HiPolicy: Hierarchical Multi-Frequency Action Chunking for Policy Learning

Jiyao Zhang[1,2], Zimu Han[3], Junhan Wang[1], Xionghao Wu[4], Shihong Lin[5], Jinzhou Li[1], Hongwei Fan[1,2], Ruihai Wu[1], Dongjiang Li[6], and Hao Dong[1,2]

[1] CFCS, School of CS, PKU, China
[2] National Key Laboratory for Multimedia Information Processing, School of CS, PKU, China
[3] XJTU, China [4] THU, China [5] BUAA, China
[6] Jingdong Technology Information Technology Co., Ltd, China
jiyaozhang@stu.pku.edu.cn

https://hipolicy.github.io

Abstract. Robotic imitation learning faces a fundamental trade-off between modeling long-horizon dependencies and enabling fine-grained closed-loop control. Existing fixed-frequency action chunking approaches struggle to achieve both. Building on this insight, we propose **HiPolicy**, a hierarchical multi-frequency action chunking framework that jointly predicts action sequences at different frequencies to capture both coarse high-level plans and precise reactive motions. We extract and fuse hierarchical features from history observations aligned to each frequency for multi-frequency chunk generation, and introduce an entropy-guided execution mechanism that adaptively balances long-horizon planning with fine-grained control based on action uncertainty. Experiments on diverse simulated benchmarks and real-world manipulation tasks show that HiPolicy can be seamlessly integrated into existing 2D and 3D generative policies, delivering consistent improvements in performance while significantly enhancing execution efficiency.

Keywords: Manipulation · Imitation learning · Hierarchical policy

1 Introduction

Learning from human demonstrations has emerged as a powerful paradigm for robotic manipulation [4, 9, 13, 33, 38–41], enabling policies to acquire complex skills without explicit reward engineering or exhaustive exploration. Imitation Learning (IL) [1, 7] has shown promise in both task-specific policy training and general-purpose vision-language-action (VLA) model [5, 12] development, providing a scalable route toward versatile robot control [2,4,17,35]. Among various IL approaches, generative policies such as

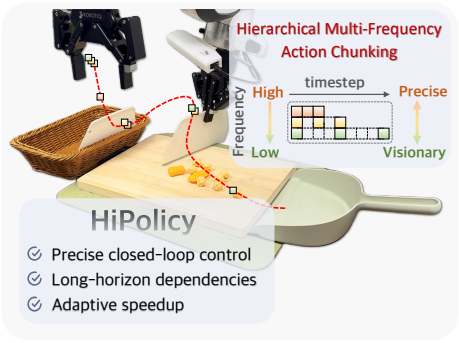

Fig. 1: We propose **HiPolicy**, a hierarchical multi-frequency action chunking for policy learning, modeling long-horizon dependency and precise closed-loop control.

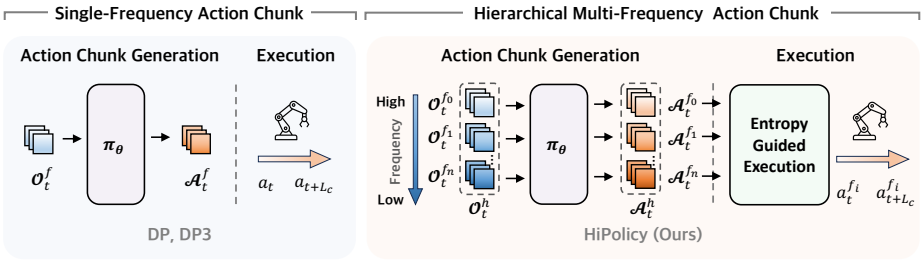

Fig. 2: Comparison of HiPolicy with Existing Methods. Existing imitation learning methods typically predict an action chunk at a single fixed frequency, leading to a trade-off between long-horizon dependency modeling and precise closed-loop control. In contrast, HiPolicy proposes hierarchical multi-frequency action chunking, enabling the capture of both long-term intentions and precise closed-loop adjustments. Additionally, the proposed action-entropy guided adaptive execution mechanism selects action frequency based on action uncertainty, balancing robustness and efficiency during task execution.

Diffusion Policy [4], ACT [42], and DP3 [40] have become dominant due to their ability to model continuous action distributions with high temporal fidelity, delivering strong performance across diverse manipulation tasks.

Despite the effectiveness of imitation learning, it still faces fundamental challenges. First, error accumulation remains a critical issue. Small discrepancies in predicted actions can compound over time, leading to substantial state deviations and eventual task failure [14,18,29]. Second, modeling long-horizon dependencies is hard, especially for non-Markovian behaviors frequently observed in human demonstrations, such as pauses or subtle oscillatory motions [19]. Recent work has sought to address this via action chunking [42], predicting multi-step sequences instead of single-step actions. While this improves temporal coherence, its effectiveness is tightly coupled to two hyperparameters: action frequency and chunk size. As shown in Figure 1, at the same chunk size, low-frequency chunks capture long-horizon dependencies but lack the temporal resolution for fine-grained closed-loop control, whereas high-frequency chunks offer fine-grained adjustments but are less effective at modeling long-horizon dependencies. This trade-off presents a significant challenge in designing effective action chunking strategies for robotic policies.

Human motor control studies reveal that complex movements naturally integrate components operating at multiple frequencies [8, 24, 34]. Long-duration, low-frequency motions encode high-level goals and stage intentions, while short-duration, high-frequency motions enable precise adjustments and reactive control. This raises a key question: *Could such a hierarchical multi-frequency structure be the key to uniting robust long-horizon planning with precise fine-grained closed-loop control in robotic policies?*

In this work, we present HiPolicy: a hierarchical multi-frequency action chunking framework for policy learning. As shown in Figure 2, our framework jointly

predicts action sequences at multiple frequencies from hierarchically aligned observation histories, fusing their representations to model long-horizon and fine-control behavior concurrently. To resolve the execution trade-off between speed and precision, we propose an action-entropy-guided adaptive execution strategy: Low-entropy frames indicate concentrated action distributions and stable predictions, prompting execution of high-frequency actions to enable precise closed-loop refinement. High-entropy frames reflect dispersed distributions and greater uncertainty, corresponding to broader phase-level decisions, where executing low-frequency actions aligns with high-level intent while increasing operational speed [10]. By dynamically selecting execution frequency based on policy uncertainty, our method speeds up the execution while balancing fine-grained control with long-horizon consistency.

