ManiFlow: A General Robot Manipulation Policy via Consistency Flow Training

Ge Yan[1] **Jiyue Zhu**[2*] **Yuquan Deng**[1*]
Shiqi Yang[2] **Ri-Zhao Qiu**[2] **Xuxin Cheng**[2] **Marius Memmel**[1]
Ranjay Krishna[1,4†] **Ankit Goyal**[3†] **Xiaolong Wang**[2†] **Dieter Fox**[1,3,4†]
[1]University of Washington [2]UC San Diego [3]Nvidia [4]Allen Institute for Artifical Intelligence
[*]Equal Contribution [†] Equal Advising

MANIFLOW-POLICY.GITHUB.IO

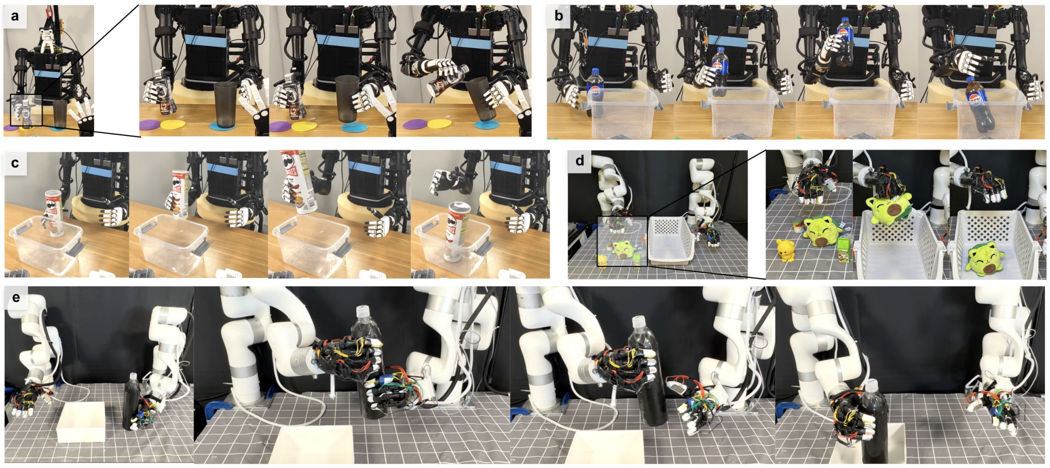


Figure 1: We introduce ManiFlow, a flow matching model excelling in complex manipulation tasks, including bimanual dexterous manipulation. *a*: Robot autonomously pours water. *b-d*: Robot grasps diverse objects and placing them into containers. *e*: Passing a bottle from one hand to the other.

Abstract: This paper introduces ManiFlow, a visuomotor imitation learning policy for general robot manipulation that generates precise, high-dimensional actions conditioned on diverse visual, language and proprioceptive inputs. We leverage flow matching with consistency training to enable high-quality dexterous action generation in just 1-2 inference steps. To handle diverse input modalities efficiently, we propose DiT-X, a diffusion transformer architecture with adaptive cross-attention and AdaLN-Zero conditioning that enables fine-grained feature interactions between action tokens and multi-modal observations. ManiFlow demonstrates consistent improvements across diverse simulation benchmarks and nearly doubles success rates on real-world tasks across single-arm, bimanual, and humanoid robot setups with increasing dexterity. The extensive evaluation further demonstrates the strong robustness and generalizability of ManiFlow to novel objects and background changes, and highlights its strong scaling capability with larger-scale datasets. Our website: maniflow-policy.github.io .

1 Introduction

The ability to reliably predict precise and dexterous actions in unstructured environments represents a fundamental challenge in robot learning. Recent advances in diffusion-based policy learning [1] have significantly enhanced robot capabilities in modeling high-dimensional and multi-modal action distributions. More recently, flow matching [2], an alternative generative modeling approach, has demonstrated improved performance and training efficiency in policy learning [3, 4] compared to diffusion-based approaches. In spite of these advances, existing flow matching policies [3, 4, 5, 6] are still limited in efficiency, robustness, and generalizability when performing complex dexterous manipulation tasks in real-world environments. They face challenges in capturing the full complexity of multi-fingered interactions, maintaining temporal coherence across action sequences, generalizing to unseen scenarios, and architectural constraints that insufficiently model multiple data sources inherent in various real-world tasks (e.g., visual, language, proprioception, etc.).

To tackle these challenges, we introduce ManiFlow, a visuomotor imitation model designed to learn robust and generalizable manipulation skills for complex real-world tasks with high dexterity. ManiFlow significantly improves previous flow matching policies [6] through two key contributions. First, we incorporate a consistency training objective into the standard flow matching loss to encourage a more consistent mapping from noisy samples to the target distribution, effectively "straightening" the flow path. As our experiments show, ManiFlow can generate accurate and dexterous actions in fewer inference steps. In contrast to previous efforts to reduce inference steps [7], ManiFlow does not rely on any pretrained teacher model, demonstrating better training efficiency. Second, we demystify the significance of different time sampling choices with valuable insights and baselines for the flow matching model through comprehensive ablations, indicating the advantage of beta and continuous-time sampling for flow matching and consistency training, as shown in Tab. 3.

Beyond the consistency flow training process, ManiFlow also improves the model architecture to handle diverse input modalities more effectively with an expressive transformer architecture DiT-X. The DiT-X block builds on the DiT block in image generation [8] with more effective AdaLN-Zero conditioning for policy learning. Specifically, we use cross-attention layers for high-dimensional visual and language input, with AdaLN-Zero conditioning for low-dimensional inputs like timestep. The learned scale and shift parameters from AdaLN-Zero conditioning are used to adjust the cross-attention input and output features in a selective manner, allowing more efficient and flexible conditioning of multimodal inputs. Our experiments show that simple yet effective modifications, such as applying AdaLN-Zero conditioning to the cross-attention layers for more adaptive conditioning, significantly improve policy performance compared to previous work, such as MDT [9].

We conduct evaluations across two setups: (1) simulation: 12 tasks in 3 dexterous benchmarks in single-task settings, 48 language-conditioned tasks in multi-task settings, and 4 bimanual dexterous tasks for robustness and generalization test in single-task settings. (2) real-world: 8 challenging tasks across three robot setups with increasing dexterity, including single-arm, bimanual, and humanoid dexterous tasks. We find that ManiFlow consistently improves over diffusion and flow matching policies, both in image-based 2D and pointcloud-based 3D settings. Specifically, ManiFlow achieves an improvement of 45.6% and 11.0% in 12 dexterous tasks with image and pointcloud input, respectively. It further achieves 31.4% improvement in the multi-task setting. Notably, ManiFlow achieves 58% improvement over the π_0 model on 4 robustness test tasks and shows superior scaling capability. Finally, ManiFlow more than doubles the success rate of 3D Diffusion Policy [10] across 8 real-world tasks. The key contributions of ManiFlow are three-fold:

- **High-quality and efficient action generation:** ManiFlow jointly optimizes flow matching with a continuous-time consistency training objective to enforce self-consistency and straightness on learned flow trajectories. This method allows the policy to generate high-dimensional, dexterous actions with high quality using only a few denoising steps, allowing faster inference speed.
- **Efficient multi-modal conditioning:** ManiFlow incorporates DiT-X, a transformer architecture that enhances multi-modal conditioning through adaptive cross-attention layers with learned scale and shift parameters. This enables selective feature modulation across different input modalities.
- **Real-world robustness and generalizability:** We evaluate ManiFlow on 3 robot setups with increasing dexterity, including challenging bimanual and humanoid dexterous manipulation tasks. ManiFlow consistently shows superior robustness in modeling complex dexterous behavior from limited human demonstrations and significantly improves generalization capability to diverse novel objects and environmental variations.

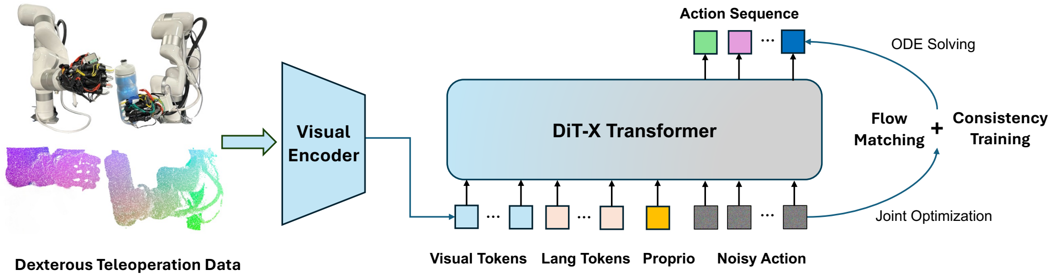


Figure 2: **Policy Architecture of ManiFlow.** Our system processes 2D or 3D visual observations, robot state, or language as inputs and outputs a sequence of actions. We leverage a DiT-X transformer architecture to efficiently optimize a flow matching model with a continuous-time consistency training objective, ensuring high-quality action generation for challenging dexterous tasks.

2 Method

Preliminaries: Flow Matching

We follow [11] to define the flow ODE forward process as straight paths between the data distribution and noise. Given a data point $x_1 \sim D$, a noise point $x_0 \sim \mathcal{N}(0, I)$ and timestep $t \sim \mathcal{U}[0,1]$, we define x_t as a linear interpolation between x_0 and x_1, i.e $x_t = (1-t)\, x_0 + t\, x_1$, and the velocity v_t as the direction from noise to data point: $v_t = x_1 - x_0$. The flow model θ is optimized to predict the velocity given a noisy sample x_t at time point t. The flow matching loss $\mathcal{L}_{\text{FM}}(\theta)$ is defined as:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{x_0, x_1 \sim D}[\|v_\theta(x_t, t) - (x_1 - x_0)\|^2] \tag{1}$$

2.1 ManiFlow Training

ManiFlow goes beyond the basic flow matching model by incorporating a continuous-time consistency training objective and improved time-space sampling strategies, as outlined below.

Continuous-time Consistency Training

Compared with standard diffusion and flow matching models that require many denoising steps during inference [1, 4], consistency training [12] provides an elegant approach to improve generation quality and achieve few-step generation without relying on pre-trained teacher models. The key insight is enforcing the consistency of partially-noisy data points along an ordinary differential equation (ODE) trajectory to the final target data points. We leverage this principle to jointly optimize the flow matching model with a consistency training objective to enhance the consistency of learned flows and thus generate high-quality action trajectories, as shown in the Fig. 3.

Similar to Shortcut Model [13], we add another argument Δt to the flow model $v_t(x_t, t, \Delta t)$, where Δt reflects the step size towards the next target point. We sample a timestep t from the discretized [0,1] interval and a step size Δt from $\mathcal{U}[0,1]$. We define the next timestep t_1 as $t + \Delta t$, ensuring that it is bounded in $[0,1]$ via clipping. The velocity v_{t_1} at point x_{t_1} toward a further timestep t_2 set as $t_1 + \Delta t'$ is predicted as $v_{\theta^-}(x_{t_1}, t_1, \Delta t')$ where θ^- is the exponential moving average (EMA) of the flow model. To enforce consistency between points x_t and x_{t_1}, we first approximate the target data point $\tilde{x}_1 = x_{t_1} + (1 - t_1) \cdot v_{t_1}$. We then further estimate the average velocity target $\tilde{v}_{\text{target}}$ from point x_t to $\tilde{x}_1$ as $\tilde{v}_{\text{target}} = (\tilde{x}_1 - x_t)\ /\ (1 - t)$. We enforce consistency by constraining the flow model to predict this estimated velocity target, with the consistency loss $\mathcal{L}_{\text{CT}}$:

$$\mathcal{L}_{\text{CT}}(\theta) = \mathbb{E}_{t, \Delta t \sim \mathcal{U}[0,1]}\left[\|v_\theta(x_t, t, \Delta t) - \tilde{v}_{\text{target}}\|^2\right] \tag{2}$$

We combine flow matching $\mathcal{L}_{\text{FM}}$ and consistency training losses $\mathcal{L}_{\text{CT}}$ in ManiFlow training: $\mathcal{L}(\theta) = \mathbb{E}[\|v_\theta(x_t, t, 0) - (x_1 - x_0)\|^2 + \|v_\theta(x_t, t, \Delta t) - \tilde{v}_{\text{target}}\|^2]$, where the third argument (Δt) in the flow model is set as 0 for the $\mathcal{L}_{\text{FM}}$ as it estimates local instant velocity [13]. Note that, unlike consistency training [12] that operates in discrete time step size Δt, we sample Δt from a continuous distribution

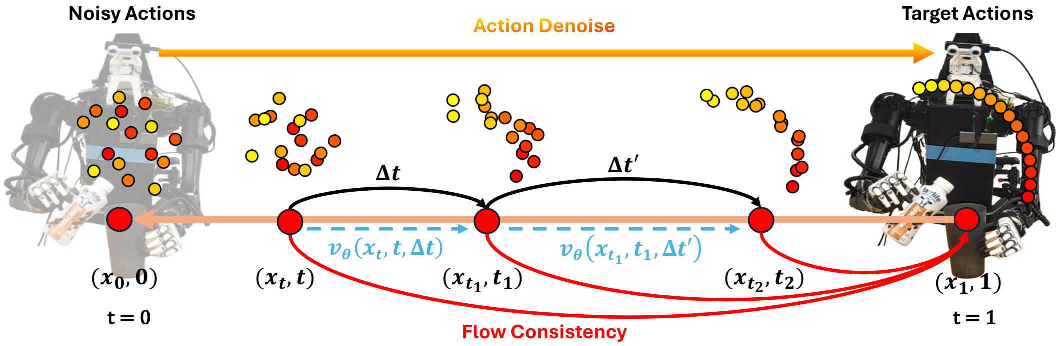


Figure 3: **ManiFlow Consistency Training.** Given a flow path that smoothly transforms action to noise, we sample multiple intermediate points via linear interpolation (e.g., x_t, x_{t_1}, and x_{t_2}). During training, we learn to map any intermediate point on the flow trajectory back to its origin x_1 and ensure the self-consistency of sampled points on the same trajectory.

to remove the undesirable bias associated with discrete-time objectives and ensure more flexible generation. The EMA model provides essential stabilization [12], with more details in the appendix.

