
FreqPolicy: Frequency Autoregressive Visuomotor Policy with Continuous Tokens

**Yiming Zhong¹, Yumeng Liu², Chuyang Xiao¹, Zemin Yang¹,
Youzhuo Wang¹, Yufei Zhu¹, Ye Shi¹, Yujing Sun^{2,3*}, Xinge Zhu⁴, Yuexin Ma^{1*}**

¹ShanghaiTech University ²The University of Hong Kong

³Digital Trust Centre, Nanyang Technological University ⁴The Chinese University of Hong Kong

{zhongym2024, mayuixin}@shanghaitech.edu.cn

Project Page: <https://freq-policy.github.io/>

Code: <https://github.com/4DVLab/Freqpolicy>

Abstract

Learning effective visuomotor policies for robotic manipulation is challenging, as it requires generating precise actions while maintaining computational efficiency. Existing methods remain unsatisfactory due to inherent limitations in the essential action representation and the basic network architectures. We observe that representing actions in the frequency domain captures the structured nature of motion more effectively: low-frequency components reflect global movement patterns, while high-frequency components encode fine local details. Additionally, robotic manipulation tasks of varying complexity demand different levels of modeling precision across these frequency bands. Motivated by this, we propose a novel paradigm for visuomotor policy learning that progressively models hierarchical frequency components. To further enhance precision, we introduce continuous latent representations that maintain smoothness and continuity in the action space. Extensive experiments across diverse 2D and 3D robotic manipulation benchmarks demonstrate that our approach outperforms existing methods in both accuracy and efficiency, showcasing the potential of a frequency-domain autoregressive framework with continuous tokens for generalized robotic manipulation.

1 Introduction

The study of visuomotor policies enable robots to learn task execution from demonstrations by leveraging raw visual inputs, such as images or point clouds, allowing them to generate effective action sequences in response to new visual observations. It has become a prevailing paradigm in robot manipulation [10]. However the requirement of high precision in robotic tasks and the sequential correlation in action space present challenges for visuomotor policy learning.

Existing methods for visuomotor policy learning can be broadly categorized into diffusion-based methods [10, 48, 38, 42, 47, 15, 43, 39] and autoregressive (AR) methods [13, 16, 49]. Diffusion-based approaches model the action distribution conditioned on observations, leveraging their powerful generative capabilities to produce reliable and diverse action sequences even from limited demonstrations. However, diffusion models typically encounter higher computational costs and increased inference latency due to their iterative sampling process, which could limit their practicality in efficiency-sensitive applications. In contrast, AR methods sequentially predict each action step conditioned on previous actions and current observations. They are generally more computationally efficient and enable faster inference, making them attractive for real-time control scenarios. Nevertheless, AR approaches may be prone to compounding errors over long horizons, and due to their common reliance on discrete representations, often struggle to accurately model inherently continuous action spaces, limiting their ability to capture complex temporal correlation. Despite

*Corresponding author.

recent advances, both diffusion-based and AR methods share a fundamental limitation: they overlook the diversity of the action space arising from task complexity and the degrees of freedom in robotic manipulation. Representing the inherent structured features of different actions is of great significance for the generalization and robustness of visuomotor policies.

To address these limitations, we rethink the robotic action representation from a frequency-domain perspective. Figure 1 shows one example of the action signals (from Adroit Door [30]) reconstructed by filtering different frequency bands. We can see that the first 30% of frequency bands is already sufficient to allow us to reconstruct a signal nearly identical to the original, preserving 95% of its energy. Retaining just the first 10% of bands can recover the general trend of the signal. Incorporating high-frequency information, up to the first 60% of frequency bands, enables the restoration of finer details, such as subtle oscillations. Meanwhile, through extensive statistical analysis of robot task execution behaviors, we found the required frequency components vary depending on the complexity of different tasks. For simple tasks, such as pick-and-place, low-frequency information is often sufficient, and high-frequency signals may be redundant, especially the collection of demonstrations in real world can introduce noise or unnatural jitter. Filtering out these components results in smoother, more natural actions. However, for complex tasks such as dexterous manipulation, high-frequency details are essential for precise control. In summary, for continuous action spaces, investigating hierarchical frequency-domain information and developing effective representation and modeling methods for spectral features are crucial for enabling robots to perform tasks with varying complexity levels.

With this observation, we propose to learn the visuomotor policies based on the modeling of hierarchical frequency domains. Notably, low-frequency information is easier to learn and captures the global structure of the motion. Rather than producing the entire frequency spectrum at once, we adopt a multi-stage progressive approach, starting from low-frequency representations and gradually extending to full-spectrum actions. This coarse-to-fine generation process not only simplifies the modeling complexity of sophisticated actions but also provides low-frequency actions with protection from high-frequency noise, ensuring stability in the fundamental motion structure. Autoregressive (AR) paradigm, with their sequential modeling capabilities, are inherently well-suited to implement this coarse-to-fine generation paradigm, effectively capturing hierarchical dependencies between frequency domains. However, previous AR-based methods [29] usually discretize the inherently continuous action space, resulting in significant information loss. Recent study [19] also suggests that discretization of originally continuous spaces is not necessary for AR modeling, and diffusion models can better represent continuous probability distributions. Motivated by these insights, we propose a coarse-to-fine robotic action generation paradigm, **FreqPolicy**, that combines continuous action representations with AR modeling in the frequency domain.

To be specific, our approach utilizes the Discrete Cosine Transform (DCT) [18] to convert action sequences into frequency components. Leveraging a masked encoder-decoder architecture, we map trajectories from various frequency bands into distinct latent codes. It predicts refined motions progressively, where low-frequency signals guide the generation of high-frequency details. Our framework bridges the gap between hierarchical frequency representations and probabilistic modeling, offering a unified and scalable solution for visuomotor policy learning. Extensive experiments show that our method yields significant improvements on challenging robotic manipulation benchmarks, demonstrating both its efficiency and state-of-the-art performance for generating precise and high-fidelity actions. In summary, our contributions are as follows:

- We have explored frequency-based action space representation and propose a novel solution for effective visuomotor policy learning by progressively modeling hierarchical frequency components to capture the structured nature of robotic motions.
- We introduce continuous latent representations with diffusion-based decoding, eliminating discretization requirements while preserving action space continuity and autoregressive efficiency.
- Our method achieves state-of-the-art performance on extensive robotic benchmarks, significantly outperforming others in both success rate and computational efficiency.

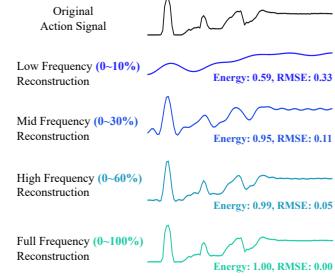


Figure 1: Action signals reconstructed across different frequency bands; *see Appendix* for energy details.

2 Related Work

2.1 Action Representations for robots

Action representation is important in robotic learning, as it determines how agents encode and generate behaviors in complex environments. Recently, transformers have achieved remarkable success in large language models (LLMs)[8, 1, 36], demonstrating exceptional versatility in sequence generation tasks. At the same time, diffusion models have shown strong generative capabilities for images[17, 44] and have garnered significant attention for their adaptation to robotic policies [10, 48]. The advancements in transformer and diffusion models have given rise to two primary paradigms for action representation: discrete and continuous. Transformer models typically employ discrete tokenization, encoding actions as sequences of tokens. In robotic tasks, discretization can be realized through straightforward quantization of each action dimension at every timestep [7, 6] or clustering approaches such as BeT [32], which enable the generation of diverse behaviors but may sacrifice fine-grained control.

In contrast, diffusion models and other continuous approaches [33, 41, 10, 48, 50, 14] leverage probabilistic frameworks—such as VAEs, diffusion processes, and normalizing flows—to model the action space directly. These methods preserve the full expressiveness and precision of the original action space, avoiding the loss of nuance that can result from discretization. However, this advantage often comes at the cost of increased computational complexity, particularly in high-dimensional settings.

Recently, frequency-domain strategies have been explored to address redundancy in both images [28, 20, 45] and action spaces. FAST [29] demonstrates that excessive high-frequency information in action sequences can hinder model training, and propose a compression-based tokenization scheme that reduce redundancy in action signals and improves training efficiency. However, both frequency-domain compression and action discretization inevitably cause information loss, which limit fine-grained, precise synthesis. In this work, we combine frequency-domain action representation with continuous sequence modeling. Our approach enables flexible transitions between spatial and frequency domains, preserving critical details for high-fidelity motion while maintaining efficiency, thus supporting more expressive and scalable robotic policy learning.

2.2 Visuomotor Policy for Robotic Manipulation

Visuomotor policies for robotic manipulation map visual inputs directly to control actions, enabling robots to interact with their environments in a closed feedback loop. Broadly, there are two main approaches to tackling this problem: one is to use diffusion based methods, and the other is to employ autoregressive methods. Diffusion-based methods typically generate action segments by modeling the conditional distribution of actions given observations using diffusion models [10, 48, 38, 42]. Autoregressive methods generate action sequences step-by-step, predicting each action based on previous outputs [21, 13, 49, 40]. While this approach is efficient, it often lacks long-horizon structural modeling. Recently, several coarse-to-fine methods have been proposed, but they primarily operate in the temporal domain. CARP [16] incorporates multi-scale reconstruction from VAR [35] for action generation, but depends on discrete representations derived from VQ-VAE [37], which limits precision. DensePolicy [34] introduces a bidirectional, BERT-inspired [11] framework for hierarchical, coarse-to-fine action prediction. However, its iterative refinement relies on progressively increasing temporal upsampling density, which does not remove high-frequency noise or compress redundant information in action signals. Unlike previous methods, our approach performs AR generation in the frequency domain, modeling signals at different frequencies independently. This separation prevents low-frequency components from being affected by high-frequency noise. Moreover, for high-dimensional and complex tasks, progressively learning high-frequency signals from low-frequency components eases the challenge of direct generation.

3 Method

3.1 Overview

Problem Formulation Given a dataset of paired sequences $\mathcal{D} = \{(\mathbf{o}, \mathbf{x})\}$, where each $\mathbf{o} = [o_0, \dots, o_{N-1}]^T$ denotes a sequence of observations (e.g., RGB images, depth maps, or point clouds)

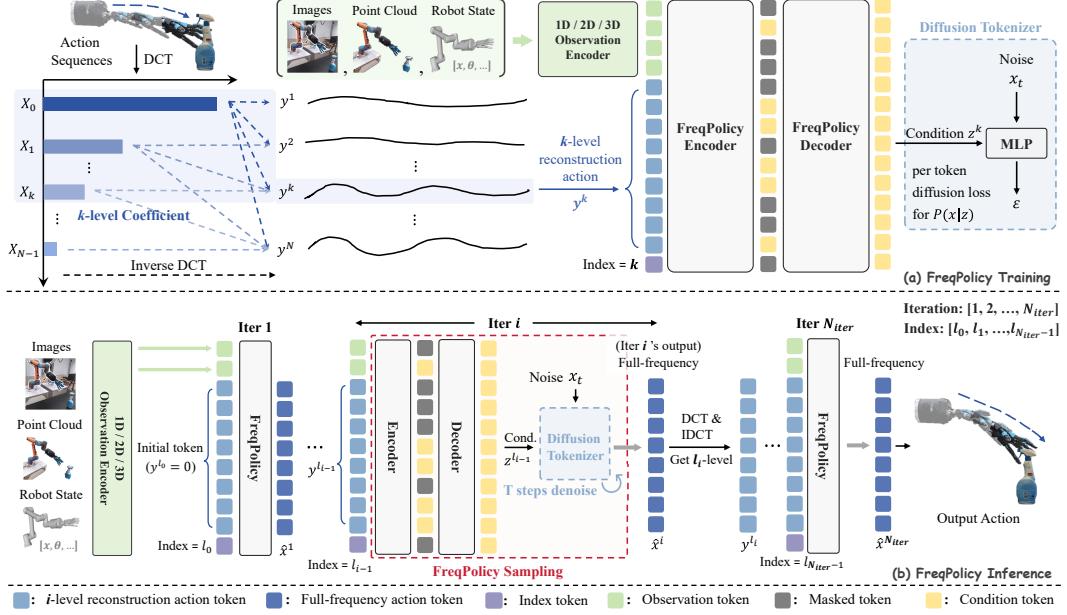


Figure 2: **Pipeline of FreqPolicy showing both training (a) and inference (b) procedures.** We first transform action trajectories into the frequency domain via DCT, and then learn latent codes for different frequency level actions using FreqPolicy, and reconstruct actions through masked prediction and a diffusion-based decoder. This enables robust, frequency-aware, and high-fidelity robotic action generation.

and each $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]^T \in \mathbb{R}^{N \times d}$ represents the corresponding sequence of robot actions with variable length N and action dimension d , our objective is to train a policy that can produce action trajectories $\hat{\mathbf{x}}$ based on new observation sequences $\hat{\mathbf{o}}$ to effectively perform the demonstrated tasks.

As shown in Figure 2, our method consists of two stages. **During training (Figure 2a),** we apply DCT to decompose action sequences into frequency components and reconstruct them at different levels via inverse DCT. For frequency level k , the FreqPolicy encoder-decoder processes the k -level reconstruction \mathbf{y}^k , observation features, and frequency index to produce a continuous token \mathbf{z}^k , which conditions a diffusion model to reconstruct the original full-frequency action \mathbf{x} . We adopt frequency-aware masking (gray blocks in the figure) to improve training efficiency. **During inference (Figure 2b),** we perform hierarchical generation through N_{iter} iterations. Starting from zero ($\mathbf{y}^{l_0} = 0$), each iteration i generates a full-frequency action $\hat{\mathbf{x}}^i$ conditioned on the previous output $\mathbf{y}^{l_{i-1}}$, then filters it to \mathbf{y}^{l_i} for the next iteration. This coarse-to-fine process progressively refines actions from low-frequency global structure to high-frequency fine details.

This section is organized as follows: Section 3.2 analyzes action trajectories in the frequency domain; Section 3.3 introduces the space-frequency transformation method employed in our approach; Section 3.4 describes the continuous token representation with diffusion models; and Section 3.5 presents the frequency-based hierarchical generation strategy.

