Dita: Scaling Diffusion Transformer for Generalist Vision-Language-Action Policy

Zhi Hou[1*] Tianyi Zhang[2,1*] Yuwen Xiong[1] Haonan Duan[5] Hengjun Pu[3,1] Ronglei Tong[5]
Chengyang Zhao[4,1] Xizhou Zhu[6,1] Yu Qiao[1] Jifeng Dai[6,1] Yuntao Chen[7†]
[1] Shanghai AI Lab [2] College of Computer Science and Technology, Zhejiang University
[3] MMLab, The Chinese University of Hong Kong [4] Peking University [5] SenseTime Research
[6] Tsinghua University [7] Center for Artificial Intelligence and Robotics, HKISI, CAS
https://robodita.github.io

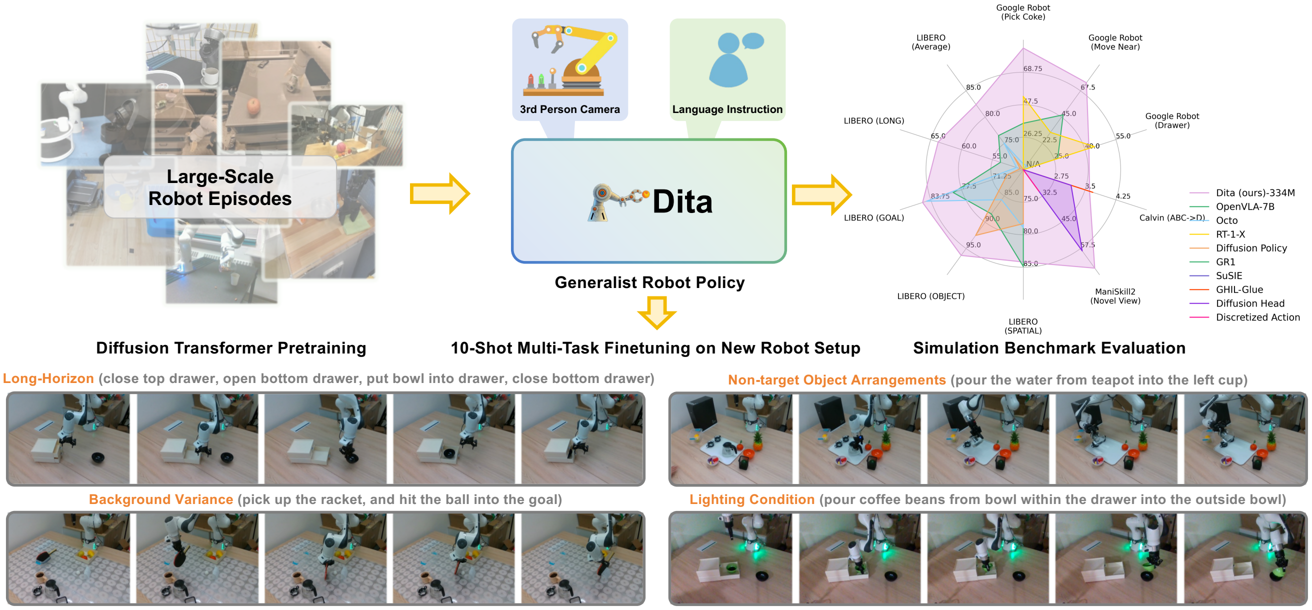


Figure 1. We introduce Dita, an open-source, simple yet effective policy for generalist robotic learning. Pretrained on large-scale cross-embodiment datasets, Dita enables 10-shot adaptation to complex, multitask, long-horizon scenarios in novel robot setups. *Particularly, Dita can complete intricate, extended-horizon tasks such as, "close the top drawer, then open the bottom drawer, subsequently place the bowl into the bottom drawer, and finally close the bottom drawer". Furthermore, Dita demonstrates remarkable robustness against complex object arrangements and even challenging lighting conditions in sophisticated 3D pick-and-rotation tasks.* In this context, the long-horizon demonstration scene serves as the training environment for all tasks. Additionally, Dita seamlessly scales to a wide range of popular simulation benchmarks, achieving state-of-the-art performance across these tasks.

Abstract

While recent vision-language-action models trained on diverse robot datasets exhibit promising generalization capabilities with limited in-domain data, their reliance on compact action heads to predict discretized or continuous actions constrains adaptability to heterogeneous action spaces. We present Dita, a scalable framework that leverages Transformer architectures to directly denoise continuous action sequences through a unified multimodal diffusion process. Departing from prior methods that condition denoising on fused embeddings via shallow networks, Dita employs in-context conditioning—enabling fine-grained alignment between denoised actions and raw visual tokens from historical observations. This design explicitly models action deltas and environmental nuances.

[*] Equal Contribution
[†] Corresponding Author

By scaling the diffusion action denoiser alongside the Transformer's scalability, Dita effectively integrates cross-embodiment datasets across diverse camera perspectives, observation scenes, tasks, and action spaces. Such synergy enhances robustness against various variances and facilitates the successful execution of long-horizon tasks. Evaluations across extensive benchmarks demonstrate state-of-the-art or comparative performance in simulation. Notably, Dita achieves robust real-world adaptation to environmental variances and complex long-horizon tasks through 10-shot finetuning, using only third-person camera inputs. The architecture establishes a versatile, lightweight and open-source baseline for generalist robot policy learning.

1. Introduction

Conventional robot learning paradigms typically depend on large-scale data collected for specific robots and tasks, yet the acquisition of data for generalized tasks remains both time-intensive and costly due to the inherent limitations of real-world robot hardware. Presently, foundational models in Natural Language Processing and Computer Vision [16, 41, 50–52, 62], pretrained on extensive, diverse, and task-agnostic datasets, have demonstrated remarkable efficacy in addressing downstream tasks either via zero-shot approaches or with minimal task-specific samples. This achievement implies that a universal robotic policy, pretrained on heterogeneous robotic data and finetuned with minimal supervision, could be instrumental in realizing true generalization in the development of vision-language-action (VLA) models. Nevertheless, training such policies across expansive cross-embodiment datasets, encompassing diverse sensors, action spaces, tasks, camera views, and environments, remains an open challenge.

In pursuit of a unified robotic policy, recent studies have directly mapped visual observations and language instructions to actions using expansive VLA models for navigation [65, 66] or manipulation [8, 9, 32, 72], thereby demonstrating zero-shot or few-shot generalization in novel environments. Robot Transformers [8, 9, 54] present policy frameworks based on Transformer architectures, achieving robust generalization by training on the extensive Open X-Embodiment (OXE) Dataset [54]. Furthermore, Octo [72] adopts an autoregressive Transformer design with a diffusion action head, while OpenVLA [32] discretizes the action space and leverages a pretrained visual-language model to construct a VLA model exposed to the OXE Dataset [54]. Nonetheless, despite the promising potential of these VLA models [32, 72] to learn robot policies from vast cross-embodiment datasets [54], the intrinsic diversity of robot configurations within these datasets continues to constrain generalization.

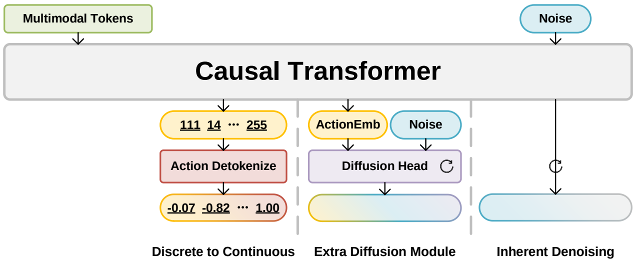


Figure 2. Illustrations of different generalist robot policy architectures. Left head: the common robot Transformer architecture with discretization actions, *e.g.*, Robot Transformer [8, 9] and OpenVLA [32]. Middle head: the Transformer architecture with diffusion action head which denoises the individual continuous action with a small network condition on each embedding from the causal Transformer, *e.g.*, Octo [72] and π_0 [5]. Right head: the proposed Dita architecture that denoises actions inherently in an in-context conditioning style.

Diffusion policies [17, 30, 61, 83] have demonstrated reliable performance in robotic policy learning under the paradigm of single-task imitation learning. Specifically, [18, 72] introduces a generalist policy that denoises actions using a network (MLP/DiT) as diffusion head conditioned on a single embedding from an auto-regressive multimodal Transformer. However, the expansive robot space within large-scale cross-embodiment datasets, encompassing diverse camera views and action spaces, presents a substantial challenge for a tiny diffusion head to effectively denoise continuous actions. Other diffusion policies [30, 61, 72] attempt to integrate historical image observations and instructions into embeddings prior to the denoising process, which might limit the denoising learning. Action anticipation typically relies more on intuitive historical observations rather than on early-fused embeddings.

