Task	10
Optimizer	AdamW
Betas (β_1, β_2)	$(0.95, 0.999)$
Learning Rate	1.0×10^{-4}
Weight Decay	1.0×10^{-6}
Epoch	1000
Learning Rate Scheduler	Cosine
Warmup Steps	500
Diffusion Training Steps	100
Inference Steps	10
Prediction Type	Sample prediction
Dynamics Loss Weight (λ)	0.05
Dynamics Loss Type	Cosine distance
Mamba Backbone Layers	8
Mamba History Encoder Layers	2
Hidden Dimension	512
SSM State Dimension	64
Evaluation Episodes	20
Maximum Evaluation Steps	1000

C.2 Hyperparameters

To account for the varying difficulty levels and unique characteristics of different benchmarks, we tailor our hyperparameter configurations to each individual dataset. The final settings, which include additional configurations for the Mamba-based history encoder, Mamba diffusion backbone, and dynamics-aware auxiliary objective and are summarized in Tables 7,8,9, are informed by established practices in prior literature [63, 65, 6].

Table 10: RoboTwin task grouping by average episode length. Short, mid, and long horizons correspond to average episode lengths of < 150, 150–250, and > 250 steps, respectively.

Horizon Group	Tasks
Long (17)	Put Bottles Dustbin, Open Microwave, Stack Blocks Three, Stack Bowls Three, Blocks Ranking Rgb, Blocks Ranking Size, Hanging Mug, Stack Blocks Two, Stack Bowls Two, Place Cans Plasticbox, Handover Block, Shake Bottle Horizontally, Put Object Cabinet, Dump Bin Bigbin, Open Laptop, Place Can Basket, Place Object Basket
Mid (15)	Shake Bottle, Place Burger Fries, Place Bread Basket, Place Dual Shoes, Handover Mic, Place Shoe, Place Empty Cup, Scan Object, Place Bread Skillet, Place Container Plate, Place A2B Left, Rotate Qrcode, Move Stapler Pad, Stamp Seal, Move Can Pot
Short (18)	Place Mouse Pad, Place Fan, Adjust Bottle, Move Pillbottle Pad, Place Object Scale, Place A2B Right, Press Stapler, Place Object Stand, Place Phone Stand, Pick Dual Bottles, Pick Diverse Bottles, Move Playingcard Away, Beat Block Hammer, Lift Pot, Turn Switch, Grab Roller, Click Bell, Click Alarmclock

C.3 RoboTwin Horizon Grouping

For horizon-wise analysis, we partition the 50 RoboTwin tasks according to their average episode length. Tasks with average length below 150 steps are categorized as short-horizon tasks, tasks between 150 and 250 steps are categorized as mid-horizon tasks, and tasks above 250 steps are categorized as long-horizon tasks. The resulting groups are listed in Table 10.

For ablation and diagnostic analysis, we further define a six-task long-horizon analysis subset from the above grouping. This subset is selected from tasks with long execution horizons and temporally dependent manipulation behaviors. The purpose of this subset is not to replace the full benchmark evaluation, but to provide a controlled set of tasks for studying how different architectural components affect history utilization. Unless otherwise specified, all ablation studies and history-utilization analyses are conducted on this subset. The selected tasks and their task-level results are shown in Table 11.

Table 11: Six-task history-sensitive analysis subset used for ablation and diagnostic studies. The table reports task-level success rates under the in-distribution (*demo clean*) evaluation setting.

Task	Avg. Steps	DP3 (%)	DSSP (%)
Put Bottles Dustbin	637	60	83
Stack Bowls Three	476	57	80
Put Object Cabinet	274	72	70
Place Dual Shoes	228	13	19
Stack Bowls Two	313	83	93
Place Can Basket	255	67	83
Average	364	58.67	71.33

D Additional Results and Failure-Mode Analysis

D.1 Per-task RoboTwin results.

Table 12 reports the per-task success rates of DP3 and DSSP on all 50 RoboTwin tasks under the clean setting. Overall, DSSP improves the average success rate from 55.24% to 62.30%, with particularly large gains on tasks requiring sequential progress tracking, such as Open Microwave, Place Cans Plasticbox, Put Bottles Dustbin, and Stack Bowls Three.