3.2 Frequency Domain Analysis

To motivate our approach, we first conduct a systematic analysis of action trajectories under various task conditions. We selected two representative benchmarks to evaluate our approach. For complex manipulation tasks, we chose three tasks from the Adroit Benchmark [30], which utilizes a dexterous robotic hand with 26 degrees of freedom. For simpler tasks, we selected three tasks from the Robomimic Benchmark [23], which features parallel grippers with only 10 degrees of freedom. On these tasks, we aim to analyze the energy distribution of each action dimension of the robotic arm across different frequency bands, as well as to investigate the impact of discarding high-frequency signals on task performance.

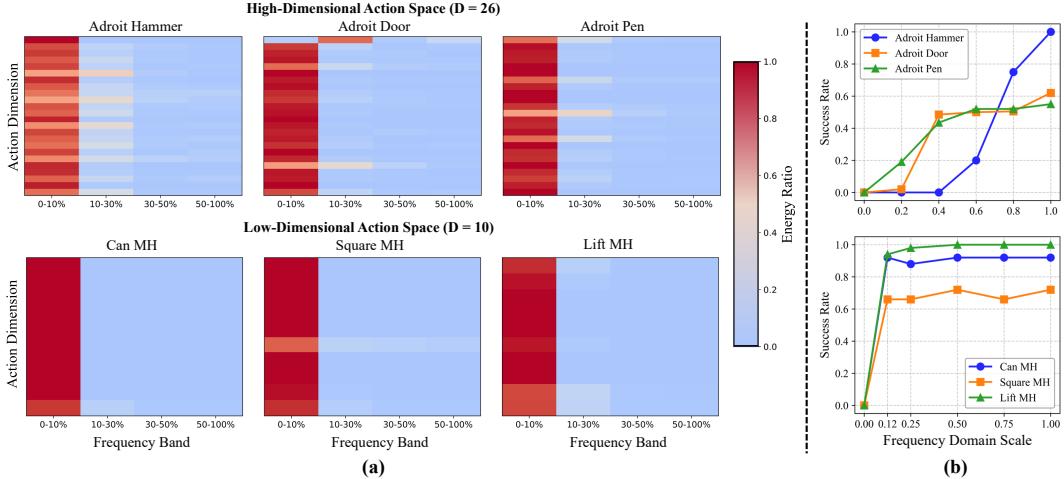


Figure 3: (a) **Heat maps of frequency band energy across action dimensions for different tasks.** The top row shows Adroit [30] tasks with high-dimensional actions (26 dimensions), while the bottom row presents Robomimic [23] tasks with low-dimensional actions (10 dimensions). (b) **Success rate of actions reconstructed with varying frequency ratios.** We reconstruct action sequences using different proportions of frequency components and evaluate their success rates on the original tasks.

Energy Distribution. We present the energy heat maps for these tasks in Figure 3(a), where the vertical axis represents different dimensions of the action space, corresponding to the joint angles of the robotic hand or arm, and the horizontal axis indicates the energy ratio across different frequency bands. The red regions indicate areas with a high energy ratio, suggesting that these frequency bands contain a greater proportion of the signal’s energy. Conversely, the blue regions represent areas with a low energy ratio. In both the Adroit tasks (top row) and the Robomimic tasks (bottom row), most of the energy is concentrated in the 0-10% low-frequency bands (leftmost columns), which is why these regions appear red. By comparing the heat map for tasks in different action space, we found that tasks with high-dimensional actions show greater variance in the energy distribution compared to low-dimensional tasks. Specifically, the energy distribution of the action space in the Adroit Benchmark shows distinct low-frequency (0-10%) patterns across its three tasks, while in the Robomimic tasks, the low-frequency (0-10%) energy distribution remains relatively uniform with little variation among the three tasks.

Performance under Frequency Compression. To further assess the role of high-frequency information on task performance across different benchmarks, we compress action signals by removing the high-frequency components from the original sequences, reconstructed the actions with varying proportions of low-frequency information, and measured their success rates on the original tasks. The results, shown in Figure 3(b), indicate that omitting high-frequency signals leads to a significant drop in success rates for Adroit tasks. In contrast, low-dimensional tasks in Robomimic maintain stable performance with as little as 12% of low-frequency information preserved. This analysis underscores the importance of frequency-aware modeling: the necessity of high-frequency signals for effective action representation varies across different scenarios.

Conclusion. These findings provide the empirical foundation for our frequency-based autoregressive generation strategy. They suggest that adjusting the granularity of frequency-domain representations can help optimize model performance for tasks with different levels of complexity. Our goal is to enable the model to learn individual task skills from expert demonstrations in diverse action spaces and with varying task complexities. Therefore, we first transform the actions into the frequency domain and recover them at different frequency scales. This allows the model to predict subsequent actions using information from lower frequency bands, enabling a coarse-to-fine generation process.

3.3 DCT Decomposition

Our spectral analysis reveals that the proportion of frequency components required for effective action representation varies across different tasks. Therefore, it is necessary to transform the action space from the time domain to the frequency domain, for more effective robotic action modeling. To efficiently achieve this, we employ the Discrete Cosine Transform (DCT), which can project

time-domain trajectories onto a set of cosine basis functions with low computational cost. This provides a compact and interpretable frequency-domain representation, enabling hierarchical modeling of actions in various tasks settings and scenarios. Specifically, given an action sequence $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]^T \in \mathbb{R}^{N \times d}$, where each $x_n \in \mathbb{R}^d$ represents the action at time step n , we transform it into the frequency domain by applying the DCT independently to each action dimension. For each dimension, the DCT is defined as:

$$X_i = \sum_{n=0}^{N-1} x_n \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) i \right], \quad i = 0, 1, \dots, N-1 \quad (1)$$

Here, X_i denotes the i -th DCT coefficient. Given any integer k such that $1 \leq k \leq N$, we refer to retaining only the first k DCT coefficients as preserving the k -level DCT. With these k coefficients, the original trajectory can be approximately reconstructed using the inverse DCT, as shown in Equation 2.

$$y_n^k = \frac{1}{N} \left[X_0 + 2 \sum_{i=1}^{k-1} X_i \cos \left(\frac{\pi}{N} \left(n + \frac{1}{2} \right) i \right) \right], \quad n = 0, 1, \dots, N-1 \quad (2)$$

Specifically, let $\mathbf{y}^k = [y_0^k, y_1^k, \dots, y_{N-1}^k]^T \in \mathbb{R}^{N \times d}$ denote the k -level reconstruction of the original sequence \mathbf{x} . If $k = N$, meaning all DCT coefficients are used, the reconstruction will recover the original trajectory without any loss of information. However, if $k < N$, only the lowest k frequency components are kept, and all higher-frequency components are discarded. As a result, the reconstructed trajectory \mathbf{y}^k becomes a compressed and smoother version of the original, preserving the essential structure of the original sequence while discarding minor fluctuations.

3.4 Continuous Tokens for FreqPolicy

We explore integrating FreqPolicy with continuous token representations using a diffusion model. Compared to discrete tokens, continuous tokens offer greater expressiveness and enable more fine-grained modeling of the action space, allowing for smoother and more accurate representation of complex trajectories. Accurately modeling the probability distribution of each token is essential for effectively incorporating a continuous tokenizer into autoregressive models. Building on MAR [19], we tackle this challenge by adopting a diffusion-based loss function for training and devising a specialized frequency-aware sampling mechanism for efficient inference. Additional details on our training and inference processes are provided in the appendix.

Training During training, given an action sequence $\mathbf{x} \in \mathbb{R}^N$ of length N , we first apply the Discrete Cosine Transform (DCT) to obtain its frequency coefficients, as defined in Equation 1, and compute reconstructions at all frequency levels as described in Equation 2. Our objective is to train a model capable of modeling token distributions across different frequency levels. For an arbitrary frequency level k , we encode the k -level reconstruction \mathbf{y}^k , the observation \mathbf{o} , and the level index k using the FreqPolicy encoder to obtain a latent representation. This latent feature is then decoded by the FreqPolicy decoder to produce a condition vector \mathbf{z}^k , which serves as a continuous token for \mathbf{y}^k . We subsequently model the conditional probability $p(\mathbf{x} \mid \mathbf{z}^k)$ using a reverse diffusion process, which allows for conditional generation based on \mathbf{z}^k to recover the original action sequence. This recovered sequence can then be further processed with DCT and inverse DCT to obtain reconstructions at any desired frequency level.

Sampling The sampling procedure follows the standard inference process of diffusion models. Beginning with initial noise sampled from a standard normal distribution, the diffusion tokenizer iteratively removes the noise to produce $\hat{\mathbf{x}} \sim p(\mathbf{x} \mid \mathbf{z}^k)$. Here, $\hat{\mathbf{x}}$ denotes a trajectory sampled from the action space of demonstration data with complete frequency components. This enables flexible reconstructions at various frequency levels, by simply specifying the desired frequency level k .

Diffusion Loss The diffusion tokenizer is optimized using the diffusion loss introduced by [19], as presented in Equation 4.

$$L(z^k, \mathbf{x}) = \mathbb{E}_{\epsilon, t} \left[|\epsilon - \varepsilon_\theta(\mathbf{x}_t \mid t, z^k)|^2 \right], \quad (3)$$

where \mathbf{x}_t is a noise-perturbed version of the original trajectory \mathbf{x} , ϵ is a noise vector sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and ε_θ is a small MLP parameterized by θ that predicts the added noise. Gradients are propagated through z^k , enabling end-to-end optimization of FreqPolicy encoder, decoder and the noise predictor.

Masked Generative Strategy To reduce training cost and enhance generation diversity, we adopt

the frequency-aware masking mechanism from FAR [45]. Since early autoregressive steps process information-sparse, low-frequency inputs, using all tokens is redundant. The frequency-aware masking strategy in FAR [45] applies higher mask ratios at lower-frequency levels, gradually increasing the number of tokens as higher-frequency information is incorporated.

3.5 Frequency-based Hierarchical Generation

Our previous analysis demonstrated that compressing actions in the frequency domain at different scales reveals hierarchical information. Actions reconstructed with higher compression ratio are smoother and better capture the overall trends of the motion, which helps reduce noise interference and lowers the difficulty of generation. Based on this observation, we propose a frequency-based autoregressive strategy for hierarchical action generation, as illustrated in Figure 2. We define an increasing frequency level sequence $\{l_0, l_1, \dots, l_{N_{\text{iter}}}\}$ where $l_0 = 0$ (starting from zero input) and $l_{N_{\text{iter}}} = N$ (full frequency). Each l_i represents the number of DCT coefficients retained at iteration i . Starting from l_0 , we iterate from low to high frequencies N_{iter} times to generate the action sequence. In each iteration $i \in \{1, 2, \dots, N_{\text{iter}}\}$, the observation tokens, the frequency level index l_{i-1} , and the corresponding l_{i-1} -level reconstruction action tokens $\mathbf{y}^{l_{i-1}}$ are encoded to produce a continuous token $\mathbf{z}^{l_{i-1}}$ that represents the current context. The diffusion model then samples a full-frequency action sequence $\hat{\mathbf{x}}^i$ conditioned on this token. For the next step, we use the l_i -level reconstruction of $\hat{\mathbf{x}}^i$, i.e., \mathbf{y}^{l_i} , as the input for the following iteration, which contains richer frequency information than the previous input.

Example. For a 16-timestep trajectory with $N_{\text{iter}} = 4$ iterations, we might use $\{l_0 = 0, l_1 = 4, l_2 = 8, l_3 = 12, l_4 = 16\}$. At iteration 1, we start with \mathbf{y}^0 (all zeros) and generate $\hat{\mathbf{x}}^1$, then filter it to keep only the first 4 DCT coefficients, obtaining \mathbf{y}^4 for iteration 2. This process continues, progressively adding more frequency components ($4 \rightarrow 8 \rightarrow 12 \rightarrow 16$) until the final full-frequency output.

This hierarchical generation paradigm first captures the basic trends and global structure of the action sequence using low-frequency components, and then progressively enriches the output by incorporating higher-frequency signals that model fine-grained motion details. By explicitly decomposing and modeling motions at different frequency levels, our approach allows the model to better learn and represent both global dynamics and subtle variations in complex action sequences. This structured frequency decomposition, combined with autoregressive conditioning, enables more efficient and expressive generation of diverse and realistic motions.

4 Experiments

This section provides a comprehensive evaluation of our proposed method. We first describe the experimental setup, including benchmarks, baseline methods, and implementation details. Next, we analyze the frequency domain requirements of different tasks and highlight their unique characteristics. We then compare our method in both the time and frequency domains. Additionally, we benchmark our approach against autoregressive methods with continuous and discrete token representations across various simulation benchmarks, and present results from real-world applications. Finally, we discuss the inference speed and ablation in sampling.

4.1 Experimental Setup

Benchmarks. We evaluate our methods on a diverse set of benchmarks that provide different types of observation data. Benchmarks with only 2D image observations are referred to as **2D tasks**, which include two single-task benchmarks, **Robomimic** [23] and **Push T** [12]. Benchmarks with 3D visual observations are referred to as **3D tasks**, consisting of **Adroit** [30], **DexArt** [4], **MetaWorld** [46], and **RoboTwin** [26], which together cover a wide range of robotic manipulation and dual-arm collaborative tasks. Tasks in DexArt are conducted using Allegro Hand [3] with 22 DoF, tasks in Adroit use ShadowHand [31] with 26-28 DoF, and tasks in RoboTwin using dual-arm grippers [2] with 14 DoF. Other tasks use parallel grippers with 10 DoF.

Baseline. For **2D tasks**, we compare against Diffusion Policy (DP) as well as two autoregressive approaches using discrete token representation: Behavior Transformer (BeT) [32] and CARP [16]. DP is available in two variants: CNN-based (DP-C) and Transformer-based (DP-T). For **3D tasks**, we use 3D Diffusion Policy (DP3) [48] and Mamba Policy [9] as baselines.

Implementation Details. Our model can be seamlessly integrated into the codebases of Diffusion Policy (DP) and 3D Diffusion Policy (DP3). To ensure fair comparisons, we use the same parameters and observation input processing as Diffusion Policy for the **2D tasks**, and as 3D Diffusion Policy for the **3D tasks**, maintaining consistency with their respective frameworks. Our approach enables flexible adjustment of autoregressive iteration counts in the frequency domain during inference. For all simulation experiments, we use 4 iterations, whereas for real-world experiments, only 1 iteration is used.