In this paper, we introduce Dita, a Diffusion Transformer (DiT) Policy that capitalizes on the Transformer architecture, as demonstrated in prior work [8, 9, 32, 54, 72], thereby ensuring scalability across extensive cross-embodiment datasets. The architecture integrates an in-context conditioning mechanism with causal transformer that intrinsically denoises action sequences, thereby enabling direct conditioning of action denoising on image tokens and empowering the model to discern subtle nuances, such as action deltas, within historical visual observations. Our objective is to provide a clean, lightweight (334M parameters), and open-source baseline model for generalist robot policy learning. The model is simple yet effective, achieving state-of-the-art or competitive results on extensive simulation benchmarks, and successfully generalizing to long-horizon tasks in novel environmental configurations—characterized by variations in background, non-target object arrangements, and lighting conditions through finetuning with a mere 10-shot set of real-world samples. Remarkably, this promising performance is achieved exclusively with a single third-person camera input, while the model's inherent flexibility affords researchers the freedom to integrate additional input modalities (e.g., wrist-camera images, target image predictions, robot state, tactile feedback, etc.) for further investigation.

2. Related Work

Diffusion Policy Denoising diffusion models [11, 19, 25, 56, 63] have demonstrated remarkable proficiency in both image generation and multi-modal robotic action modeling [12, 14, 17, 30, 39, 42, 61, 74, 75, 77, 83]. Nevertheless, existing diffusion-based manipulation policies predominantly rely on U-Net architectures or shallow cross-attention networks designed for single tasks, limiting their scalability to multi-modal applications. Recent generalist models [72, 76] employ VLM embeddings combined with compact MLP diffusers, while others, like RDT [42] and [18], utilize cross-attention Transformers or DiT decoders for bimanual manipulation. In contrast, we propose a scalable DiT with in-context conditioning, which directly processes historical observations through a causal Transformer architecture, thereby providing enhanced expressiveness and generalization capabilities for multi-modal action generation.

Generalist Robot Policies Language-conditioned policies [15, 23, 44, 49, 60, 73, 84] have gained prominence for their adaptability in real-world applications, enabling robots to interpret and execute natural language instructions. Recent advancements in generalist robot policies leverage foundation multi-modal models across both navigation [6, 27, 65, 66, 71, 82] and manipulation [3, 7–9, 20, 21, 29, 32, 45, 46, 54, 55, 60, 64, 67, 68, 72, 79, 80], with scalable VLA models emerging as a dominant framework [1, 8, 9, 32, 54, 58, 72]. Some approaches incorporate large-scale video backbones trained on internet-scale data [13, 28, 34, 78] to improve temporal visual reasoning. While these methods enhance visual representation learning, our focus is on action generation, where diffusion-based models provide a more expressive alternative. Another crucial factor in generalist policies is the choice of pretrained VLM models for action generation. Unlike recent works [5, 38] that employ PaliGemma [2] to enhance vision-language understanding, we adopt a LLaMA-style causal Transformer for policy learning. This approach is both simple and highly scalable, demonstrating effectiveness across a wide range of benchmarks. Furthermore, by aligning robot actions with language instructions and visual observations in an in-context conditional manner, our method significantly enhances generalization across diverse robotic embodiments.

3. Method

In this section, we describe Dita in detail. We begin by detailing the architecture of the model, which is a scalable DiT with in-context conditioning. We then define the training objective for generating multi-modal actions. Finally, we present the data and implementation specifics for the pretraining of our model.

3.1. Architecture

Multi-modal Input Tokenization. Dita only takes language instructions and third-person camera images as input. The language instructions are tokenized using a frozen CLIP [59] model, while the image observations are first processed by DINOv2 [53] to extract image patch features. Notably, DINOv2 is trained on web data, which differs from robot-specific data. Thus, we jointly optimize the DINOv2 parameters alongside Dita in an end-to-end fashion. To mitigate computational costs, we incorporate a Q-Former [33] with FiLM [57] conditioning to select image features from the DINOv2 patch features based on the instruction context.

Action Preprocess. We represent the end-effector action as a 7D vector, comprising 3 dimensions for the translation vector, 3 dimensions for the rotation vector, and 1 dimension for the gripper position. To align the dimensionality with the image and language tokens, we pad the continuous action vector with zeros to form the action representation. Noise is only introduced into the 7D action vector during the denoising diffusion optimization process.

Model Design. Our core design is the DiT structure [56], which denoises action token chunks rather than individual action tokens. This is achieved by conditioning directly on image observations and instruction tokens through an in-context conditioning approach using a causal Transformer. Specifically, we concatenate language tokens, image features, and timestamp embeddings at the beginning of the sequence, treating the noisy action in conjunction with the instruction tokens, as illustrated in Figure 3. This design preserves the scalability of Transformer networks and enables denoising to be conditioned directly on image patches, thereby allowing the model to capture nuanced changes in action over historical observations. The model is supervised by the noise introduced into the continuous actions. In other words, we directly apply the diffusion objective in the action chunk space with a large Transformer model, in contrast to the diffusion action head approach [18, 61, 72]. Notably, our proposed Dita presents a versatile and scalable design, adaptable to diverse datasets for both pretraining and finetuning, while achieving promising performance. Furthermore, additional observation tokens and input can be seamlessly integrated into the Transformer architecture. Further details are provided in Appendix A.

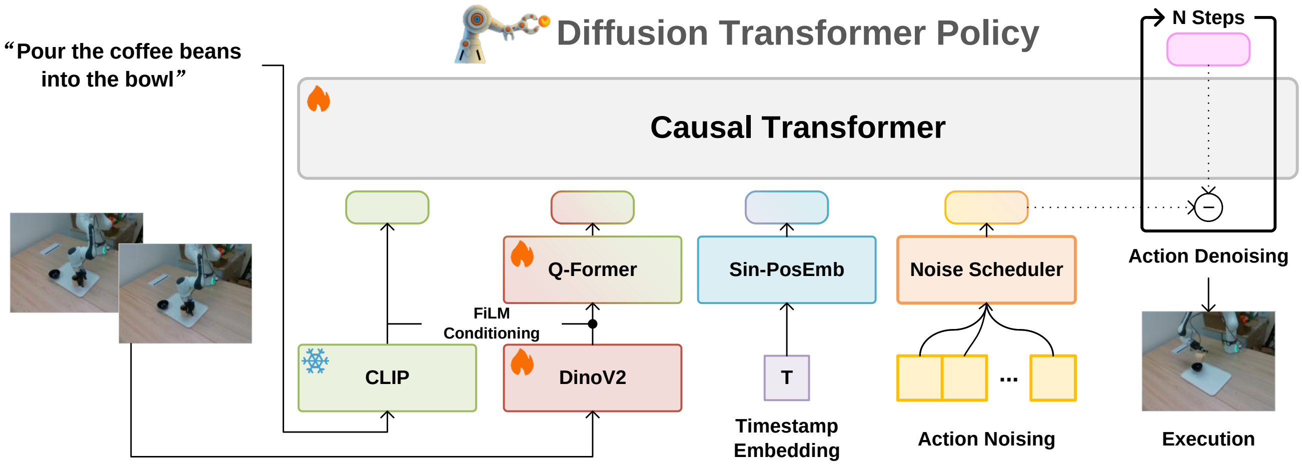


Figure 3. Our model employs a Transformer-based diffusion architecture, integrating a pretrained CLIP network to extract language instruction tokens. The DinoV2 [53] model encodes image observations, followed by a Q-Former that queries features for each image. The instruction tokens, image features, timestep embeddings, and noised action are concatenated to construct a token sequence, which is then fed into the network to denoise the raw actions.

3.2. Training Objective

The denoising network $\mathcal{E}_\theta(c_{lang}, c_{obs}, t, \boldsymbol{x}^t)$ is constructed upon a causal Transformer, where c_{obs} represents the image observation, c_{lang} denotes the language instruction, and $t \in 1, 2, \ldots, T_{train}$ is the timestamp index within the total denoising steps T_{train}. During training, a Gaussian noise vector $\boldsymbol{x}^t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is sampled at each timestamp t and added to the action $\boldsymbol{a}$ to form the noised action token $\hat{\boldsymbol{a}}$. The network $\mathcal{E}_\theta$ is trained to predict the noise vector $\hat{\boldsymbol{x}}$, with randomly sampled t . The optimization objective of Dita is to minimize the mean squared error (MSE) loss between $\boldsymbol{x}^t$ and $\hat{\boldsymbol{x}}^t$.

The inference procedure is delineated as follows, α, γ, and σ constitute the noise scheduler [25]. The denoising process is iterated over N_{eval} steps to yield a reliable action.

$$\boldsymbol{x}^{t-1} = \alpha(\boldsymbol{x}^t - \gamma\mathcal{E}_\theta(c_{lang}, c_{obs}, t, \boldsymbol{x}^t) + \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})). \quad (1)$$

3.3. Pretraining Data

To evaluate the proposed Dita policy, we select the OXE datasets [32, 54] for model pretraining. We primarily adhere to the method detailed in [32, 72] for dataset selection and weight assignment. Actions are normalized and filtered similar to [54].