Table 12: Per-task success rate comparison between RoboTwin DP3 and DSSP on 50 RoboTwin tasks under the clean setting.

Task	DP3	DSSP
Adjust Bottle	**99.00**	96.00
Beat Block Hammer	72.00	**79.00**
Blocks Ranking RGB	3.00	**6.00**
Blocks Ranking Size	2.00	**4.00**
Click Alarmclock	77.00	**99.00**
Click Bell	90.00	**100.00**
Dump Bin Bigbin	**85.00**	84.00
Grab Roller	**98.00**	**98.00**
Handover Block	70.00	**95.00**
Handover Mic	**100.00**	93.00
Hanging Mug	17.00	**24.00**
Lift Pot	**97.00**	96.00
Move Can Pot	70.00	**86.00**
Move Pillbottle Pad	41.00	**58.00**
Move Playingcard Away	68.00	**71.00**
Move Stapler Pad	12.00	**16.00**
Open Laptop	82.00	**88.00**
Open Microwave	61.00	**97.00**
Pick Diverse Bottles	52.00	**53.00**
Pick Dual Bottles	60.00	**66.00**
Place A2B Left	**46.00**	40.00
Place A2B Right	49.00	**52.00**
Place Bread Basket	26.00	**29.00**

Task	DP3	DSSP
Place Bread Skillet	19.00	**39.00**
Place Burger Fries	72.00	**81.00**
Place Can Basket	67.00	**83.00**
Place Cans Plasticbox	48.00	**88.00**
Place Container Plate	86.00	**95.00**
Place Dual Shoes	13.00	**19.00**
Place Empty Cup	65.00	**86.00**
Place Fan	36.00	**40.00**
Place Mouse Pad	**4.00**	**4.00**
Place Object Basket	**65.00**	62.00
Place Object Scale	**15.00**	10.00
Place Object Stand	60.00	**61.00**
Place Phone Stand	44.00	**60.00**
Place Shoe	**58.00**	49.00
Press Stapler	**69.00**	66.00
Put Bottles Dustbin	60.00	**83.00**
Put Object Cabinet	**72.00**	70.00
Rotate QRcode	**74.00**	66.00
Scan Object	**31.00**	29.00
Shake Bottle	98.00	**100.00**
Shake Bottle Horizontally	**100.00**	**100.00**
Stack Blocks Three	1.00	**2.00**
Stack Blocks Two	24.00	**30.00**
Stack Bowls Three	57.00	**80.00**
Stack Bowls Two	83.00	**93.00**
Stamp Seal	18.00	**32.00**
Turn Switch	46.00	**57.00**
Average	55.24	**62.30**

D.2 Memory-Dependent Evaluation on RMBench

To further evaluate the memory capabilities of DSSP, we conduct additional experiments on RM-Bench [7], a benchmark specifically designed for memory-dependent robotic manipulation. We select three representative tasks with distinct memory requirements: *Put Back Block*, which requires remembering an object's initial position; *Swap T*, which requires retaining historical position and orientation information; and *Blocks Ranking Try*, which requires tracking previous attempts and their outcomes.

We compare DSSP with imitation-learning baselines, including DP[8], MambaPolicy[3], Diff-Control[36], MTIL[68], and CycleManip[54], as well as memory-oriented VLA baselines, including $\pi_{0.5}$[22], Mem-0[7], MemoryVLA[48], and EventVLA[59]. Our policy uses RGB observations from three camera views, consisting of one main camera and two wrist-mounted cameras.

As shown in Table 13, DSSP achieves the highest average success rate of 93.33%, outperforming the strongest competing baseline, EventVLA (87.67%), by 5.66 percentage points. In particular, DSSP achieves 100% on Put Back Block and 93% on Swap T, demonstrating its ability to retain task-relevant spatial information. It also achieves 87% on Blocks Ranking Try, indicating its capability to utilize historical feedback across multiple attempts. These results further demonstrate the effectiveness of full-history encoding for memory-dependent manipulation.