Table 1: **Success rate (%) comparison on 10 simulation tasks in Adroit, DexArt and Meta-World.** To ensure fair comparison, all tasks employ full-spectrum outputs and identical frequency level progression. * denotes results we reproduced using the same expert demonstrations as ours.

Alg \ Task	Adroit [30]			DexArt [4]			Meta-World [46]			Average	
	Hammer	Door	Pen	Laptop	Faucet	Toilet	Bucket	Assembly	Disassemble		
Diffusion Policy	45 \pm 5	37 \pm 2	13 \pm 2	69 \pm 4	23 \pm 8	58 \pm 2	46 \pm 1	15 \pm 1	43 \pm 7	63 \pm 3	44.0
DP3	100 \pm 0	62 \pm 4	43 \pm 6	83 \pm 1	63 \pm 2	82 \pm 4	46 \pm 2	99 \pm 1	69 \pm 4	97 \pm 4	74.4
Mamba Policy	100 \pm 0	68 \pm 1	41 \pm 2	80 \pm 4	33 \pm 2	76 \pm 0	27 \pm 1	100 \pm 0	76 \pm 4	100 \pm 0	70.1
DP3*	100\pm0	53 \pm 2	50 \pm 5	83 \pm 3	33 \pm 2	70 \pm 6	24 \pm 4	95 \pm 3	87 \pm 3	83 \pm 3	67.8
Mamba Policy*	100 \pm 0	59 \pm 3	55 \pm 2	79 \pm 3	35\pm6	65 \pm 5	23 \pm 2	96 \pm 2	90 \pm 2	82 \pm 5	68.4
ours(<i>w/o</i> DCT)	100 \pm 0	51 \pm 5	47 \pm 3	55 \pm 6	23 \pm 2	45 \pm 5	30\pm3	95 \pm 0	88 \pm 3	80 \pm 3	61.4
ours	100 \pm 0	65\pm5	59\pm5	85\pm4	30 \pm 3	77\pm3	25 \pm 3	97\pm2	92\pm6	85\pm5	71.5

Table 2: **Main results on 48 simulation tasks.** The table reports average success rates (%) across all tasks. * denotes results we reproduced using the same expert demonstrations for fair comparison. Success rates for individual tasks are in Appendix.

Algorithm \ Task	Adroit (3)	DexArt (4)	MetaWorld Easy (20)	MetaWorld Medium (11)	MetaWorld Hard (5)	MetaWorld Very Hard (5)	Average
DP3	68.3	68.5	91.7	61.6	38.0	49.0	62.9 \pm 16.9
Diffusion Policy	31.7	49.0	86.8	31.1	10.8	26.6	39.3 \pm 23.9
DP3*	67.7	52.5	89.9	66.8	42.8	68.0	64.6 \pm 14.7
Mamba Policy*	71.3	50.5	90.2	67.4	46.4	69.0	65.8 \pm 14.4
ours	74.7	54.3	92.4	67.4	48.8	70.2	67.9\pm14.2

Table 3: **Success rate (%) comparison on the RoboTwin Benchmark for Dual-Arm Manipulation with D435 Camera Setting.** We evaluated our approach on 7 tasks using 20 expert demonstrations and 3 seeds (0, 1, 2) and report the success rate. Our method was compared against DP3 (XYZ+RGB) and DP, all tested under the same conditions. Both DP and our method are trained for 500 epochs, while DP3 is trained for 3,000 epochs.

Task	RoboTwin [26]						
	Block Hammer Beat	Block Handover	Bottle Adjust	Container Place	Empty Cup Place	Pick Apple Messy	Dual Bottles Pick(Hard)
DP	0.0 \pm 0.0	0.0 \pm 0.0	6.3 \pm 5.9	1.7 \pm 0.6	0.0 \pm 0.0	5.3 \pm 2.5	8.0 \pm 2.0
DP3(XYZ+RGB)	47.7\pm4.0	86.0\pm1.0	25.0 \pm 5.0	37.3 \pm 2.1	23.7 \pm 5.5	6.0 \pm 2.6	28.0 \pm 4.4
Ours(XYZ+RGB)	42.0 \pm 4.2	80.7 \pm 9.7	27.7\pm11.4	39.7\pm3.3	29.3\pm9.4	7.0\pm1.0	30.0\pm4.2

4.2 Simulation Result

High Success Rate. In Table 1, we compare our method on 10 tasks from Adroit, DexArt, and MetaWorld, while Table 3 presents results on 7 RoboTwin tasks using colored point clouds as observations. In Table 1, our method outperforms both the state-of-the-art (SOTA) methods and a baseline (without DCT decomposition) on 8 tasks, and achieves comparable results on the remaining 2. Similarly, as shown in Table 3, our method achieves superior performance in RoboTwin, with 5 tasks reaching higher success rates than DP and DP3. We also evaluated all 48 tasks in Adroit, DexArt, and MetaWorld and reported the average success rate for each benchmark. As shown in Table 2, our approach maintains an average success rate of 67.9%, surpassing Diffusion Policy 3D (64.6%) and Mamba Policy (65.8%). These results demonstrate the effectiveness of our approach across a variety of task settings and scenarios. *More detailed results for each individual task are provided in the appendix.*

Generalization Ability. Results in Table 4 demonstrate that our method achieves consistently strong performance in unseen DexArt test environments, maintaining high success rates under both limited (10 demonstrations) and abundant (100 demonstrations) data settings. These results demonstrate the method’s robustness and generalization ability in diverse scenarios.

Table 4: **Generalization Results.** Success rates (%) on unseen DexArt test data after 3000 training epochs, averaged over 100 trials \times 3 random seeds.

Alg \ Task	10 Demonstrations				100 Demonstrations			
	Laptop	Faucet	Toilet	Bucket	Laptop	Faucet	Toilet	Bucket
DP3	12.2 \pm 2	11.5 \pm 3	8.5 \pm 1	15.0 \pm 3	36.4 \pm 5	17.4\pm2	25.4 \pm 1	29.2\pm6
Mamba Policy	11.9 \pm 3	8.7 \pm 3	11.2 \pm 4	23.6 \pm 4	30.9 \pm 5	16.6 \pm 4	26.8 \pm 5	25.9 \pm 3
ours	14.5\pm3	12.4\pm4	16.6\pm4	32.3\pm4	40\pm3	12.1 \pm 3	27.8\pm3	26.4 \pm 4

Table 5: **Noise Robustness Evaluation.** Success rates (%) with relative change vs. high-quality baseline (see Appendix Table 10) shown as superscripts. We evaluate two noise conditions: (1) Low-Quality demonstrations using suboptimal trajectories (5 tasks, hammer excluded), and (2) Gaussian Noise (std=0.025, 0.05, 0.1) applied to high-quality baseline demonstrations (6 tasks).

Method	Low-Quality Demonstrations						Gaussian Noise std=0.025							
	Pick	Soccer	Stick	Hammer	Door	Pen	Avg.	Pick	Soccer	Stick	Hammer	Door	Pen	Avg.
DP3	10 ⁻¹⁷	15 ⁻³²	41 ⁻²⁸	—	47 ⁻¹¹	33 ⁻³⁴	29 ⁻⁴⁰	26 ⁺¹⁷	25 ⁺¹⁴	63⁺¹¹	0 ⁻¹⁰⁰	55⁺⁴	53 ⁺⁶	37 ⁻²⁴
Mamba	16 ⁻³⁶	20⁻²⁹	40 ⁻²⁷	—	44 ⁻²⁵	34⁻³⁸	31 ⁻⁴³	18 ⁻²⁸	33⁺¹⁸	61 ⁺¹¹	0 ⁻¹⁰⁰	50 ⁻¹⁵	57⁺⁴	37 ⁻³²
Ours	20⁻³³	17 ⁻⁴⁷	43⁻³¹	—	49⁻²⁵	32 ⁻⁴⁵	32⁻⁴⁴	32⁺⁷	33 ⁺³	60 ⁻³	70⁻³⁰	52 ⁻²⁰	56 ⁻³	51⁻¹³

Method	Gaussian Noise std=0.05						Gaussian Noise std=0.1							
	Pick	Soccer	Stick	Hammer	Door	Pen	Avg.	Pick	Soccer	Stick	Hammer	Door	Pen	Avg.
DP3	21 ⁺⁷⁵	19 ⁻¹⁴	51 ⁻¹¹	0 ⁻¹⁰⁰	27 ⁻⁴⁹	55 ⁺¹⁰	29 ⁻⁴¹	7 ⁻⁴²	18 ⁻¹⁸	50 ⁻¹²	0 ⁻¹⁰⁰	24 ⁻⁵⁵	38 ⁻²⁴	23 ⁻⁵³
Mamba	23 ⁻⁸	25 ⁻¹¹	60⁺⁹	0 ⁻¹⁰⁰	51⁻¹⁴	57⁺⁴	36 ⁻³³	3 ⁻⁸⁸	14 ⁻⁵⁰	39 ⁻²⁹	0 ⁻¹⁰⁰	21 ⁻⁶⁴	39 ⁻²⁹	19 ⁻⁶⁴
Ours	25⁻¹⁷	30⁻⁶	60⁻³	51⁻⁴⁹	46 ⁻²⁹	55 ⁻⁵	45⁻²³	20⁻³³	27⁻¹⁶	52⁻¹⁶	15⁻⁸⁵	32⁻⁵¹	49⁻¹⁶	33⁻⁴⁴

Noise Robustness. We evaluate robustness under two noise scenarios (Table 5): (1) training on low-quality demonstrations with reduced reward thresholds (5 tasks used, hammer task cannot generate low-reward expert demonstrations), and (2) adding Gaussian noise (std=0.025, 0.05, 0.1) to high-quality demonstrations. On low-quality demonstrations, all methods show significant performance decline, validating the importance of high-quality training data. More importantly, under Gaussian noise conditions, our method demonstrates exceptional robustness: at high noise (std=0.1), we achieve 33% success versus 19-23% for baselines. The Hammer task is particularly revealing—baselines completely fail (0% success) under any Gaussian noise, while we maintain 70%, 51%, and 15% at increasing noise levels. This superior performance stems from our low-frequency to high-frequency progressive generation mechanism’s natural anti-noise advantages: (1) *Low-frequency anti-noise characteristics*—low-frequency components inherently have stronger noise resistance, establishing a stable action foundation in initial stages; (2) *Progressive filtering*—each frequency domain level’s reconstruction process acts as natural noise filtering; (3) *Structured constraints*—the coarse-to-fine generation process ensures that key action structures are preserved even under noise interference. Interestingly, moderate noise (std=0.025) can promote generalization in certain tasks: for Pen spinning and Pick Out of Hole, low-intensity Gaussian noise unintentionally increases hand movement variation, adding action diversity and leading to higher success rates than noise-free baselines. This suggests beneficial noise augmentation can enrich the action distribution.

Table 6: **Success rates (%) comparisons on Robomimic benchmark [24] and Push-T task [12].**

Tokenizer	Model	Lift(mh) State		Can(mh) State		Square(mh) State		Transport(mh) State		Push-T State		Time	
Diffusion Models													
Continuous	DP-C	1.00/1.00	1.00/0.97	1.00/0.96	1.00/0.96	0.98/0.84	0.97/0.82	0.89/0.69	0.68/0.46	0.91/0.84	0.95/0.91	2.11	
	DP-T	1.00/0.99	1.00/1.00	1.00/0.98	1.00/0.94	0.94/0.80	0.95/0.81	0.73/0.50	0.62/0.35	0.78/0.66	0.95/0.79	1.35	
Autoregressive Models													
Discrete	BET	0.94/0.90	1.00/0.97	0.74/0.68	1.00/0.90	0.88/0.85	0.46/0.42	0.44/0.37	0.00/0.00	0.21/0.06	—	0.79/0.70	0.007
	CARP	0.94/0.90	1.00/0.97	0.74/0.68	1.00/0.90	0.88/0.84	0.46/0.42	0.44/0.37	0.00/0.00	0.21/0.06	0.88/0.83	0.85/0.83	0.09
Continuous	Ours	1.00/1.00	1.00/0.98	0.98/0.94	1.00/0.90	0.84/0.78	0.88/0.74	0.58/0.50	0.50/0.38	0.82/0.76	0.92/0.85	0.21	

Discrete vs Continuous. As shown in Table 6, our systematic comparison between discrete autoregressive and continuous diffusion models reveals a key finding: discrete representation methods (CARP, BET) exhibit significant limitations in modeling continuous action spaces, primarily due to information loss during the discretization process, particularly evident in hard tasks requiring precise control. In contrast, our proposed approach that integrates autoregressive and diffusion mechanisms in a continuous representation framework not only maintains consistent high performance across various tasks, but also inherits the computational efficiency advantages of autoregressive methods (requiring only 1/10 of DP’s inference time), conclusively demonstrating its superior expressiveness,

computational efficiency, and environmental adaptability in modeling continuous action spaces.

Inference Efficiency. As shown in Table 6, our continuous token AR method with integrated diffusion mechanisms achieves an order of magnitude faster inference (0.21s vs. 2.11s) compared to diffusion policy while maintaining comparable success rates on the Robomimic benchmark.

Flexible Sampling. Our method allows flexible selection of the number of autoregressive iterations in the frequency domain during inference. We evaluated the impact of different iteration counts on task success rates and inference times on the Adroit benchmark, and provide a Pareto frontier visualization [22] in Figure 4, which displays the tradeoff between inference time and success rate, methods closer to the top-left corner achieve better trade-offs between efficiency and performance. It shows that our method maintaining competitive performance even at minimal iterations.

4.3 Real-World Result

To validate our method in real-world applications, we collected demonstration data for an object handover task using the teleoperation system of the ShadowHand [31]. In this setup, a human subject holds an object, and the robot is required to stably receive it. The dynamic interactions between the hand and object necessitate high inference speed for real-time responsiveness. Our real-world tests, conducted on an RTX 4090 GPU, show that our single-iteration implementation achieves 70 FPS, significantly outperforming DP3’s 25 FPS. As illustrated in Figure 5, our robotic hand successfully receives the object from the human subject. These results demonstrate the practical advantages of our method for real-time robotic control applications with strict temporal constraints. *Additional experimental details for real-world scenarios are provided in the appendix.*



Figure 5: **Real-World Experiments on Robotic Handover Task.** The robotic hand stably receives an object from a human subject during real-world testing.