3.4. Pretraining Details

We employ the DDPM diffusion objective [25] with $T_{train} = 1000$ timestamps for pretraining, while adopting DDIM [69] with $T_{eval} = 20$ timestamps during zero-shot evaluation to accelerate inference. Based on preliminary experiments reported in ManiSkill2 [22], we utilize 2-frame image observations to predict 16 action chunks. The network is optimized by AdamW [43] for 100,000 steps, with learning rates of $1e{-}4$ for both the causal Transformer and Q-Former, and $1e{-}5$ for DINOv2. Training is conducted with a batch size of 8192 across 32 NVIDIA A100 GPUs, allocating 256 samples per GPU. Additional pretraining configurations are detailed in Appendix A.

4. Simulation Experiments

We strive to develop a robust foundational VLA model that is both scalable across diverse simulation benchmarks and adaptive to new complex tasks in unseen robot environments with as few as 10 or even fewer samples. To assess the capabilities of the pretrained model, we conduct evaluations across four simulation benchmarks in this section: 1) SimplerEnv [37] (Google Robot) demonstrates the model's zero-shot adaptation to simulation environments; 2) LIBERO [40] assesses finetuning adaptability with a single-camera setup; 3) Calvin [47] evaluates long-horizon task performance in novel environments; and 4) ManiSkill2 [22] is re-rendered to illustrate generalization across unseen camera views. Across all four benchmarks, Dita pretrained on OXE datasets $\mathcal{E}_{\theta\sim OXE}$ is evaluated in a zero-shot manner on SimplerEnv, while it is finetuned on the remaining three benchmarks using their respective datasets.

4.1. Baselines

Diffusion Action Head The diffusion head for the action generation $\mathcal{E}^{Diff}_{\theta\sim s}$ [72] is also implemented. Specifically, we employ a three-layer MLP network as the denoising module, conditioned on each action token embedding outputed by the same causal Transformer architecture (as Mid-

Table 1. Success rate comparison with RT-1-X [8], Octo-Base [72] and OpenVLA-7B [32] on SimplerEnv (both match and variant results of Google Robot [8]).

Method	coke_can		move_near		drawer	
	match	variant	match	variant	match	variant
RT-1-X [8]	56.7%	49.0%	31.7%	32.3%	**59.7%**	29.4%
Octo-Base [72]	17.0%	0.6%	4.2%	3.1%	22.7%	1.1%
OpenVLA-7B [32]	16.3%	54.5%	46.2%	47.7%	35.6%	17.7%
Dita (Ours)	**83.7%**	**85.5%**	**76.0%**	**73.0%**	46.3%	**37.5%**

dle head illustrated in Figure 2). This approach introduces additional parameters (the extra MLP) compared to Dita.

Octo & OpenVLA We also reproduce these two open-source VLA models using their released checkpoints, as they employ the same multimodal inputs (language instruction and third-person camera image) as our approach.

4.2. SimplerEnv

SimplerEnv [37] is a Real-to-Sim platform designed to evaluate policies learned from real robot data within a simulation environment. In this section, we compare our approach with leading generalist policies, including RT-1-X [8, 54], Octo [72], and OpenVLA [32], under both match and variant scenarios. For a fair comparison, we adhere to the evaluation protocol of SimplerEnv [37], which includes tasks “pick up coke can”, “move an object near to others”, “open drawer”, “close drawer”.

Table 1 demonstrates that Dita achieves strong generalization performance under zero-shot evaluation across various types of variations, including background, texture, objects, spatial positions, and more. Leveraging the in-context conditioning design, Dita exhibits enhanced robustness, relying solely on third-person view images to detect subtle nuances and generate more reliable actions. The qualitative results are listed in Appendix B.1.

4.3. LIBERO

LIBERO [40] is a comprehensive benchmark for knowledge transfer in multitask and lifelong robot learning. It consists of four sub-datasets: LIBERO-SPATIAL, LIBERO-OBJECT, LIBERO-GOAL, and LIBERO-100. Notably, LIBERO-100 is further divided into LIBERO-90 and LIBERO-LONG, with the latter featuring 10 long-horizon tasks that encompass diverse object interactions and versatile motor skills. We employ the modified version of LIBERO from OpenVLA [32] as the data source for fine-tuning and evaluation.

Comparisons. Table 2 demonstrates that Dita outperforms baseline methods on most LIBERO sub-datasets, achieving an overall increase in average success rate by nearly 6%.

Table 2. Comparison with Diffusion Policy (denoted as DP*, training from scratch) [17], Octo [72], and OpenVLA [32] on LIBERO [40]. Except for Dita results, all other results are sourced from [32].

Method	SPATIAL	OBJECT	GOAL	LONG	Averge
DP*[17]	78.3%	92.5%	68.3%	50.5%	72.4%
Octo [72]	78.9%	85.7%	84.6%	51.1%	75.1%
OpenVLA [32]	**84.9%**	88.4%	79.2%	53.7%	76.5%
Dita (Ours)	84.2%	**96.3%**	**85.4%**	**63.8%**	**82.4%**

Notably, Dita exhibits a 10% improvement on LIBERO-LONG, highlighting its strong potential for tackling long-horizon tasks.

4.4. CALVIN

CALVIN [47] is an open-source simulated benchmark designed for learning long-horizon, language-conditioned tasks. It consists of four distinct scenes (A, B, C, and D) and introduces the ABC→D evaluation protocol, where models are trained on environments A, B, and C and evaluated on environment D. The benchmark aims to solve up to 1,000 unique task sequences, each comprising five distinct subtasks. The primary evaluation metric is the success sequence length, which measures the ability to complete five consecutive subtasks within a sequence. To assess the long-horizon generalization of Dita, we adopt the ABC→D setting while only utilizing static RGB images as perception inputs. Additionally, we also implement a diffusion policy baseline $\mathcal{E}_{\theta\sim s}^{Diff}$ by introducing a three-layer MLP diffusion head, further demonstrating the effectiveness of Dita’s in-context conditioning mechanism.

Comparisons. Table 3 presents a comparative analysis of prior approaches and the proposed Dita on CALVIN. Without whistles and bells, the proposed Dita achieves comparable performance among methods relying solely on a single RGB camera for observation. Noticeably, only 1% of the trajectories were labeled with text in Calvin [47]. The rest are unstructured play data collected by untrained users, with no information for downstream tasks. *Dita does not utilize the play data which provides external trajectory data compared to the labeled data, while GR-MG uses it for training the policy.* Remarkably, GHIL-Glue [4, 24], which builds upon SuSIE [4] with further finetuned generative models [10, 81], results in significantly larger models. Furthermore, Dita surpasses its non-pretrained variant by a margin of 1.23, underscoring its superior transferability. In contrast, employing diffusion head underperforms Dita by 0.45 points with similar pretrained weights, highlighting the efficacy of Dita’s in-context conditioning mechanism. The results illustrate that Dita excels at discerning subtle visual

Table 3. The comparisons with state-of-the-art approaches on Calvin (ABC→D) with the metrics of success rate and average success length. The abbreviations denote different input modalities: S-RGB for Static RGB, G-RGB for Gripper RGB, S-RGBD for Static RGB-D, G-RGBD for Gripper RGB-D, P for proprioceptive arm position, and Cam for camera parameters.

Method	**Input**	**No. Instructions in a Row (1000 chains)**					
		1	2	3	4	5	Avg.Len.
RoboFlamingo [36]	S-RGB,G-RGB	82.4%	61.9%	46.6%	33.1%	23.5%	2.47
GR-1 [78]	S-RGB,G-RGB,P	85.4%	71.2%	59.6%	49.7%	40.1%	3.06
3D Diffuser [30]	S-RGBD,G-RGBD,P,Cam	92.2%	78.7%	63.9%	51.2%	41.2%	3.27
GR-MG [34]	S-RGBD,G-RGBD,P	**96.8%**	**89.3%**	**81.5%**	**72.7%**	**64.4 %**	**4.04**
SuSIE [4]	S-RGB	87.0%	69.0%	49.0%	38.0%	26.0%	2.69
GHIL-Glue [4, 24]	S-RGB	**95.2%**	**88.5%**	**73.2%**	**62.5%**	**49.8%**	**3.69**
$\mathcal{E}^{Diff}_{\theta\sim s}$ w/o Pretrain	S-RGB	75.5%	44.8%	25.0%	15.0%	7.5%	1.68
$\mathcal{E}^{Diff}_{\theta\sim s}$	S-RGB	94.3%	77.5%	62.0%	48.3%	34.0%	3.16
Ours w/o Pretrain	S-RGB	89.5%	63.3%	39.8%	27.3%	18.5%	2.38
Ours	S-RGB	**94.5%**	**82.5%**	**72.8%**	**61.3%**	**50.0%**	**3.61**

nuances in long-horizon tasks and generalizes proficiently across diverse environments, effectively transferring knowledge from extensive, real-world pretraining datasets to the CALVIN benchmark.