Our contributions are threefold:

- We introduce HiPolicy, a novel hierarchical multi-frequency action chunking framework for policy learning that addresses the trade-off between long-horizon dependency modeling and high-frequency reactive control.
- We propose an action-entropy-guided adaptive execution mechanism that dynamically selects action frequency based on policy uncertainty, enhancing both robustness and execution efficiency.
- We validate our approach on extensive simulation benchmarks and real-world manipulation tasks, demonstrating consistent improvements in both performance and efficiency across 2D and 3D generative policies.

2 Related Work

2.1 Imitation Learning for Manipulation

Imitation learning (IL) has become a dominant learning paradigm for robots to acquire manipulation skills from human demonstrations [4, 40, 42]. Generative models such as Diffusion Models [11, 31] and Conditional Variational Autoencoders (CVAE) [16,27] are the primary drivers of this progress, enabling accurate and robust action distribution modeling [20, 27, 32, 36].

Diffusion Policy [4] integrates the diffusion process into generating the action sequence of robots, enabling accurate action prediction. 3D Diffusion Policy [40] and 3D Diffuser Actor [13] incorporate point cloud representations into the diffusion policy, enhancing the spatial perception ability for manipulation. Our Hierarchical Policy proposes a hierarchical modeling method based on different frequencies, which can be easily integrated into Diffusion Policy [4] and 3D Diffusion Policy [40], compensating their weakness in modeling non-Markovian processes by jointly predicting at hierarchical temporal resolutions. Furthermore, our action entropy-guided execution method chooses the suitable execution frequency adaptively, significantly improves the efficiency of manipulation, and relieves the low-speed problem of diffusion-based policies.

2.2 Hierarchical Manipulation Policy

Hierarchical structures in the area of computer vision can effectively perceive and integrate environmental information at different semantic levels, thereby enabling the model to take into account both global information and local details, and improving model performance.

In the robot learning community, recent research has sought to integrate hierarchical structures into imitation learning framework. [21, 23] H^3DP [22] proposes a visuomotor learning framework that explicitly strengthens the coupling between visual features and action generation by hierarchically conditioning on multi-scale visual representations. Reactive Diffusion Policy [39] introduces a hierarchical slow-fast visual-tactile imitation learning algorithm that uses a slow latent diffusion policy for high-level action chunks and a fast asymmetric tokenizer for low-level control to enable quick responses in contact-rich manipulation tasks. CARP [9] applies a next-scale autoregressive structure that decouples action generation into multi-scale representation learning via an action autoencoder. Dense Policy [33] introduces a new bidirectionally expanded learning approach for autoregressive policies, employing an encoder-only architecture to hierarchically unfold the action sequence in a coarse-to-fine manner.

However, the hierarchical structure of these policies only models perceptual input information at different levels, and few work studies have been conducted on imitation learning algorithms with a hierarchical structure for different temporal resolutions. In contrast, our Hierarchical Policy features temporal hierarchical modeling, which hierarchically divides the perceptual input and action sequence simultaneously according to different frequencies and trains the model to predict future action sequences at different time scales.

3 Method

3.1 Problem Formulation

We consider an imitation learning setting where the agent learns a generative policy:

$$\boldsymbol{\mathcal{A}}_{t,L_c} = \pi_\theta(\boldsymbol{\mathcal{O}}_{t,L_h}), \tag{1}$$

where t denotes the current time step, and L_c and L_h denote the action chunk length and observation history length, respectively. The policy π maximizes the likelihood of the actions given the observation by $P(\boldsymbol{\mathcal{A}}_{t,L_c}|\boldsymbol{\mathcal{O}}_{t,L_h})$. θ are the learnable parameters of the policy π_θ, and $\boldsymbol{\mathcal{O}}_{t,L_h}$ denotes the observation history over L_h time steps:

$$\boldsymbol{\mathcal{O}}_{t,L_h} = \{\boldsymbol{o}_{t-L_h+1}, \ldots, \boldsymbol{o}_t\}, \tag{2}$$

with $\boldsymbol{o}_t = \{v_t, s_t\}$, where v_t denotes the visual observation (either a 2D RGB image or a 3D point cloud) and $\boldsymbol{s}_t$ denotes proprioceptive states such as joint positions, velocities, and gripper status. The *action chunk* $\boldsymbol{\mathcal{A}}_{t,L_c}$ is defined as:

$$\boldsymbol{\mathcal{A}}_{t,L_c} = \{\boldsymbol{a}_t, \boldsymbol{a}_{t+1}, \ldots, \boldsymbol{a}_{t+L_c-1}\}, \tag{3}$$

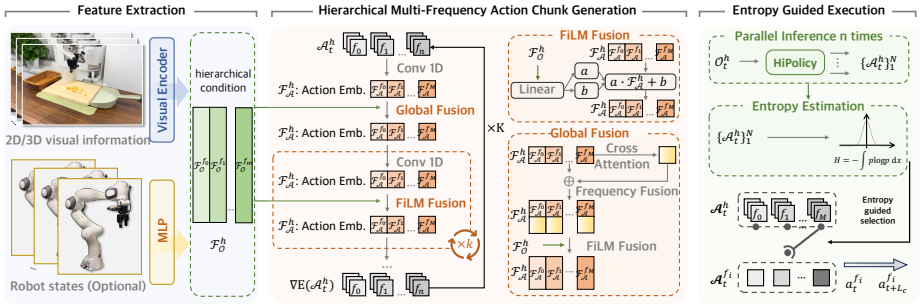

Fig. 3: Overview of HiPolicy. We propose HiPolicy, a hierarchical multi-frequency action chunk policy with an entropy-guided adaptive execution strategy. Given a hierarchical observation history, HiPolicy predicts multi-frequency action chunks simultaneously through a diffusion-based model. During inference, HiPolicy estimates the action entropy through multiple samplings and adaptively chooses the execution frequency according to the estimated entropy.

where $\boldsymbol{a}$ is a low-level control action and L_c is the chunk size, i.e., the number of actions executed before receiving the next observation and generating the subsequent chunk.

Prior formulation (single frequency). In conventional fixed-frequency chunking, both observation histories and action chunks are sampled at a single frequency f:

$$\boldsymbol{\mathcal{O}}^{f}_{t,L_h} = \{\boldsymbol{o}^{f}_{t-L_h+1}, \ldots, \boldsymbol{o}^{f}_{t}\}, \tag{4}$$

$$\boldsymbol{\mathcal{A}}^{f}_{t,L_c} = \{\boldsymbol{a}^{f}_{t}, \boldsymbol{a}^{f}_{t+1}, \ldots, \boldsymbol{a}^{f}_{t+L_c-1}\}. \tag{5}$$

This formulation poses a challenge in selecting the appropriate frequency f: low f supports long-horizon dependency modeling, but reduces opportunities for precise and closed-loop control. High f allows for fine-grained control but limits the field of view for long-horizon tasks.