Time Space Sampling Strategy

Time scheduling in generative models significantly impacts learning dynamics and final performance. We evaluate five representative timestep t sampling strategies in flow matching as denoted in Eq. 1 with visualization and pseudo-code in Fig. 14 and Alg. 1: (1) Uniform sampling [11], which draws timesteps uniformly from [0,1] and serves as a straightforward baseline; (2) Logit-normal sampling (lognorm) [14], which emphasizes intermediate timesteps through a logit-normal distribution with tunable location and scale parameters; (3) Mode sampling [15], which allows explicit control over whether to favor midpoint or endpoints during training through a scale parameter s; (4) CosMap sampling [16], which adapts the cosine schedule from diffusion models to the flow matching setting through a specialized mapping function; and (5) Beta distribution sampling [17], which places more weight on lower timesteps corresponding to noisier actions, with a cutoff threshold $s = 0.999$ to avoid sampling timesteps that contribute minimal learning value. As we find in Tab. 3, while lognorm sampling shows strong performance, the beta distribution's focus on the high-noise regime proves particularly effective for robotic control tasks, outperforming other scheduling strategies across diverse manipulation scenarios. We further ablate the step size choice Δt in consistency training, denoted in Eq. 2, and continuous time shows improved performance as shown in Tab. 3.

2.2 Perception

Our 3D visual encoder builds upon [10] while introducing a key modification to prioritize the preservation of fine-grained geometric information in 3D point cloud representations. The key insight is that maintaining detailed spatial relationships throughout the encoding process is crucial for precise manipulation tasks. While previous works like [10] used max pooling operations to compress point cloud features into a compact representation, we found this compression can lead to loss of important geometric details. Our architecture deliberately avoids such pooling operations, instead preserving point-wise features throughout the network. This design choice allows the encoder to maintain richer spatial relationships and detailed geometric information of the input point cloud, which we found particularly beneficial for tasks requiring precise object interaction and spatial reasoning.

Empirical observations show that scene configuration significantly impacts the optimal point density for representation efficiency. In well-calibrated scenes with cropped points, ManiFlow achieves strong performance with sparse point clouds of 128 points, demonstrating the efficiency of the network. For uncalibrated egocentric views, denser representations of 4096 points are sufficient, sug-

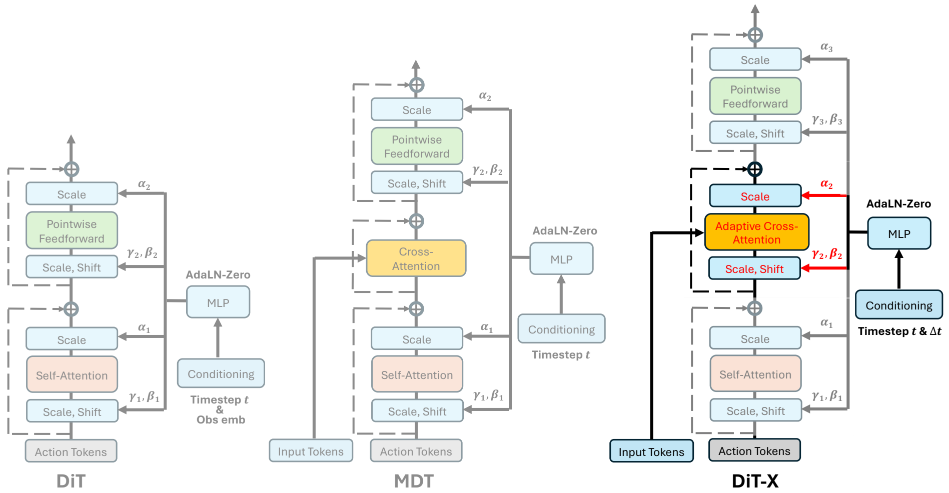


Figure 4: **DiT-X Block.** Unlike DiT (self-attention only) and MDT (basic cross-attention), DiT-X applies AdaLN-Zero conditioning to low-dimensional robot state inputs, and adjusts cross-attention input and output with learned scaling and shift parameters, ensuring adaptive and fine-grained feature interactions between action tokens and multi-modal input tokens. This design enables efficient handling of both low-dimensional control signals and high-dimensional perceptual inputs.

gesting the benefit of increased point density in less structured environments. Note that proper color augmentation is helpful for optimal results without overfitting, as elaborated in the appendix.

2.3 ManiFlow Policy Architecture

For the lack of adaptive conditioning in standard cross-attention mechanisms (e.g., MDT [9]), we introduce **DiT-X**, a transformer architecture that effectively processes low-dimensional signals and high-dimensional multi-modal inputs for general robotic control. Our design is motivated by the inherent challenges in generative models for handling diverse input modalities: low-dimensional signals require precise encoding of high-frequency dynamics, visual inputs contain rich spatial-semantic information, and language instructions introduce fine-grained language understanding. We follow the principles below to design an expressive architecture for multi-modality conditioning.

Adaptability & Granularity: Being capable of generating highly adaptive actions is essential for robot manipulation in a dynamic environment, requiring a reactive adjustment with precision. Additionally, the integration of high-dimensional visual and language features with low-dimensional signal demands fine-grained understanding and adaptive interaction collectively. We address this through a dedicated adaptive cross-attention mechanism that enables direct token-level interactions between actions and multi-modal inputs, facilitating precise spatial and semantic alignment.

DiT-X block with Adaptive Cross-attention Conditioning: We introduce adaptive cross-attention layers to process visual and language tokens with low-dimensional input. Specifically, given low-dimensional inputs like timesteps, we employ AdaLN-Zero conditioning [8] to generate conditioning scale and shift parameters (α, γ, β) for dynamic adaptation of network behavior while ensuring stable training through zero initialization. In particular, instead of only applying scale and shift parameters to self-attention and feedforward layers, we also adjust the input and output of cross-attention layers with the same modulation. This design empowers the network to manipulate fine-grained visual and language tokens by scaling them down or up in a selective manner, which is crucial for tasks requiring a precise understanding of visual cues and language instructions. While this introduces a modest computational overhead, the enhanced representational capability proves valuable for complex manipulation tasks.

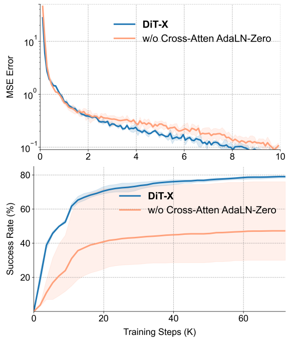


Figure 5: Training action error and success rate of DiT-X vs w/o cross-attention AdaLN-zero conditioning in 10 Metaworld tasks with language conditioning.

Table 1: **Main Simulation Results.** Success rates on 12 dexterous tasks in 3 benchmarks. ManiFlow achieves superior performance on both image and point cloud-based inputs.

Algorithm \ Task	Obs.	RoboTwin 5 tasks	Adroit 3 tasks	DexArt 4 tasks	Average
Diffusion Policy	Img	28.8±2.3	38.1±2.9	53.6±2.1	39.4±2.3
Flow Matching Policy	Img	27.1±2.7	39.0±2.2	53.3±2.4	38.8±2.5
2D ManiFlow Policy	Img	**46.1**±2.7	**74.3**±1.9	**56.3**±2.3	**56.5**±2.4
3D Diffusion Policy	PC	42.7±3.3	77.8±2.4	60.6±0.7	57.4±2.2
3D Flow Matching Policy*	PC	48.1±6.3	77.1±3.3	61.7±1.1	59.9±2.8
3D ManiFlow Policy	PC	**61.9**±2.5	**78.6**±2.3	**63.2**±2.7	**66.5**±2.5

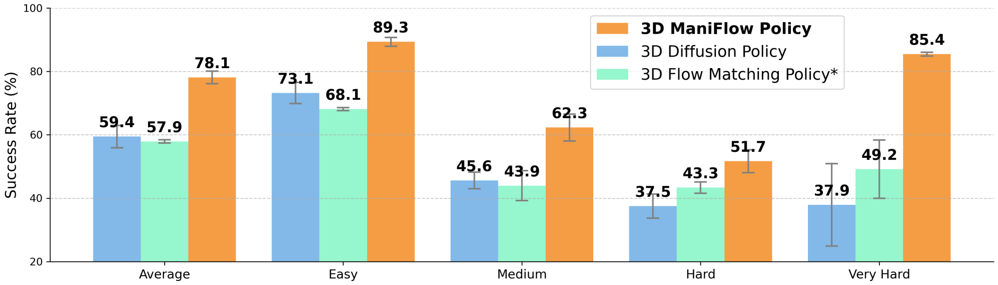


Figure 6: **Comparison on language-conditioned multi-task learning on 48 MetaWorld tasks.** ManiFlow achieves superior performance across all difficulty levels compared to the 3D diffusion and flow matching policy, with an average **31.4%** and **34.9%** relative improvement.

As shown in Fig. 5, the DiT-X block shows faster convergence during training and better performance than w/o cross-attention AdaLN-Zero conditioning. Furthermore, we provide a detailed illustration of the evolving DiT and MDT architecture baselines in Fig. 4. Our architecture provides greater expressiveness than the DiT and MDT blocks on multi-modality conditioning in Fig. 13.

3 Experiments

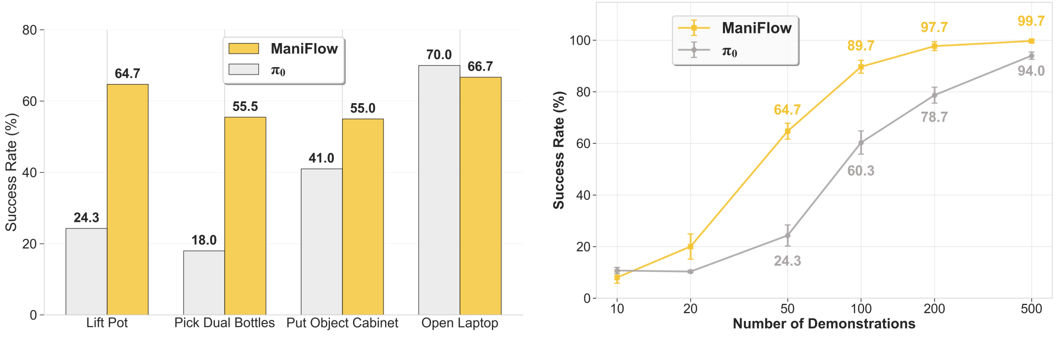
3.1 Simulation Experiments

Benchmarks: We select three diverse dexterous manipulation benchmarks (Adroit [18], Dexart [19], and RoboTwin 1.0 [20] to comprehensively evaluate ManiFlow in 12 dexterous tasks that assess a wide spectrum of manipulation capabilities. Furthermore, with the MetaWorld benchmark [21] comprising 48 tasks, we specifically focus on the challenging language-conditioned multi-task learning scenario to provide a comprehensive assessment of model performance when conditioning on visual and language input. We further use the RoboTwin 2.0 benchmark [22] to fully test the policy robustness and generalizability. More details are provided in the appendix.

Baselines: For 2D image inputs, we compare ManiFlow with diffusion policy [1] and flow matching policy [6] with the same ResNet-18 encoder [23]. For 3D pointcloud-based methods, we primarily compare against 3D Diffusion Policy (DP3) [10], which has demonstrated superior performance over 2D Diffusion Policy across various simulation environments. Since the flow matching policy [6] is only image-based in the original paper, we add the same 3D encoder from [10] to it in order to get a baseline for the 3D-based flow matching model, denoted as 3D Flow Matching Policy*. For the robustness test and scaling experiment on the RoboTwin 2.0 benchmark, we compare with the π_0 model, which takes multi-view images as input and is fine-tuned on the domain randomized data.