4.5. ManiSkill2

ManiSkill2 [22], the next generation of the SAPIEN ManiSkill benchmark [48], serves as a widely recognized platform for assessing the generalized manipulation capabilities of embodied models. It encompasses 20 distinct manipulation task families and over 4M demonstration frames across various configurations. Leveraging ManiSkill2, we establish a *novel camera view generalization* benchmark to evaluate the effectivenes of Dita.

Setup. To construct the benchmark, we select 5 tasks (PickCube-v0, StackCube-v0, PickSingleYCB-v0, PickClutterYCB-v0, PickSingleEGAD-v0) from ManiSkill2 and create a camera pool comprising 300K random cameras. 20 cameras are sampled each time to render each trajectory, resulting in over 40K trajectories, which are utilized to train Dita from scratch. The generated dataset is divided into training and validation sets with a 19:1 ratio, ensuring that each category in task family, and trajectories rendered from different camera views are assigned to both, thereby preventing data leakage. During training, the number of data samples is balanced across task families by duplicating trajectories for task families with fewer samples. To construct a closed-loop evaluation dataset, we randomly sample 100 trajectories from the validation set for each task family. This evaluation dataset with 500 trajectories is used to assess the success rate for each task family and demonstrate the camera-view generalization capabilities of Dita. We also implement RT-1 [8] style baseline model $\mathcal{E}^{Disc}_{\theta\sim s}$ with an architecture similar to ours for comparison. Unlikely, we discretize each action dimension into 256 bins [8] and utilize a Transformer network to predict the corresponding bin indices.

Table 4. Comparison of our model with two baseline methods (discretization and diffusion head) on ManiSkill2 success rate. The abbreviations denote the task names: S-YCB for PickSingleYCB, C-YCB for PickClutterYCB, EGAD for PickSingleEGAD.

Method	**Avg.**	**PickC**	**StackC**	**S-YCB**	**C-YCB**	**EGAD**
$\mathcal{E}^{Disc}_{\theta\sim s}$	30.2%	41.0%	33.0%	22.0%	1.0%	54.0%
$\mathcal{E}^{Diff}_{\theta\sim s}$	58.6%	**86.0%**	76.0%	37.0%	24.0%	70.0%
Dita (ours)	**65.8%**	79.0%	**80.0%**	**62.0%**	**36.0%**	**72.0%**

Comparisons. Table 4 compares the proposed method with the discretization action head and the diffusion action head. The experiments demonstrate that Dita outperforms the $\mathcal{E}^{Disc}_{\theta\sim s}$ in large-scale, novel camera view scenarios. Additionally, Dita shows superior performance on more complex tasks and outperforms $\mathcal{E}^{Diff}_{\theta\sim s}$ by 20% in the PickSinglYCB task and by 12% in the PickClutterYCB task. The results highlight that Dita offers better scalability on large, diverse datasets, while also achieving enhanced camera view generalization.

4.6. Ablation Study

In this section, we conduct an ablation study on key factors in the model architecture design, including observation length, trajectory length, and denoising steps.

Observation length. The length of historical observation images significantly impacts performance. As shown in Table 5, success rate drops sharply when the observation length is increased to 3. This could be due to the increased difficulty in model convergence, as the number of corresponding image tokens also rises. Additionally, we observe that using 2-frame observations enhances performance, particularly when the prediction horizon is extended. When

Table 5. Ablation on ManiSkill2 about the observation length (# obs) and the trajectory length (# traj).

# obs	# traj	All	PickC	StackC	S-YCB	C-YCB	EGAD
2	2	40.8%	68.0%	54.0%	33.0%	9.0%	40.0%
2	4	51.6%	81.0%	69.0%	44.0%	11.0%	53.0%
2	8	62.4%	**89.0** %	78.0%	54.0%	25.0%	66.0%
2	16	65.6%	83.0%	**80.0** %	**70.0** %	25.0%	70.0%
2	32	**65.8** %	79.0%	**80.0** %	62.0%	**36.0** %	**72.0**%
1	32	61.6%	78.0%	76.0%	64.0%	24.0%	66.0%
1	1	51.0%	79.0%	66.0%	42.0%	19.0%	49.0%
3	3	35.4%	54.0%	49.0%	27.0%	5.0%	42.0%

the trajectory length is 32, Dita with 2-frame observations achieves superior performance. We argue that 2-frame observations strike an optimal balance, providing sufficient visual distinction between objects in the workspace and the robot states.

Trajectory length. Trajectory length is the sum length of observation and action prediction chunks, which is also a critical factor influencing performance. Table 5 shows that performance improves as trajectory length increases. Notably, the performance of more complex tasks, such as PickClutterYCB, increases substantially with longer trajectory lengths, while simpler tasks, like PickCube, maintain high performance once the trajectory length exceeds 4. Long trajectory length significantly boosts performance, as this optimization allows the model to better anticipate the target object's position and gain awareness of more future states.

Denoising steps. Typically, diffusion models require multiple denoising steps in image generation [62]. For diffusion-based policies in robot learning, the number of denoising steps during inference can impact control frequency. Surprisingly, we find that DDIM significantly reduces the denoising steps to 10 without compromising performance on the ”Pick Coke” task as described in Table 6. With only 2 denoising steps, the model still achieves a 70.4% success rate. We attribute model performs best with 10 steps to the reduction of overfitting when fewer denoising steps are used. Unlike image generation, the action dimension in robot learning is much smaller, allowing effective denoising with fewer steps without requiring advanced techniques [70]. These results suggest that the in-context conditioning used by Dita does not hinder inference speed.

5. Real-Robot Experiments

Few-shot real-world robot adaptation is a critical metric for evaluating the effectiveness of a generalist policy in practical applications. For real-robot experiments, We employ 10-shot finetuning to assess the model's adaptability in complex, long-horizon, multi-modal tasks within unseen robot environments.

Table 6. The impact of the number of denoising steps (# Steps) of DDIM on Google Robot Simulation during inference, trained with 1000 DDPM denoising steps.

# Steps	100	50	20	10	5	2
Pick Coke (variant)	76.4	79.1	**85.5**	85.3	82.7	70.4
Pick Coke (match)	79.7	83.3	**83.7**	82.0	82.	73.3
Move Near (variant)	52.1	66.0	**73.0**	69.5	63.5	51.6
Move Near (match)	49.1	72.0	**76.0**	74.0	72.0	65.0

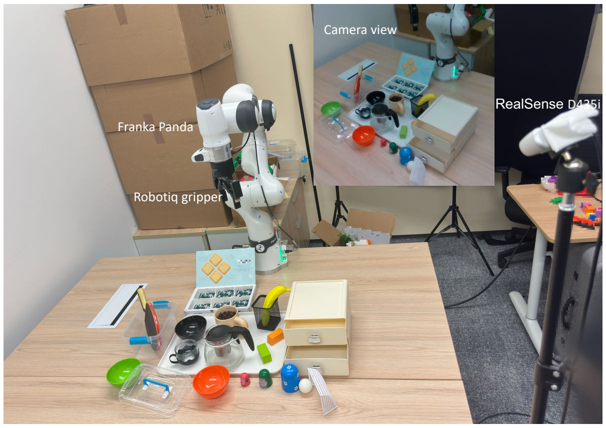


Figure 4. The experimental platform consists of Franka Emika Panda robot arm, Robotiq 2F-85 gripper and RealSense D435i positioned in third-person view.

5.1. Real-Robot Task Finetuning

Setup. To deploy the Dita model, as shown in Figure 4, the robot setup consists of a 7-DoF tabletop Franka Emika Panda robot arm and a Robotiq 2F-85 gripper. A RealSense D435i RGB-D camera, positioned approximately 1.5m away from the robot in a third-person view, captures RGB scenes with cluttered background at each inference timestamp. Robot control is managed from a desktop computer running ROS, communicating with the model-deploy server with 1 NVIDIA A100 GPU. The system is operating under control frequency of 3Hz. Given the data domain gap between our robot platform and the pretrain dataset, we primarily evaluate Dita on 10-shot generalization for the following challenging tasks relevant to current VLA approaches:

- **Pick & Place**. Two pick-and-place tasks with target object banana and kiwifruit are evaluated. 10 samples are collected for each task, with position variances introduced during evaluation to assess generalization performance.
- **Pour**. We design two pouring tasks to evaluate the complex rotation finetuning: “pour the coffee beans into the bowl”, and “pour the water from the teapot into the cup”.