D.3 Effect of History Latent Dimension

We investigate the effect of history latent dimensionality on the six-task RoboTwin analysis subset, keeping all other training and evaluation settings unchanged. As shown in Table 14, reducing the dimension from 512 to 256 decreases the average success rate from 71.33% to 66.67%, while increasing it to 1024 yields 69.83%. These results suggest that the default dimension of 512 provides

sufficient representation capacity for the evaluated tasks without introducing unnecessary model complexity.

D.4 Dynamics Predictor Analysis

To assess whether the dynamics-aware objective encourages the learned context to capture action-dependent state transitions, we conduct additional evaluations on the six-task RoboTwin analysis subset.

Specifically, we compare predictions conditioned on the executed action with those conditioned on a temporally mismatched action sampled from another timestep of the same rollout, while keeping the history context fixed. We evaluate prediction quality using two metrics: (1) *Transition Cosine*, which measures the alignment between the predicted latent transition $\hat{z}_{t+1} - z_t$ and the actual transition $z_{t+1} - z_t$; and (2) *Decoded Joint-state RMSE*, which measures the error in predicting 12 arm-joint positions through an evaluation-only linear probe trained on offline demonstrations and frozen during evaluation.

As shown in Table 15, conditioning on the executed action yields higher transition alignment and substantially lower joint-state prediction error than using a mismatched action. These results indicate that the dynamics-aware predictor captures action-dependent transition information, supporting its role in learning predictive history representations.

D.5 Failure-Mode Analysis on Real-World Tasks

We further analyze the failure modes of each real-world task to better understand where the gains of DSSP come from.

Morse Tapping. DSSP achieves the largest relative improvement on this task. The policy is required to tap the target three times before returning, which demands accurate tracking of the number of completed taps. However, the visual observation before each tap is nearly identical, making the task progress ambiguous when only short-term context is available. As a result, DP3 often stops at an incorrect stage or performs redundant taps. By maintaining a history context, DSSP better tracks the tapping progress and executes the correct number of taps. The remaining failures of DSSP mainly come from inaccurate target localization, which can cause missed or imprecise contacts.

Put Bottles. This long-horizon task requires the robot to sequentially place multiple bottles into the basket. A common failure mode of DP3 is premature termination after placing only one bottle. This happens because intermediate states, where one bottle has already been placed, can appear visually similar to the final completion state when the policy only observes a short recent context. With historical context, DSSP better infers the overall task progress and distinguishes intermediate subgoals from true task completion. Its failures are more often caused by local manipulation errors, such as inaccurate grasping or placement, rather than losing track of the task stage.

Table 13: **Results on memory-dependent RMBench tasks.** Success rates (%) across three tasks requiring different forms of historical memory. The best result in each column is highlighted in bold.

Method	Put Back Block	Swap T	Blocks Ranking Try	Average
DP	0	2	0	0.67
MambaPolicy	0	2	12	4.67
Diff-Control	0	14	9	7.67
MTIL	37	35	41	37.67
CycleManip	0	88	**94**	60.67
$\pi_{0.5}$	11	15	6	10.67
Mem-0	90	14	18	40.67
MemoryVLA (OpenVLA)	50	9	12	23.67
MemoryVLA (QwenOFT)	81	9	53	47.67
EventVLA (visual anchors only)	95	87	81	87.67
DSSP (Ours)	**100**	**93**	87	**93.33**

Table 14: **Effect of history latent dimension.** Average success rates (%) across six RoboTwin analysis tasks.

Latent Dimension	256	512	1024
Success Rate (%)	66.67	**71.33**	69.83

Table 15: **Dynamics predictor evaluation.** Comparison of state transition prediction under executed and temporally mismatched actions.

Metric	Correct Action	Mismatched Action
Transition Cosine ↑	$\mathbf{0.108}$	0.056
Joint-state RMSE (rad) ↓	$\mathbf{0.036}$	0.201

Object Swap. This task requires swapping two objects through an intermediate buffer slot, with demonstrations collected in both swap directions. When an object is located in the buffer, the current observation alone is insufficient to determine whether it should be moved to the left or to the right. Consequently, DP3 may move the object back to its original location, leading to a reversal of progress. In contrast, DSSP uses the historical context to infer the previous movement direction and resolve this ambiguity. This allows the policy to maintain consistent task progress across visually aliased intermediate states.