3.2 Key Findings

As shown in Tab. 1, ManiFlow outperforms both 2D image and 3D point cloud-based diffusion and flow matching policies on all 3 dexterous benchmarks, with an average 43.4% and 45.6% improvement on 2D input, and 15.9% and 11.0% improvement on 3D input. ManiFlow further achieves 78.1% success rate in language-conditioned multi-task learning on 48 MetaWorld tasks, demonstrating 31.4% and 34.9% improvement (see Fig. 6). Notably, ManiFlow achieves 58% improvement



(a) (b)

Figure 7: **(a) Efficiency & Generalization.** We evaluate ManiFlow and π_0 with 4 bimanual tasks on RoboTwin 2.0 benchmark (Fig. 8), after training with 50 domain randomized demonstrations per task. Compared to the large-scale pre-trained π_0 model, ManiFlow shows superior learning efficiency and generalization capability to novel objects and backgrounds, while learning from scratch with pointcloud input. **(b) Scaling Behavior.** Results show the scaling performance on the task ”lift pot” with demonstration numbers varying from 10 to 500. ManiFlow consistently outperforms π_0 on both the low data regime and final scaling to 500 demos, achieving 99.7% success eventually.

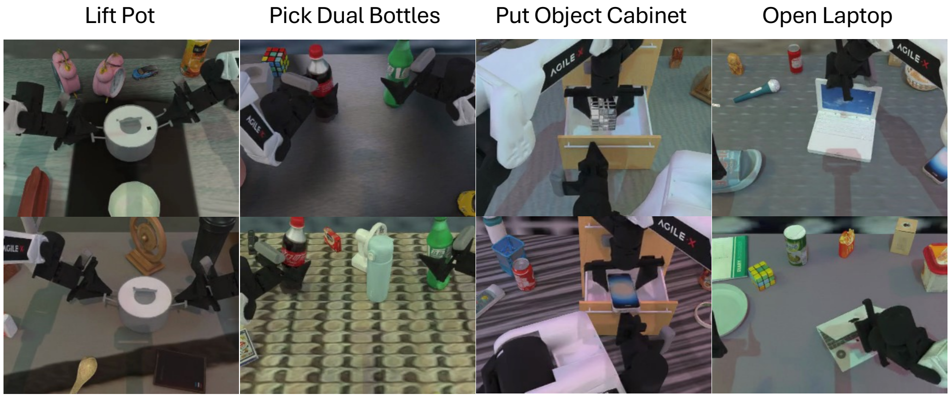


Figure 8: **Visualization of Domain Randomized Evaluation.** To fully test the robustness and generalizability of our policy, we evaluate both ManiFlow and π_0 on the RoboTwin 2.0 benchmark with challenging domain randomizations, including cluttered scenes with random distractors, novel objects and diverse background textures, various lighting conditions, and table height changes.

over the π_0 model on 4 bimanual tasks with point cloud input, also demonstrating superior scaling capability. We discuss the key takeaways below and provide further ablations in the appendix.

High-quality action generation. Dexterous manipulation poses a significant challenge in the model's ability to capture high-dimensional behaviors. We observe that ManiFlow consistently achieves higher success rates compared to the 3D diffusion and flow matching policy. This performance advantage is clearly demonstrated in the most challenging bimanual dexterous tasks in the RoboTwin 1.0 benchmark, where ManiFlow achieves a success rate of 61.9% with only 50 demonstrations, while DP3 achieves 42.7% success rate (see Tab. 1). The performance gap is particularly notable given the challenging nature of bimanual coordination.

Robust visual and language conditioning. ManiFlow demonstrates better visual conditioning capability than diffusion and flow matching policies for both 2D and 3D visual input. Notably, for the Adroit 3 tasks in Tab. 1, 2D ManiFlow achieves 73.2% success rate, while both 2D baselines struggle in this benchmark. Additionally, for language conditioning, we evaluate against 3D-based baselines on 48 MetaWorld tasks with multi-task learning in Fig. 6. ManiFlow outperforms 3D dif-

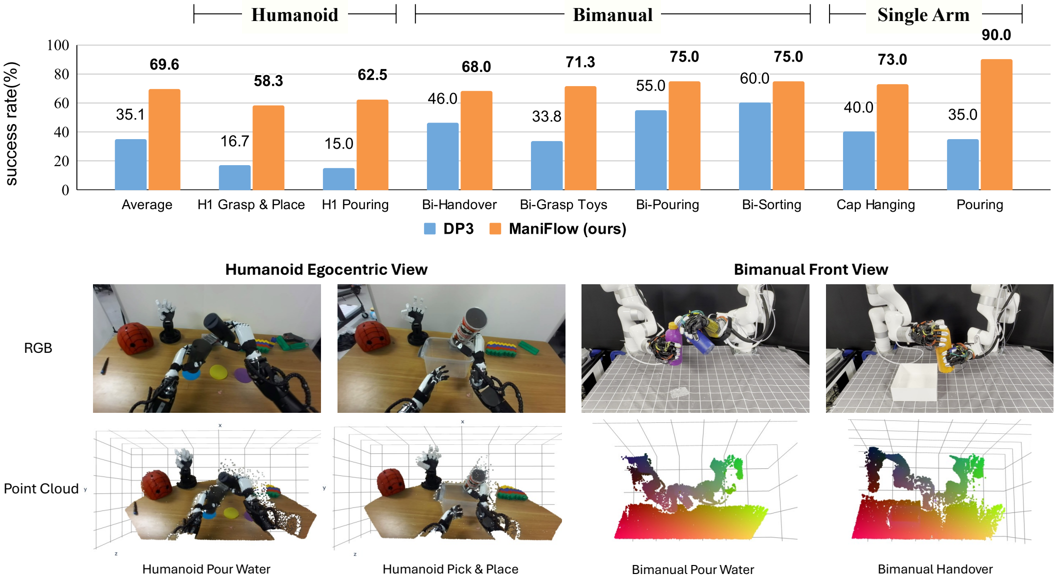


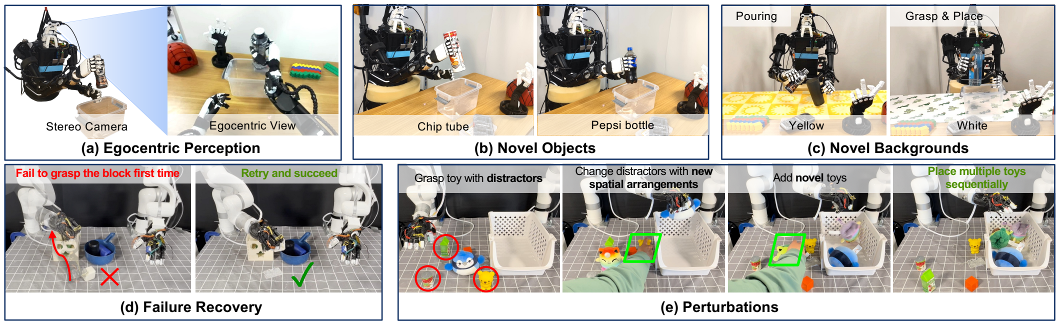
Figure 9: **Real-Robot Results:** (Top) We test 8 real-robot tasks across 3 robot platforms, including Franka with gripper, bimanual xArm with ability hands, and Unitree H1 humanoid with bimanual anthropomorphic hands. ManiFlow succeeds **69.6%** on average, almost doubling DP3's performance. **Visualizations:** (Bottom) 3D point cloud visualizations of sampled 4 real robot tasks.

fusion and flow matching baselines on all task difficulty levels by a large margin: 31.4% and 34.9% relative improvement on average, and notable 125% and 73.6% on the very hard tasks.

Enhanced performance through DiT-X architecture. Our experimental results on 10 language-conditioned MetaWorld tasks demonstrate the significant advantages of ManiFlow's DiT-X block over the DiT and MDT architectures. As shown in Fig. 13, DiT-X achieves faster learning and better final performance on various tasks. DiT-X's adaptive cross-attention AdaLN-Zero conditioning mechanism enables more fine-grained interactions between visual features, language instructions, and action sequences, which is crucial for language-conditioned tasks where success depends on a precise understanding of both visual cues and natural language commands.

Learning Efficiency & Generalization. As demonstrated in Fig. 7(a), ManiFlow achieves superior learning efficiency compared to the fine-tuned π_0 model across 4 challenging bimanual dexterous tasks on the RoboTwin 2.0 benchmark. Training from scratch with only 50 domain randomized demonstrations per task, ManiFlow substantially outperforms π_0: 64.7% vs 24.3% on *Lift Pot*, 55.5% vs 18.0% on *Pick Dual Bottles*, 55.0% vs 41.0% on *Put Object Cabinet*, and 66.7% vs 70.0% on *Open Laptop*, achieving 58% relative improvement on average. Beyond learning efficiency, ManiFlow demonstrates robust generalization to environmental variations including novel objects, diverse backgrounds, cluttered scenes with distractors, and varying lighting conditions as shown in Fig. 8. This combination of efficiency and generalization capability suggests that ManiFlow effectively learns robust and generalizable manipulation skills from limited demonstrations, outperforming even large-scale pre-trained models in challenging unseen scenarios.

ManiFlow Scaling Behavior. ManiFlow exhibits strong scaling capability across different data regimes, as shown on the *lift pot* task in Fig. 7(b). Starting from comparable performance at 10 demonstrations ($\sim$10% for both methods), ManiFlow shows a clear performance advantage in the low-data regime: achieving 64.7% success rate at 50 demonstrations compared to π_0's 24.3%, and quickly reaching $\sim$90% success with 100 demonstrations while π_0 achieves 60.3%. Notably, ManiFlow demonstrates better data scaling behavior by achieving 97.7% success with 200 demonstrations, while π_0 requires 500 demonstrations to reach 94.0%, still below ManiFlow's 200-demo performance. ManiFlow continues to improve to 99.7% at 500 demos. The consistent upward scaling


Stereo Camera
Egocentric View
Chip tube
Pepsi bottle
Pouring
Grasp & Place
Yellow
White
Fail to grasp the block first time
Retry and succeed
Grasp toy with distractors
Change distractors with new spatial arrangements
Add novel toys
Place multiple toys sequentially

(a) Egocentric Perception

(b) Novel Objects

(c) Novel Backgrounds

(d) Failure Recovery

(e) Perturbations

Figure 10: **Real World Robustness.** We test the policy robustness with varying perturbations during real-world deployment, such as different egocentric viewpoints, novel objects and backgrounds, recovering from failure, and adding diverse distractors with human perturbed locations. ManiFlow is robust against these perturbations with limited data. Please check our website for more details.

trajectory indicates that ManiFlow leverages larger scale demonstration data more effectively than π_0, suggesting better scaling properties for learning complex dexterous behaviors with more data.

ManiFlow excels in few-step inference. Due to the costly iterative denoising steps, few-step inference is essential for sufficiently fast policy generation in the real world. As shown in Tab. 4 in the appendix, ManiFlow achieves 63.7% and 64.5% success rate using only 1 and 2 inference steps, respectively, compared to 3D Diffusion and Flow Matching Policies using 10 inference steps to achieve 42.7% and 48.1% success rate on 5 bimanual dexterous tasks in the RoboTwin benchmark.

3.3 Real World Experiments

We evaluate ManiFlow on 8 real-robot tasks across 3 robot setups with increasing dexterity (see Fig. 9 and Tab. 2). Each setup is evaluated on a unique set of tasks designed to assess ManiFlow's capabilities across diverse scenarios. We provide an overview of the robot setups in Fig. 16 and task visualizations in the appendix. We compare ManiFlow against DP3, the previous state-of-the-art dexterous manipulation policy. Both ManiFlow and DP3 take point clouds as visual input. As can be seen, ManiFlow consistently outperforms DP3 by a significant margin: 88.8% relative improvement for in-distribution environment configurations and 116.7% on unseen objects, leading to 98.3% relative improvement on average.

High Dexterity: As shown in Tab. 2, ManiFlow excels in tasks requiring high dexterity, particularly evident in its performance with anthropomorphic hands on the Unitree H1 humanoid and bimanual setups. ManiFlow demonstrates strong capability in tasks such as pouring, where it must precisely control multi-finger positions to grasp the bottle without missing and aligning the bottle opening with the cup carefully, showing improved success rate from 20% to 65% on the humanoid platform. The additional complexity of bimanual coordination, requiring synchronization between two independent dexterous hands, further highlights ManiFlow's superiority. As shown in the handover task that requires the left hand to grasp the bottle first and hand it to the right hand, ManiFlow succeeds on 22 out of 30 runs (73% success rate) compared to DP3's success on 14 out of 30 runs (47%).

Generalization: ManiFlow is able to handle unseen object types and geometries (e.g., varying bottle heights, appearances, and shapes) along with changes in the environment without any significant drop in performance (see Tab. 2). On the other hand, DP3 often halted mid-motion or failed to recognize and adapt to new objects during task execution. This inability to handle unfamiliar objects was particularly evident when DP3 was tasked with manipulating unseen objects in the Toy Grasping tasks. In contrast, our method was able to adapt to novel objects and successfully executed the tasks with minimal disruption. Furthermore, ManiFlow demonstrated robustness to changes in the scene, such as distractor objects, cluttered environments, and varying backgrounds. On the other hand, in tasks like Toy Grasping with randomly placed distractors, DP3 showed a tendency to overfit to the specific end-effector trajectories seen during training.