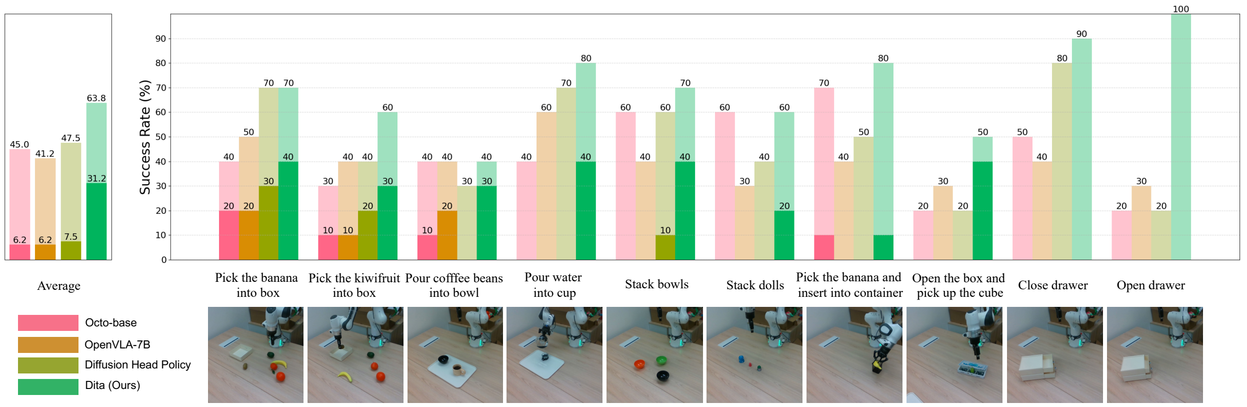


Figure 5. Quantitative results in real-robot experiments. Each task is manually divided into two sequential steps, except for the last two single-step tasks. In each stacked bar, the light-colored region represents the model's success rate in the first stage, while the dark-colored region indicates the contribution of second-stage success to the overall success rate. A larger proportion of the dark-colored region signifies a stronger capability of the model in completing long-horizon tasks. Since the open/close drawer tasks are single-step, they are excluded from the calculation of the average success rate.

- **Stack**. We design two stacking tasks for long-horizon pick-and-place: stacking three bowls and stacking three Russian dolls.
- **Pick & Rotation**. These skills combine two tasks: "pick the banana and insert it into the small pen container" and "open the flip-top door box and then pick up the small cube".
- **Pull & Push**. We design the task "open/close the drawer" to evaluate pull and push abilities of our models.
- **Long-Horizon Tasks.** We further devise several long-horizon tasks (more than 3 steps), including "Pick up the bowl within the drawer and pour the coffee beans into the outside bowl", "pick up the racket and hit the ball into the goal", "open the top drawer, then pick the cube into the drawer, and finally close the drawer", "close the top drawer, then open the bottom drawer and put the bowl into the drawer, and finally close the drawer" and "open the box and move the green cube into the box then close the box", to demonstrate the long-horizon manipulation ability of Dita (detailed in demo videos in Supplementary Materials).

The diffusion policy [17] has demonstrated strong capabilities in learning to mimic single tasks. Therefore, in addition to Octo and OpenVLA, we design a multimodal diffusion policy baseline based on a causal Transformer for comparison, which incorporates a similar diffusion head as described in Section 4.1.

Optimization. We finetune the Dita on the aforementioned multiple manipulation tasks, with data collected on the same platform, using LoRA [26] for fair comparison and AdamW for 20,000 steps with image augmentations. The number of timestamps is set to 100 for DDPM [25], and the batch size of 512.

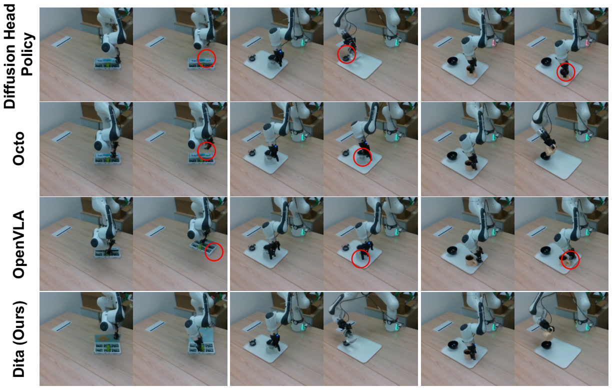


Figure 6. Qualitative comparison in real-robot experiments. Failures are highlighted with red circles. For a direct comparison, we initialize the layout consistently across all methods.

10-shot Finetuning. We directly use the model pretrained on OXE datasets to evaluate 10-shot finetuning generalization for real-robot experiments. To ensure consistency with OpenVLA [32], we finetune the network with one observation and one-step prediction. We compare the proposed method with OpenVLA [32] and Octo-base [72] models. Overall, Dita achieves a 63.8% success rate on two-step tasks, with the second stage contributing nearly half, as shown in Figure 5. Dita consistently outperforms both Octo and OpenVLA, demonstrating superior performance on all complex tasks. For long-horizon tasks, OpenVLA effectively completes the first task but fails to handle the long-horizon task, such as completely misunderstanding the insert operation. In contrast, Octo performs better with rotation tasks and approaches the second step of the task more effectively. Supplementary materials provide extensive qualitative comparisons.

Variance Robustness. To evaluate the robustness of Dita, we further validate its performance under different variances, including background changes, non-target object arrangements and lighting conditions. As illustrated in Figure 1, it is surprising to find that Dita not only excels in completing complex long-horizon tasks but also demonstrates resilience to a wide range of variations.

5.2. Qualitative Comparison

Figure 6 presents a comparison between Dita, diffusion head baseline, Octo [72], and OpenVLA [32] as evaluated in real-robot experiments under 10-shot finetuning setting. According to the visualized comparison, those baseline methods usually fail to grasp the correct position under the 10-shot setting, *e.g*., "fail to insert the gap of the box", "grasp when the gripper is not in the correct grasping position (the hand of cup or teapot)". Meanwhile, the baseline methods, *e.g*., the second raw (pouring water) in Figure 6, sometimes misunderstand the lifting action and get stuck after grasp. In contrast, Dita is able to effectively complete all the complex tasks with extreme 3D rotations.

6. Conclusion

In this paper, we present Dita, an architecture for generalist robot learning that leverages a Transformer-based diffusion model to denoise continuous action sequences through an in-context conditioning mechanism. By harnessing the scalability of Transformers, Dita effectively models diverse robot behaviors across extensive cross-embodiment datasets, enabling robust generalization across multiple simulation benchmarks within a unified framework. Additionally, Dita demonstrates strong few-shot adaptation capabilities, successfully transferring to novel real-world robot setups and long-horizon tasks with minimal in-domain samples. Notably, the model is clean, lightweight, and open-source, and its promising performance—achieved exclusively with a single third-person camera input—underscores its potential as a scalable and flexible solution for learning generalist robot policies.

References

[1] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024. 3

[2] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 3

[3] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4788–4795. IEEE, 2024. 3, 5

[4] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023. 5, 6

[5] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 2, 3

[6] Denis Blessing, Onur Celik, Xiaogang Jia, Moritz Reuss, Maximilian Li, Rudolf Lioutikov, and Gerhard Neumann. Information maximizing curriculum: A curriculum-based approach for learning versatile skills. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[7] Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X Lee, Maria Bauza, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, et al. Robocat: A self-improving foundation agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*, 2023. 3

[8] 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. 2, 3, 5, 6, 4

[9] 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. 2, 3, 4

[10] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023. 5

[11] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 3

[12] 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. 3

[13] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 3

[14] Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *arXiv preprint arXiv:2407.01392*, 2024. 3, 5

[15] Lili Chen, Shikhar Bahl, and Deepak Pathak. Playfusion: Skill acquisition via diffusion from language-annotated play. In *Conference on Robot Learning*, pages 2012–2029. PMLR, 2023. 3

[16] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024. 2

[17] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. 2, 3, 5, 8, 4

[18] Sudeep Dasari, Oier Mees, Sebastian Zhao, Mohan Kumar Srirama, and Sergey Levine. The ingredients for robotic diffusion transformers. *arXiv preprint arXiv:2410.10088*, 2024. 2, 3

[19] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 3

[20] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 3

[21] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A robotic dataset for learning diverse skills in one-shot. In *RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023. 3

[22] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023. 4, 6, 2

[23] Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, pages 3766–3777. PMLR, 2023. 3

[24] Kyle B Hatch, Ashwin Balakrishna, Oier Mees, Suraj Nair, Seohong Park, Blake Wulfe, Masha Itkina, Benjamin Eysenbach, Sergey Levine, Thomas Kollar, et al. Ghil-glue: Hierarchical control with filtered subgoal images. *arXiv preprint arXiv:2410.20018*, 2024. 5, 6