5 Conclusion

In this paper, we introduced FreqPolicy, a novel visuomotor policy framework leveraging hierarchical frequency-domain representations and continuous tokens for effective robotic manipulation. By decomposing action signals into frequency components using DCT, our method allows for flexible reconstruction of actions under different levels of action details. By integrating continuous latent representations with an autoregressive paradigm, our method enables precise and efficient modeling of action spaces via coarse-to-fine generation, eliminating discretization losses. Extensive experiments demonstrate that our approach achieves state-of-the-art performance, outperforming existing methods in both success rate and computational efficiency. In future work, we plan to extend our research to the VLA framework, further exploring how frequency domain representations influence action spaces in multi-task training environments.

6 Limitations

Since all of our experiments used condition inputs consistent with DP3 or DP, we have not yet explored the potential impact of altering condition input methods on model performance. Additionally, it should be noted that our method still has room for improvement in 2D tasks, and performance tends to decrease when frequency domain partitioning becomes too fine-grained.

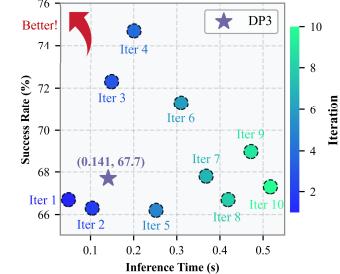


Figure 4: **Pareto Analysis on Adroit benchmark.** The x-axis represents inference time and the y-axis indicates task success rate.

7 acknowledgements

This work was supported by NSFC (No.62206173), Shanghai Frontiers Science Center of Human-centered Artificial Intelligence (ShangHAI), and MoE Key Laboratory of Intelligent Perception and Human-Machine Collaboration (KLIP-HuMaCo).

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] AgileXRobotics. Agilex robotics. <https://global.agilex.ai/>, 2025. Accessed: 2025-05-16.
- [3] AllegroHand. Allegro hand | robot hand. <https://www.allegrohand.com/>, 2024. Accessed: 2025-05-16.
- [4] Chen Bao, Helin Xu, Yuzhe Qin, and Xiaolong Wang. Dexart: Benchmarking generalizable dexterous manipulation with articulated objects. In *CVPR*, 2023a.
- [5] NVIDIA: Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzheng Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025.
- [6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.
- [7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.
- [8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [9] Jiahang Cao, Qiang Zhang, Jingkai Sun, Jiaxu Wang, Hao Cheng, Yulin Li, Jun Ma, Yecheng Shao, Wen Zhao, Gang Han, et al. Mamba policy: Towards efficient 3d diffusion policy with hybrid selective state models. *arXiv preprint arXiv:2409.07163*, 2024.
- [10] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *RSS*, 2023.
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.
- [12] Pete Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. *Conference on Robot Learning (CoRL)*, November 2021.
- [13] Letian Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, William Chung-Ho Panitch, Fangchen Liu, Hui Li, and Ken Goldberg. In-context imitation learning via next-token prediction. *arXiv preprint arXiv:2408.15980*, 2024.

[14] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *arXiv*, 2024.

[15] Niklas Funk, Julen Uraín, Joao Carvalho, Vignesh Prasad, Georgia Chalvatzaki, and Jan Peters. Actionflow: Equivariant, accurate, and efficient policies with spatially symmetric flow matching. *arXiv preprint arXiv:2409.04576*, 2024.

[16] Zhefei Gong, Pengxiang Ding, Shangke Lyu, Siteng Huang, Mingyang Sun, Wei Zhao, Zhaoxin Fan, and Donglin Wang. Carp: Visuomotor policy learning via coarse-to-fine autoregressive prediction. *arXiv preprint arXiv:2412.06782*, 2024.

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020.

[18] Syed Ali Khayam. The discrete cosine transform (dct): theory and application. *Michigan State University*, 114(1):31, 2003.

[19] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024.

[20] Xiaoyi Liu and Hao Tang. Diffffno: Diffusion fourier neural operator. *arXiv preprint arXiv:2411.09911*, 2024.

[21] Yumeng Liu, Yaxun Yang, Youzhuo Wang, Xiaofei Wu, Jiamin Wang, Yichen Yao, Sören Schwertfeger, Sibei Yang, Wenping Wang, Jingyi Yu, Xuming He, and Yuexin Ma. Realdex: towards human-like grasping for robotic dexterous hand. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI '24*, 2024.

[22] Alexander V Lotov and Kaisa Miettinen. Visualizing the pareto frontier. In *Multiobjective optimization: interactive and evolutionary approaches*, pages 213–243. Springer, 2008.

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

[24] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021.

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

[26] Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, et al. Robotwin: Dual-arm robot benchmark with generative digital twins. *arXiv preprint arXiv:2504.13059*, 2025.

[27] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots, 2024.

[28] Mang Ning, Mingxiao Li, Jianlin Su, Haozhe Jia, Lanmiao Liu, Martin Beneš, Albert Ali Salah, and Itir Onal Ertugrul. Dctdiff: Intriguing properties of image generative modeling in the dct space. *arXiv preprint arXiv:2412.15032*, 2024.

[29] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[30] 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. *arXiv*, 2017.

[31] ShadowHand. Shadow dexterous hand series – research and development tool. <https://www.shadowrobot.com/dexterous-hand-series/>, 2024. Accessed: 2025-05-16.

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

[33] Avi Singh, Huihan Liu, Gaoyue Zhou, Albert Yu, Nicholas Rhinehart, and Sergey Levine. Parrot: Data-driven behavioral priors for reinforcement learning. *arXiv preprint arXiv:2011.10024*, 2020.

[34] Yue Su, Xinyu Zhan, Hongjie Fang, Han Xue, Hao-Shu Fang, Yong-Lu Li, Cewu Lu, and Lixin Yang. Dense policy: Bidirectional autoregressive learning of actions. *arXiv preprint arXiv:2503.13217*, 2025.

[35] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

[36] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[37] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[38] Chenxi Wang, Hongjie Fang, Hao-Shu Fang, and Cewu Lu. Rise: 3d perception makes real-world robot imitation simple and effective. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2870–2877. IEEE, 2024.

[39] Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. *arXiv preprint arXiv:2407.01812*, 2024.

[40] Yuqing Wang, Zhijie Lin, Yao Teng, Yuanzhi Zhu, Shuhuai Ren, Jiashi Feng, and Xihui Liu. Bridging continuous and discrete tokens for autoregressive visual generation. *arXiv preprint arXiv:2503.16430*, 2025.

[41] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *ICLR*, 2023.

[42] Shijie Wu, Yihang Zhu, Yunao Huang, Kaizhen Zhu, Jiayuan Gu, Jingyi Yu, Ye Shi, and Jingya Wang. Afforddp: Generalizable diffusion policy with transferable affordance. *arXiv preprint arXiv:2412.03142*, 2024.

[43] Jingyun Yang, Zi-ang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning. *arXiv preprint arXiv:2407.01479*, 2024.

[44] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. arxiv 2022. *arXiv preprint arXiv:2209.00796*, 2022.

[45] Hu Yu, Hao Luo, Hangjie Yuan, Yu Rong, and Feng Zhao. Frequency autoregressive image generation with continuous tokens. *arXiv preprint arXiv:2503.05305*, 2025.

[46] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, 2020.

[47] Yanjie Ze, Zixuan Chen, Wenhao Wang, Tianyi Chen, Xialin He, Ying Yuan, Xue Bin Peng, and Jiajun Wu. Generalizable humanoid manipulation with improved 3d diffusion policies. *arXiv preprint arXiv:2410.10803*, 2024.

- [48] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.
- [49] Xinyu Zhang, Yuhan Liu, Haonan Chang, Liam Schramm, and Abdeslam Boularias. Autoregressive action sequence learning for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025.
- [50] Zihan Zhou, Animesh Garg, Dieter Fox, Caelan Garrett, and Ajay Mandlekar. Spire: Synergistic planning, imitation, and reinforcement learning for long-horizon manipulation. *arXiv preprint arXiv:2410.18065*, 2024.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer **[Yes]** , **[No]** , or **[NA]** .
- **[NA]** means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "**[Yes]**" is generally preferable to "**[No]**", it is perfectly acceptable to answer "**[No]**" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "**[No]**" or "**[NA]**" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer **[Yes]** to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

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

Answer: **[Yes]**

Justification: The abstract and introduction accurately describe the proposed framework FreqPolicy, contributions, and experimental scope which align with the content presented in the paper section 3, section 4.

Guidelines:

- The answer NA 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 NA 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 will discuss the limitations of our work in the Appendix.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA 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 elaborate our methodology in detail in section 3 as well as disclose our implementing details in section 4.

Guidelines:

- The answer NA 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: **[No]**

Justification: Currently, the code is not publicly available, but we will release it along with detailed instructions upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: The implementation details (data splits, hyperparameters, optimizers, etc.) of all the simulation experiments in this paper remain exactly the same as in previous work (presented in subsection 4.1), while the specific details of the real-world experiments are described in the Appendix.

Guidelines:

- The answer NA 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: The experimental part of this paper is consistent with previous work, i.e., we report the error bars of our experiments where they report the error bars.

Guidelines:

- The answer NA 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 will report the computational resources required for all experiments in the Appendix.

Guidelines:

- The answer NA 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 confirm that the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA 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: Our method does not pose any negative societal impacts. The potential positive societal impacts have been discussed in the section 1: our approach can advance the development of vision-driven strategies for robotics, thereby facilitating progress in real-world robotic applications.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks.

Guidelines:

- The answer NA 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: For all the priors works we quote and compare, we make reference to their original papers.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not release new assets.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method does not involve LLMs.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

This document contains supplementary materials for our main paper. We provide further technical details, additional experimental results, and more qualitative examples to complement the findings presented in the main text. We hope this supplementary information will help readers better understand our approach and results.

The remainder of this supplementary material is organized as follows. In Section A, we provide the hardware specifications used in our experiments. In Section B, we list the hyperparameters employed. Section C presents the detailed algorithms for training and inference. In Section D, we describe the ablation studies conducted. Section E outlines the details of our real-world experiments. Finally, in Section F, we offer further discussion on VLA models and frequency domain analysis.

A Computational Resources

To ensure reproducibility, we provide detailed information on the computational resources used in our experiments. For all simulation environment experiments including training, inference, and time benchmarking tests, we used NVIDIA RTX 2080Ti GPUs. Our model has 63M parameters, with DP3 at 255M, consuming approximately 4.5GB of memory during operation. For real-world environment experiments, we employed NVIDIA RTX 4090 GPUs for training, inference, and time benchmarking tests.

B Hyperparameters

In Table 7, we present the hyperparameters used in our experiments. For the baseline methods DP and DP3, we use their default hyperparameters. For the Adroit, DexArt, and MetaWorld benchmarks, our models are trained for 3,000 epochs. For the Robomimic and Push-T tasks, we use 1,000 training epochs, and for the RoboTwin benchmark, our models are trained for 500 epochs.

Table 7: Hyperparameters used for various benchmark.

Hyperparameter	Value
Horizon (T_h)	16 (8 for RoboTwin)
Action step (T_a)	8 (6 for RoboTwin)
Observation step (T_o)	2 (3 for RoboTwin)
point_feature_dim	64
state_mlp_size	64
Batchsize	128
Num_iter	4
Num_training_steps(Diffusion training)	100
Num_sampling_steps(Diffusion sampling)	ddim10
Diffloss_d	3
Diffloss_w	1024
encoder_embed_dim	512
decoder_embed_dim	512
encoder_depth	4
decoder_depth	4
encoder_num_heads	8
decoder_num_heads	8
Optimizer	AdamW
Betas (β_1, β_2)	[0.95, 0.999]
Learning Rate	1.0e-4
Weight Decay	1.0e-6
Learning Rate Scheduler	Cosine

C Training and Inference Details

The training process for our FreqPolicy is outlined in Algorithm 1. At each epoch, we first encode the input observations using an observation encoder. The ground truth action sequence is then transformed into the frequency domain via the Discrete Cosine Transform (DCT), and a frequency index is randomly sampled. Conditional reconstruction is performed by applying the inverse DCT up to the sampled frequency level, enabling the model to focus on different frequency components during training. An adaptive mask ratio is determined based on the frequency index, and a mask is sampled accordingly. The masked observation and conditional reconstruction are then encoded and subsequently decoded by the FreqPolicy encoder and decoder, respectively. The diffusion model is trained to predict noise added to the actions at randomly sampled diffusion steps, using a standard mean squared error loss. Model parameters are updated by minimizing this loss throughout the training epochs. This procedure effectively leverages masked autoregressive modeling and diffusion-based generation, enabling the policy to learn robust representations across the frequency domain.

Algorithm 1 FreqPolicy Training

Require: Number of training epochs K , Observation Encoder \mathcal{E}_{obs} , FreqPolicy Encoder \mathcal{E} , FreqPolicy Decoder \mathcal{D} , diffusion model ϵ_θ , Observation \mathcal{O} , Ground truth action \mathbf{x} , Horizon T , diffusion steps T_{diff} , initial mask ratio m .

