

# FlowPolicy: Enabling Fast and Robust 3D Flow-Based Policy via Consistency Flow Matching for Robot Manipulation

Qinglun Zhang<sup>1,2,\*</sup>, Zhen Liu<sup>1,2,\*</sup>, Haoqiang Fan<sup>2</sup>, Guanghui Liu<sup>1</sup>, Bing Zeng<sup>1</sup>, Shuaicheng Liu<sup>1,2,†</sup>

<sup>1</sup>University of Electronic Science and Technology of China

<sup>2</sup>Megvii Technology

{zhangqinglun26@std., liuzhen03@std., guanghUILIU@, eezeng@, liushuaicheng@}uestc.edu.cn, fhq@megvii.com

## Abstract

Robots can acquire complex manipulation skills by learning policies from expert demonstrations, which is often known as vision-based imitation learning. Generating policies based on diffusion and flow matching models has been shown to be effective, particularly in robotic manipulation tasks. However, recursion-based approaches are inference inefficient in working from noise distributions to policy distributions, posing a challenging trade-off between efficiency and quality. This motivates us to propose **FlowPolicy**, a novel framework for fast policy generation based on consistency flow matching and 3D vision. Our approach refines the flow dynamics by normalizing the self-consistency of the velocity field, enabling the model to derive task execution policies in a single inference step. Specifically, FlowPolicy conditions on the observed 3D point cloud, where consistency flow matching directly defines straight-line flows from different time states to the same action space, while simultaneously constraining their velocity values, that is, we approximate the trajectories from noise to robot actions by normalizing the self-consistency of the velocity field within the action space, thus improving the inference efficiency. We validate the effectiveness of FlowPolicy in Adroit and Metaworld, demonstrating a  $7\times$  increase in inference speed while maintaining competitive average success rates compared to state-of-the-art methods.

**Code** — <https://github.com/zql-kk/FlowPolicy>

## Introduction

Enabling robots to acquire diverse manipulation tasks by learning from expert demonstrations, commonly referred to as imitation learning, has been a long-standing objective in robotic learning and embodied AI (Atkeson and Schaal 1997; Argall et al. 2009; Wang et al. 2024a; Zare et al. 2024; Li et al. 2024). Prior work attempts to tackle this problem either by explicitly defining it as a supervised regression task (Zhang et al. 2018; Florence, Manuelli, and Tedrake 2019; Toyer et al. 2020; Rahmatizadeh et al. 2018), or by implicitly modeling action distributions using energy-based models (Florence et al. 2022; Jarrett, Bica, and van der

\*These authors contributed equally.

†Corresponding authors

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

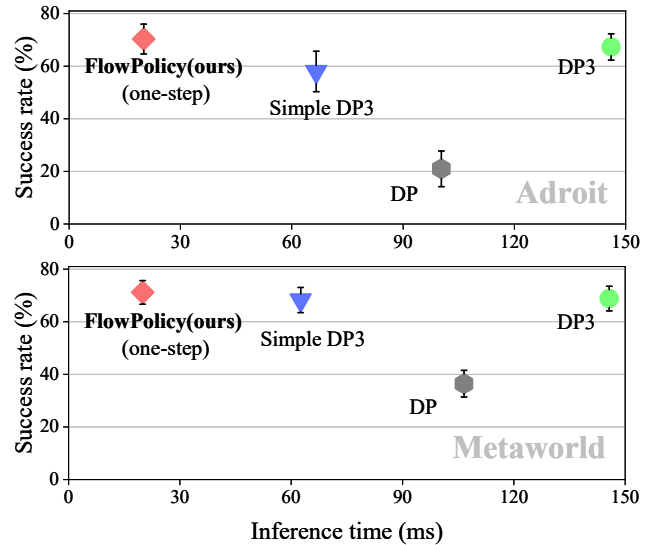


Figure 1: Comparison of FlowPolicy with 2D-based method DP (Chi et al. 2023) and 3D-based methods DP3 (Ze et al. 2024) and Simple DP3 in terms of inference time and average success rate on Adroit and Metaworld.

Schaar 2020). The former learns the mapping from observations to actions, while the latter seeks to identify the minimal energy action through energy optimization. However, explicit regression approaches are often inadequate for capturing multi-modal demonstration behaviors and struggle with high-precision tasks (Florence et al. 2022). Conversely, energy-based models face challenges with training stability, primarily due to the necessity of negative sample extraction during the training process (Chi et al. 2023).