E Theoretical Analysis

We provide the formal proofs for the propositions presented in the main text regarding the theoretical advantages of history conditioning in imitation learning. We analyze the diffusion noise-prediction loss directly. This distinction is important: its Bayes-optimal predictor is a conditional denoiser, not the conditional mean of the clean expert action. As a result, the loss is sensitive to the entire conditional action distribution, including changes that preserve both its mean and variance.

Let $A := A_t \in \mathbb{R}^d$, $H := H_t$, and $O := O_t$, with H and O taking values in standard Borel spaces so that regular conditional laws exist. Let K be a diffusion step sampled from a finite schedule, independently of (A, H, O), and let $\epsilon \sim \mathcal{N}(0, I_d)$ be independent noise. At step k, the noised action is

$$Z_k = \alpha_k A + \sigma_k \epsilon. \tag{17}$$

We assume $\mathbb{E}\|A\|^2 < \infty$ and that at least one step k with $P(K = k) > 0$ is nondegenerate, i.e., $\alpha_k \neq 0$ and $\sigma_k > 0$. Standard diffusion schedules satisfy this condition. For $X \in \{O, H\}$ and a measurable denoiser f with finite second moment, define

$$\mathcal{L}_{\text{diff}}(f; X) := \mathbb{E}\Big[\|\epsilon - f(Z_K, K, X)\|^2\Big], \qquad \mathcal{L}^*(X) := \inf_f \mathcal{L}_{\text{diff}}(f; X). \tag{18}$$

Here the infimum is over all such measurable denoisers, so $\mathcal{L}^*$ is a population Bayes risk. If the parameterized denoiser class realizes the corresponding Bayes predictors, then $\mathcal{L}^*(X) = \inf_\theta \mathcal{L}_{\text{diff}}(\pi_\theta; X)$. For restricted neural classes, the non-strict result below still follows when the history-conditioned class contains the reactive class, whereas strictness additionally requires sufficient realizability.

E.1 History cannot increase the optimal population diffusion loss.

Proposition E.1. *The optimal population diffusion loss satisfies*

$$\mathcal{L}^*(H) \leq \mathcal{L}^*(O). \tag{19}$$

Proof. We first identify the optimal denoiser for an arbitrary conditioning variable X. Let

$$f_X^*(Z_K, K, X) := \mathbb{E}[\epsilon \mid Z_K, K, X]. \tag{20}$$

For any measurable denoiser $f(Z_K, K, X)$ with finite second moment, add and subtract f_X^* inside the prediction error:

$$\epsilon - f = (\epsilon - f_X^*) + (f_X^* - f). \tag{21}$$

Expanding the squared norm and taking expectations gives

$$\begin{aligned}\mathcal{L}_{\text{diff}}(f; X) = \mathbb{E}\|\epsilon - f_X^*\|^2 + \mathbb{E}\|f_X^* - f\|^2 \\ + 2\mathbb{E}\big[(\epsilon - f_X^*)^\top (f_X^* - f)\big].\end{aligned} \tag{22}$$

The last term is zero. Indeed, $f_X^* - f$ is measurable with respect to $\sigma(Z_K, K, X)$, whereas the defining property of conditional expectation gives

$$\mathbb{E}[\epsilon - f_X^* \mid Z_K, K, X] = 0. \tag{23}$$

Thus, by the tower property,

$$\begin{aligned}&\mathbb{E}\big[(\epsilon - f_X^*)^\top (f_X^* - f)\big] \\ &\quad = \mathbb{E}\big[\mathbb{E}\big[(\epsilon - f_X^*)^\top (f_X^* - f) \mid Z_K, K, X\big]\big] \\ &\quad = \mathbb{E}\big[\mathbb{E}[\epsilon - f_X^* \mid Z_K, K, X]^\top (f_X^* - f)\big] = 0.\end{aligned} \tag{24}$$

Substituting this equality into (22) yields

$$\mathcal{L}_{\text{diff}}(f; X) = \mathbb{E}\|\epsilon - f_X^*\|^2 + \mathbb{E}\|f_X^* - f\|^2. \tag{25}$$