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3, 4, 8

[26] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 8

[27] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023. 3

[28] Siyuan Huang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Zhengkai Jiang, Yue Hu, Peng Gao, Hongsheng Li, Maoqing Yao, and Guanghui Ren. Enerverse: Envisioning embodied future space for robotics manipulation. *arXiv preprint arXiv:2501.01895*, 2025. 3

[29] Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023. 3

[30] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024. 2, 3, 6

[31] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 1

[32] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 2, 3, 4, 5, 8, 9

[33] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 3

[34] Peiyan Li, Hongtao Wu, Yan Huang, Chilam Cheang, Liang Wang, and Tao Kong. Gr-mg: Leveraging partially-annotated data via multi-modal goal-conditioned policy. *IEEE Robotics and Automation Letters*, 2025. 3, 6

[35] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024. 5

[36] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023. 6

[37] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024. 4, 5

[38] Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024. 3

[39] Zhixuan Liang, Yao Mu, Hengbo Ma, Masayoshi Tomizuka, Mingyu Ding, and Ping Luo. Skilldiffuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16467–16476, 2024. 3

[40] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36, 2024. 4, 5, 2

[41] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 2

[42] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024. 3

[43] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[44] Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*, 2020. 3

[45] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. PMLR, 2020. 3

[46] Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters*, 7(4):11205–11212, 2022. 3

[47] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3): 7327–7334, 2022. 4, 5

[48] Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021. 6, 2

[49] Vivek Myers, Andre Wang He, Kuan Fang, Homer Rich Walke, Philippe Hansen-Estruch, Ching-An Cheng, Mihai Jalobeanu, Andrey Kolobov, Anca Dragan, and Sergey Levine. Goal representations for instruction following: A semi-supervised language interface to control. In *Conference on Robot Learning*, pages 3894–3908. PMLR, 2023. 3

[50] OpenAI. Dall·e: Creating images from text, 2021. 2

[51] OpenAI. Chatgpt, 2022.

[52] OpenAI. Gpt-4: Multimodal capabilities, 2023. 2

[53] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 3, 4

[54] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 2, 3, 4, 5

[55] Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. *arXiv preprint arXiv:2301.10677*, 2023. 3

[56] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 3

[57] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 3

[58] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025. 3

[59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[60] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023. 3

[61] Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024. 2, 3, 4

[62] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Tim Schmidt. Stable diffusion: High-resolution image synthesis with latent diffusion models, 2021. 2, 7

[63] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[64] Paul Maria Scheikl, Nicolas Schreiber, Christoph Haas, Niklas Freymuth, Gerhard Neumann, Rudolf Lioutikov, and Franziska Mathis-Ullrich. Movement primitive diffusion: Learning gentle robotic manipulation of deformable objects. *IEEE Robotics and Automation Letters*, 2024. 3

[65] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023. 2, 3

[66] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. *arXiv preprint arXiv:2306.14846*, 2023. 2, 3

[67] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. Mutex: Learning unified policies from multimodal task specifications. *arXiv preprint arXiv:2309.14320*, 2023. 3

[68] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023. 3

[69] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 4

[70] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023. 7

[71] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 63–70. IEEE, 2024. 3

[72] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 2, 3, 4, 5, 8, 9

[73] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. *arXiv preprint arXiv:2412.15109*, 2024. 3

[74] Yixiao Wang, Yifei Zhang, Mingxiao Huo, Ran Tian, Xiang Zhang, Yichen Xie, Chenfeng Xu, Pengliang Ji, Wei Zhan, Mingyu Ding, et al. Sparse diffusion policy: A sparse, reusable, and flexible policy for robot learning. *arXiv preprint arXiv:2407.01531*, 2024. 3

[75] Zhendong Wang, Zhaoshuo Li, Ajay Mandlekar, Zhenjia Xu, Jiaojiao Fan, Yashraj Narang, Linxi Fan, Yuke Zhu, Yogesh Balaji, Mingyuan Zhou, et al. One-step diffusion policy: Fast visuomotor policies via diffusion distillation. *arXiv preprint arXiv:2410.21257*, 2024. 3

[76] Junjie Wen, Minjie Zhu, Yichen Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Chengmeng Li, Xiaoyu Liu, Yaxin Peng, Chaomin Shen, et al. Diffusion-vla: Scaling robot foundation models via unified diffusion and autoregression. *arXiv preprint arXiv:2412.03293*, 2024. 3

[77] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025. 3

[78] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pretraining for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023. 3, 6

[79] Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, and Katerina Fragkiadaki. Unifying diffusion models with action detection transformers for multi-task robotic manipulation. In *7th Annual Conference on Robot Learning*, page 5, 2023. 3

[80] Ted Xiao, Harris Chan, Pierre Sermanet, Ayzaan Wahid, Anthony Brohan, Karol Hausman, Sergey Levine, and Jonathan Tompson. Robotic skill acquisition via instruction augmentation with vision-language models. *arXiv preprint arXiv:2211.11736*, 2022. 3

[81] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, pages 399–417. Springer, 2024. 5

[82] Jonathan Yang, Catherine Glossop, Arjun Bhorkar, Dhruv Shah, Quan Vuong, Chelsea Finn, Dorsa Sadigh, and Sergey Levine. Pushing the limits of cross-embodiment learning for manipulation and navigation. *arXiv preprint arXiv:2402.19432*, 2024. 3

[83] 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 *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024. 2, 3

[84] Edwin Zhang, Yujie Lu, William Wang, and Amy Zhang. Language control diffusion: Efficiently scaling through space, time, and tasks. *arXiv preprint arXiv:2210.15629*, 2022. 3

[85] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. 5

Dita: Scaling Diffusion Transformer for Generalist Vision-Language-Action Policy

Supplementary Material

A. Model and Training Scheme

The architecture of our model is illustrated in Figure 3. The language instruction is encoded using a pretrained CLIP model, with its encoder frozen during training. Input images are resized to 224 × 224 and processed by a pretrained DINOv2 model, with all parameters being finetuned. A Q-Former, trained from scratch with a depth of 4, is employed to reduce the dimensionality of the image features to a length of 32; within each block, text tokens are injected as FiLM conditions to augment the image features with linguistic information. The action is perturbed with noise via a DDPM scheduler with 100 timesteps, and a timestamp index is embedded using a sinusoidal positional embedding module. These multimodal inputs are then fed into a causal Transformer, which predicts the added noise. The Transformer adopts a LLaMA2-style architecture, trained from scratch, comprising 12 self-attention blocks with a hidden size of 768. All components are trained except for the CLIP text encoder. In total, the model comprises 334M parameters, with 221M being trainable. Achieving this level of performance with such a compact model represents a pioneering advancement in the field, underscoring the efficacy of the architectural design.

B. Simulation Benchmarks

B.1. SimplerEnv

Results. As described in Figure 7, leveraging the in-context conditioning design, Dita exhibits enhanced robustness, relying solely on third-person view images to detect subtle nuances and generate more reliable actions.

B.2. LIBERO

LIBERO comprises four subtasks: LIBERO-SPATIAL, LIBERO-OBJECT, LIBERO-GOAL, and LIBERO-100, each designed to evaluate different model capabilities. LIBERO-SPATIAL assesses spatial relationship understanding, containing data with identical object sets but varying layouts. LIBERO-OBJECT evaluates object transferability, featuring data with consistent layouts but different object sets. LIBERO-GOAL examines task comprehension and transferability, maintaining the same object sets and layouts while varying tasks. LIBERO-100 is further divided into LIBERO-90 and LIBERO-10 (also referred to as LIBERO-LONG), designated for policy pretraining and long-horizon task evaluation, respectively. LIBERO-100 encompasses a diverse range of objects, layouts, and backgrounds, providing a comprehensive benchmark for generalization in robot learning.

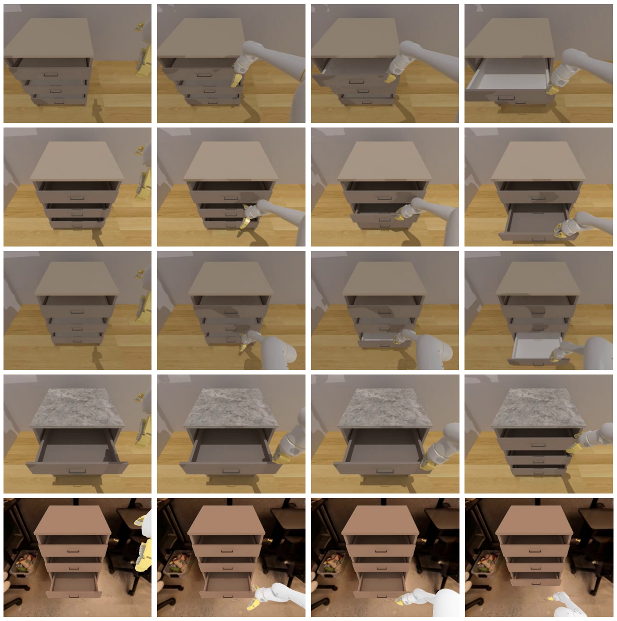


Figure 7. Qualitative results of Dita under variances in Google Robot.