- 1: **for** $e = 1$ to K **do**
- 2: $z_{obs} \leftarrow \mathcal{E}_{obs}(\mathcal{O})$ ▷ Encode observations
- 3: $\{X_0, X_1, \dots, X_{T-1}\} \leftarrow \text{DCT}(\mathbf{x})$, $k \sim \mathcal{U}(0, T)$ ▷ Apply DCT, sample index
- 4: $y^k \leftarrow \begin{cases} \text{IDCT}(\{X_0, X_1, \dots, X_{k-1}\}) & \text{if } k > 0 \\ \mathbf{0} & \text{if } k = 0 \end{cases}$ ▷ Extract k-level reconstruction
- 5: $\text{mask_ratio} \leftarrow m \cdot (1 - k/T)$ ▷ Adaptive mask ratio
- 6: $\text{mask} \sim \text{TruncNorm}(\text{mask_ratio})$ ▷ Encode with mask
- 7: $z_{mask} \leftarrow \mathcal{E}(z_{obs}, y^k, k, \text{mask})$ ▷ Decode with mask
- 8: $z^k \leftarrow \mathcal{D}(z_{obs}, z_{mask}, k, \text{mask})$
- 9: $t \sim \mathcal{U}(1, T_{diff})$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
- 10: $\mathbf{x}_t \leftarrow \sqrt{\bar{\alpha}_t} \mathbf{x} + \sqrt{1 - \bar{\alpha}_t} \epsilon$
- 11: $\mathcal{L} \leftarrow \mathbb{E}_{\epsilon, t} [\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, k, z^k)\|^2]$ ▷ Diffusion loss
- 12: Update model parameters by minimizing \mathcal{L}
- 13: **end for**
- 14: **return** Trained models \mathcal{E}_{obs} , \mathcal{E} , \mathcal{D} , and ϵ_θ

The inference process for FreqPolicy is detailed in Algorithm 2. Given an input observation, we first encode it using the observation encoder. The action tokens are initialized to zeros and are fully masked at the beginning. For each iteration, the model progressively predicts actions at increasing frequency levels, guided by the current frequency index.

At each step k , the partially reconstructed tokens and the current mask are passed through the encoder and decoder to obtain the continuous latent code z^k . The diffusion sampler then generates an updated action prediction conditioned on z^k . If it is not the final step, the predicted actions are transformed into the frequency domain using the Discrete Cosine Transform (DCT), and the reconstruction is refined up to the next frequency level via inverse DCT. The masking ratio is adaptively reduced at each iteration, gradually revealing more of the reconstructed action sequence. This process continues until the entire action sequence is fully reconstructed. By progressively incorporating information across different frequency bands, our inference procedure enables the policy to generate high-fidelity predictions.

D Ablation

To verify the effectiveness of our proposed method and the contribution of each component, we conducted a series of ablation experiments.

The results in Table 8 present the ablation study on the prediction horizon, it shows that the performance is not sensitive to this parameter, demonstrating the robustness of our method. When $T_h = 8$

Algorithm 2 FreqPolicy Inference

Require: Observation Encoder \mathcal{E}_{obs} , FreqPolicy Encoder \mathcal{E} , FreqPolicy Decoder \mathcal{D} , DiffusionSampler \mathcal{F} , diffusion steps T_{diff} , Observation \mathcal{O} , Horizon T , Number of iterations N_{iter} , Frequency indices $\{i_0, i_1, \dots, i_{N_{iter}-1}\}$.

- 1: **Initialize:**
- 2: $mask \leftarrow \mathbf{1}_{B \times T}$ ▷ Full masking initially
- 3: $tokens \leftarrow \mathbf{0}_{B \times T \times D_{action}}$ ▷ Zero initialization
- 4: $z_{obs} \leftarrow \mathcal{E}_{obs}(\mathcal{O})$ ▷ Encode observations
- 5: **for** step = 0 to $N_{iter} - 1$ **do**
- 6: $k \leftarrow i_{\text{step}}$ ▷ Current frequency index
- 7: $z_{mask} \leftarrow \mathcal{E}(z_{obs}, tokens, k, mask)$
- 8: $z^k \leftarrow \mathcal{D}(z_{obs}, z_{mask}, k, mask)$
- 9: $\hat{x} \leftarrow \mathcal{F}(z^k, k, T_{diff})$ ▷ Generate prediction via diffusion
- 10: **if** step < $N_{iter} - 1$ **then**
- 11: $\{X_0, X_1, \dots, X_{T-1}\} \leftarrow \text{DCT}(\hat{x})$ ▷ Transform to frequency domain
- 12: $\text{next_k} \leftarrow i_{\text{step}+1}$ ▷ Next frequency level
- 13: $tokens \leftarrow \text{IDCT}(\{X_0, X_1, \dots, X_{\text{next_k}-1}\})$
- 14: **else**
- 15: $tokens \leftarrow \hat{x}$
- 16: **end if**
- 17: $mask_ratio \leftarrow \cos\left(\frac{\pi}{2} \cdot \frac{\text{step}+1}{N_{iter}}\right)$
- 18: $mask \leftarrow \text{GenerateMask}(mask_ratio)$
- 19: **end for**
- 20: **return** tokens ▷ Final action sequence

or $T_h = 16$, the model achieves slightly better performance with an average score over 58 points, while both smaller and larger horizons yield comparable results. This robustness across a range of temporal scales highlights the flexibility of our approach. Notably, only at extremely long horizons ($T_h = 64$) do we observe a noticeable decline in performance.

Table 9 presents the results of the ablation study on masking strategies. Our frequency policy mask significantly improves performance across all tasks, increasing the average score from 40 to 58—an improvement of approximately 45%. These results clearly demonstrate the importance of the frequency-based masking strategy for prediction tasks.

Table 8: **Ablation study on prediction horizon.** Analysis of Horizon (T_h), Action step (T_a), and Observation step (T_o).

T_h	T_o	T_a	Hammer	Door	Pen	Pick Out of Hole	Soccer	Stick Pull	Average
4	2	1	65±6	76±3	55±5	37±4	23±4	60±2	53±17
4	2	2	62±4	71±4	53±6	31±3	38±3	64±2	53±14
8	2	4	100±0	68±2	52±4	25±2	38±4	62±0	58±24
8	2	6	100±0	72±4	50±5	29±3	31±2	64±3	58±25
16	2	8	100±0	65±5	59±5	30±2	32±4	62±0	58±23
16	2	12	100±0	59±4	51±3	35±2	35±4	55±5	56±21
32	2	16	98±2	58±5	38±2	34±3	19±2	52±4	50±25
64	2	32	80±4	35±4	42±2	35±5	32±3	38±4	44±17

Table 9: **Ablation study on mask.** This experiment analyzes the model performance with and without mask.

	Hammer	Door	Pen	Pick Out of Hole	Soccer	Stick Pull	Average
W/o mask	99±1	35±4	31±3	15±0	27±2	34±2	40±27
Freqpolicy	100±0	65±5	59±5	30±2	32±4	62±0	58±23

Table 10: **Main results on 48 simulation tasks.** Success rates (%) for each task are provided in this table.

Alg \ Task	Meta-World [46] (Easy)												
	Button Press	Button Press Wall	Coffee Button	Dial Turn	Door Close	Reach Wall	Door Open	Door Unlock	Drawer Close	Drawer Open			
DP3	100\pm0	99 \pm 1	100\pm0	66 \pm 1	100\pm0	68 \pm 3	99 \pm 1	100\pm0	100 \pm 0	100\pm0			
Diffusion Policy	99 \pm 1	97 \pm 3	99 \pm 1	63 \pm 10	100 \pm 0	59 \pm 7	98 \pm 3	98 \pm 3	100 \pm 0	93 \pm 3			
DP3*	100\pm0	100\pm0	100\pm0	58 \pm 5	100 \pm 0	47 \pm 5	100\pm0	100\pm0	100 \pm 0	100\pm0			
Mamba Policy*	100 \pm 0	100 \pm 0	100 \pm 0	56 \pm 4	100 \pm 0	50 \pm 3	100 \pm 0	100 \pm 0	100 \pm 0	100 \pm 0			
ours	100 \pm 0	100 \pm 0	100 \pm 0	72\pm4	100 \pm 0	71\pm4	100 \pm 0	100 \pm 0	100 \pm 0	100 \pm 0			
Alg \ Task	Meta-World (Easy)												
	Faucet Open	Handle Press	Lever Pull	Plate Slide	Plate Slide Back	Plate Slide Back Side	Plate Slide Side	Reach	Window Close	Window Open			
DP3	100\pm0	100\pm0	79 \pm 8	100 \pm 1	99 \pm 0	100\pm0	100\pm0	24 \pm 1	100\pm0	100\pm0			
Diffusion Policy	100 \pm 0	81 \pm 4	49 \pm 5	83 \pm 4	99 \pm 0	100\pm0	100 \pm 0	18 \pm 2	100 \pm 0	100 \pm 0			
DP3*	100 \pm 0	86 \pm 5	84\pm2	100\pm0	100\pm0	100\pm0	100 \pm 0	22 \pm 4	100 \pm 0	100 \pm 0			
Mamba Policy*	100 \pm 0	83 \pm 5	74 \pm 6	100\pm0	100\pm0	100\pm0	100\pm0	17 \pm 3	100 \pm 0	100 \pm 0			
ours	100 \pm 0	90 \pm 3	84\pm4	100\pm0	100\pm0	100\pm0	100\pm0	30 \pm 2	100 \pm 0	100 \pm 0			
Alg \ Task	Meta-World (Medium)												
	Hammer	Peg Insert Side	Push Wall	Soccer	Sweep	Sweep Into	Basketball	Bin Picking	Box Close	Coffee Pull	Coffee Push		
DP3	76 \pm 4	69\pm7	49 \pm 8	18 \pm 3	96 \pm 3	15 \pm 5	98 \pm 2	34 \pm 30	42 \pm 3	87 \pm 3	94\pm3		
Diffusion Policy	15 \pm 6	34 \pm 7	20 \pm 3	14 \pm 4	18 \pm 8	10 \pm 4	85 \pm 6	15 \pm 4	30 \pm 5	34 \pm 7	67 \pm 4		
DP3*	80 \pm 5	62 \pm 5	87 \pm 5	22 \pm 3	100\pm0	17 \pm 2	100\pm0	35\pm10	50 \pm 6	95 \pm 0	87 \pm 5		
Mamba Policy*	90 \pm 5	63 \pm 4	92 \pm 2	28 \pm 3	95 \pm 5	15 \pm 3	100\pm0	26 \pm 3	48 \pm 10	96 \pm 2	89 \pm 3		
ours	96\pm1	51 \pm 4	97\pm3	32\pm4	85 \pm 4	19\pm5	83 \pm 8	31 \pm 2	56\pm4	100\pm0	91 \pm 4		
Alg \ Task	Meta-World (Hard)					Meta-World (Very Hard)							
	Assembly	Hand Insert	Pick Out of Hole	Pick Place	Push	Shelf Place	Disassemble	Stick Pull	Stick Push	Pick Place Wall			
DP3	99\pm1	14 \pm 4	14 \pm 9	12 \pm 4	51 \pm 3	17 \pm 10	69 \pm 4	27 \pm 8	97\pm4	35 \pm 8			
Diffusion Policy	15 \pm 1	9 \pm 2	0 \pm 0	0 \pm 0	30 \pm 3	11 \pm 3	43 \pm 7	11 \pm 2	63 \pm 3	5 \pm 1			
DP3*	95 \pm 3	11 \pm 2	12 \pm 7	42\pm6	54 \pm 4	31 \pm 4	87 \pm 3	57 \pm 3	83 \pm 3	82 \pm 6			
Mamba Policy*	96 \pm 2	15 \pm 3	25 \pm 5	36 \pm 10	60 \pm 4	38\pm4	90 \pm 2	55 \pm 2	82 \pm 5	80 \pm 5			
ours	97 \pm 2	17\pm3	30\pm2	37 \pm 4	63\pm2	27 \pm 3	92\pm6	62\pm0	85 \pm 5	85\pm3			
Alg \ Task	Adroit [30]					DexArt [4]			Average (41+3+4)				
	Hammer	Door	Pen	Laptop	Faucet	Toilet	Bucket						
DP3	100\pm0	62 \pm 4	43 \pm 6	83 \pm 1	63\pm2	82\pm4	46\pm2				71.36		
Diffusion Policy	45 \pm 5	37 \pm 2	13 \pm 2	69 \pm 4	23 \pm 8	58 \pm 2	46\pm1				53.25		
DP3*	100\pm0	53 \pm 2	50 \pm 5	83 \pm 3	33 \pm 2	70 \pm 6	24 \pm 4				72.91		
Mamba Policy*	100\pm0	59 \pm 3	55 \pm 2	79 \pm 3	35 \pm 6	65 \pm 5	23 \pm 2				73.71		
ours	100\pm0	65\pm5	59\pm5	85\pm4	30 \pm 3	77 \pm 3	25 \pm 3				75.54		

E Real-world Experiment Details

In real-world experiments, we employ an RGB-D camera (Kinect) to capture environmental point clouds at a rate of 30 Hz. The FoundationPose model serves as our object pose estimation module, which processes the point clouds from the RGB-D camera to predict the 6D object pose, achieving a 93.22% ADD AUC score on the YCBInEOAT dataset at 30 FPS. By sampling 4,096 points from the object’s mesh and applying the estimated pose from FoundationPose, we obtain cleaned object point clouds for the past 5 frames ($T_0 = 5$).

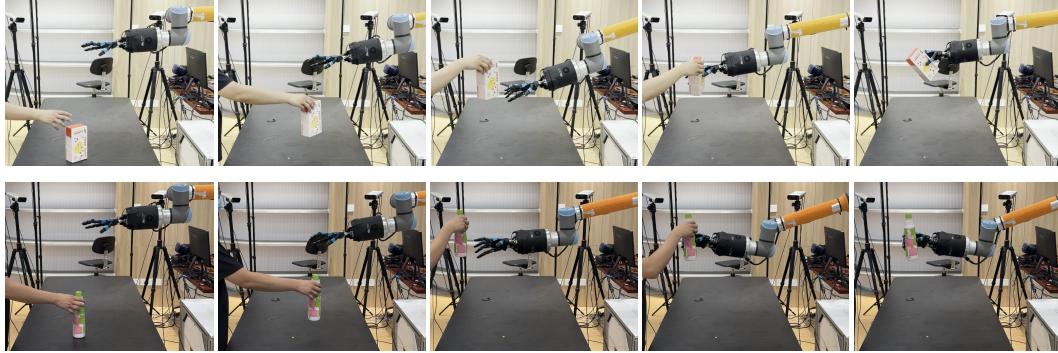


Figure 6: **Additional Visualization of Real-World Experiments on 2 Robotic Handover Tasks.**

Our algorithm takes these cleaned object point clouds along with the shadow hand poses from the previous 5 frames as input conditions, and predicts the hand’s poses for the next 3 frames ($T_a = 3$). These predicted poses are executed directly on the physical robot, enabling a fully end-to-end action generation system. In terms of system performance, the perception module operates at approximately 30 FPS, while the action prediction module achieves over 70 FPS with a single iteration setting. The

complete end-to-end system maintains a comprehensive operating rate of approximately 25 FPS, meeting real-time interaction requirements. Figure 6 shows the results of two additional sets of handover tasks.