Table 2: Detailed Comparison of DP3 and ManiFlow on 8 real robot tasks across 3 robot platforms

Real Robot Setup	**Task**	**In Distribution**		**Unseen Objects**	
		DP3	**ManiFlow**	DP3	**ManiFlow**
Humanoid	Grasp & Place	7/40	**23/40**	3/20	**12/20**
	Pouring	4/20	**13/20**	2/20	**12/20**
Bimanual	Handover	14/30	**22/30**	9/20	**12/20**
	Pouring	21/40	**30/40**	12/20	**15/20**
	Toy Grasping	17/50	**37/50**	7/30	**20/30**
	Sorting	7/10	**8/10**	5/10	**7/10**
Single-Arm	Cap Hanging	4/10	**7/10**	2/5	**4/5**
	Pouring	5/10	**9/10**	2/10	**9/10**
Average Success Rate		37.6%	**71.0%**	31.1%	**67.4%**

4 Related Work

Generative Models for Policy Learning: Diffusion models, a family of generative models that iteratively transform random noise into a data sample, have achieved great success in generating high-resolution images and videos. Owning to this impressive success, they have also been applied in various robotics domains. Notably, Diffusion Policies [1] have been effective in modeling multi-modal action distributions. Building on them, Consistency Policies [7] used a pre-trained diffusion model to distill a student model. By using this two-stage pipeline, they demonstrated faster inference with fewer denoising steps. Recently, flow matching has demonstrated improved performance and training efficiency in policy learning [6, 3]. However, these methods still face limitations in modeling more complex and high-dimensional dexterous behaviors. We improve upon the flow matching model by using a consistency training objective. ManiFlow shows strong capability in generating high-quality actions with only a few inference steps, demonstrating both robustness and efficiency in challenging dexterous tasks. Notably, ManiFlow can be trained end-to-end in a single run without requiring an additional teacher model, unlike other methods [7, 24, 25] that typically require pre-training models for teacher-student distillation or multiple training stages for inference acceleration, making them computationally expensive and more cumbersome to work with.

Visual Imitation Learning. Prior works have shown that visual observations are essential for robots to have an accurate understanding of the environment. While 2D image-based imitation learning policies have been widely adopted due to the simplicity and easy access of RGB images, policies that take in 3D input have demonstrated better performance and generalizability. Recent works [26, 27, 28, 29, 30, 31] have shown success in leveraging 3D data for manipulation tasks. However, these methods are typically restricted to low-dimensional 6-DoF end-effector control with coarse temporal keypoints prediction. Hence, they are not suitable for highly dynamic and dexterous tasks. Beyond these methods, 3D Diffuser Actor [32] can predict continuous dense actions, but is still restricted to 6-DoF end-effector control and not applicable for high-dimensional dexterous manipulation. 3D Diffusion Policy [10] leverages an efficient 3D encoder and achieves superior performance for dexterous tasks. Compared to this line of works, we aim to develop a general robot policy that is capable of learning robust manipulation skills from either 2D or 3D observations.

Architecture for Multi-modality Conditioning Recent advancements in robotic manipulation have leveraged data from different modalities to improve robustness and sample efficiency in complex real-world environments. Prior works have developed a high-capacity diffusion transformer (DiT) [33] and applied it to manipulation tasks [34], demonstrating better visual conditioning compared to traditional transformer architecture. A related work MDT [9] showed improved performance by incorporating cross-attention layers to fuse multimodal conditioning information. ManiFlow builds upon these prior works and improves them further through the DiT-X block. We add a simple yet effective modification: introducing the AdaLN-Zero conditioning to the cross-attention layer with learned scaling and shift parameters to better manipulate the conditioned network's features in a selective manner, allowing more flexible and efficient multimodal conditioning.

5 Conclusion

In this work, we introduce ManiFlow, a robust and efficient dexterous manipulation model. ManiFlow improves upon prior flow matching policies by introducing a continuous-time consistency training objective, a superior time sampling strategy, and a novel DiT-X block. The proposed DiT-X architecture effectively handles diverse input modalities through its dual conditioning mechanisms, enabling strong performance across varied manipulation tasks. Our comprehensive evaluation spanning 64 simulation tasks and 8 real-world scenarios demonstrates ManiFlow's effectiveness, particularly in challenging real-world bimanual dexterous manipulation, where it achieves a 98.3% relative improvement over existing approaches.

6 Limitation

While ManiFlow demonstrates strong performance across diverse manipulation tasks, there are several promising avenues for future work. The success in real-world robot tasks depends heavily on the quality and diversity of training demonstrations. Incorporating ManiFlow into a reinforcement learning framework could potentially reduce the burden on the demonstration data. Furthermore, while the design choices for ManiFlow are inspired by dexterous manipulation tasks, none of these are limited to robot manipulation, and we believe that ManiFlow could be equally beneficial for tasks such as navigation or mobile manipulation. Finally, we only scratched the surface of ManiFlow's multi-modal capabilities, and the incorporation of further modalities such as tactile information or VLM-based conditioning via points, trajectories, or bounding boxes is an interesting extension.

Acknowledgments

Part of this work was funded by the Army Research Lab and award #W911NF-24-2-0191.

References

[1] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[2] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *ICLR*, 2023.

[3] E. Chisari, N. Heppert, M. Argus, T. Welschehold, T. Brox, and A. Valada. Learning robotic manipulation policies from point clouds with conditional flow matching. *CoRL*, 2024.

[4] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control, 2024. *URL https://arxiv. org/abs/2410.24164*.

[5] M. Braun, N. Jaquier, L. Rozo, and T. Asfour. Riemannian flow matching policy for robot motion learning. In *IROS*, 2024.

[6] F. Zhang and M. Gienger. Affordance-based robot manipulation with flow matching. *arXiv preprint arXiv:2409.01083*, 2024.

[7] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. *RSS*, 2024.

[8] W. S. Peebles and S. Xie. Scalable diffusion models with transformers. 2023 ieee. In *CVF International Conference on Computer Vision (ICCV)*, volume 4172, 2022.

[9] M. Reuss, Ö. E. Yağmurlu, F. Wenzel, and R. Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. *RSS*, 2024.

[10] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy. *RSS*, 2024.

[11] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[12] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[13] K. Frans, D. Hafner, S. Levine, and P. Abbeel. One step diffusion via shortcut models. *ICLR*, 2025.

[14] J. Atchison and S. M. Shen. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261–272, 1980.

[15] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

[16] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[17] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[18] V. Kumar. *Manipulators and Manipulation in high dimensional spaces*. PhD thesis, University of Washington, Seattle, 2016. URL https://digital.lib.washington.edu/researchworks/handle/1773/38104.

[19] C. Bao, H. Xu, Y. Qin, and X. Wang. Dexart: Benchmarking generalizable dexterous manipulation with articulated objects. In *CVPR*, 2023.

[20] Y. Mu, T. Chen, S. Peng, Z. Chen, Z. Gao, Y. Zou, L. Lin, Z. Xie, and P. Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). *arXiv preprint arXiv:2409.02920*, 2024.

[21] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[22] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Q. Liang, Z. Li, X. Lin, Y. Ge, Z. Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[24] G. Lu, Z. Gao, T. Chen, W. Dai, Z. Wang, W. Ding, and Y. Tang. Manicm: Real-time 3d diffusion policy via consistency model for robotic manipulation. *arXiv preprint arXiv:2406.01586*, 2024.

[25] B. Jia, P. Ding, C. Cui, M. Sun, P. Qian, S. Huang, Z. Fan, and D. Wang. Score and distribution matching policy: Advanced accelerated visuomotor policies via matched distillation. *arXiv preprint arXiv:2412.09265*, 2024.

[26] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *CoRL*, 2023.

[27] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox. Rvt: Robotic view transformer for 3d object manipulation. In *CoRL*, 2023.

[28] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox. Rvt-2: Learning precise manipulation from few demonstrations. *RSS*, 2024.

[29] Y. Ze, G. Yan, Y.-H. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *CoRL*, 2023.

[30] G. Yan, Y.-H. Wu, and X. Wang. Dnact: Diffusion guided multi-task 3d policy learning. *arXiv preprint arXiv:2403.04115*, 2024.

[31] Y. Li, G. Yan, A. Macaluso, M. Ji, X. Zou, and X. Wang. Integrating lmm planners and 3d skill policies for generalizable manipulation. *arXiv preprint arXiv:2501.18733*, 2025.

[32] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *CoRL*, 2024.

[33] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[34] S. Dasari, O. Mees, S. Zhao, M. K. Srirama, and S. Levine. The ingredients for robotic diffusion transformers. *arXiv preprint arXiv:2410.10088*, 2024.

[35] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[36] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[37] J. Zhou, J. Wang, B. Ma, Y.-S. Liu, T. Huang, and X. Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023.

[38] L. Yang, Z. Zhang, Z. Zhang, X. Liu, M. Xu, W. Zhang, C. Meng, S. Ermon, and B. Cui. Consistency flow matching: Defining straight flows with velocity consistency. *arXiv preprint arXiv:2407.02398*, 2024.

[39] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *ICLR*, 2021.

[40] X. Li, M. Liu, H. Zhang, C. Yu, J. Xu, H. Wu, C. Cheang, Y. Jing, W. Zhang, H. Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023.

[41] K. Black, M. Nakamoto, P. Atreya, H. Walke, C. Finn, A. Kumar, and S. Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[42] H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023.

[43] R. Ding, Y. Qin, J. Zhu, C. Jia, S. Yang, R. Yang, X. Qi, and X. Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. *arXiv preprint arXiv:2407.03162*, 2024.

[44] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang. Open-television: Teleoperation with immersive active visual feedback. *CoRL*, 2024.

A Policy Implementation Details

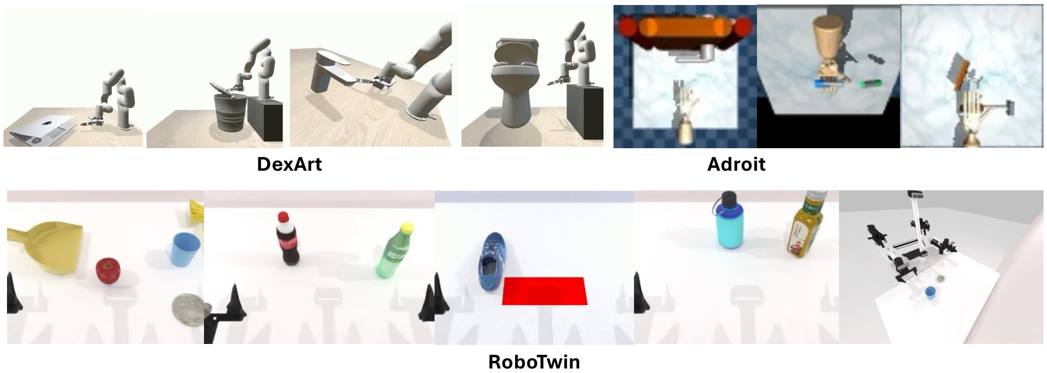


Figure 11: Simulation Tasks Visualization. 12 dexterous manipulation tasks, including 4 DexArt tasks, 3 Adroit tasks, and 5 bimanual dexterous RoboTwin 1.0 tasks.

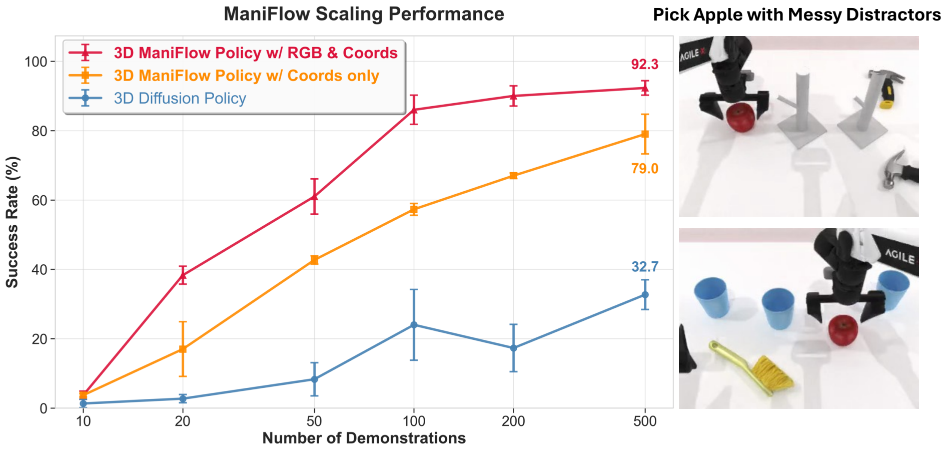


Figure 12: **Scaling Comparison.** We evaluate 3D ManiFlow Policy and 3D Diffusion Policy across 10 to 500 demos on the *Pick Apple Messy* task from the RoboTwin 1.0 benchmark. ManiFlow achieves a 79.0% success rate with 500 demonstrations using point cloud coordinates only, significantly outperforming the diffusion baseline at 32.7%. Adding RGB information further improves performance to 92.3%, demonstrating superior data efficiency and scaling capability of ManiFlow.