The second term is nonnegative and vanishes for $f = f_X^*$. Therefore f_X^* is a Bayes-optimal denoiser and

$$\mathcal{L}^*(X) = \mathbb{E}\|\epsilon - f_X^*(Z_K, K, X)\|^2. \tag{26}$$

We now compare observation conditioning with history conditioning. For brevity, write

$$f_O^* := f_O^*(Z_K, K, O), \qquad f_H^* := f_H^*(Z_K, K, H). \tag{27}$$

Because O is a measurable function of H, any function of (Z_K, K, O) is also a function of (Z_K, K, H). In particular, $f_H^* - f_O^*$ is measurable with respect to $\sigma(Z_K, K, H)$. Decompose the observation-conditioned residual as

$$\epsilon - f_O^* = (\epsilon - f_H^*) + (f_H^* - f_O^*). \tag{28}$$

After squaring and taking expectations, we obtain

$$\begin{aligned}\mathcal{L}^*(O) = \mathbb{E}\|\epsilon - f_H^*\|^2 + \mathbb{E}\|f_H^* - f_O^*\|^2 \\ + 2\mathbb{E}\big[(\epsilon - f_H^*)^\top (f_H^* - f_O^*)\big].\end{aligned} \tag{29}$$

Again, the cross term is zero. To see this, condition on all the information available to the history-conditioned denoiser:

$$\begin{aligned}&\mathbb{E}\big[(\epsilon - f_H^*)^\top (f_H^* - f_O^*) \mid Z_K, K, H\big] \\ &\quad = \mathbb{E}[\epsilon - f_H^* \mid Z_K, K, H]^\top (f_H^* - f_O^*) = 0.\end{aligned} \tag{30}$$

Taking the outer expectation preserves zero. Moreover, $\mathbb{E}\|\epsilon - f_H^*\|^2 = \mathcal{L}^*(H)$ by (26). Consequently,

$$\mathcal{L}^*(O) = \mathcal{L}^*(H) + \mathbb{E}\Big[\|f_H^*(Z_K, K, H) - f_O^*(Z_K, K, O)\|^2\Big]. \tag{31}$$

Rearranging gives

$$\mathcal{L}^*(O) - \mathcal{L}^*(H) = \mathbb{E}\Big[\|f_H^*(Z_K, K, H) - f_O^*(Z_K, K, O)\|^2\Big] \geq 0. \tag{32}$$

The final inequality follows because a squared Euclidean norm is nonnegative. Hence additional conditioning information cannot increase the Bayes risk. Equality holds if and only if $f_H^*(Z_K, K, H) = f_O^*(Z_K, K, O)$ almost surely. □

E.2 Distributional aliasing strictly decreases the optimal loss.

Proof structure. Proposition E.1 already shows that the two optimal losses are equal exactly when the two optimal denoisers are equal. The only technically difficult step is to show that equality of the denoisers is equivalent to equality of the complete conditional action distributions. We isolate that step in the following lemma. Proposition E.2 then follows from the lemma and the definition of conditional mutual information.

Lemma 1 (The optimal denoiser identifies the conditional action distribution)**.** *Under the assumptions above,*

$$f_H^*(Z_K, K, H) = f_O^*(Z_K, K, O) \quad \textit{almost surely} \tag{33}$$

if and only if

$$P_E(A \in \cdot \mid H) = P_E(A \in \cdot \mid O) \quad \textit{almost surely}. \tag{34}$$

Proof. We prove the two directions separately. The forward direction, from equal denoisers to equal conditional action distributions, contains the main technical argument and has four steps. First, we select a diffusion step that contains both clean-action signal and Gaussian noise. Second, we derive an explicit formula connecting the optimal denoiser to the conditional density of the noisy action. Third, we use this formula to show that equal denoisers imply equal conditional distributions of the noisy action. Fourth, we show that equality of the noisy-action distributions implies equality of the clean-action distributions. The reverse direction is Step 5 and follows by applying the same diffusion corruption to equal clean-action distributions.