HiPolicy formulation (hierarchical multi-frequency). HiPolicy jointly encodes multi-frequency observation histories:

$$\boldsymbol{\mathcal{O}}^{\mathrm{h}}_{t,L_h} = \bigcup_{m=1}^{M} \boldsymbol{\mathcal{O}}^{f_m}_{t,L_h}, \tag{6}$$

and predicts a single hierarchical multi-frequency action chunk:

$$\boldsymbol{\mathcal{A}}^{\mathrm{h}}_{t,L_c} = \bigcup_{m=1}^{M} \boldsymbol{\mathcal{A}}^{f_m}_{t,L_c}, \tag{7}$$

where each f_m represents a frequency component within the same chunk, and M is the number of frequencies. Unlike single-frequency approaches, the observation and action components at different frequencies are jointly generated and fused

within the chunk, enabling coarse high-level planning and fine-grained reactive control to communicate and synergize.

HiPolicy Overview. As shown in Figure 3, our HiPolicy is designed as a diffusion-based model, predicting action chunks with hierarchical time resolutions simultaneously. First, we extract hierarchical observation features from visual and proprioception input. Then we obtain the overall action chunk by concatenating noisy action chunks at different frequencies and predict the noise through a 1D-CNN-based U-Net [28]. Hierarchical FiLM fusion [26] and global feature fusion are two key fusion modules designed to promote fusing observation features with action and injecting global information across different frequencies. After predicting action chunks at hierarchical temporal resolutions, we perform the inference in parallel to estimate the action entropy and select the execution frequency according to the entropy adaptively.

We follow Diffusion Policy [4] to design our training process, which is detailed described in Appendix C.

3.2 Multi-Frequency Action Chunk Prediction

Hierarchical feature extraction. To obtain the hierarchical observation feature, we first extract the observation frames of different frequencies from the raw observation chunk. Then the hierarchical feature is obtained through the observation encoder. Input vision observation is selectable between 2D images and 3D point clouds. Robot state is also optional as proprioception information, and is encoded through MLP.

Hierarchical FiLM fusion. To model observation-action mappings at different frequencies, we use features of different frequencies as conditions for action chunks of different frequencies. As shown in Figure 3, we adopt the FiLM condition [26] method and impose the corresponding frequency of observation features to each frequency of action chunk. Then we get the hierarchical action feature $\boldsymbol{\mathcal{F}}_{\boldsymbol{\mathcal{A}}}^{\mathrm{h}}$ as follows:

$$\boldsymbol{\mathcal{F}}_{\boldsymbol{\mathcal{A}}}^{\mathrm{h}} \leftarrow \bigoplus_{m=1}^{M} \mathrm{FiLM}(\boldsymbol{\mathcal{F}}_{\boldsymbol{\mathcal{O}}}^{f_m}, \boldsymbol{\mathcal{F}}_{\boldsymbol{\mathcal{A}}}^{f_m}) \tag{8}$$

where $\bigoplus$ represents the concatenation on the temporal dimension, $\boldsymbol{\mathcal{F}}_{\boldsymbol{\mathcal{A}}}^{\mathrm{h}} \in \mathbb{R}^{B\times(MT)\times C}$ denotes the hierarchical action feature, and $\boldsymbol{\mathcal{F}}_{\boldsymbol{\mathcal{O}}}^{f_m} \in \mathbb{R}^{B\times T\times C}$,$\boldsymbol{\mathcal{F}}_{\boldsymbol{\mathcal{A}}}^{f_m} \in \mathbb{R}^{B\times T\times C}$ are the observation and action feature at frequency f_m. B is the batch size, C denotes the hidden feature dimension, and T refers to the temporal dimension. This hierarchical condition design aims to capture feature mapping relations at different temporal resolutions in a corresponding way.

Global feature fusion. As shown in Figure 3, we design a global feature fusion to facilitate information communication across different frequencies. We first apply frequency fusion, and a cross attention [37] module is adopted to get the global feature across different frequencies. Cross attention [37] module is adopted for its performance in capturing context information. Specifically, we add a CLS

Algorithm 1: Entropy-Guided Execution

Input : Observation $\boldsymbol{\mathcal{O}}^{\mathrm{h}}_{t,L_h}$
Output: Action for Execution $\boldsymbol{\mathcal{A}}^{\mathrm{exec}}_{t,L_c}$
$\boldsymbol{\mathcal{A}}^{\mathrm{h}}_{t,L_c} \leftarrow \emptyset$
for $i \leftarrow 1$ **to** N **do**
  $\boldsymbol{\mathcal{A}}^{i}_{t,L_c} \leftarrow \pi_{\mathrm{HiPolicy}}(\boldsymbol{\mathcal{O}}^{\mathrm{h}}_{t,L_h})$ ▷ Independently sample for N times
  $\boldsymbol{\mathcal{A}}^{\mathrm{h}}_{t,L_c} \leftarrow \boldsymbol{\mathcal{A}}^{\mathrm{h}}_{t,L_c} \cup \boldsymbol{\mathcal{A}}^{i}_{t,L_c}$
$\boldsymbol{\mathcal{A}}^{\mathrm{h^{exec}}}_{t,L_c} \leftarrow \boldsymbol{\mathcal{A}}^{1}_{t,L_c}$ ▷ Use the first sample output for execution
$H \leftarrow \mathit{CalculateEntropy}(\boldsymbol{\mathcal{A}}^{\mathrm{h}}_{t,L_c})$ ▷ Calculate action entropy
$\boldsymbol{\mathcal{A}}^{\mathrm{exec}}_{t,L_c} \leftarrow \mathit{SelectFrequency}(H, \boldsymbol{\mathcal{A}}^{\mathrm{h^{exec}}}_{t,L_c})$ ▷ Select action according to entropy
return $\boldsymbol{\mathcal{A}}^{\mathrm{exec}}_{t,L_c}$

token [6] to obtain the global feature token. This process is written as follows.