Recent advancements in diffusion-based policy learning frameworks (Wang, Hunt, and Zhou 2022; Pearce et al. 2023; Reuss et al. 2023; Chi et al. 2023; Ze et al. 2024) have elevated this task to new heights, leveraging their powerful capability to model complex, high-dimensional robotic trajectories. Within this context, Diffusion Policy (Chi et al. 2023) defines policy learning as a conditional denoising diffusion process over the robot action space, conditioned on 2D observation features. The subsequent 3D Diffusion Pol-

icy (Ze et al. 2024), also known as DP3, further extends the conditional input to more complex 3D visual representations and is thus capable of acquiring generalizable visuomotor policies with only a few dozen expert demonstrations. Nevertheless, diffusion-based solutions are inevitably plagued by substantial runtime inefficiencies, as they typically require numerous sampling steps during inference to generate high-quality actions. While various sampling acceleration techniques (Song, Meng, and Ermon 2020; Song et al. 2023) have been proposed, the critical challenge of balancing efficiency and policy quality persists, severely limiting the practical application of these learned policies.

On the other hand, Flow Matching (FM) (Lipman et al. 2022), another class of generative model frameworks that directly defines probability paths via Ordinary Differential Equations (ODEs) to transform between noise and data samples, has shown greater numerical stability and inference efficiency than diffusion families. The latest advancement in this area, Consistency Flow Matching (Consistency-FM) (Yang et al. 2024), further improves efficiency by defining a straight-line flow from any point in time to the same endpoint, making one-step generation possible, which offers a potential direction for real-time policy generation. However, the conditional generation capability of Consistency-FM, particularly for embodied tasks involving complex 3D vision representations as conditions, remains underexplored.

In this paper, we address these challenges in policy generation by leveraging the concept of consistency flow matching, introducing a novel 3D flow-based framework for real-time robotic manipulation policy generation, named FlowPolicy. More specifically, we make the first attempt at conditional consistent flow matching in 3D robotic manipulation tasks. FlowPolicy allows for the fastest possible generation of target actions by directly defining straight-line flow conditions on the initial 3D point clouds. In addition, multi-segment training is applied to ensure the quality of action generation, aiming at a better trade-off between efficiency and effectiveness. We evaluated our framework on 37 robot manipulation tasks from Adroit and Metaworld. As illustrated in Fig. 1, our approach achieves a  $7\times$  reduction in average inference time while maintaining a competitive average success rate compared to state-of-the-art methods based on 3D visual conditional representation. By combining consistency flow matching with 3D condition-based robot manipulation, we enable the real-time decoding of high-quality robot actions with one-step inference. In summary, our main contributions are threefold:

- We first propose a 3D flow-based policy generation framework that conditions the 3D visual representation and can generate robust robotic actions with few demonstrations, namely **FlowPolicy**.
- The proposed FlowPolicy significantly narrows the gap between efficiency and quality in vision-based imitation learning algorithms, enabling **real-time policy generation** across diverse manipulation tasks.
- We conduct extensive experiments on 37 simulation tasks from Adroit and Metaworld, demonstrating that our approach achieves **state-of-the-art average success rates**.

## Related Work

### Application of Diffusion Models to 3D Robotics

Diffusion models are a family of generative models that are highly generalizable and easy to train, which are initially utilized for image generation tasks (Batzolis et al. 2021; Epstein et al. 2023). As a subversive approach, diffusion models progressively refine Gaussian noise by recursively refining it to the point where it satisfies the data distribution. Recently, it has been introduced to the field of robotics, including reinforcement learning (Römer et al. 2024), imitation learning (Wang et al. 2024b), trajectory optimization (Carvalho et al. 2023) and grasping (Urain et al. 2023), and have shown impressive performance. Denoising Diffusion Probabilistic Models (DDPMs) (Ho, Jain, and Abbeel 2020) generate a clean sample from noise by solving a Stochastic Differential Equation (SDE). However, numerical simulations of diffusion processes typically involve hundreds of steps, greatly slowing the inference speed. Denoising Diffusion Implicit Models (DDIMs) (Song, Meng, and Ermon 2020) speeds up inference by solving the inverse process as an Ordinary Differential Equation (ODE), but this sacrifices generation quality. Although adaptive ODE solvers are available, they do not meaningfully reduce the number of inference steps. In 3D robot manipulation tasks involving high-dimensional control, achieving real-time inference speed with high-quality actions using diffusion models is challenging. The primary obstacles arise from the inherent uncertainty of the diffusion process and its recursive nature.

To improve the efficiency of the diffusion process, Song (Song et al. 2023) proposes the consistency model. The consistency model directly maps noise to a data distribution by learning to map points at arbitrary moments on the probabilistic flow ODE (PF-ODE) trajectory to the origin (i.e., clean data) to support one-step or several-step generation. It has been utilized in several domains such as high-fidelity image generation (Luo et al. 2023), text-to-3D generation (Seo et al. 2023), and video editing (Chai et al. 2023). Although consistency models support one-step generation (Lu et al. 2024), they typically require distillation to achieve the desired results, which restricts their applicability in real-world robotics. Compared to the above methods, our FlowPolicy does not require numerical simulation or distillation.