A.1 Perception

ManiFlow is designed as a general robot policy capable of learning robust manipulation skills from either 2D or 3D visual observations. Our experiments demonstrate consistent improvements over baseline methods in both modalities, with 32.9% relative improvement on 2D image inputs and 16.2% improvement on 3D point cloud inputs across dexterous manipulation benchmarks. We detail our 2D and 3D visual encoding approaches below.

2D Visual Encoding. For 2D image-based inputs, we train a ResNet-18 encoder from scratch to process RGB images and extract adaptive visual features optimized by policy gradient. The resulting visual tokens are fed into our DiT-X transformer for cross-attention conditioning with action tokens. We apply a set of image augmentations, including random crop (ratio 0.95), random rotation (±5 degrees), and color jitter (brightness 0.3, contrast 0.4, saturation 0.5, hue 0.08) to improve generalization and robustness. This design choice is primarily for less noisy simulation environments. For in-the-wild real-world environments, we recommend using larger, pre-trained visual encoders to learn more robust and reactive behavior, as demonstrated in UMI [35].

3D Visual Encoding. Our 3D visual encoder builds upon a lightweight pointnet encoder [10] while removing max pooling operations to preserve point-wise 3D features for fine-grained geometric understanding. We elaborate the key design choices for deploying 3D-based ManiFlow subsequently.

Point Cloud Density. ManiFlow can learn from varying point cloud densities efficiently. In well-calibrated and cropped scenes, ManiFlow only needs very sparse point clouds with as few as 128 to 256 points. For more cluttered environments, ManiFlow adopts denser 2048 to 4096 points to ensure adequate spatial coverage and preserve important geometric details in complex scenes.

Point Cloud Augmentation. We found that SE3 spatial augmentation is detrimental to performance and do not use it in our training. In most simulation tasks, we use point cloud coordinates only unless specifically noted. However, as demonstrated in Fig. 12, adding color information can substantially improve performance in cluttered environments as it provides rich semantics regarding various objects and surroundings. In real-world experiments, color jitter augmentation becomes essential for generalizing to environment changes and preventing overfitting to specific lighting conditions. We apply the same color jitter parameters as in image augmentation to the RGB in point clouds with 0.2 probability, significantly improving robustness and generalizability in real-world deployment.

Learn from Egocentric View. ManiFlow is applied to both third-person view cameras with static viewpoints and egocentric view with active sensing cameras that have moving viewpoints. For third-person setups, cameras are positioned externally to provide consistent, fixed perspectives of the manipulation workspace, as seen in our real-world bimanual and single-arm experiments. For egocentric setups, such as the humanoid configuration with gimbal-mounted stereo cameras, the visual perspective dynamically changes as the robot's head moves during data collection, requiring the policy to handle varying viewpoints and coordinate head-arm movements simultaneously.

More capable 3D Encoders. While our current lightweight PointNet-based encoder prioritizes simplicity and efficiency for dexterous manipulation, it may be limited in highly complex in-the-wild scenes that require richer semantic understanding. Future enhancements could address these limitations through two primary directions: (1) integrating pre-trained 3D foundation models [36, 37] to leverage large-scale geometric and semantic priors for improved generalization to novel objects and environments, and (2) lifting 2D semantic features from vision-language models into 3D space [29, 30, 32], to combine our efficient geometric processing with rich semantic understanding. These approaches would strengthen ManiFlow's robustness and adaptability to more challenging real-world scenarios with diverse objects, cluttered environments, and varying lighting conditions.

A.2 ManiFlow & Baseline Model Details.

Language Encoding. For language-conditioned tasks, we use a frozen pre-trained T5 language model to encode instructions into 512-dimensional embeddings, then project to token dimensions for cross-attention.

Proprioception Encoding. Proprioception is encoded through a 2-layer MLP. We notice that progressively masking proprioceptive inputs with probability p during training helps alleviate overfitting to proprioception only and prevents the model from learning shortcuts that bypass visual understanding. This masking strategy can be important for dexterous manipulation tasks where robots might otherwise rely too heavily on proprioceptive feedback rather than developing robust visual-motor coordination, ultimately leading to more generalizable policies that can handle sensor noise and partial state observability in real-world deployment.

Action Generation: We predict action sequences of varying lengths depending on task complexity and use a 2-layer MLP to decode action tokens into continuous actions. We use action horizons of 4 steps for short-horizon simulation tasks (Adroit, DexArt, MetaWorld) and 16 steps for dexterous tasks in RoboTwin requiring bimanual coordination. For real-world tasks, we use 64 steps to account for execution delays and employ temporal ensembling to aggregate predicted actions over multiple timesteps, ensuring smoother temporal transitions and avoiding abrupt motion discontinuities for better stability and safety. We use an observation history of 2 timesteps for all tasks to provide temporal context while maintaining computational efficiency.

Baseline Architecture. We use the U-Net architecture as the diffusion network for both 2D and 3D diffusion/flow matching policies, following their original papers and code. While Diffusion Policy has both CNN and transformer variants available, we use the U-Net version as it demonstrates superior performance in our experiments.

A.3 ManiFlow Training Details.

We employ a single-stage training approach that jointly optimizes flow matching and consistency objectives without requiring pre-trained teacher models. Rather than directly constraining velocities at intermediate points to be identical along the flow path, which often yields trivial solutions and unstable training, we learn mappings from any partially-noised data point to the final target data point, ensuring self-consistency throughout the ODE trajectory. We provide the pseudocode for different times sampling strategies in Alg. 1 and ManiFlow training in Alg. 2.

Joint Training Strategy. To reduce the training cost of ManiFlow, our training batch consists of two components with different batch ratios: 75% for flow matching training and 25% for consistency training. During flow matching training, we set $\Delta t = 0$ to predict instantaneous velocity at timestep t, while consistency training uses sampled Δt from a continuous uniform distribution to enforce consistency across different points on the same trajectory. Additionally, we use different time sampling strategies for t: Beta distribution for flow matching to emphasize the high-noise regime, and discrete uniform sampling for consistency training to cover the full denoising trajectory.

Target Time Conditioning. A key design choice in our velocity prediction is the target timestep conditioning. We evaluate two modes: *absolute* mode, where the model predicts velocity toward $t + \Delta t$, and *relative* mode, where it predicts velocity for step size Δt. Empirically, we find that the relative mode (Δt conditioning) achieves better performance than the absolute mode, as it provides more direct step-size information for the model to learn appropriate velocity magnitudes.

EMA Stabilization. The exponential moving average (EMA) model plays a crucial role in stabilizing consistency training [12]. During consistency training, we require reliable velocity predictions at future timesteps to compute consistency targets, but using the current model (which is being updated) can lead to training instability due to rapidly changing predictions. Instead, we maintain an EMA version of the model parameters $\theta^- = \mu\theta^- + (1-\mu)\theta$, where μ is the momentum coefficient. This EMA model provides stable target generation for consistency training by offering slowly-evolving, more reliable velocity predictions at intermediate timesteps. The EMA mechanism ensures that consistency targets remain relatively stable across training iterations, preventing oscillations and enabling smooth convergence of the joint flow matching and consistency objectives.

A.4 Failure Cases.

We observe that ManiFlow fails in tasks that require detailed contact information and precise force feedback, such as delicate assembly operations or compliant insertion tasks. This limitation stems from ManiFlow's design focus on kinematic control rather than force-based interactions, lacking the tactile sensing and force control capabilities necessary for tasks where contact dynamics are critical for success. However, we believe incorporating tactile feedback as an additional modality would significantly enhance ManiFlow's capability to handle more contact-rich manipulation tasks and broaden its applicability.

B Simulation Experiments.

B.1 Training Details.

We collect varying amounts of demonstrations across benchmarks based on task complexity: 10 demonstrations per task for Adroit and MetaWorld, 50 for RoboTwin, and 100 for DexArt. To en-

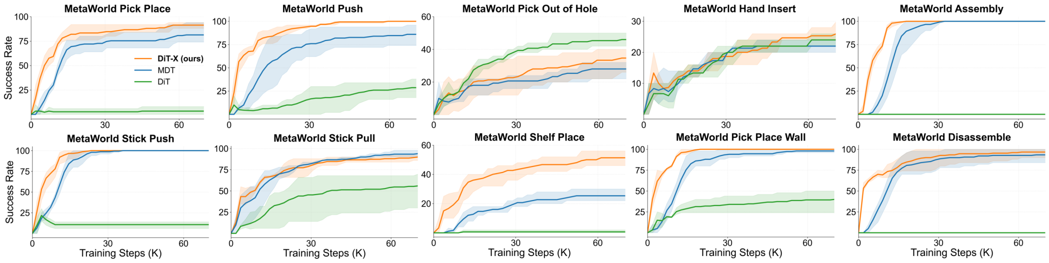


Figure 13: **Comparison between DiT, MDT, and ManiFlow's DiT-X block:** Language-conditioned multi-task learning curves for 10 MetaWorld hard tasks. DiT-X demonstrates faster convergence towards higher accuracy, highlighting superior multi-modal conditioning capabilities.

sure rigorous and fair evaluation, all models are trained and tested under identical conditions across multiple benchmarks. For the RoboTwin benchmark, models are trained for 2000 epochs, with performance evaluated on the final checkpoint over 100 episodes. For the Adroit and DexArt benchmarks, models are trained for 3000 epochs, with performance assessed every 50 epochs over 20 episodes. The final performance metric is computed as the average of the top five success rates to account for potential performance variations. In the MetaWorld benchmark, we specifically focus on the more challenging language-conditioned multi-task learning scenario rather than single-task evaluation. This decision stems from the observation that both baseline 3D diffusion policy and our method consistently achieve near-perfect success rates (approximately 90% to 100%) in single-task settings for most tasks, making it difficult to meaningfully differentiate their capabilities. The language-conditioned multi-task setting provides a more nuanced assessment of model performance. For all benchmarks, we report both mean success rates and standard deviations across three independent training seeds to provide a comprehensive view of model performance and stability. This evaluation protocol, with consistent metrics and multiple seeds, ensures robust and reliable performance comparisons across all tested approaches.

B.2 Simulation Benchmark

MetaWorld contributes single-arm manipulation scenarios such as door opening and tool use, while Adroit specializes in dexterous manipulation using a shadow dexterous hand for precise finger control tasks like in-hand manipulation and pen twirling. DexArt introduces challenging dexterous tasks tested on unseen articulated objects, such as lifting a bucket and turning on a faucet with a revolute joint, while RoboTwin complements the suite with realistic simulation environments and a variety of bimanual dexterous manipulation tasks. This carefully curated benchmark selection enables a thorough evaluation of our policy's generalization capabilities across different environments, task complexities, and skill sets, providing comprehensive insight into its robustness and adaptability while maintaining direct relevance to real-world applications. The visualization of simulation tasks across these 4 benchmarks is shown in Fig. 11.

B.3 Ablation

More Scaling Comparison. We evaluate both 3D ManiFlow Policy and 3D Diffusion Policy across varying numbers of demonstrations on the Pick Apple Messy dexterous task from the RoboTwin benchmark, which requires picking apples from cluttered environments with distractors and random positions. As shown in Fig. 12, ManiFlow exhibits strong scaling performance, increasing from 3.7% with 10 demos to 57.3% with 100 demos, and reaching 79.0% with 500 demos using point cloud coordinates only, significantly outperforming the 3D diffusion policy baseline, which plateaus at 32.7%. Adding RGB information further enhances performance, achieving 86.0% at 100 demos and continuing to improve to 92.3% at 500 demos. This scaling capability stems from ManiFlow's more capable DiT-X architecture and efficient consistency training objective that better leverages

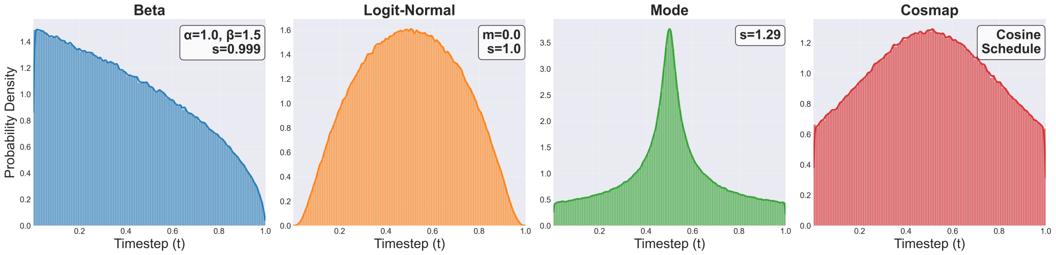


Figure 14: **Comparison of timestep sampling strategies for flow matching models**. We show the sample probability density of different timestep $t \in [0, 1]$. The Beta distribution ($\alpha = 1.0, \beta = 1.5, s = 0.999$) concentrates samples near t=0 (early noise levels), the logit-Normal distribution ($m = 0.0, s = 1.0$) provides balanced sampling around $t = 0.5$, the Mode distribution ($s = 1.29$) strongly favors midpoint during training through a scale parameter s, and Cosmap follows a cosine schedule. Histograms represent empirical sample frequencies, while smooth curves show estimated probability distributions. We provide pseudo code for each sampling strategy in Alg. 1.