$$\boldsymbol{\mathcal{F}}^{\mathrm{global}}_{\boldsymbol{\mathcal{A}}} = \mathrm{CrossAttention}(\boldsymbol{\mathcal{F}}^{f_0}_{\boldsymbol{\mathcal{A}}}, \boldsymbol{\mathcal{F}}^{f_1}_{\boldsymbol{\mathcal{A}}}, ..., \boldsymbol{\mathcal{F}}^{f_M}_{\boldsymbol{\mathcal{A}}}) \tag{9}$$

where $\boldsymbol{\mathcal{F}}^{f_i}_{\boldsymbol{\mathcal{A}}} \in \mathbb{R}^{B\times T\times C}$ is the local action feature for frequency f_i, $\boldsymbol{\mathcal{F}}^{\mathrm{global}}_{\boldsymbol{\mathcal{A}}} \in \mathbb{R}^{B\times T\times C}$ is the global feature across different frequencies. Then we concatenate the global feature and the local feature to obtain hierarchical features.

$$\boldsymbol{\mathcal{F}}^{\mathrm{h}}_{\boldsymbol{\mathcal{A}}} \leftarrow \boldsymbol{\mathcal{F}}^{\mathrm{h}}_{\boldsymbol{\mathcal{A}}} \oplus \boldsymbol{\mathcal{F}}^{\mathrm{global}}_{\boldsymbol{\mathcal{A}}} \tag{10}$$

After frequency fusion, we use a FiLM condition, which is performed as Equation 8, to promote the fusion of action feature and observation feature.

3.3 Entropy-Guided Adaptive Execution

To balance the fine-grained closed-loop control and the capability to capture time-dependent information, while speeding up the execution speed of HiPolicy, we propose an entropy-guided execution strategy to perform our policy at an adaptive frequency. As shown in Figure 4, low entropy indicates high consistency (e.g., during grasping), justifying the execution of high-frequency actions for precise, closed-loop control. Conversely, high entropy signals casual motion (e.g., during approaching); here, adaptively switching to low-frequency actions enables high-level planning and significantly accelerates execution. The complete algorithm of entropy-guided execution is shown in Algorithm 1.

To obtain the entropy of an action chunk, we first sample in parallel for N times independently to get N predicted actions when evaluating our HiPolicy, described in the right part of Figure 3. Considering action chunks predicted at a certain frequency, we model the j-th action in the chunk as a continuous variable following a Gaussian distribution with probability distribution function:

$$\hat{p}(\boldsymbol{a}^j|\boldsymbol{o}) = \frac{1}{\sqrt{2\pi}\hat{\sigma}^j}\exp(-\frac{(\boldsymbol{a}^j - \bar{\boldsymbol{a}}^j)^2}{2(\hat{\sigma}^j)^2}) \tag{11}$$

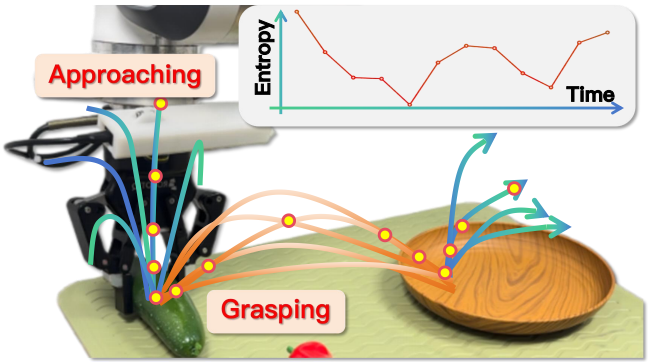

Fig. 4: Entropy-Guided Execution. HiPolicy leverages action entropy, estimated through parallel stochastic inference, as a dynamic gating signal to arbitrate between predictions at multiple frequencies, ensuring a balance between long-horizon planning and reactive control.

where the mean and variance are calculated as follows:

$$\bar{\boldsymbol{a}}^j = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{a}_i^j \tag{12}$$

$$\hat{\sigma}^j = \frac{1}{N-1}\sum_{i=1}^{N} (\boldsymbol{a}_i^j - \bar{\boldsymbol{a}}^j)^2 \tag{13}$$

We use Shannon entropy [30] to measure the entropy of the predicted action:

$$\hat{H}^j = -\int \hat{p}(\boldsymbol{a}^j|\boldsymbol{o}) \log(\hat{p}(\boldsymbol{a}^j|\boldsymbol{o}))\mathrm{d}\boldsymbol{a}^j \tag{14}$$

The simplified formula of action entropy is as follows:

$$\hat{H}^j = \log(\sqrt{2\pi \mathrm{e}} \cdot \hat{\sigma}^j) \tag{15}$$

Then the overall action entropy at time t is the average entropy of all frequencies and all time steps:

$$\hat{H}_t = \frac{1}{ML_c}\sum_{m=1}^{M}\sum_{j=t}^{t+L_c-1} \hat{H}_t^{f_m,j} \tag{16}$$

where $\hat{H}_t^{f_m,j}$ denotes action entropy for j-th action in the chunk at frequency f_m, calculated following Equation 15.

After estimating the action entropy, we choose execution frequency by comparing the estimated entropy with preset ascending thresholds $\{\hat{H}_{\mathrm{th},1}, ..., \hat{H}_{\mathrm{th},M+1}\}$, with frequencies sorted in descending order:

$$f_i = f_k, \text{ if } H_t \in (H_{\mathrm{th},k}, H_{\mathrm{th},k+1}) \tag{17}$$

Table 1: Detailed evaluation results on RoboTwin 1.0 and RoboTwin 2.0 Benchmark. Results (success rate) are based on 100 evaluation episodes, distinguishing between precise and other tasks. Bold red highlights the relative percentage improvement over baseline methods, while color shades are used to categorize tasks by their horizon (short, long, and super long). Tasks are sorted according to demo steps.