Optimization. We optimize the network using AdamW for 100,000 steps on LIBERO. The learning rate is set to $1e-4$ for LIBERO-SPATIAL, LIBERO-OBJECT, and LIBERO-GOAL, and $5e-4$ for LIBERO-LONG. Across all subdatasets, a half-cycle cosine scheduler is applied to decay the learning rate. Denoising timestamps are set to 100 during finetuning, and training is conducted with a batch size of 512 across 8 NVIDIA A100 GPUs.

Results. Table 7 shows that Dita achieves a success rate of 77.93% on the most challenging task in LIBERO, *i.e.*, SPATIAL-LONG. We argue that the Droid dataset [31] serves as a more suitable pretraining dataset for LIBERO, as our model (334M) lacks the capacity to fully accommodate the entire OXE dataset. We anticipate that performance on the OXE-pretrained model can be significantly improved by scaling up the model size.

B.3. CALVIN

Setup. We directly apply the proposed method to CALVIN using a single static RGB camera to predict the end-effector action, which includes three dimensions for translation,

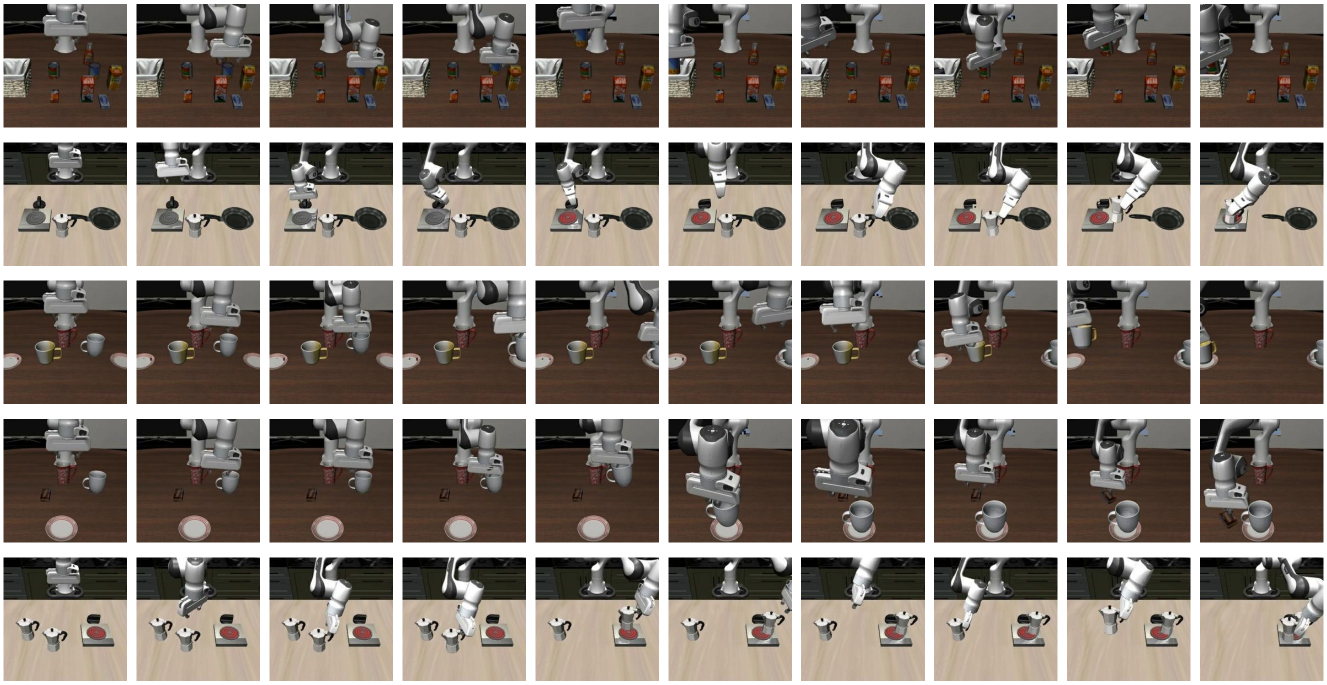


Figure 8. Qualitative results of Dita on LIBERO benchmark.

Table 7. Comparison with Diffusion Policy [17], Octo [72], and OpenVLA [32] on LIBERO [40]. Dita (OXE) denotes the use of a pretrained model on OXE, while Dita (Droid) refers to the use of a pretrained model on Droid.

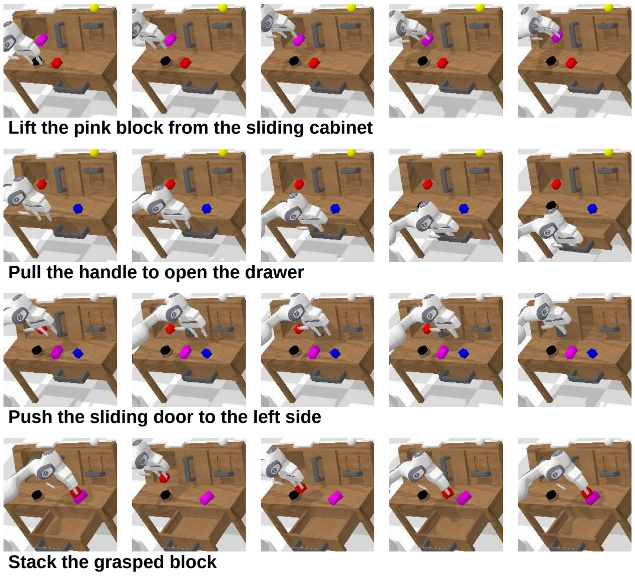
Method	LIBERO-LONG
Diffusion Policy* [17]	50.5%
Octo [72]	51.1%
OpenVLA [32]	53.7%
Dita (pretrained on OXE)	63.8%
Dita (pretrained on Droid)	**77.9%**

three dimensions for Euler angle rotation, and one dimension for gripper position (open or close). We evaluate Dita and $\mathcal{E}_S^{Diff}$ on CALVIN, leveraging the pretrained model on the OXE dataset to initialize the model for CALVIN.

Optimization. For each training iteration, the model predicts 10 future action chunks supervised by MSE loss. An AdamW optimizer is used together with a decayed learning rate with half-cycle cosine scheduler after several steps of warming up. The learning rate is initialized as $1e-4$. Model is trained for 15 epochs with batch size of 128 across 4 NVIDIA A100 GPUs.

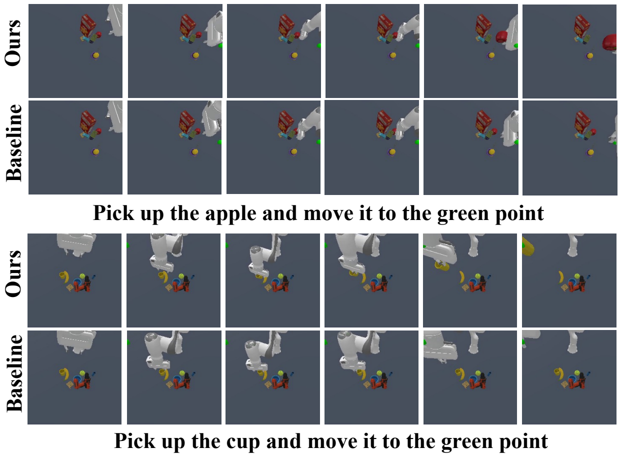
B.4. Maniskill2

ManiSkill2 [22], the next generation of the SAPIEN ManiSkill benchmark [48], serves as a widely recognized platform for assessing the generalized manipulation capabilities of embodied models. It encompasses 20 distinct manipulation task families and over 4M demonstration frames across various configurations. Leveraging ManiSkill2, we establish a *novel camera view generalization* benchmark to evaluate the effectivenes of Dita.



Lift the pink block from the sliding cabinet
Pull the handle to open the drawer
Push the sliding door to the left side
Stack the grasped block

Figure 9. Qualitative results of Dita on CALVIN ABC→D benchmark.



Pick up the apple and move it to the green point

Pick up the cup and move it to the green point

Figure 10. Qualitative comparison between Dita (top) and Diffusion Action Head baseline $\mathcal{E}_{\theta\sim s}^{Diff}$ (bottom) on ManiSkill2 (PickClutterYCB).

Optimization. The network is optimized using AdamW for 50,000 steps on ManiSkill2, with a learning rate set to $1e-4$. The number of denoising timestamps is set to 100, and the batch size is 1024 distributed across 16 NVIDIA A100 GPUs.

C. Real-Robot Experiments

C.1. Real-Robot Setup

Optimization. We apply image augmentations using ColorJitter from the torchvision library, with brightness set to 0.3, contrast ranging from 0.7 to 1.3, saturation ranging from 0.7 to 1.3, and hue set to 0.07. Further details are provided in the code.