Table 3: **Ablation on Time Scheduler.** We compare different time schedulers of timestep t for flow matching and stepsize Δt for consistency training with our ManiFlow policy.

Time	Time Scheduler	Door	Pen	shelf-place	pick-place-wall	stick-pull	stick-push	disassemble	**Average**
t	**Beta**	**80.3±1.2**	**55.5±5.8**	44.0±9.1	**95.3±0.9**	90.7±0.9	**100.0±0.0**	**80.0±1.6**	**78.0±2.8**
	Uniform	77.7±0.9	55.0±2.9	40.0±4.3	94.7±5.0	87.3±4.7	**100.0±0.0**	80.0±4.3	76.4±3.2
	Lognorm	79.5±2.0	55.0±2.8	43.3±8.4	94.7±2.5	90.7±0.9	**100.0±0.0**	**81.0±3.0**	77.7±2.8
	Cosmap	80.3±2.9	52.0±2.5	44.0±5.9	93.3±5.0	88.0±2.8	**100.0±0.0**	82.0±5.9	77.1±3.6
	Mode	78.8±5.9	53.0±2.7	35.3±2.5	94.7±6.2	89.3±3.4	**100.0±0.0**	82.0±3.3	76.2±3.4
Δt	**continuous**	**80.3±1.2**	**55.5±5.8**	**44.0±9.1**	**95.3±0.9**	**90.7±0.9**	**100.0±0.0**	80.0±1.6	**78.0±2.8**
	discrete	78.7±2.0	52.0±3.9	37.3±10.6	**95.3±6.6**	90.0±4.0	**100.0±0.0**	**80.7±2.5**	76.3±4.2

more abundant data for learning complex dexterous behaviors. We expect ManiFlow to achieve even better performance with larger, more diverse datasets.

Ablation on Time Scheduler. We ablate the scheduler choices of timestep t and stepsize Δt on 7 tasks from Adroit and MetaWorld benchmarks. For sampling t, as shown in Tab. 3, while other schedulers like uniform, Cosmap, Mode, and especially logit-normal achieve reasonable results, the beta scheduler consistently outperforms them. The key advantage stems from its emphasis on lower timesteps with higher noise levels, which is particularly beneficial for robotic action prediction. This finding aligns with the insight that robot observations provide rich constraints on possible actions, making the learning of noise-conditioned policies especially important in the high-noise regime. For Δt sampling, continuous time sampling shows better performance than discrete sampling.

Comparison Across Diffusion and Flow-Matching Training Objectives. We evaluate ManiFlow against representative generative models with different training objectives. Diffusion Policy [1] serves as our primary diffusion-based baseline given its strong performance in robotic control. For flow matching approaches, we include Rectified Flow [11], which introduces a simplified training objective optimizing straight trajectories in latent space, Consistency-FM [38], which leverages velocity consistency to improve sample quality, and the shortcut model [13]. which conditions on the additional step size and enforces self-consistency to improve generation quality. As shown in Tab. 5, ManiFlow consistently outperforms these baselines across diverse manipulation scenarios, demonstrating the effectiveness of our proposed training objective for robotic control tasks.

ManiFlow as a Versatile and Effective Policy Head. The broad applicability of ManiFlow is demonstrated through its successful integration into the established 3D Diffuser Actor [32] (3D-DA) architecture as a policy head. As shown in Tab. 6, single-step inference with ManiFlow (avg sequence 3.67) outperforms the original 25-step DDPM (avg sequence 3.35), achieving 25 times inference speedup. The performance advantage becomes more pronounced for longer instruction sequences, where our 10-step ManiFlow achieves a 0.68 higher average sequence length. Notably, the improvement is particularly significant for longer-horizon tasks, with ManiFlow showing substantial gains in completing 4-instruction (73.0% vs 53.3%) and 5-instruction chains (65.7% vs 41.2%).

Table 4: **Few-step Inference.** ManiFlow achieves better efficiency with only a few inference steps compared to 3D Diffusion and Flow Matching Policy across 5 bimanual dexterous tasks on the RoboTwin benchmark.

Algorithm	Inference Step	Pick	Diverse	Dual	Empty	Shoe	Average
3D Diffusion Policy	10	9.3±3.7	38.3±7.1	46.3±2.5	73.0±0.8	46.5±2.5	42.7±3.3
3D Flow Matching Policy*	10	16.0±7.1	56.3±6.6	46.5±0.5	82.3±1.7	39.3±15.5	48.1±6.3
3D ManiFlow Policy	1	42.7±1.9	75.3±1.7	53.7±0.5	**83.0±0.0**	63.7±2.6	63.7±2.2
	2	**43.3±2.1**	**76.3±1.7**	54.0±1.6	82.0±1.4	66.7±2.9	**64.5±1.9**
	4	38.3±1.2	72.7±1.9	**54.3±1.9**	75.3±2.4	67.3±4.9	61.6±2.5
	8	41.3±0.5	72.7±2.5	53.7±1.7	72.3±3.3	**68.3±2.9**	61.7±2.2
	10	42.0±0.8	72.3±1.7	54.0±2.2	72.7±4.8	**68.3±2.9**	61.9±2.5

Table 5: **Ablation on more generative models.** We include more diffusion and flow matching baselines for comparison. All variants use the same encoder and DiT-X architecture.

Algorithm \ Task	Door	Pen	shelf-place	pick-place-wall	stick-pull	stick-push	disassemble	**Average**
ManiFlow	**80.3±1.2**	**55.5±5.8**	44.0±9.1	**95.3±0.9**	90.7±0.9	**100.0±0.0**	**80.0±1.6**	**78.0±2.8**
DDIM [39]	79.3±5.2	53.8±1.0	44.0±5.7	90.7±7.4	**94.0±1.6**	**100.0±0.0**	78.7±1.9	77.2±3.3
Rectified Flow [11]	78.2±3.6	49.7±5.0	**46.0±4.9**	88.7±9.8	88.0±4.3	**100.0±0.0**	79.3±1.9	75.7±4.2
Consistency-FM [38]	79.7±1.9	52.2±1.9	42.0±5.9	92.0±8.5	88.7±5.2	**100.0±0.0**	79.3±3.4	76.3±3.8
Shortcut Model [13]	80.0±1.4	52.2±5.7	40.7±5.2	93.3±5.7	89.3±4.1	**100.0±0.0**	78.0±2.8	76.2±3.6

These promising results demonstrate ManiFlow's potential as an efficient and effective replacement for existing diffusion-based policy heads across robotic learning frameworks.

	Instruction completed in a row (1000 chains)					
	1	2	3	4	5	Avg. Len
RoboFlamingo [40]	82.4	61.9	46.6	33.1	23.5	2.48
SuSIE [41]	87.0	69.0	49.0	38.0	26.0	2.69
GR-1 [42]	85.4	71.2	59.6	49.7	40.1	3.06
3D-DA (DDPM 25 steps)	93.8$_{\pm0.01}$	80.3$_{\pm0.0}$	66.2$_{\pm0.01}$	53.3$_{\pm0.02}$	41.2$_{\pm0.01}$	3.35$_{\pm0.04}$
3D-DA (ManiFlow 1-step)	92.7$_{\pm0.6}$	82.4$_{\pm1.5}$	72.0$_{\pm3.5}$	64.4$_{\pm3.5}$	55.9$_{\pm4.8}$	3.67$_{\pm0.13}$
3D-DA (ManiFlow 10-step)	**95.1**$_{\pm0.3}$	**88.0**$_{\pm1.3}$	**81.0**$_{\pm1.7}$	**73.0**$_{\pm3}$	**65.7**$_{\pm3.2}$	**4.03**$_{\pm0.09}$

Table 6: **Zero-shot long-horizon evaluation on CALVIN** on 3 random seeds. 3D-DA [32] with ManiFlow policy head achieves superior performance with fewer inference steps, especially for longer instruction sequences.

C Real World Experiment

C.1 Real-World Setups

We evaluate ManiFlow's performance on three distinct robot setups: the Unitree H1 humanoid robot, the bimanual xArm 7 robot configuration, and the Franka Emika Panda robot. Each setup is evaluated on a unique set of tasks designed to assess ManiFlow's manipulation capabilities across diverse scenarios. Fig. 16 provides a visual overview of the experimental setups, including robot configurations and task environments. The details of each setup are as follows:

(a) **Humanoid Setup.** The Unitree H1 is a full-sized humanoid robot equipped with two 7-DoF arms and anthropomorphic hands featuring 28-DoF (two 7-DoF arms + two 6-DoF anthropomorphic Inspire hands + 2-DoF active head). It is equipped with a gimbal-mounted ZED stereo camera, enabling active perception and spatial awareness. The humanoid's anthropomorphic hand design with 12 total DoF per hand (6 actuated, 6 underactuated through linkage mechanisms) requires sophisticated multi-finger coordination.

(b) **Bimanual Setup.** This setup consists of two UFACTORY xArm 7 robotic arms paired with two 6-DoF PSYONIC Ability Hands featuring 26-DoF in total, following the experiment

Table 7: **Main results on 3 dexterous manipulation benchmarks.**

Algorithm \ Task	Obs.	Adroit (10 demos)				DexArt (100 demos)				
		hammer	door	pen	**Average**	laptop	faucet	bucket	toilet	**Average**
Diffusion Policy	Img	54.0±3.6	41.8±2.7	18.5±2.5	38.1±2.9	81.7±2.1	29.3±2.1	26.0±2.4	77.3±1.9	53.6±2.1
Flow Matching Policy	Img	55.7±4.2	40.0±1.6	21.2±0.8	39.0±2.2	81.7±2.5	31.3±3.7	24.0±2.2	76.3±1.2	53.3±2.4
2D ManiFlow	Img	**100.0±0.0**	**67.0±2.2**	**56.0±3.6**	**74.3±1.9**	**85.7±2.1**	**32.3±0.5**	**29.7±3.4**	**77.7±3.3**	**56.3±2.3**
3D Diffusion Policy	PC	100.0±0.0	76.7±4.7	**56.7±2.6**	77.8±2.4	89.7±0.9	41.7±0.5	31.3±0.5	**79.7±0.9**	60.6±0.7
3D Flow Matching*	PC	100.0±0.0	77.7±6.1	53.5±3.9	77.1±3.3	92.7±1.2	42.0±0.8	32.3±1.9	**79.7±0.5**	61.7±1.1
3D ManiFlow	PC	**100.0±0.0**	**80.3±1.2**	55.5±5.8	**78.6±2.3**	**93.0±1.6**	**45.0±3.6**	**35.3±2.1**	79.3±3.3	**63.2±2.7**

Algorithm \ Task	Obs.	RoboTwin (50 demos)						**Overall Avg.**
		Pick Apple Messy	Diverse Bottles Pick	Dual Bottles Pick Hard	Empty Cup Place	Shoe Place	**Average**	
Diffusion Policy	Img	17.0±0.8	36.3±2.4	41.3±3.7	42.0±1.6	7.3±2.9	28.8±2.3	39.4±2.3
Flow Matching Policy	Img	15.3±1.9	32.0±4.5	43.0±0.0	38.0±5.4	7.3±1.7	27.1±2.7	38.8±2.5
2D ManiFlow	Img	**37.3±4.8**	**37.0±1.6**	**47.3±2.1**	**63.7±1.2**	**45.3±3.7**	**46.1±2.7**	**56.5±2.4**
3D Diffusion Policy	PC	9.3±3.7	38.3±7.1	46.3±2.5	73.0±0.8	46.5±2.5	42.7±3.3	57.4±2.2
3D Flow Matching*	PC	16.0±7.1	56.3±6.6	46.5±0.5	**82.3±1.7**	39.3±15.5	48.1±6.3	59.9±2.8
3D ManiFlow	PC	**42.0±0.8**	**72.3±1.7**	**54.0±2.2**	72.7±4.8	**68.3±2.9**	**61.9±2.5**	**66.5±2.5**

Table 8: **Language-conditioned Multi-task results on 48 Meta-World simulation tasks.** Results for using 10 demonstrations for each task are provided in this table.