Task name	Task Source	Precise	Demo steps	DP [4]	DP+HiPolicy	DP3 [40]	DP3+HiPolicy
Blocks stack hard	RoboTwin 1.0	✓	665	0	**10**	4	**7**
Blocks stack easy	RoboTwin 1.0	✓	437	0	**15**	27	**44**
Block handover	RoboTwin 1.0	✓	415	61	**100**	45	**99**
Dual shoes place	RoboTwin 1.0	✓	384	4	**15**	13	**14**
Shoe place	RoboTwin 1.0	✓	251	33	**48**	49	**66**
Place bread basket	RoboTwin 2.0	✓	231	41	**64**	15	**47**
Block hammer beat	RoboTwin 1.0	✓	211	0	**67**	57	**94**
Scan object	RoboTwin 2.0	✓	170	21	**44**	12	**52**
Place bread skillet	RoboTwin 2.0	✓	162	38	**55**	45	**58**
Stamp seal	RoboTwin 2.0	✓	151	5	**19**	15	**74**
Place mouse pad	RoboTwin 2.0	✓	149	13	**29**	3	**15**
Place phone stand	RoboTwin 2.0	✓	130	45	**79**	55	**64**
Average				22	**45 (105%↑)**	28	**53 (89%↑)**
Put bottles dustbin	RoboTwin 2.0	×	637	38	**81**	66	**80**
Stack bowls three	RoboTwin 2.0	×	476	80	**88**	84	**86**
Empty cup place	RoboTwin 1.0	×	269	89	**95**	54	**92**
Open laptop	RoboTwin 2.0	×	258	69	**86**	82	**86**
Bottle adjust	RoboTwin 1.0	×	194	46	**52**	81	**85**
Dual bottles pick hard	RoboTwin 1.0	×	192	60	**72**	53	**60**
Diverse bottles pick	RoboTwin 1.0	×	182	24	**65**	44	**50**
Dual shoes pick easy	RoboTwin 1.0	×	179	82	**94**	56	**89**
Pick apple messy	RoboTwin 1.0	×	167	40	**81**	**10**	3
Average				59	**79 (34%↑)**	59	**70 (19%↑)**
Total average				37	**60 (62%↑)**	41	**59 (44%↑)**

where f_i is the selected frequency. Finally, the execution action chunk is:

$$\boldsymbol{\mathcal{A}}_{t,L_c}^{\text{exec}} = \{\boldsymbol{a}_t^{f_i}, \ldots, \boldsymbol{a}_{t+L_c-1}^{f_i}\}, \tag{18}$$

4 Experiments

4.1 Experiments on Simulation Benchmarks

Benchmarks and demonstrations. We evaluate the HiPolicy on two widely-used robotic manipulation simulation benchmarks covering a diverse set of tasks ranging from fine-grained to long-horizon manipulation:

- **RoboTwin 1.0** [25]: RoboTwin 1.0 is a digital twin framework introduced to address the scarcity of diverse demonstration data. We remove some simple tasks that baseline policies are fully capable of, and evaluate HiPolicy on **12** challenging tasks from the RoboTwin 1.0 benchmark.
- **RoboTwin 2.0** [3]: RoboTwin 2.0 is a scalable framework for the automated generation of realistic synthetic data. We removed tasks with fewer than 100 demo steps and with low precision requirements (such as simple pick-and-place) and evaluated HiPolicy on **9** selected long-horizon or precise tasks from the RoboTwin 2.0 benchmark.

To highlight HiPolicy's ability to jointly model long-horizon dependencies and high-precision closed-loop controls shown in Table1, tasks are further categorized, and we present various temporal dependency profiles and levels of control granularity (precise tasks or non-precise tasks). Precise tasks require tight end-effector pose tolerances to succeed (e.g., cube stacking, QR-code scanning), while non-precise tasks have more lenient success criteria.

All demo data was collected according to the methods provided by the official. Details of the selected tasks, demo collection, and preprocessing are provided in Appendix A.

Baselines. We compare our method against two representative generative imitation learning policies, selected to cover different input modalities:

- **Diffusion Policy (DP)** [4]: A conditional generative policy that operates purely on 2D image observations, predicting continuous actions via a diffusion process. This baseline reflects the vision-only setting where the policy relies entirely on visual inputs.
- **DP3** [40]: Extends diffusion-based policies to 3D perception, capturing the multi-modal setting typical in 3D manipulation, leveraging both color texture cues and geometric shape information.

Evaluation protocols. We follow the standard evaluation protocols defined by RoboTwin 1.0 for the tasks from RoboTwin 1.0 [25] and RoboTwin 2.0 [3]. Training is conducted for a fixed number of epochs using identical data splits for all methods, ensuring fair comparison. For RoboTwin 1.0 and 2.0, we report *success rate* averaged over $N = 100$ episodes per task.

Hyperparameters. In this paper, we use the unified hyperparameter provided in Appendix D for all simulation and real-world tasks.

4.2 Results and Analysis in Simulation

As the simulation results shown in Table 1, HiPolicy shows significant relative improvements compared with DP and DP3 baselines. Specifically, in RoboTwin 1.0 and RoboTwin 2.0, it shows a dramatic relative advantage of **62%** and **44%** versus DP and DP3, respectively.

The most significant improvement appears in tasks such as *block hammer beat* and *stamp seal*, which are representatives of high-precision tasks; *put bottles dustbin*, which is challenging for its super-long horizon properties. These experimental results strongly demonstrate that our HiPolicy successfully reconciles the often-conflicting objectives of maintaining high control accuracy and enabling long-horizon sequential prediction within a single architecture.

Operations such as *blocks stack hard* and *place bread basket* possess both long-horizon and high-precision features. Results on these tasks validate that HiPolicy, whether based on DP or DP3, dramatically enhances the precision of manipulation and improves the modeling capability for long-horizon time-dependent tasks by simultaneously predicting action chunks at hierarchical temporal resolutions.

Table 2: Ablation Experiment. In the RoboTwin 1.0 platform, we compare our complete HiPolicy with that removing the hierarchical condition, including all conditions only with the highest frequency (High) or the lowest (Low) frequency, the fusion module, and the hierarchical frequency structure (Hier.), respectively.

Task name	Condition			w/o Fusion	w/o Hier.
	Hier.	Low	High		
Blocks stack hard	10	8	4	8	0
Blocks stack easy	15	12	10	8	0
Shoe place	48	43	14	43	33
Dual shoes place	15	12	8	11	4
Block handover	100	100	89	98	61
Block hammer beat	67	66	42	56	0
Dual bottles pick hard	72	70	62	67	60
Dual shoes pick easy	94	94	94	93	82
Pick apple messy	81	80	67	66	40
Empty cup place	95	96	95	92	89
Bottle adjust	52	52	50	46	46
Diverse bottles pick	65	61	54	61	24
Average	**60**	58	49	54	37

4.3 Ablation Study

We conduct ablation studies to isolate the contributions of each component:

No hierarchical observation condition. The hierarchical condition is replaced by a fixed-frequency condition, where the policy is conditioned solely on high-frequency or low-frequency observations. As shown in Table 2, taking observations at a fixed frequency leads to a performance decline, with more obvious drop when taking a higher fixed frequency of observation.

No hierarchical feature fusion. Remove the attention module for fusing features across different frequencies. From the results in Table 2, the hierarchical feature fusion part promotes feature communication across different frequency chunks, contributing to performance improvement by 6%.