Step 1: Select a diffusion step containing both signal and noise. Suppose first that the two optimal denoisers are equal almost surely. We restrict this equality to one sampled diffusion step at which the clean action has a nonzero coefficient and the Gaussian noise has a positive coefficient. Choose a diffusion step k such that

$$P(K = k) > 0, \qquad \alpha_k \neq 0, \qquad \sigma_k > 0. \tag{35}$$

Such a step exists by assumption. Since $P(K = k) > 0$, equality of the denoisers in the statement of the lemma implies

$$\mathbb{E}[\epsilon \mid Z_k, H] = \mathbb{E}[\epsilon \mid Z_k, O] \quad \text{almost surely}. \tag{36}$$

Step 2: Relate each optimal denoiser to its noisy-action density. In this step, we derive the optimal noise prediction from the conditional density of the noisy action, first under history conditioning and then under observation conditioning. We next express both sides of (36) in terms of the conditional density of Z_k. Fix a history value h, and let

$$\mu_h(B) := P_E(A \in B \mid H = h) \tag{37}$$

be the conditional action distribution for each measurable set $B \subseteq \mathbb{R}^d$. Define the Gaussian density

$$g_{\sigma_k}(x) := \frac{1}{(2\pi\sigma_k^2)^{d/2}} \exp\left(-\frac{\|x\|^2}{2\sigma_k^2}\right). \tag{38}$$

Conditional on $A = a$ and $H = h$, the random variable $Z_k = \alpha_k A + \sigma_k \epsilon$ has density $g_{\sigma_k}(z - \alpha_k a)$. Averaging this density over $A \mid H = h$ gives the density of $Z_k \mid H = h$:

$$q_h(z) = \int_{\mathbb{R}^d} g_{\sigma_k}(z - \alpha_k a)\, \mu_h(da). \tag{39}$$

The integrand is strictly positive for every z and a, so $q_h(z) > 0$ for every z. Moreover, differentiating the Gaussian density gives

$$\nabla_z g_{\sigma_k}(z - \alpha_k a) = -\frac{z - \alpha_k a}{\sigma_k^2} g_{\sigma_k}(z - \alpha_k a). \tag{40}$$

Using (40) inside (39), we obtain

$$\begin{aligned} \nabla_z q_h(z) &= -\frac{1}{\sigma_k^2} \int_{\mathbb{R}^d} (z - \alpha_k a) g_{\sigma_k}(z - \alpha_k a)\, \mu_h(da) \\ &= -\frac{q_h(z)}{\sigma_k^2} \left(z - \mathbb{E}[\alpha_k A \mid Z_k = z, H = h]\right). \end{aligned} \tag{41}$$

The second equality follows from the conditional density of A after observing $Z_k = z$ and $H = h$: for every integrable function r,

$$\mathbb{E}[r(A) \mid Z_k = z, H = h] = \frac{\int_{\mathbb{R}^d} r(a) g_{\sigma_k}(z - \alpha_k a)\, \mu_h(da)}{q_h(z)}. \tag{42}$$

Dividing (41) by the positive number $q_h(z)$ gives

$$\nabla_z \log q_h(z) = -\frac{1}{\sigma_k^2} \left(z - \mathbb{E}[\alpha_k A \mid Z_k = z, H = h]\right). \tag{43}$$

On the other hand, the forward diffusion equation implies

$$\epsilon = \frac{Z_k - \alpha_k A}{\sigma_k}. \tag{44}$$

Therefore,

$$\begin{aligned} \mathbb{E}[\epsilon \mid Z_k = z, H = h] &= \frac{z - \mathbb{E}[\alpha_k A \mid Z_k = z, H = h]}{\sigma_k} \\ &= -\sigma_k \nabla_z \log q_h(z). \end{aligned} \tag{45}$$

The same calculation can be performed while conditioning only on $O = o$. Let

$$\mu_o(B) := P_E(A \in B \mid O = o) \tag{46}$$

and define

$$q_o(z) := \int_{\mathbb{R}^d} g_{\sigma_k}(z - \alpha_k a)\, \mu_o(da). \tag{47}$$

Then $q_o(z) > 0$ for every z, and the identical calculation yields

$$\mathbb{E}[\epsilon \mid Z_k = z, O = o] = -\sigma_k \nabla_z \log q_o(z). \tag{48}$$