	Meta-World (Easy)						
Alg \ Task	Button Press	Button Press Topdown	Button Press Topdown Wall	Button Press Wall	Coffee Button	Dial Turn	Door Close
3D Diffusion Policy	62±15	**100±0**	**100±0**	72±25	73±34	53±15	**100±0**
3D Flow Matching*	0±0	**100±0**	**100±0**	67±31	97±5	**70±7**	**100±0**
3D ManiFlow	**100±0**	**100±0**	**100±0**	**100±0**	**100±0**	67±13	**100±0**

	Meta-World (Easy)						
Alg \ Task	Door Lock	Door Open	Door Unlock	Drawer Close	Drawer Open	Faucet Close	Faucet Open
3D Diffusion Policy	0±0	**100±0**	98±2	88±13	98±2	92±8	83±12
3D Flow Matching*	0±0	**100±0**	**100±0**	5±7	**100±0**	92±6	**100±0**
3D ManiFlow	**78±14**	**100±0**	**100±0**	**100±0**	**100±0**	**100±0**	**100±0**

	Meta-World (Easy)						
Alg \ Task	Handle Press	Handle Pull	Handle Pull Side	Lever Pull	Plate Slide	Plate Slide Back	Plate Slide Back Side
3D Diffusion Policy	**100±0**	22±17	43±6	60±12	20±18	92±12	**100±0**
3D Flow Matching*	**100±0**	15±18	20±7	45±11	0±0	88±10	**100±0**
3D ManiFlow	100±0	**42±10**	**65±7**	**63±19**	**100±0**	**93±9**	**100±0**

	Meta-World (Easy)			Meta-World (Medium)			
Alg \ Task	Plate Slide Side	Reach	Reach Wall	Window Close	Window Open	Basketball	Bin Picking
3D Diffusion Policy	92±6	48±2	25±4	**100±0**	83±17	**100±0**	0±0
3D Flow Matching*	82±14	57±10	35±11	**100±0**	92±6	90±4	18±6
3D ManiFlow	**100±0**	**58±19**	**67±9**	**100±0**	**97±2**	**100±0**	**33±2**

	Meta-World (Medium)						
Alg \ Task	Box Close	Coffee Pull	Coffee Push	Hammer	Peg Insert Side	Push Wall	Soccer
3D Diffusion Policy	18±8	52±23	55±0	**77±6**	58±6	60±8	8±5
3D Flow Matching*	18±8	67±2	58±15	75±22	68±5	15±15	**10±4**
3D ManiFlow	**45±12**	**97±2**	**82±6**	42±24	**88±8**	**93±2**	7±6

	Meta-World (Medium)		Meta-World (Hard)				
Alg \ Task	Sweep	Sweep Into	Assembly	Hand Insert	Pick Out of Hole	Pick Place	Push
3D Diffusion Policy	70±4	3±2	77±16	7±9	**20±11**	42±5	55±14
3D Flow Matching*	63±21	0±0	88±10	0±0	38±2	53±17	62±2
3D ManiFlow	**92±2**	**7±2**	**100±0**	**12±9**	13±5	**68±5**	**88±9**

	Meta-World (Very Hard)				**Average**
Alg \ Task	Shelf Place	Disassemble	Stick Pull	Pick Place Wall	
3D Diffusion Policy	25±8	55±19	28±14	55±27	59.4±3.5
3D Flow Matching*	18±10	**67±5**	43±25	40±11	57.9±0.5
3D ManiFlow	**28±5**	63±8	**83±5**	**98±2**	**78.1±2.0**

configuration used in Bunny-VisionPro [43]. An Intel RealSense LiDAR L515 camera, positioned in front of the setup, provides visual observations.

(c) **Single-Arm Setup.** This configuration uses a 7-DoF Franka Emika Panda robot with a Robotiq parallel gripper. The robot is mounted statically, and an Intel RealSense D455 RGB-D camera provides external visual observations.

Humanoid vs. Bimanual Setup. The key differences include both perception and hardware complexity:

(a) **Perception:** The Humanoid Setup uses first-person active sensing with a 2-DoF gimbal-mounted stereo camera that moves with the operator's head during data collection, requiring the policy to learn coordinated head-arm movements and handle dynamic viewpoints from training data. In contrast, the Bimanual Setup uses a static third-person view camera, providing a consistent but relatively limited perspective. The humanoid's active perception adds complexity as the policy must learn optimal head movements while managing visual instabilities from camera motion.

(b) **Hardware Complexity:** Humanoid setup present greater control challenges due to quasi-direct-drive motors that have gear clearance and reduced accuracy compared to precision industrial arms (UFactory xArms) used in bimanual setups. They also feature complex anthropomorphic kinematic chains with additional singularities and workspace limitations from human-like proportions. These mechanical imprecisions and kinematic constraints create significant challenges for policy learning in dexterous manipulation, as the learned policies must compensate for hardware inconsistencies and coordinate more complex joint configurations for fine-grained tasks.

C.2 Task Descriptions

We evaluate ManiFlow on eight real-world tasks, consisting of **(i)** two tasks evaluated on *Humanoid Setup*: **Humanoid Grasp & Place**, **Humanoid Pouring**, **(ii)** four tasks on *Bimanual Setup*: **Bimanual Handover**, **Bimanual Pouring**, **Bimanual Toy Grasping**, **Bimanual Sorting**, and **(iii)** two tasks on *Single-Arm Setup*: **Single-Arm Cap Hanging**, and **Single-Arm Pouring**. Notably, Bimanual Toy Grasping is a Single-Arm task executed within the bimanual setup. The first word of each task name specifies the corresponding real-world setup. Fig. 15 shows the trajectories of the tasks. Detailed task descriptions are provided below:

(a) **Humanoid Grasp & Place**: The right hand grasps a water bottle and places it into a container. Evaluated on *Humanoid Setup*.

(b) **Humanoid Pouring**: The left hand grasps a cup and holds it. The right hand grasps a bottle and pours it into the cup accurately. Evaluated on *Humanoid Setup*.

(c) **Bimanual Handover**: The left hand grasps a bottle and hands it over to the right hand. The right hand then places the bottle into a box. Evaluated on *Bimanual Setup*.

(d) **Bimanual Pouring**: Both hands grasp separate water bottles. The left hand performs a pouring motion above the bottle held by the right hand. Evaluated on *Bimanual Setup*.

(e) **Bimanual Toy Grasping**: The right hand grasps a toy and places it into a basket, while randomly placed distractors interfere with the grasp. Evaluated on *Bimanual Setup*.

(f) **Bimanual Sorting**: Continuously sorts three objects, with the right hand placing cubes into a box and the left hand sorting cylinders into a pot. Evaluated on *Bimanual Setup*.

(g) **Single-Arm Cap Hanging**: The gripper grasps a cap and precisely positions it onto a hook. Evaluated on *Single-Arm Setup*.

(h) **Single-Arm Pouring**: The gripper grasps a bottle and pours it into a cup on the table. Evaluated on *Single-Arm Setup*.

C.3 Data Collection

(a) **Humanoid Setup**: We follow the data collection approach outlined in Open-TeleVision [44], using the Apple Vision Pro for teleoperation.

(b) **Bimanual Setup**: We adopt the same data collection methods as Bunny-VisionPro [43], using Apple Vision Pro to teleoperate the bimanual hand-arm setup. Approximately 50 demonstrations are collected for each task.

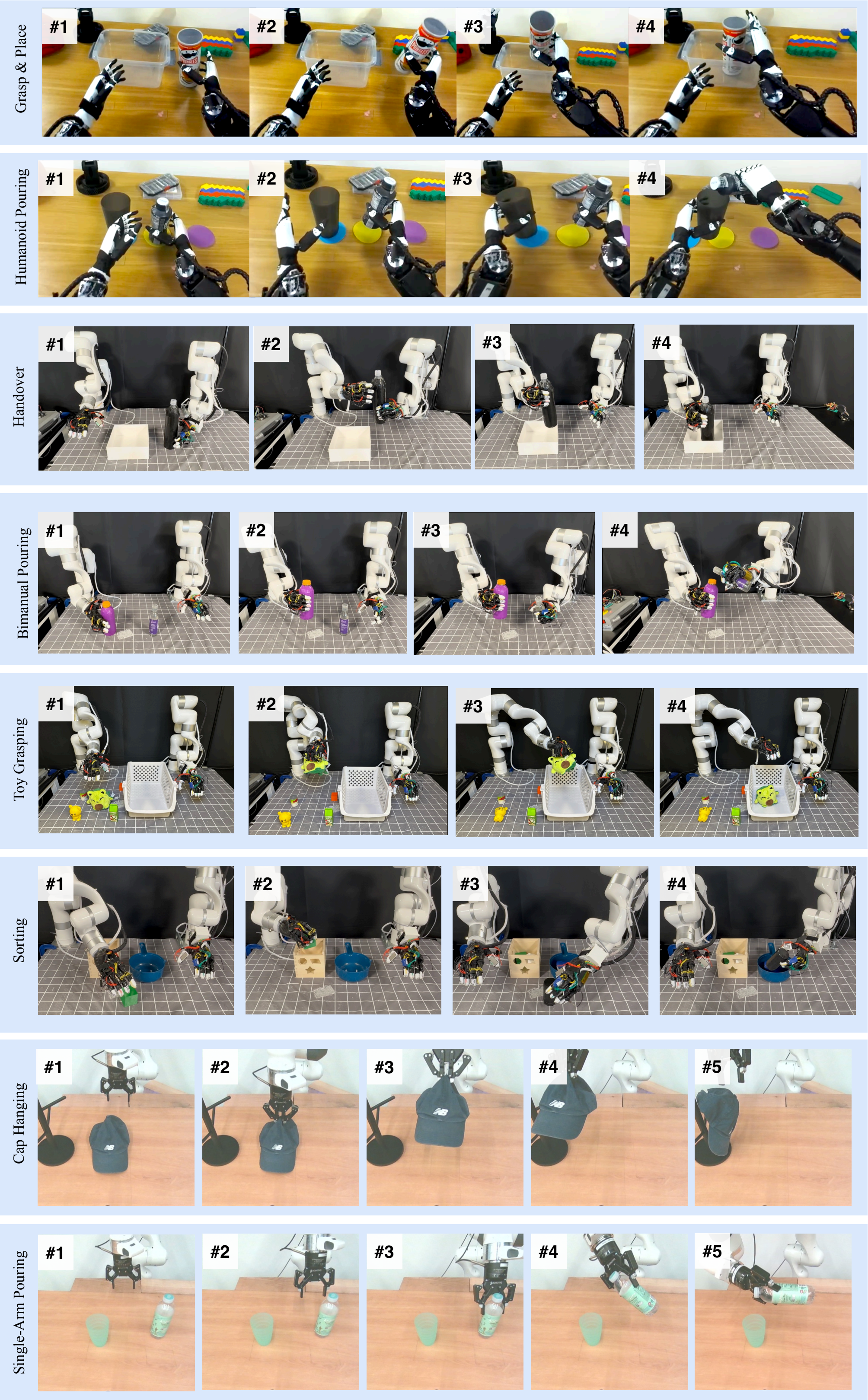


Figure 15: **Tasks Trajectories.** Illustration of the task trajectories, including Humanoid Grasp & Place, Humanoid Pouring, Bimanual Handover, Bimanual Pouring, Bimanual Toy Grasping, Single-Arm Cap Hanging, and Single-Arm Pouring.

(c) **Single-Arm Setup**: We use Oculus VR teleoperation, collecting 70–80 demonstrations per task. During data collection, objects are varied in type, location, and orientation to encourage generalization.

C.4 Evaluation Metrics

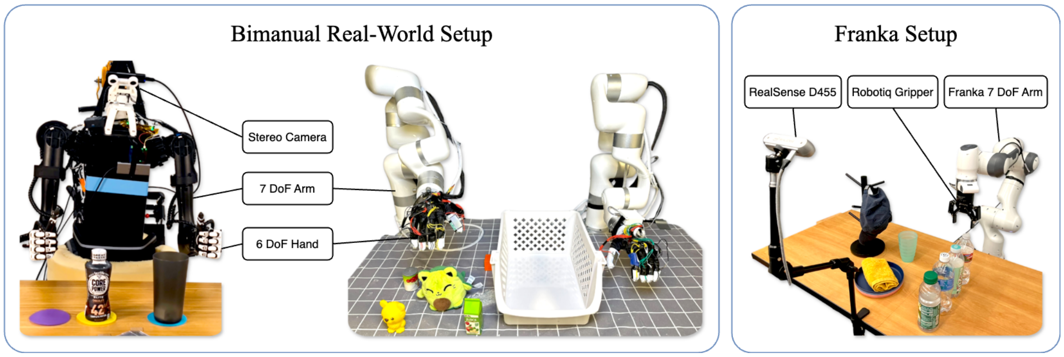
To assess generalization, we evaluate the model under the following categories:

- **Seen Object:** Using objects and configurations from the training dataset.
- **Unseen Objects:** Using novel object types not present in training.
- **Perturbations:** Including Distractors in the Scene.

Task	# of Seen Objs	# of Unseen Objs	# of Eval Trials/Obj
Humanoid Grasp & Place	4	2	10
Humanoid Pouring	2	2	10
Bimanual Handover	3	2	10
Bimanual Pouring	4	2	10
Bimanual Toy Grasping	5	3	10
Bimanual Sorting	6	4	2.8
Single-Arm Pour Water	4	2	3.3
Single-Arm Cap Hanging	2	1	5

Table 9: **Number of Seen/Unseen objects for Each Task.** In Bimanual Toy Grasping, each trial involves a mixed set of objects, so we report the average number of trials per object. For other tasks, we record the exact number of trials per object.

For both Seen and Unseen Object, we evaluate each object with the number of trials as shown in Tab. 9. Overall, except for Bimanual Sorting, we evaluated each method in 305 rollouts (29 objects with 10 trials each for the Humanoid and Bimanual settings and 15 trials total for the Single-Arm setting).