Single-frequency chunking. Remove multi-frequency actions, only predicting and executing actions at a fixed frequency. As the core design of our HiPolicy, removing the hierarchical frequency structure causes significant drop in success rate, as evidenced by the last column of Table 2.

No entropy guidance execution. Entropy-guided execution provides a clear advantage. As shown in Table 3, the strategy improves execution speed by 25% while incurring only a very small drop in success rate, relative to DP+HiPolicy without entropy-guided execution.

Sample Times. In this paper, we set $N = 100$ as the unified hyperparameter and inference in parallel. Since the condition features are extracted once and reused, increasing N from 1 to 100 introduces only 2.0 ms of latency and 63 MB

Table 3: Acceleration Experiment. We measure the success rate (SR) and the execution steps for DP baseline, DP+HiPolicy without entropy-guided (EG) execution, and DP+HiPolicy across 8 tasks from the RoboTwin 2.0 platform. We use red to highlight our improvement against DP in the success rate and execution speed.

Task name	DP [4]		DP+HiPolicy w/o EG		DP+HiPolicy	
	SR↑	Steps↓	SR↑	Steps↓	SR↑	Steps↓
Stamp seal	5	140	19	189	25	140
Place mouse pad	13	164	29	150	28	90
Scan object	21	146	44	139	37	103
Place phone stand	45	103	79	102	73	82
Place bread skillet	38	114	55	113	42	85
Average	24	133	45	139	41(71%↑	100(25%↓)

Table 4: Sample times ablation. Ablation study on the number of sampled action candidates N.

N	Time(ms)↓	VRAM(MB)↓	SR(%)↑	Steps↓
1	105.5	7394	–	–
5	106.0	7403	35	104
10	106.3	7414	36	104
100	107.5	7457	41	100
500	147.9	7629	39	103

of GPU memory overhead, as shown in Table 4, which is negligible. Increasing N improves the accuracy of entropy estimation and the task success rate, and saturates around $N = 100$.

In general, all hierarchical designs and the entropy-guided execution strategy is essential to balance the long-horizon modeling ability and execution efficiency.

Runtime and Resource Cost Evaluation HiPolicy improves efficiency by reducing executed steps and the resulting rollout wall-clock time, rather than by reducing per-inference latency. To evaluate the computational cost of HiPolicy, we measured DP and HiPolicy on 5 RoboTwin 2.0 tasks (Stamp seal, Place mouse pad, Scan object, Place phone stand, and Place bread skillet) using an RTX 4090. Let S denote executed steps, L_c=8 the chunk length, f_{ctrl}=15Hz the control frequency, and τ_{infer} the per-chunk inference latency. Since inference is performed once per executed chunk, we report rollout wall-clock time as $T_{\text{wall}} = \lceil S/L_c \rceil \tau_{\text{infer}} + S/f_{\text{ctrl}}$. As shown in Table 5, DP+HiPolicy adds only modest per-call runtime/memory cost over DP (8ms and 897MB VRAM), while reducing steps by 25% and rollout wall-clock time by 24%, and improving SR from 24% to 41%.

Table 5: Runtime and memory comparison between DP and DP with HiPolicy on an RTX 4090 GPU.

Method	VRAM↓	Infer.↓	Steps↓	Wall-clock↓	SR↑
DP	6560MB	100ms	133	10.57s	24%
DP + HiPolicy	7457MB	108ms	**100** (25%↓)	**8.07s** (24%↓)	**41%** (+71%↑)

4.4 Real-world Experiments

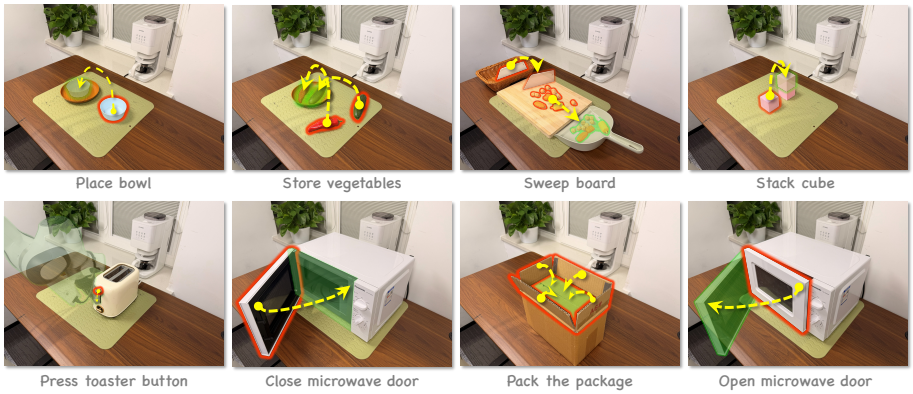

Fig. 5: Real-world Robot Tasks. We evaluate our Hierarchical Policy on 8 real-world manipulation tasks with the Franka Panda robot arm.

Setup. We use a Franka Emika Panda robotic arm equipped with a Robotiq 2F-85 gripper for our experiments. The setup includes two Zed 2i cameras positioned to provide third-person views and one Zed Mini camera mounted on the robot's end-effector to capture first-person perspectives of the environment, following the setup in DROID [15]. Detailed information can be found in Appendix G.
Tasks. As shown in Fig. 5, we evaluate our Hierarchical Policy on eight real-world manipulation tasks:

- **Place Bowl**: The robot is required to grasp the edge of the light blue bowl precisely and place it on top of the plate.
- **Close Microwave Door**: The robot's end effector pushes the door of the white microwave oven, closing it tightly.
- **Store Vegetables**: The robot needs to accurately pick up the chili peppers and cucumbers and place them on a plate. This task serves as an effective test of the policy's viability over a long-horizon manipulation.
- **Pack the Package**: The robot uses its end effector to fold 4 edges of the box, and this task can also effectively test the performance of a policy in long-horizon manipulation.

Table 6: Detailed evaluation results on 8 real-world tasks. Our results are reported based on an average of 10 evaluation trials per task, each conducted with randomized object initializations to ensure statistical robustness. And the bold red color is used to highlight the relative percentage improvement over baseline methods. In *close microwave oven*, we respectively report the execution length of closing the door to nearly-closed position and closing the door tightly.