Step 3: Use equal denoisers to obtain equal noisy-action distributions. The formulas from Step 2 turn equality of the two denoisers into equality of derivatives of the two log densities. We then show that two normalized densities with these equal derivatives must themselves be equal. We now return to the equality of the denoisers. For almost every history h, let o be the current-observation component of h. Equations (36), (45), and (48) imply

$$\nabla_z \log q_h(z) = \nabla_z \log q_o(z) \quad \text{for } q_h(z)\,dz\text{-almost every } z. \tag{49}$$

Because $q_h(z) > 0$ everywhere, an equality holding for $q_h(z)\,dz$-almost every z also holds for ordinary Lebesgue-almost every z. Both sides of (49) are continuous in z; hence they are equal for every z. It follows that

$$\nabla_z(\log q_h(z) - \log q_o(z)) = 0 \quad \text{for every } z \in \mathbb{R}^d. \tag{50}$$

Since $\mathbb{R}^d$ is connected, a differentiable function with zero gradient everywhere is constant. Thus, for some constant $c_h > 0$,

$$q_h(z) = c_h q_o(z) \quad \text{for every } z. \tag{51}$$

Both sides are probability densities, so their integrals over $\mathbb{R}^d$ equal one:

$$1 = \int q_h(z)\,dz = c_h \int q_o(z)\,dz = c_h. \tag{52}$$

Consequently,

$$q_h(z) = q_o(z) \quad \text{for every } z \quad \text{and for almost every } h. \tag{53}$$

Step 4: Recover equality of the clean-action distributions. Step 3 shows that the two clean-action distributions become equal after the same Gaussian noise is added. We now show explicitly that this can happen only when the two clean-action distributions were already equal. It remains to show that equality after adding Gaussian noise implies equality before adding noise. For any random vector $Y \in \mathbb{R}^d$, define

$$\varphi_Y(u) := \mathbb{E}[e^{iu^\top Y}], \qquad u \in \mathbb{R}^d. \tag{54}$$

This function uniquely determines the probability distribution of Y. From (53), the conditional distributions of $Z_k \mid H = h$ and $Z_k \mid O = o$ are equal, and hence

$$\mathbb{E}[e^{iu^\top Z_k} \mid H = h] = \mathbb{E}[e^{iu^\top Z_k} \mid O = o] \quad \text{for every } u \in \mathbb{R}^d. \tag{55}$$

Using $Z_k = \alpha_k A + \sigma_k \epsilon$ and the independence of ϵ from (A, H, O), the left-hand side is

$$\begin{aligned}\mathbb{E}[e^{iu^\top Z_k} \mid H = h] &= \mathbb{E}[e^{i\alpha_k u^\top A} \mid H = h]\mathbb{E}[e^{i\sigma_k u^\top \epsilon}] \\ &= \varphi_{A|h}(\alpha_k u) \exp\left(-\frac{\sigma_k^2 \|u\|^2}{2}\right).\end{aligned} \tag{56}$$

Similarly,

$$\mathbb{E}[e^{iu^\top Z_k} \mid O = o] = \varphi_{A|o}(\alpha_k u) \exp\left(-\frac{\sigma_k^2 \|u\|^2}{2}\right). \tag{57}$$

Substituting (56) and (57) into (55) gives

$$\varphi_{A|h}(\alpha_k u) e^{-\sigma_k^2 \|u\|^2/2} = \varphi_{A|o}(\alpha_k u) e^{-\sigma_k^2 \|u\|^2/2} \quad \text{for every } u. \tag{58}$$

The factor $e^{-\sigma_k^2 \|u\|^2/2}$ is strictly positive, so it can be cancelled. Because $\alpha_k \neq 0$, every $v \in \mathbb{R}^d$ can be written as $v = \alpha_k u$. We therefore obtain

$$\varphi_{A|h}(v) = \varphi_{A|o}(v) \quad \text{for every } v \in \mathbb{R}^d. \tag{59}$$

Since the function in (54) uniquely determines a probability distribution, this proves

$$P_E(A \in \cdot \mid H = h) = P_E(A \in \cdot \mid O = o) \tag{60}$$

for almost every h, which is (34).