F Discussion

F.1 Further Discussion on Vision-Language-Action (VLA) Models

In the main sections of this paper, we have detailed and validated the superior performance of FreqPolicy in learning specific robotic manipulation tasks. Its innovative frequency-domain autoregressive mechanism and the use of continuous tokens have demonstrated significant advantages in both precision and efficiency for single-task learning. However, a natural and promising extension is to investigate the adaptability of FreqPolicy in more complex and generalized multitask learning scenarios, especially when task instructions are given in natural language. Vision-Language-Action (VLA) models provide a powerful framework for achieving such language-driven multitask robot control.

Therefore, this section aims to conduct a preliminary exploration of FreqPolicy’s potential when applied to VLA models. Given that policy learning in multitask environments is still a multifaceted and challenging research problem, we do not aim to provide a solution here. Instead, our goal is to assess FreqPolicy on a multitask benchmark and to provide a forward-looking discussion on whether the core ideas of FreqPolicy, such as frequency-domain decomposition and hierarchical learning, can benefit VLA models. We also outline possible directions for future research. We believe that this discussion can help to provide a more comprehensive understanding of FreqPolicy’s potential and its future development.

Multitask Benchmark. To preliminarily assess FreqPolicy’s performance in a multitask setting, we selected RoboCasa [27] as the benchmark platform. RoboCasa comprises a suite of tasks defined within a simulated kitchen environment, representing complex interaction scenarios that robots might encounter in the real world. In this exploratory experiment, we focused on 24 "atomic" tasks, which cover fundamental sensorimotor skills such as pick-and-place, opening and closing doors, pressing buttons, and turning faucets. Placing our method in such a simulated environment, which possesses a certain level of difficulty and a rich variety of tasks, helps us to preliminarily understand its potential and areas for exploration in multitask learning.

Baseline. For effective comparison, we first selected the Diffusion Policy and the autoregressive method BC-Transformer [25] implemented in RoboCasa as direct baselines for our FreqPolicy. Considering the characteristics of VLA models, we further introduced GR00T-N1 [5] as a reference. GR00T-N1 is a VLA model with a tightly coupled dual-system, where its vision-language module is responsible for understanding the environment and instructions, and the subsequent Diffusion Transformer module employs Flow Matching technology to generate smooth action sequences in real-time. Selecting these baselines helps us to more clearly position the relative performance of FreqPolicy within existing VLA frameworks.

Implementation Details. In this initial exploration, our FreqPolicy model is directly integrated on top of the Diffusion Policy framework within RoboCasa. To ensure a fair comparison, the main parameters and observation inputs used by FreqPolicy are kept consistent with those of Diffusion Policy. This simplified integration aims to quickly validate the basic adaptability of FreqPolicy’s core mechanisms in a multitask scenario, rather than to perform deep customization and optimization.

Results and Discussion. The experimental results, as shown in Table 11, offer an initial insight into FreqPolicy’s performance on the RoboCasa multitask benchmark. The data indicate that, compared to Diffusion Policy and BC-Transformer, our method demonstrates a certain advantage in overall multitask success rates. This preliminarily suggests that FreqPolicy’s frequency-domain processing mechanism, and its hierarchical modeling approach to action sequences, are not only effective for single-task learning but may also bring positive impacts to scenarios requiring the simultaneous handling of multiple tasks. However, when compared to the GR00T-N1 model, which is specifically designed for VLA, FreqPolicy’s current performance still shows a gap in multi-task success rates. We attribute this primarily to the fact that FreqPolicy, in its current design and similar to Diffusion Policy, focuses more on action generation and optimization. It has not been specifically enhanced for deep understanding of complex language instructions and multi-modal scene perception to

Table 11: **Multitask results on RoboCasa.** Experimental results of BC-Transformer, Diffusion Policy and GR00T-N1 are from the GR00T-N1 paper.

	BC-Transformer	Diffusion Policy	GR00T-N1	FreqPolicy(Ours)
Success Rate	26.3%	25.6%	32.1%	27.4%

the same extent as GR00T-N1. One of the core strengths of VLA models lies in their powerful semantic understanding and scene perception capabilities, enabling them to generalize better to unseen instruction and environment combinations.

Nevertheless, these preliminary results also provide us with important insights: could FreqPolicy’s unique frequency-domain analysis and modeling approach serve as a beneficial supplement when integrated into more powerful VLA frameworks? For instance, FreqPolicy’s ability to capture the smoothness and structural information of action signals might assist VLA models in generating more stable and physically plausible action sequences. Exploring how to effectively combine the frequency-domain strengths of FreqPolicy with the semantic understanding capabilities of VLA models, with the aim of further enhancing overall multi-task learning performance, will be a highly valuable research direction for us in the future. This might involve designing new fusion mechanisms or tailoring FreqPolicy’s frequency decomposition strategies and autoregressive processes to the specific characteristics of VLA tasks. In summary, while FreqPolicy was not natively designed for VLA tasks, its core ideas demonstrate a potential worthy of further investigation in the broader field of multi-task robot learning.

F.2 Discussion on Frequency Domain

In the preceding discussions, we have preliminarily shown that the action signals in robotic manipulation tasks exhibit significant compressibility in the frequency domain, with most critical information concentrated in the lower frequency bands. To make this statement precise, we quantify “compressibility” by the energy proportion:

$$E(p) = \sum_{k=0}^{\lfloor (N-1)p\% \rfloor} |X_k|^2 / \sum_{k=0}^{N-1} |X_k|^2, \quad (4)$$

i.e. the share of total signal energy that is preserved when only the lowest $p\%$ DCT coefficients are retained (see Figure 1).

This section aims to provide a more in-depth discussion and analysis of the frequency-domain characteristics of action signals, based on a broader set of tasks and more detailed visualizations (as shown in Figures 7 to 30). These supplementary figures provide action visualizations (a), frequency band energy heatmaps for each action dimension (b), and success rate curves for actions reconstructed with varying frequency ratios (c) for each task, thereby offering more robust support for our core arguments.

High-Dimensional Tasks. In high-dimensional action spaces (22 dimensions) within Dexart tasks (Figures 7-10, Dexart Bucket, Faucet, Laptop and Toilet), we observe consistent trends. The success rate curves (c) for these tasks generally show that even using only 30%-70% of the low-frequency components is often sufficient to reconstruct action sequences capable of task completion, strongly supporting the core hypothesis that high-frequency components contribute relatively little to the macroscopic success of these complex tasks. Concurrently, energy heatmaps (b) clearly demonstrate that different action dimensions exhibit varying dependencies on frequency components; some dimensions (like large-range arm movements) have energy highly concentrated in very low-frequency bands, while others (like fine finger postures) might retain significant energy in relatively higher bands. For instance, in Dexart Faucet (Figure 8b), energy distribution in the 10-30% or even higher frequency bands for some dimensions might correspond to fine adjustments for turning a faucet. Although overall trends are similar, the minimum frequency ratio for high success rates varies slightly across Dexart tasks, with Dexart Laptop (Figure 9c), for example, reaching a success plateau around a 0.4-0.6 frequency ratio, suggesting subtle differences in action signal fidelity requirements for various complex manipulations.

Low-Dimensional Tasks. Compared to high-dimensional Dexart tasks, low-dimensional MetaWorld tasks (Figures 11-30, typically 4-dimensional action spaces) exhibit a more pronounced

low-frequency dominance. In most Meta-World tasks, success rate curves (c) indicate that only 10%-40% of the low-frequency ratio is sufficient for near-perfect task success, with tasks like Meta-World Coffee-Pull (Figure 13c) and Meta-World Disassemble (Figure 15c) requiring only about 20% low-frequency signal. This suggests higher compressibility in action signals for these simpler robotic tasks, corroborated by their energy heatmaps (b) where most action dimensions show energy concentrated in the lowest 0-10% band, consistent with their typically smoother, direct motion trajectories. Minor exceptions, such as Meta-World Shelf-Place (Figure 28c) or Meta-World Push-Wall (Figure 26c), might need slightly more frequency components due to potentially higher precision demands at the end-effector.

Discussion. Synthesizing these analyses, we discover the universality and variability of frequency compression across task types: action signals are generally compressible in both high-dimensional complex and low-dimensional structured tasks, though the required frequency bandwidth varies with task dimensionality, complexity, and operational specifics. The heterogeneity of action dimensions highlighted by heatmap analysis suggests that future work on dimension-adaptive frequency processing, rather than global uniform cutoffs, could be a promising optimization, such as dynamically allocating frequency components based on each dimension’s energy to balance performance and efficiency. A deeper understanding of task action frequency characteristics can also guide policy learning algorithm design, for example, by using low-frequency biases or stronger regularization for low-frequency dominated tasks to accelerate learning and enhance generalization.

In conclusion, the additional frequency domain analysis in this section provides more comprehensive empirical support for FreqPolicy’s core mechanisms, points to valuable directions for optimizing frequency-based robot learning, deepens our understanding of robotic action nature, and underscores the potential of the frequency-domain perspective in building efficient, robust robotic agents.

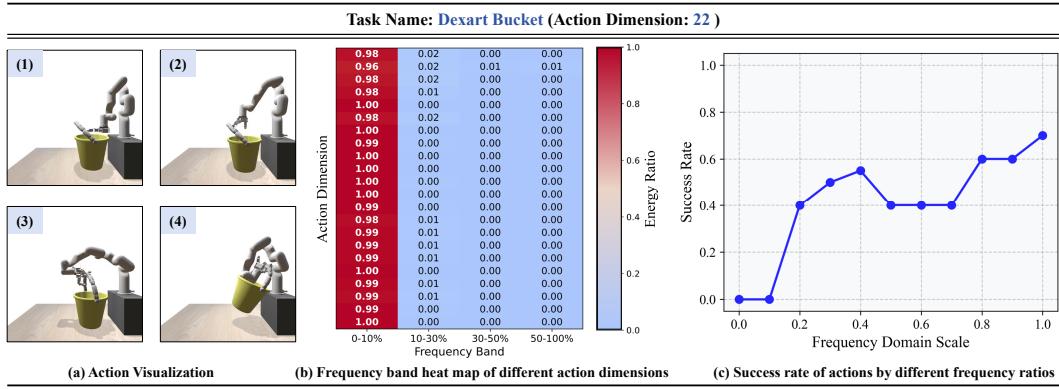


Figure 7: Frequency Domain Analysis of Dexart Bucket.

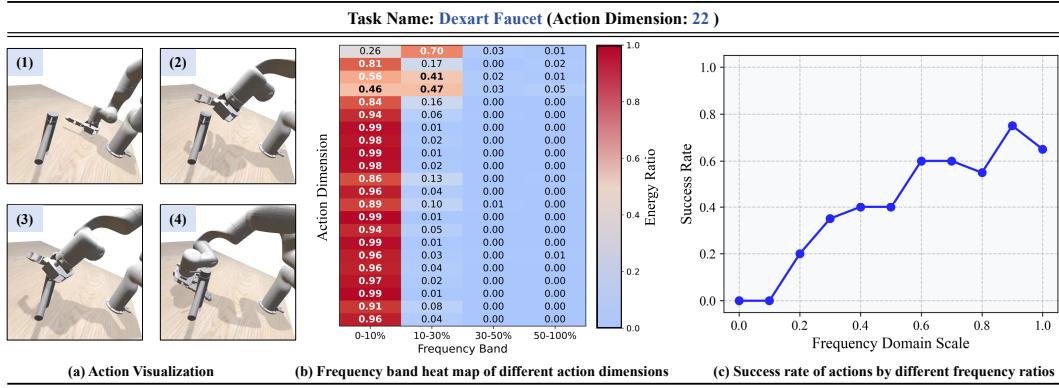


Figure 8: Frequency Domain Analysis of Dexart Faucet.

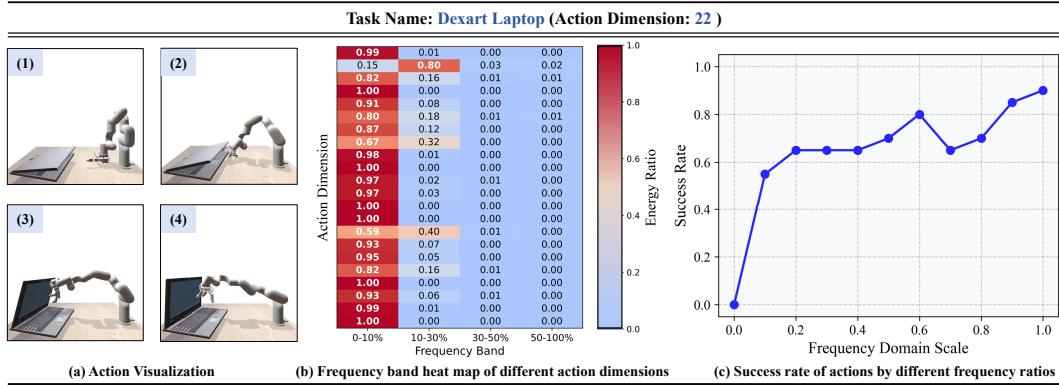


Figure 9: Frequency Domain Analysis of Dexart Laptop.

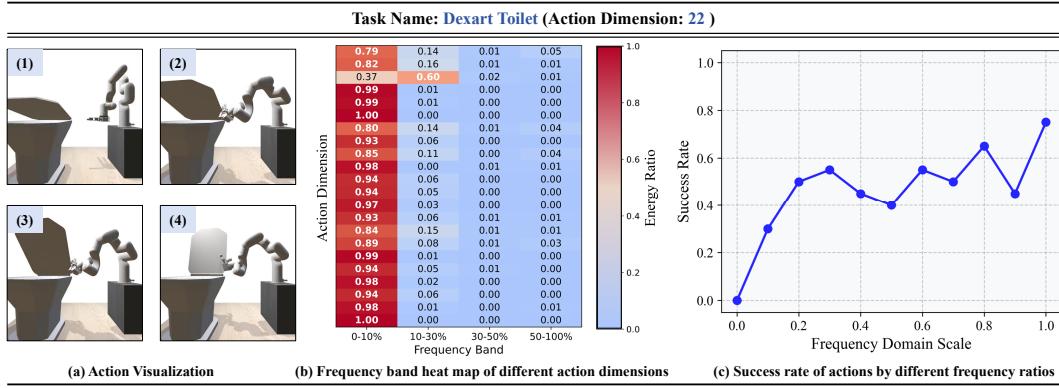


Figure 10: Frequency Domain Analysis of Dexart Toilet.