	Success Rate↑		**Avg. Execution Steps↓**	
Task name	**DP [4]**	**DP+HiPolicy**	**DP [4]**	**DP+HiPolicy**
Place bowl	70%	**90%**	98	**87**
Store vegetables	60%	**90%**	184	**159**
Sweep board	**60%**	**60%**	202	**185**
Stack cube	70%	**80%**	137	**108**
Press toaster button	70%	**80%**	62	**58**
Open microwave door	80%	**90%**	117	**82**
Close microwave door	0%	**100%**	60	**58/67**
Pack the package	70%	**90%**	173	**154**
Avg.	60%	**85%** (**42%**↑)	129	**111/113**

- **Sweep Board**: The robot is required to grasp a plate and sweep the snacks into the dustpan accurately. The entire process effectively evaluates the robot's performance in fine-grained and long-horizon manipulation tasks.
- **Stack Cube**: The robot needs to place one pink block on top of another pink block stably, which poses a significant challenge to the robot's manipulation precision.
- **Press Toaster Button**: The robot needs to press the very small toaster button all the way down, which needs robot to perform at extremely high manipulation precision.
- **Open Microwave Door**: Initially, the microwave oven door has only a tiny opening, and the robot must precisely open the door within an extremely limited area to succeed.

Results and analysis in real-world robot experiments. As shown in Table 6, HiPolicy shows **42%** relatively higher success rate and **14%** faster execution speed compared to DP baseline, strongly proving that our HiPolicy balance the execution speed and high control accuracy in challenging real-world tasks.

The most notable increase is observed in *close microwave door*. The significant difference lies in the fact that closing a microwave oven requires locking its internal latches, while DP only closes the door to a near-closed position without locking the latches, leaving it stuck. Our HiPolicy, however, can perform continuously until the microwave oven is fully closed. Closing the door requires relatively greater force, thus, the demonstration often pauses before closing it tightly, forming a non-Markovian process. The experiment result indicates that DP, running at a fixed high frequency (15Hz) to pursue higher control accuracy, fails to keep the long-horizon modeling ability. In contrast, our HiPolicy

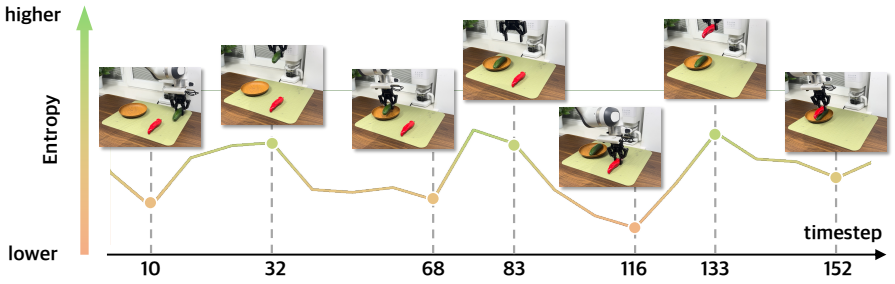

Fig. 6: Action entropy curve in *store vegetables*. We display 7 key frames with high or low precision requirements through the task above the entropy curve.

solves the dilemma between improving manipulation precision and efficiency by modeling with a hierarchical multi-frequency structure.

Tasks such as *store vegetables* and *pack the package*, featuring long-horizon manipulation, also witness significant improvement in the success rate or execution speed of our HiPolicy. These results clearly demonstrate that our HiPolicy successfully balances the execution quality and efficiency in long-horizon tasks by modeling with hierarchical temporal resolutions.

Furthermore, the entropy curve presented in Figure 6 validates our hypothesis that a higher entropy value within a chunk is indicative of actions requiring lower precision (e.g., fetch the vegetable), while a lower entropy value implies high-precision movement (e.g., pick the vegetables).

5 Conclusion

We presented HiPolicy, a hierarchical multi-frequency action chunking framework that jointly generates coarse low-frequency and precise high-frequency actions within each chunk, balancing long-horizon dependency modeling and fine-grained closed-loop control. Across diverse simulated and real-world manipulation tasks, HiPolicy consistently improved both performance and efficiency.

Limitations. Our evaluation is limited to relatively small models and datasets, without exploring integration with large vision-language-action frameworks or other scalable policy architectures. We believe that extending HiPolicy to large-scale multimodal settings could unlock substantial new capabilities for general-purpose embodied agents.

6 Acknowledgment

This work is supported by the Beijing Natural Science Foundation (26L080330) and the National Natural Science Foundation of China (62376006).

References

1. Andrychowicz, M., Raichuk, A., Stanek, P., Shipley, T., Debiak, H., Chociej, S., Durkaya, M., Sherborne, R., Trzciński, B., Tabaka, M., et al.: Learning dexterous in-hand manipulation. arXiv preprint arXiv:1808.00177 (2018)
2. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Shi, L.X., Tanner, J., Vuong, Q., Walling, A., Wang, H., Zhilinsky, U.: $\pi 0$: A vision-language-action flow model for general robot control. ArXiv **abs/2410.24164** (2024)
3. Chen, T., Chen, Z., Chen, B., Cai, Z., Liu, Y., Li, Z., Liang, Q., Lin, X., Ge, Y., Gu, Z., et al.: Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. arXiv preprint arXiv:2506.18088 (2025)
4. Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research **44**, 1684 – 1704 (2023)
5. Contributors, I.M.: Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy. arXiv preprint arXiv:2510.13778 (2025)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: North American Chapter of the Association for Computational Linguistics (2019)
7. Finn, C., Levine, S.: Learning to see, seeing to act: Emergent visual skills for robot manipulation. In: Conference on Robot Learning (CoRL). pp. 1–13 (2017)
8. Flanagan, J.R., Bowman, M.C., Johansson, R.S.: Control strategies in object manipulation tasks. Current Opinion in Neurobiology **16**, 650–659 (2006)
9. Gong, Z., Ding, P., Lyu, S., Huang, S., Sun, M., Zhao, W., Fan, Z., Wang, D.: Carp: Visuomotor policy learning via coarse-to-fine autoregressive prediction. ArXiv **abs/2412.06782** (2024)
10. Guo, L., Xue, Z., Xu, Z., Xu, H.: Demospeedup: Accelerating visuomotor policies via entropy-guided demonstration acceleration. ArXiv **abs/2506.05064** (2025)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
12. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Galliker, M.Y., Ghosh, D., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., LeBlanc, D., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Ren, A.Z., Shi, L.X., Smith, L., Springenberg, J.T., Stachowicz, K., Tanner, J., Vuong, Q., Walke, H.R., Walling, A., Wang, H., Yu, L., Zhilinsky, U.: $\pi 0.5$: a vision-language-action model with open-world generalization. ArXiv **abs/2504.16054** (2025)
13. Ke, T.W., Gkanatsios, N., Fragkiadaki, K.: 3d diffuser actor: Policy diffusion with 3d scene representations. In: Proceedings of Robotics: Conference on Robot Learning(CoRL) (2024)
14. Kelly, M., Sidrane, C., Driggs-Campbell, K., Kochenderfer, M.J.: Hg-dagger: Interactive imitation learning with human experts. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 8077–8083. IEEE (2019)
15. Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M.K., Chen, L.Y., Ellis, K., Fagan, P.D., Hejna, J., Itkina, M., Lepert, M., Ma, Y.J., Miller, P.T., Wu, J., Belkhale, S., Dass, S., Ha, H., Jain, A., Lee, A., Lee, Y., Memmel, M., Park, S., Radosavovic, I., Wang, K., Zhan, A.,