Step 5: Prove the reverse direction. We finally assume that the conditional clean-action distributions are equal and show that the corresponding optimal denoisers must be equal. This direction does not require the density calculations above. Suppose (34) holds. Then, for almost every history h and its current observation o, the conditional laws of $A \mid H = h$ and $A \mid O = o$ are equal. Since ϵ is independent of (A, H, O), the conditional laws of (A, ϵ) are also equal under these two conditioning events. Applying the same deterministic map

$$(a, e) \longmapsto (\alpha_k a + \sigma_k e, e) \tag{61}$$

shows that the conditional joint laws of (Z_k, ϵ) are equal for every sampled diffusion step k. Equal joint laws give equal conditional means, so

$$\mathbb{E}[\epsilon \mid Z_k, H] = \mathbb{E}[\epsilon \mid Z_k, O] \tag{62}$$

for every diffusion step k of positive sampling probability. Hence $f_H^*(Z_K, K, H) = f_O^*(Z_K, K, O)$ almost surely, which proves the reverse direction. Steps 1–4 prove the forward direction and Step 5 proves the reverse direction, so the two statements in the lemma are equivalent. □

Proposition E.2. *Under the assumptions above, history conditioning strictly reduces the optimal population diffusion loss if and only if the available history contains additional information about the expert action:*

$$\mathcal{L}^*(H) < \mathcal{L}^*(O) \quad \Longleftrightarrow \quad I_E(A; H \mid O) > 0. \tag{63}$$

In particular, the strict inequality holds under the history-resolvable aliasing condition in definition 1, even when the conditional action laws have identical means and variances.

Proof. We first characterize when the two losses are equal. From (32),

$$\mathcal{L}^*(H) = \mathcal{L}^*(O) \quad \Longleftrightarrow \quad f_H^*(Z_K, K, H) = f_O^*(Z_K, K, O) \quad \text{almost surely.} \tag{64}$$

By lemma 1, this is equivalent to

$$P_E(A \in \cdot \mid H) = P_E(A \in \cdot \mid O) \quad \text{almost surely.} \tag{65}$$

On the other hand, by (2),

$$I_E(A; H \mid O) = \mathbb{E}[\mathrm{KL}(P_E(A \in \cdot \mid H) \,\|\, P_E(A \in \cdot \mid O))]\,. \tag{66}$$

The quantity inside this expectation is nonnegative and equals zero exactly when its two probability distributions are equal. Therefore,

$$I_E(A; H \mid O) = 0 \quad \Longleftrightarrow \quad P_E(A \in \cdot \mid H) = P_E(A \in \cdot \mid O) \quad \text{almost surely.} \tag{67}$$

Combining these equivalences gives

$$\mathcal{L}^*(H) = \mathcal{L}^*(O) \quad \Longleftrightarrow \quad I_E(A; H \mid O) = 0. \tag{68}$$

Taking the complementary cases and using $\mathcal{L}^*(H) \le \mathcal{L}^*(O)$ and $I_E(A; H \mid O) \ge 0$ proves

$$\mathcal{L}^*(H) < \mathcal{L}^*(O) \quad \Longleftrightarrow \quad I_E(A; H \mid O) > 0. \tag{69}$$

□

Scope of the result. The propositions compare optimal population diffusion training losses under the expert data distribution. They do not by themselves guarantee lower finite-sample error, successful optimization, or higher closed-loop task return. Those conclusions require additional assumptions on model capacity, estimation, optimization, and distribution shift. The result isolates the precise distributional condition under which the available history can, in principle, improve diffusion-based imitation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our contributions and scope are completely and accurately reflected in the end of the introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have included the section of limitations in Appendix 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We have included in Section E the theoretical proof of the propositions proposition 4.1 and proposition 4.2.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have included implementation details in Section C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We include implementation details such as hyperparameter in Section C and plan to release the code upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We have specified the training and test details in Section C and Section 5.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Our evaluation detail is in Section 5.1 and the experiment figure also shows the statistical significance in Table 2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have stated our compute resources in section 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the NeurIPS Code of Ethics carefully and conformed with it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have included it in the Section 6.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work has no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: In our work, we do experiments on RoboTwin, Adroit and Meta-World and have cited them properly in the paragraph of experimental setup in Section 5.1.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: Our work does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research. The core method development in our work does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.