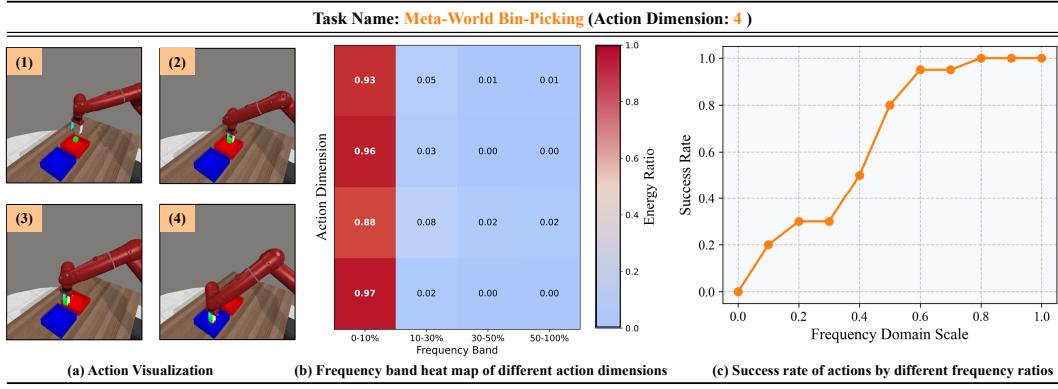


Figure 11: Frequency Domain Analysis of Meta-World Bin-Picking.

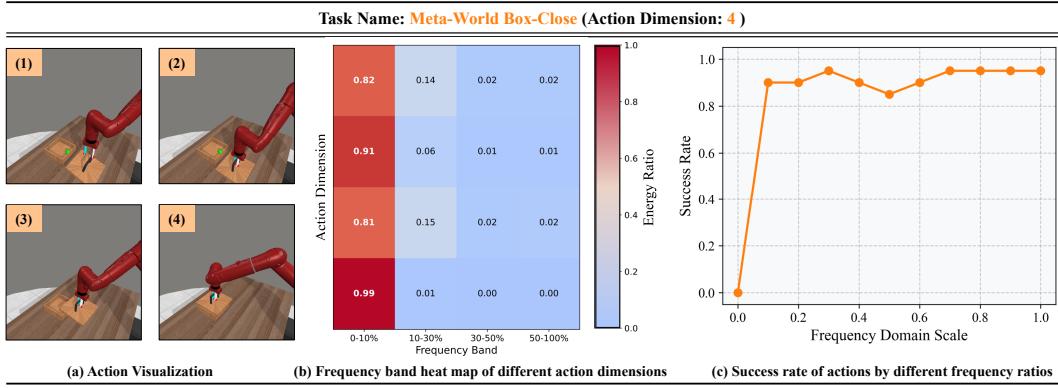


Figure 12: Frequency Domain Analysis of Meta-World Box-Close.

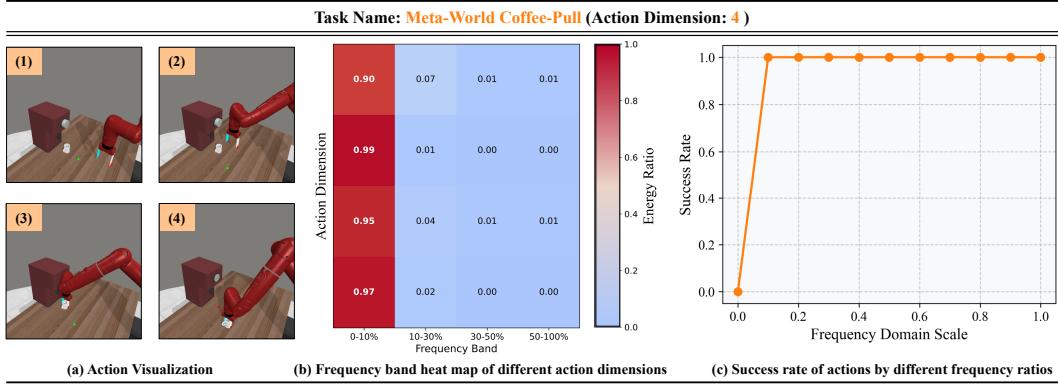


Figure 13: Frequency Domain Analysis of Meta-World Coffee-Pull.

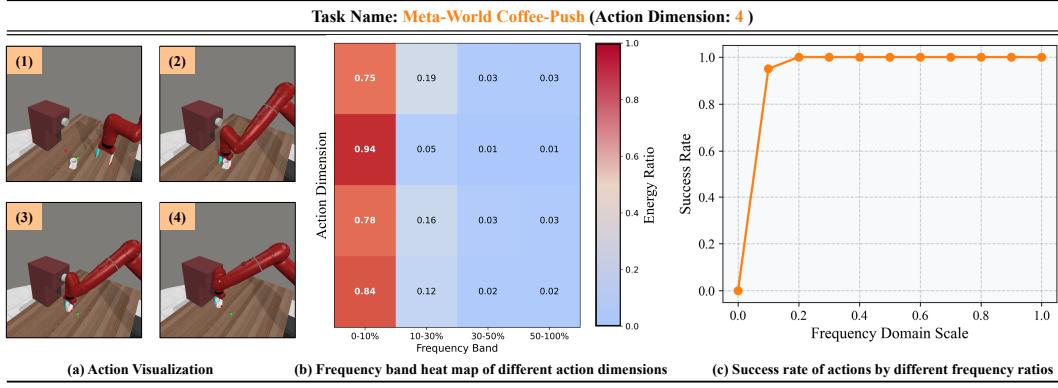


Figure 14: Frequency Domain Analysis of Meta-World Coffee-Push.

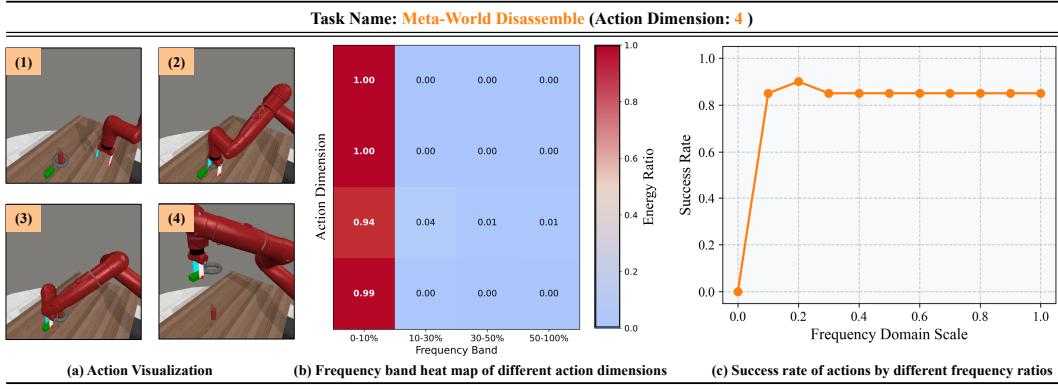


Figure 15: Frequency Domain Analysis of Meta-World Disassemble.

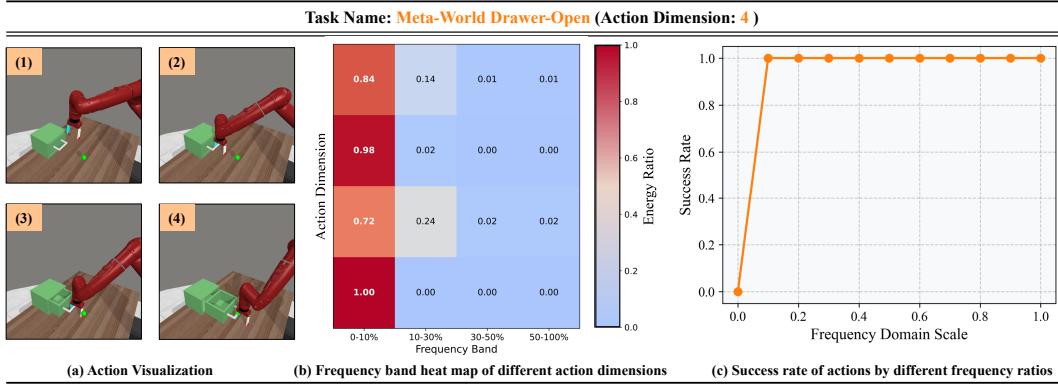


Figure 16: Frequency Domain Analysis of Meta-World Drawer-Open.

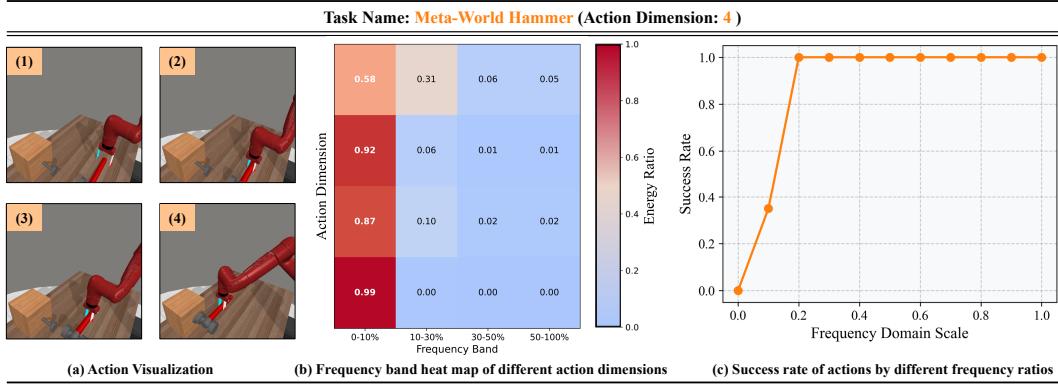


Figure 17: Frequency Domain Analysis of Meta-World Hammer.

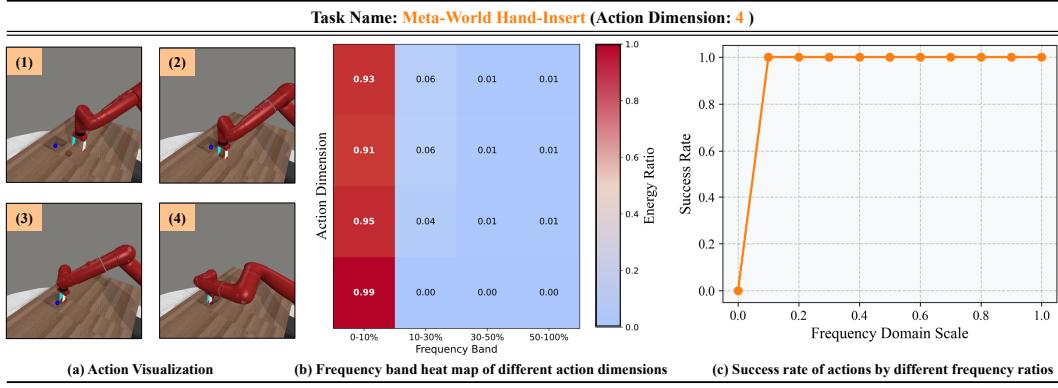


Figure 18: Frequency Domain Analysis of Meta-World Hand-Insert.

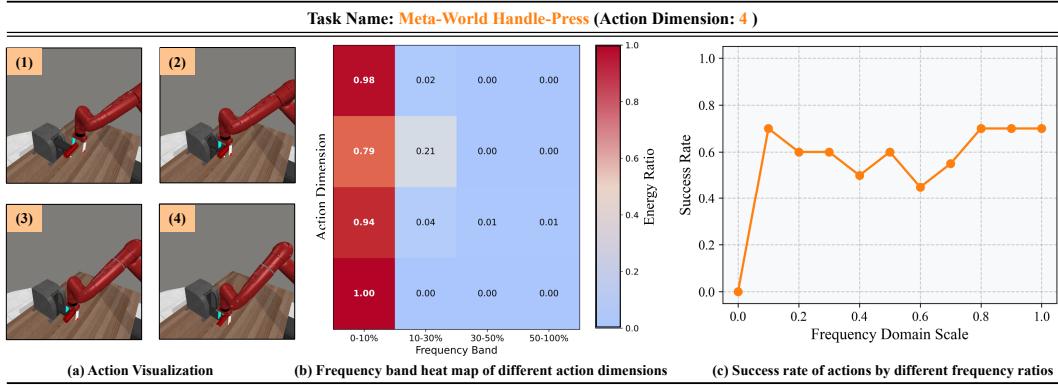


Figure 19: Frequency Domain Analysis of Meta-World Handle-Press.

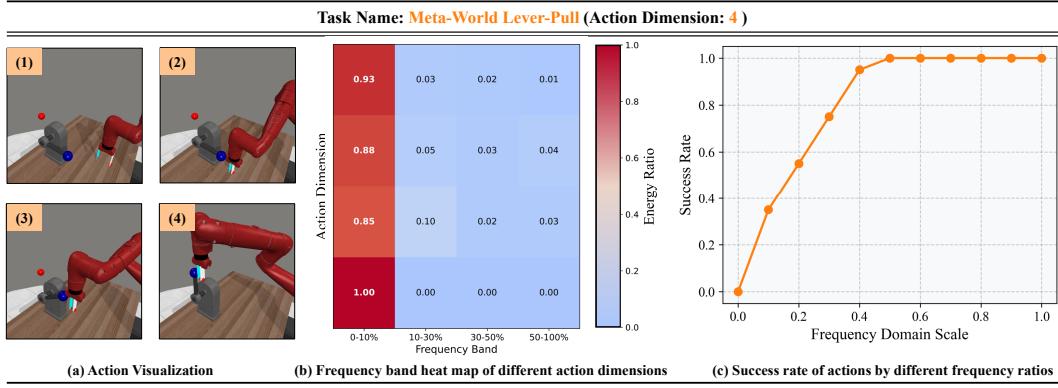


Figure 20: Frequency Domain Analysis of Meta-World Lever-Pull.