Black, K., Chi, C., Hatch, K.B., Lin, S., Lu, J., Mercat, J., Rehman, A., Sanketi, P.R., Sharma, A., Simpson, C., Vuong, Q., Walke, H.R., Wulfe, B., Xiao, T., Yang, J.H., Yavary, A., Zhao, T.Z., Agia, C., Baijal, R., Castro, M.G., Chen, D., Chen, Q., Chung, T., Drake, J., Foster, E.P., Gao, J., Guizilini, V., Herrera, D.A., Heo, M., Hsu, K., Hu, J., Irshad, M.Z., Jackson, D., Le, C., Li, Y., Lin, K., Lin, R., Ma, Z., Maddukuri, A., Mirchandani, S., Morton, D., Nguyen, T., O'Neill, A., Scalise, R., Seale, D., Son, V., Tian, S., Tran, E., Wang, A.E., Wu, Y., Xie, A., Yang, J., Yin, P., Zhang, Y., Bastani, O., Berseth, G., Bohg, J., Goldberg, K., Gupta, A., Gupta, A., Jayaraman, D., Lim, J.J., Malik, J., Martín-Martín, R., Ramamoorthy, S., Sadigh, D., Song, S., Wu, J., Yip, M.C., Zhu, Y., Kollar, T., Levine, S., Finn, C.: Droid: A large-scale in-the-wild robot manipulation dataset (2024)

16. Kim, J., Kong, J., Son, J.: Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In: International Conference on Machine Learning. pp. 5530–5540. PMLR (2021)
17. Kim, M., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., Finn, C.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
18. Laskey, M., Lee, J., Fox, R., Dragan, A., Goldberg, K.: Dart: Noise injection for robust imitation learning. In: Conference on robot learning. pp. 143–156. PMLR (2017)
19. Li, W., Zhang, R., Shao, R., He, J., Nie, L.: Cogvla: Cognition-aligned vision-language-action model via instruction-driven routing & sparsification. ArXiv **abs/2508.21046** (2025)
20. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
21. Liu, R., Zhou, P., Luo, Q., Sun, L., Cen, J., Song, Y., Yang, Y.: Himacon: Discovering hierarchical manipulation concepts from unlabeled multi-modal data. Advances in neural information processing systems (2025)
22. Lu, Y., Tian, Y., Yuan, Z., Wang, X., Hua, P., Xue, Z., Xu, H.: H3dp: Triply-hierarchical diffusion policy for visuomotor learning. ArXiv **abs/2505.07819** (2025)
23. Ma, X., Patidar, S., Haughton, I., James, S.: Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 18081–18090 (2024)
24. Miller, J.A., Constantinidis, C.: Timescales of learning in prefrontal cortex. Nature Reviews Neuroscience **25**, 597–610 (2024). `https://doi.org/10.1038/s41583-024-00836-8`
25. Mu, Y., Chen, T., Chen, Z., Peng, S., Lan, Z., Gao, Z., Liang, Z., Yu, Q., Zou, Y., Xu, M., Lin, L., Xie, Z., Ding, M., Luo, P.: Robotwin: Dual-arm robot benchmark with generative digital twins. In: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR). pp. 27649–27660 (June 2025)
26. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
27. Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., Carin, L.: Variational autoencoder for deep learning of images, labels and captions. Advances in neural information processing systems **29** (2016)
28. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)

29. Ross, S., Gordon, G., Bagnell, D.: A reduction of imitation learning and structured prediction to no-regret online learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 627–635. JMLR Workshop and Conference Proceedings (2011)
30. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal **27**(3), 379–423 (Jul 1948). https://doi.org/10.1002/j.1538-7305.1948.tb01338.x, part I of two parts.
31. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (ICLR) (2021), https://openreview.net/forum?id=St1giarCHLP
32. Song, Y., Ermon, S.: Improved techniques for training score-based generative models. Advances in neural information processing systems **33**, 12438–12448 (2020)
33. Su, Y., Zhan, X., Fang, H., Xue, H., Fang, H., Li, Y.L., Lu, C., Yang, L.: Dense policy: Bidirectional autoregressive learning of actions. ArXiv **abs/2503.13217** (2025)
34. Taniguchi, T., Hirai, Y., Suzuki, M., Murata, S., Horii, T., Tanaka, K.: System 0/1/2/3: Quad-process theory for multi-timescale embodied collective cognitive systems. ArXiv **abs/2503.06138** (2025)
35. Tian, Y., Yang, S., Zeng, J., Wang, P., Lin, D., Dong, H., Pang, J.: Predictive inverse dynamics models are scalable learners for robotic manipulation. arXiv preprint arXiv:2412.15109 (2024)
36. Vahdat, A., Kreis, K., Kautz, J.: Score-based generative modeling in latent space. Advances in neural information processing systems **34**, 11287–11302 (2021)
37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
38. Wang, C., Fang, H., Fang, H., Lu, C.: Rise: 3d perception makes real-world robot imitation simple and effective. 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) pp. 2870–2877 (2024)
39. Xue, H., Ren, J., Chen, W., Zhang, G., Fang, Y., Gu, G., Xu, H., Lu, C.: Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. In: Proceedings of Robotics: Science and Systems (RSS) (2025)
40. Ze, Y., Zhang, G., Zhang, K., Hu, C., Wang, M., Xu, H.: 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In: Proceedings of Robotics: Science and Systems (RSS) (2024)
41. Zhang, W., Hu, T., Qiao, Y., Zhang, H., Qin, Y., Li, Y., Liu, J., Kong, T., Liu, L., Ma, X.: Chain-of-action: Trajectory autoregressive modeling for robotic manipulation. ArXiv **abs/2506.09990** (2025)
42. Zhao, T., Kumar, V., Levine, S., Finn, C.: Learning fine-grained bimanual manipulation with low-cost hardware. In: Proceedings of Robotics: Science and Systems (RSS) (2023)