Variance Robustness. To evaluate the robustness of Dita, we further validate its performance under different variances, including:

- *Background changes.* The background includes both the tabletop color and the backdrop. We introduce variance in both aspects by using tablecloths in colors different from the tabletop and a black backdrop.
- *Non-target object arrangements.* We randomly place non-target objects in arbitrary poses within the robot's workspace to create a cluttered scene, whereas it remains clean during demonstration recording.
- *Lighting conditions.* We modify the lighting by turning off one of the two lights in the room to introduce variation in illumination.

C.2. Details of Real-Robot Tasks

In addition to the fundamental tasks used for quantitative comparison with prior approaches, we incorporate complex long-horizon tasks that previous methods fail to complete for illustrative purposes. Below, we present all tasks along with their step-wise decomposition.

- *Pick the banana into the box.* We divided this task into two steps: first, successfully picking up the banana, and second, successfully placing it into the box.
- *Pick the kiwifruit in the box.* We divided this task into two steps: first, successfully picking up the kiwifruit, and second, successfully placing it into the box.
- *Pouring the coffee beans into the bowl.* This task is divided into two steps: first, successfully picking up the cup, and second, successfully pouring the coffee beans within the cup into the box.
- *Pouring the water from the teapot into the cup.* This task is divided into two steps: first, successfully picking up the teapot, and second, successfully pouring the water into the cup.
- *Stacking three bowls.* This task is divided into two steps: Stacking the first bowl successfully, and second stacking the left bowl into previous stacked bowls.
- *Stacking three nesting dolls.* This task is divided into two steps: Stacking the first two small dolls successfully, and second stacking the large doll into previous stacked dolls.
- *Pick the banana and insert into the small pen container.* This task is divided into two steps: first, successfully picking up the banana, and second, successfully inserting the banana into the pen container.
- *Open the Flip-top door box and the pick up the small cube inside.* This task is divided into two steps: first, successfully open the door box, and second, successfully picking up the small cube.
- *Open the drawer.* This task has only one step.
- *Close the drawer.* This task has only one step.
- *Pick up the bowl within the drawer and pouring the coffee beans into the outside bowl.* This is a long horizon task, and we demonstrate it with video in the supplementary appendix given that previous approaches fail to complete the task.
- *Pick up the racket and hit the ball into the goal* This is a long horizon task, and we demonstrate it with video in the supplementary appendix given that previous approaches fail to complete the task.
- *Open the top drawer, then pick the cube into the drawer, and finally close the drawer.* This is a long horizon task, and we demonstrate it with video in the supplementary appendix given that previous approaches fail to complete the task.
- *Close the top drawer, then open the bottom drawer and put the bowl into the drawer, and finally close the drawer.* This is a long horizon task, and we demonstrate it with video in the supplementary appendix given that previous approaches fail to complete the task.
- *Open the box and move the green cube into the box then close the box.* This is a long horizon task, and we demon-

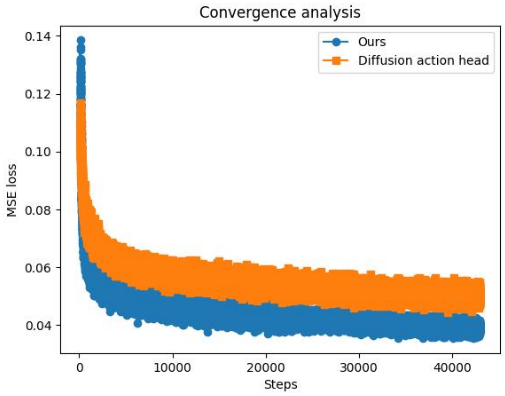


Figure 11. Convergence Analysis on OXE dataset [9]. The blue line is DiT Policy, and the orange line is Diffusion action head strategy with the same number of parameters.

strate it with video in the supplementary appendix given that previous approaches fail to complete the task.

C.3. Finetuning Details

We adhere to [32] and employ LoRA for 10-shot finetuning. However, Dita comprises only 221M trainable parameters, with merely 5% (approximately 11M) remaining trainable under LoRA finetuning. We contend that this limited capacity is inadequate to effectively accommodate image augmentations, thereby compromising robustness against environmental variances. To this end, we evaluate robustness through full finetuning and observe a substantial improvement in the success rate for long-horizon tasks, alongside greater resilience to variances such as background changes, non-target object arrangements, and lighting conditions. Quantitatively, full finetuning achieves a success rate of 20%, whereas LoRA finetuning fails to complete tasks under extreme variances.

D. Analysis, Ablations, and Discussions

D.1. Practices on reproducing Octo and OpenVLA

We observe that OpenVLA [32] demonstrates superior pick-up performance compared to Octo [72]. However, for tasks requiring the learning of rotational operations, such as opening a box, Octo achieves better performance. We attribute this to Octo's ability to predict continuous actions, which are less sensitive to action normalization, whereas OpenVLA relies on action discretization based on action statistics. We compute the statistics from the 10-shot training samples across all tasks and find it challenging to obtain suitable statistics for discretization values, which are unnecessary for the diffusion policy.

Table 8. The ablation study on the learning rate scheduler in the Calvin benchmark.

Strategy	**No. Instructions in a Row (1000 chains)**					
w lr decay	**94.5%**	**82.5%**	**72.8%**	**61.3%**	**50.0%**	**3.61**
w/o lr decay	91.8%	80.0%	68.0%	56.9%	45.9%	3.43

Table 9. More action designs w/o pretraining on Calvin (ABC→D). MDT is from issue 9 of its GitHub repo and GR-MG.

Methods	No. Instructions in a Row (1000 chains)					
MDT* [61]	61.7%	40.6%	23.8%	14.7%	8.7%	1.54
Unet1D head [17]	76.8%	46.5%	28.8%	18.5%	10.0%	1.80
Transformer head [17]	75.8%	44.8%	26.5%	16.5%	8.0%	1.72
8-layer MLP head	69.8%	42.5%	26.3%	16.8%	11.0%	1.66
3-layer MLP head	75.5%	44.8%	25.0%	15.0%	7.5%	1.68
Single token act chunks	56.5%	18.3%	6.0%	2.8%	0.8%	0.84
Ours	**89.5%**	**63.3%**	**39.8%**	**27.3%**	**18.5%**	**2.38**

Table 10. The effect of the number of execution steps (# Steps) on ManiSkill2.

# Steps	1	2	4	8	16
All	**61.6%**	60.8 %	60.6 %	60.0 %	58.0 %

D.2. Convergence Analysis

Figure 11 illustrates the convergence comparison between the diffusion head baseline $\mathcal{E}_{\theta}^{Diff}$ and Dita. Dita achieves clearly faster convergence than $\mathcal{E}_{\theta}^{Diff}$. We believe this further highlights the scalability of Dita.

D.3. More Ablations

Learning Rate Scheduler. As outlined in the main text, we utilize a standard learning rate scheduler to decay the learning rate during the experiments in Calvin, rather than using a fixed learning rate of $1e-4$ as in the pretraining stage. This adjustment results in a slight performance improvement, as shown in Table 8.

Table 11. Ablation study of shuffle buffer size on SimplerEnv (both math and variant results of Google Robot [8]).

Shuffle Buffer Size	**coke_can**	
	match	variant
128000	71.2%	73.6%
256000	**83.7%**	**85.5%**

Diffusion Head. We implement several policies inspired by the core idea of the diffusion action head, which differ slightly from Octo [72] in predicting action chunks. Specifically, Octo [72] flattens action chunks into a single vector with a unified embedding. For instance, when predicting 8 actions, it generates a $8 \times 7 = 56$-dimensional vector. In contrast to the Octo-style diffusion action head, we adopt a diffusion action head, akin to Diffusion Loss [35] and Diffusion Force [14], which are more effective. We evaluate multiple diffusion heads, including Unet1D, Transformer, and MLP on Calvin without pretraining.

Table 9 shows that Dita achieves the better generalization on Calvin (ABC→D), compared to other diffusion head strategies [3, 17, 85].

Execution steps. Since Dita can anticipate multiple future actions, we can execute multiple steps within a single inference. Here, we analyze the impact of execution steps under a model with a trajectory length of 32, as presented in Table 10. The ablation study reveals that shorter execution steps yield slightly better results than longer ones; that is, the further the prediction extends from the current frame, the lower its accuracy. Nevertheless, the slight performance drop demonstrates that even with only 2-frame image observations, Dita can generate reliable action trajectories, underscoring its scalability.

Shuffle Buffer Size. The shuffle buffer size of TensorFlow datasets has a significant impact on performance. Following OpenVLA [32, 54], we utilize TensorFlow datasets for network optimization, where the shuffle buffer size similarly influences performance (Table 11), as observed in Octo [72].