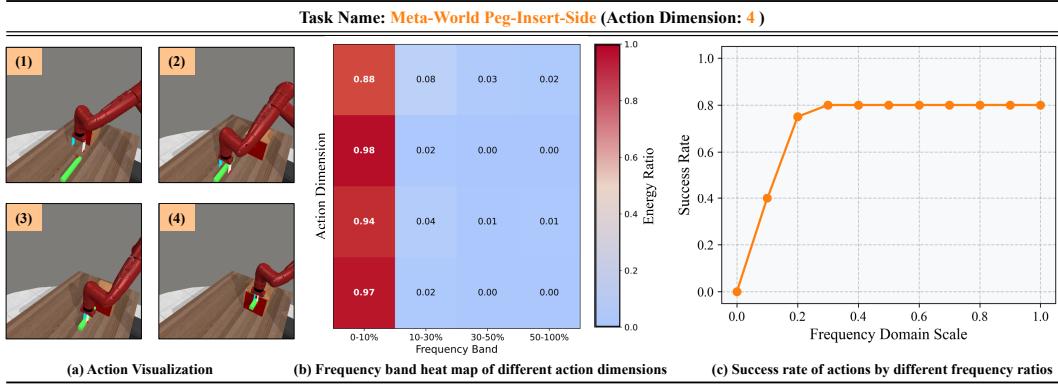


Figure 21: Frequency Domain Analysis of Meta-World Peg-Insert-Side.

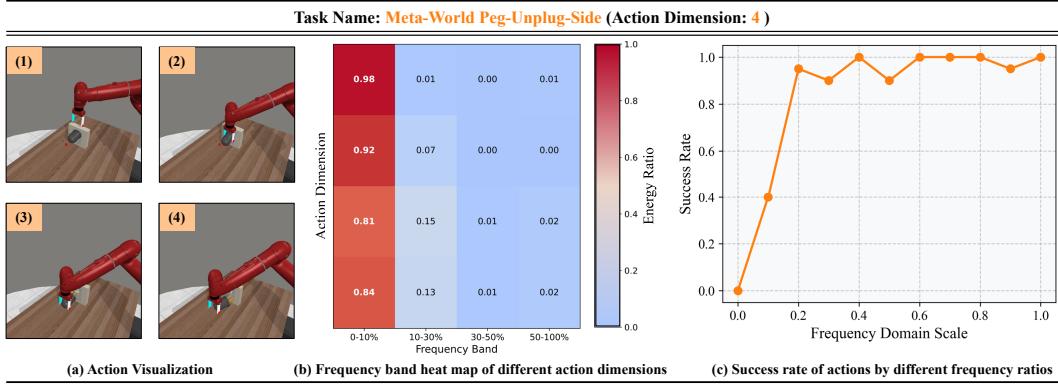


Figure 22: Frequency Domain Analysis of Meta-World Peg-Unplug-Side.

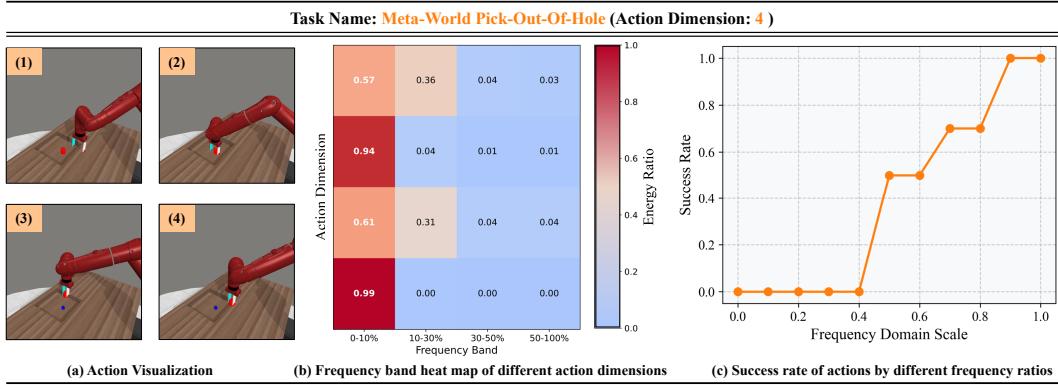


Figure 23: Frequency Domain Analysis of Meta-World Pick-Out-Of-Hole.

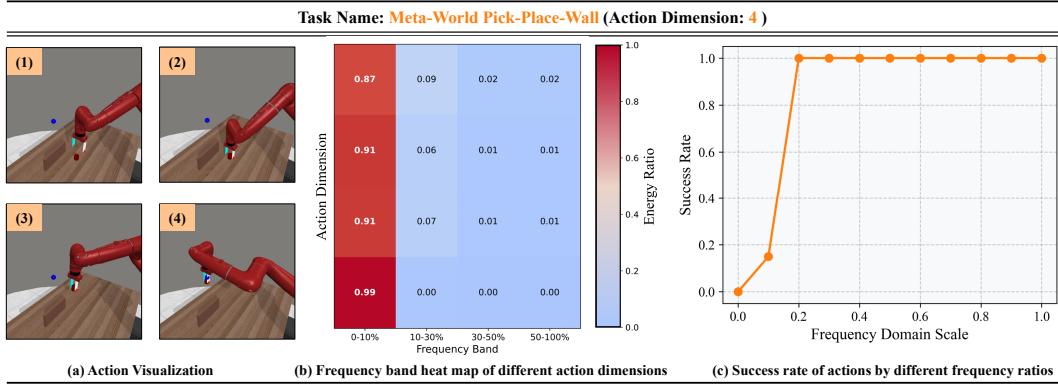


Figure 24: Frequency Domain Analysis of Meta-World Pick-Place-Wall.

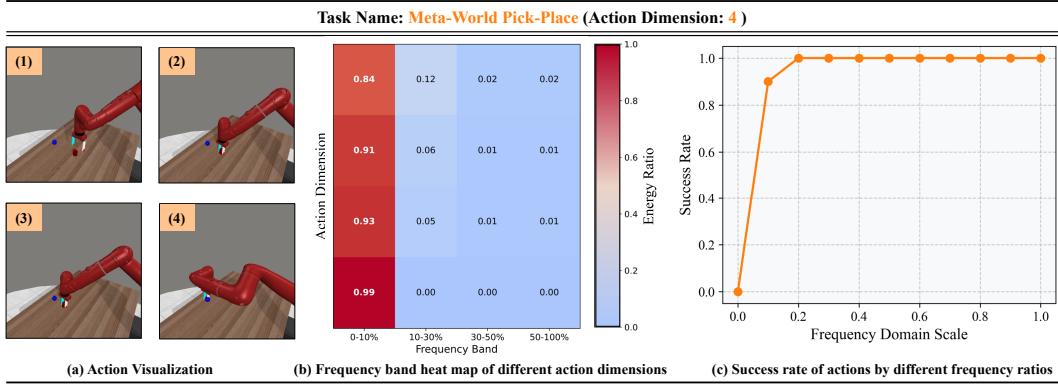


Figure 25: Frequency Domain Analysis of Meta-World Pick-Place.

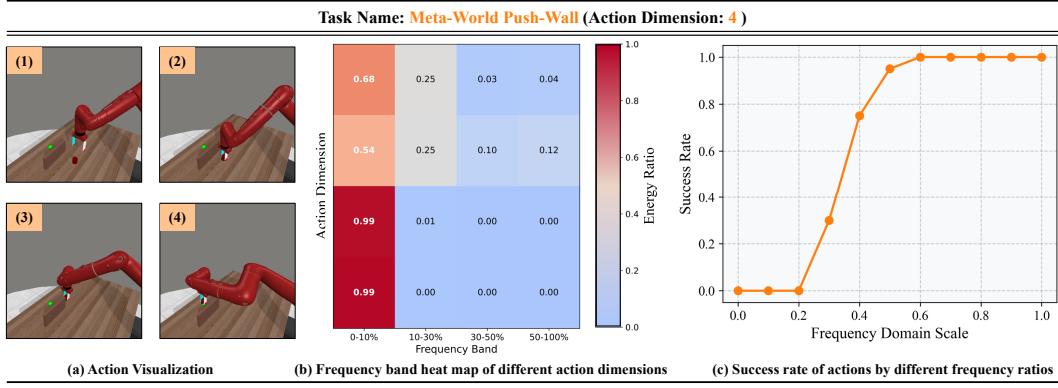


Figure 26: Frequency Domain Analysis of Meta-World Push-Wall.

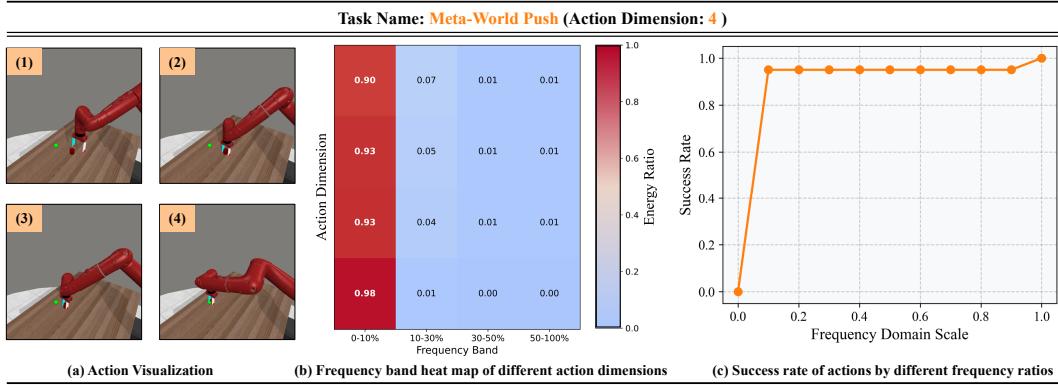


Figure 27: Frequency Domain Analysis of Meta-World Push.

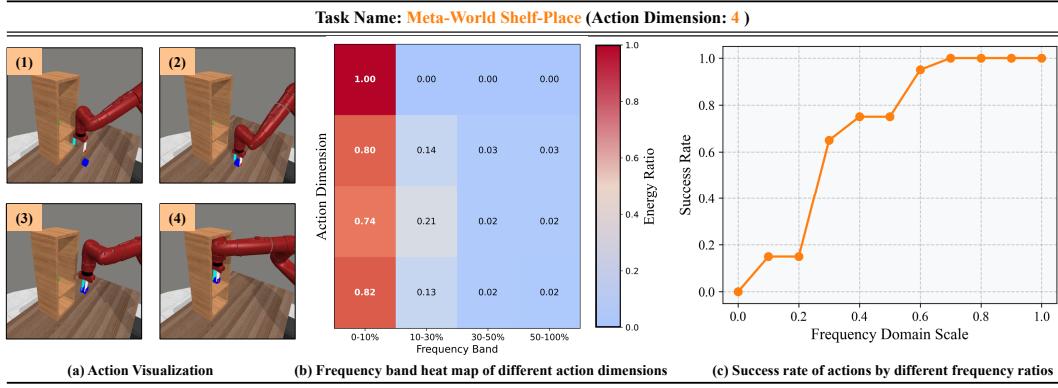


Figure 28: Frequency Domain Analysis of Meta-World Shelf-Place.

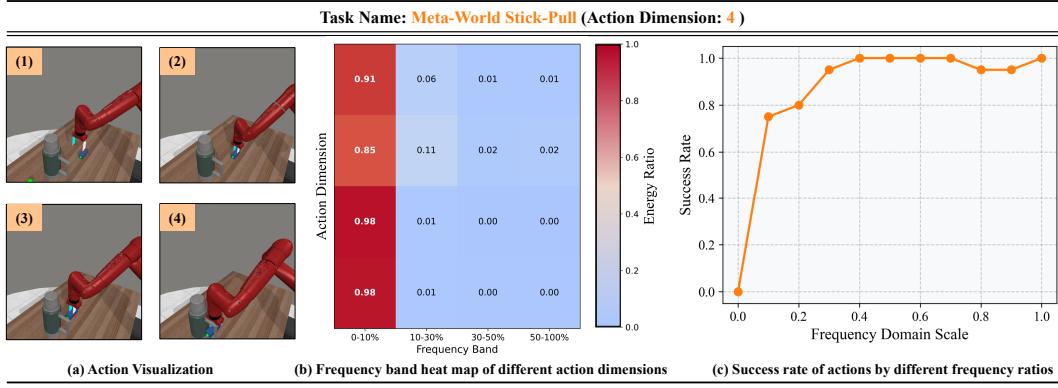


Figure 29: Frequency Domain Analysis of Meta-World Stick-Pull.

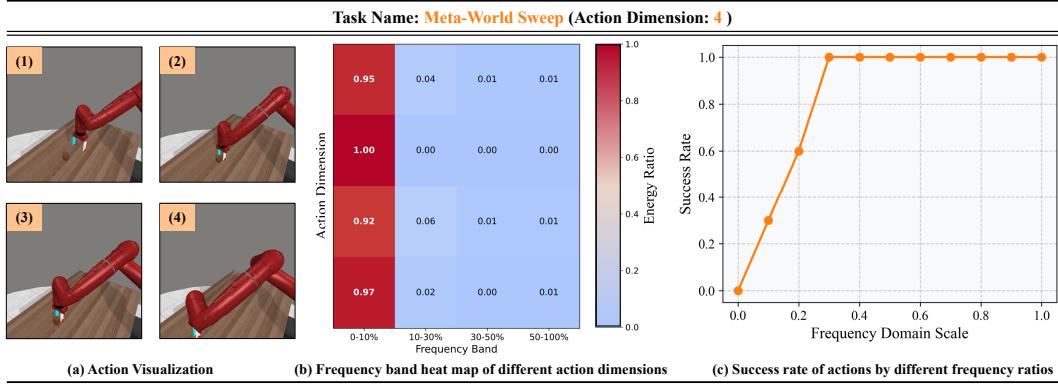


Figure 30: Frequency Domain Analysis of Meta-World Sweep.