Bimanual Real-World Setup

Franka Setup

Figure 16: **Real-World Setup.** The experimental setup includes three configurations: (1) a bimanual Unitree H1 humanoid robot with 7-DoF arms, anthropomorphic hands, and a gimbal-mounted stereo camera; (2) a Bimanual 7DoF xArm setup with PSYONIC Ability Hands and an Intel RealSense L515 camera; and (3) a Franka Emika Panda robot with a Robotiq gripper and a statically positioned Intel RealSense D455 camera.

C.5 Evaluation Details for Bimanual and Humanoid tasks

Fig. 17 illustrates our sets of seen and unseen objects. (i) Bimanual Pouring involves one bottle serving as the target while another pours into it. The task demands precise grasping and rim alignment, so we choose bottles of varying sizes, shapes, and textures to evaluate the policy's generalizability. (ii) Handover requires the robot to accurately grasp and transfer bottles. Thus, we select bottles of

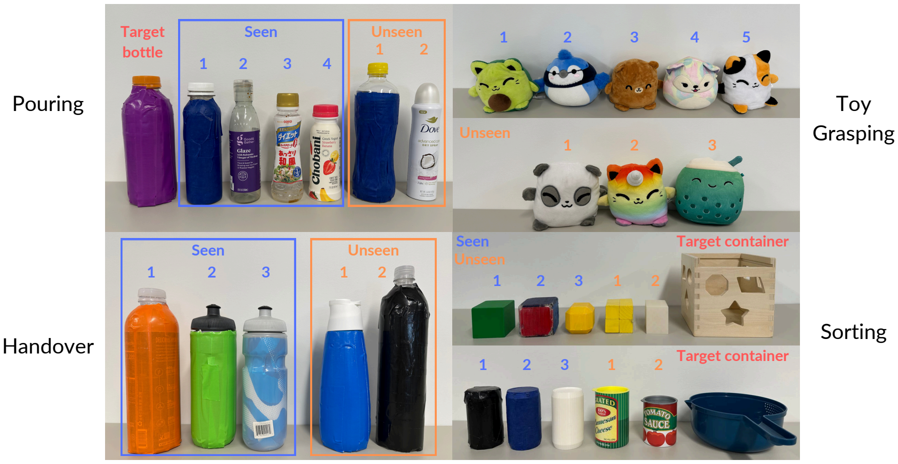


Figure 17: **Objects in Bimanual Setting.** The objects observed during the demonstration collection and the unseen objects are shown above. The objects selected represent a variety of geometries, with many differing in scale. (i) Pouring: The left hand grasps a seen or unseen bottle and performs a pouring motion above a target bottle held by the right hand. (ii) Handover: The left hand grasps a seen or unseen bottle and hands it over to the right hand. The right hand then places the bottle into a box. (iii) Toy Grasping: The right hand grasps a seen or unseen toy and places it into a basket. (iv) Sorting: The right hand sorts cubes and the left hand sorts cylinders into their respective containers.

different shapes and sizes to assess performance. (iii) Toy Grasping primarily tests the policy's spatial generalizability and its ability to operate amidst distractors. To this end, we select toys of similar sizes but diverse shapes. (iv) Sorting requires the policy to distinguish between the geometries of cubes and cylinders. We select cubes and cylinders with subtle differences in shape and scale.

Fig. 19 shows our seen and unseen objects in Humanoid setting. Grasp & Place requires accurately grasping the object and placing them into the basket, while Pouring requires the robot to grasp accurately the objects and align their poses well with the cup. We carefully select objects of varying shapes and scales, ensuring that the system encounters a wide range of object properties and tests its ability to handle different geometries and dimensions effectively.

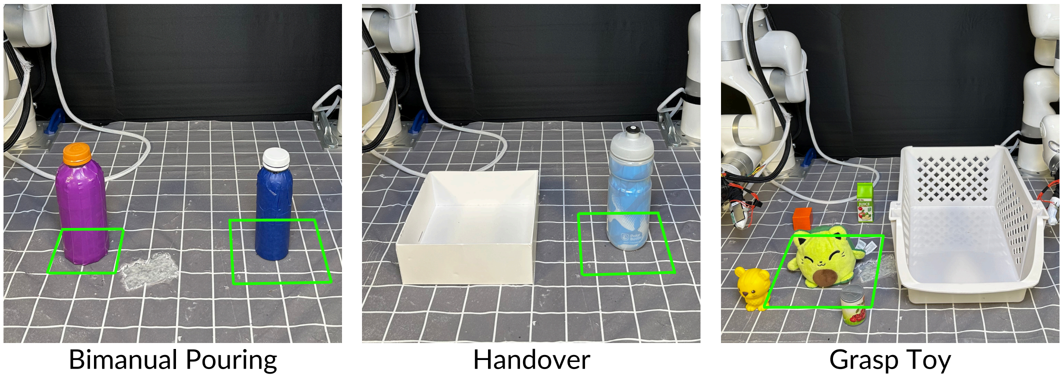
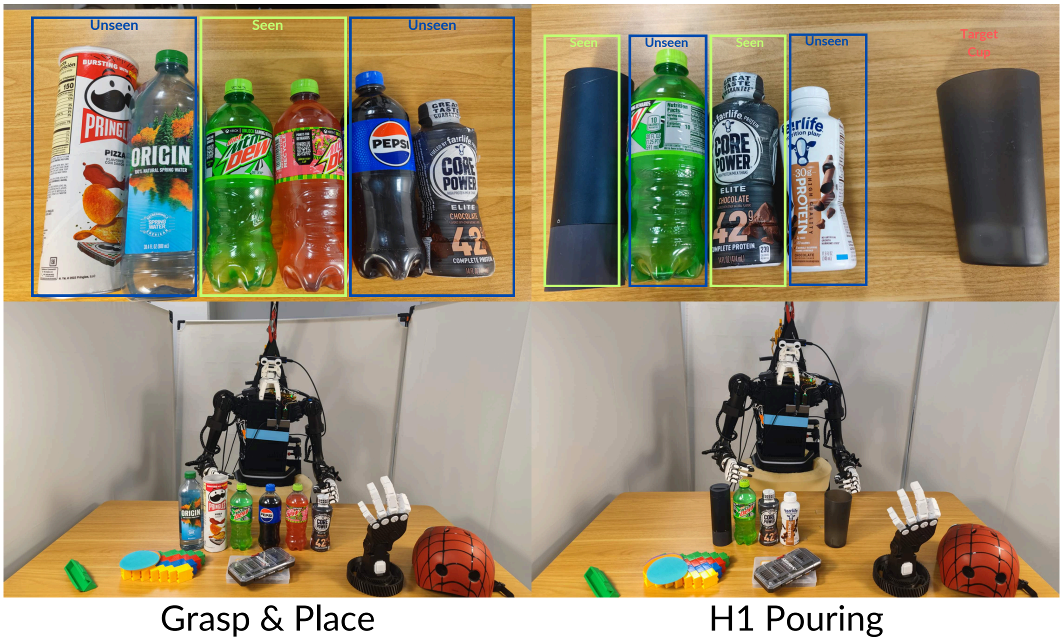

Bimanual Pouring
Handover
Grasp Toy

Figure 18: **Testing area of Bimanual Tasks.** The testing areas for our bimanual tasks are highlighted as green quadrilaterals. (i) Bimanual Pouring designates a 10.5cm × 10.5cm area for the target bottle and a 15cm × 15cm area for the pouring bottle. (ii) Handover positions the bottle within a 15cm × 15cm area, while the box may experience displacement perturbations of approximately 1.5cm in all directions. (iii) Toy Grasping places the toy within a 21cm × 21cm area, with distractors randomly arranged around it. Additionally, the basket may undergo front-back displacement perturbations of around 2.5cm in each direction.

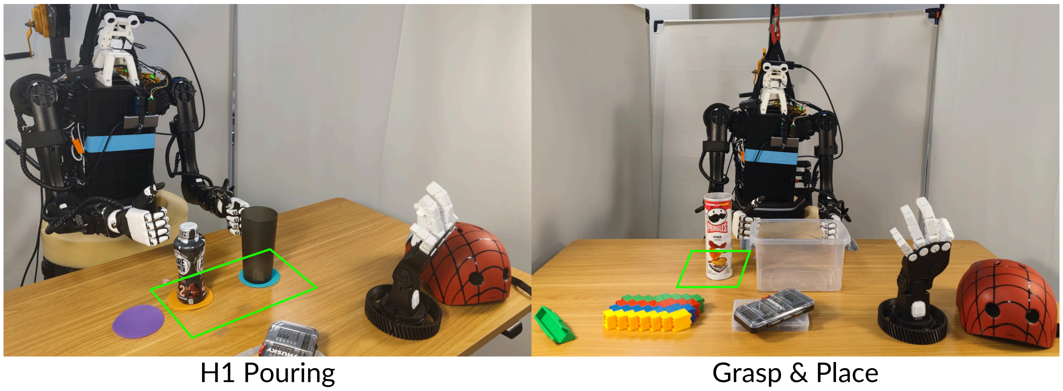
Single-Arm Setup Data Collection Details. Specific data collection details for each task are provided below:



Grasp & Place

H1 Pouring

Figure 19: **Objects in Humanoid Setting.** The seen and unseen objects in the H1 setting are shown above, along with their relative sizes compared with H1 and the experiment environment.



H1 Pouring

Grasp & Place

Figure 20: **Testing area of H1 Tasks.** The testing areas for our H1 tasks are highlighted as green quadrilaterals. (i) H1 Pouring positions the bottle and cup in front of H1, with variations in placement across different directions. (ii) Grasp & Place situates the object to one side of H1, also allowing for positional variations in multiple directions.

- **Pouring Water:** The gripper grasps a water bottle and pours water into a cup. Three types of water bottles are used. Bottle locations are randomized within a 10 cm × 20 cm area, and cup locations within an 8.5 cm × 20 cm area.
- **Cap Hanging:** Two types of caps are used. Cap locations are randomized within a 24 cm × 30 cm area, with orientations varying within 10–20 degrees.

D ManiFlow Training Algorithms

Algorithm 1 Timestep Sampling Strategies for Flow Matching

```
**Beta Sampling** ($\alpha = 1.0$, $\beta = 1.5$, $s = 0.999$):
    Sample $u \sim \text{Beta}(\alpha, \beta)$
    $t \leftarrow s \cdot u$

**Logit-Normal Sampling** ($m = 0.0$, $s = 1.0$):
    Sample $z \sim \mathcal{N}(m, s^2)$
    $t \leftarrow \frac{1}{1+e^{-z}}$

**Mode Sampling** ($s = 1.29$):
    Sample $u \sim \text{Uniform}(0, 1)$
    $t \leftarrow 1 - u - s \cdot \left(\cos^2\left(\frac{\pi u}{2}\right) - 1 + u\right)$
    $t \leftarrow \max(0, \min(1, t))$

**Cosmap Sampling**:
    Sample $u \sim \text{Uniform}(0, 1)$
    $t \leftarrow 1 - \frac{1}{\tan\left(\frac{\pi u}{2}\right)+1}$
    $t \leftarrow \max(0, \min(1, t))$
```

Algorithm 2 ManiFlow Model Training

```
**while** not converged **do**
  **Sample Data Points:**
    $x_0 \sim \mathcal{N}(0, I)$ {Sample from noise distribution}
    $x_1 \sim D$ {Sample from data distribution}
    **if** Flow Matching Training **then**
      $t \sim \text{Beta}(\alpha, \beta)$ {Sample time from Beta distribution}
      $\Delta t \leftarrow 0$ {No time step size for Flow matching training}
    **else if** Consistency Training **then**
      $t \sim \mathcal{U}\{0, \frac{1}{T}, \frac{2}{T}, \ldots, \frac{T-1}{T}\}$ {Sample time from discretized [0,1) interval}
      $\Delta t, \Delta t' \sim \mathcal{U}[0, 1]$ {Sample time interval from uniform distribution}
  **Construct Linear Interpolation Path:**
    $x_t \leftarrow (1-t)x_0 + tx_1$ {Current interpolated point}
    **if** Consistency Training **then**
      $t_1 \leftarrow t + \Delta t$ {Next time step}
      $x_{t_1} \leftarrow (1-t_1)x_0 + t_1 x_1$ {Next interpolated point}
  **Compute Target Velocities:**
    **if** Flow Matching Training **then**
      $v_{\text{target}} \leftarrow x_1 - x_0$ {Direct flow target}
    **else if** Consistency Training **then**
      $v_{t_1} \leftarrow v_{\theta^-}(x_{t_1}, t_1, \Delta t')$ {Velocity from flow EMA model}
      $\tilde{x}_1 \leftarrow x_{t_1} + v_{t_1} \cdot (1 - t_1)$ {ODE integration step}
      $v_{\text{target}} \leftarrow (\tilde{x}_1 - x_t)/(1-t)$ {Average velocity as consistency target}
  **Update Parameters:**
    $\mathcal{L} \leftarrow \|v_\theta(x_t, t, \Delta t) - v_{\text{target}}\|^2$ {Compute loss}
    $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}$ {Gradient update}
```