### Flow Matching

Flow Matching (FM) (Lipman et al. 2022) is an emerging family of generative models based on optimal transport theory (Mozaffari et al. 2016), which aims to estimate a vector field that yields an ODE and admit the desired probabilistic path. FM allows direct regression to the goal of estimating the vector field, and hence it is simpler and more numerically stable than DDPM and DDIM. By avoiding estimating noise and instead matching a path from the noise to the target, FM enables faster inference, which is crucial in real-time robot manipulations. Previous works have demonstrated the effectiveness of FM for inverse reinforcement learning (Chang et al. 2022), multi-support motion (Rouxel et al. 2024), and robot deployment (Feng et al. 2023). However, FM-based generation of complex robot action has rarely been reported,

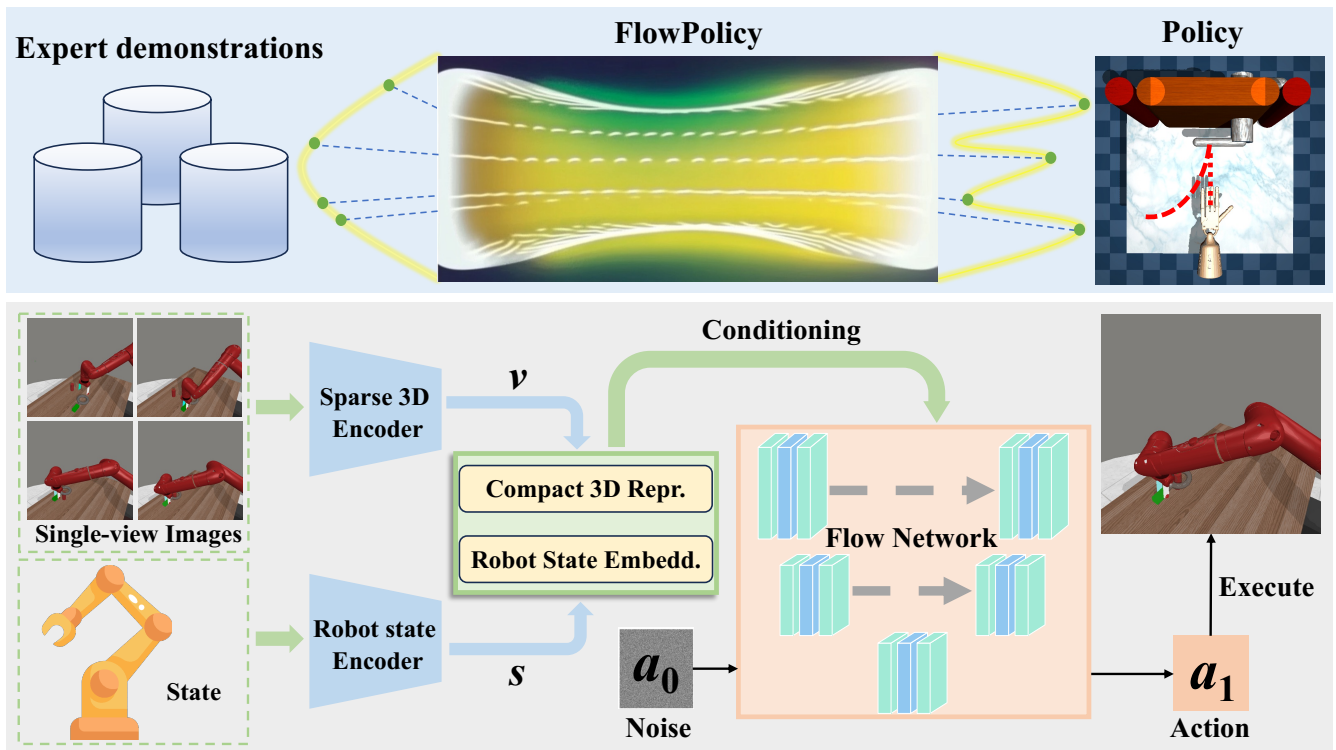


Figure 2: **Overall pipeline.** The top section visualizes FlowPolicy, where a straight-line flow enables the fastest data transition from the noise distribution to the action distribution (Adroit: Open the door). The bottom section shows the details of Flow-Policy: expert demonstrations are converted to 3D point clouds, which, along with the robot state, are encoded into compact 3D visual representations and state embeddings. A straight-line flow is then learned via conditional consistency flow matching, generating high-quality actions for tasks (Metaworld: Assembly) at real-time inference speed.

especially involving complex 3D spatial representations. In this work, we explore the feasibility of applying FM to 3D robot manipulation and hope to achieve real-time generation through consistency constraints.

Recently, Yang (Yang et al. 2024) proposed to perform consistency in velocity field space rather than sample space to improve the flow matching process (Consistency-FM). Consistency-FM directly defines a straight-line flow from any time to the same endpoint and imposes constraints on its velocity values. Also, the multi-segment training technique allows Consistency-FM to achieve a better trade-off between sampling quality and efficiency. Therefore, Consistency-FM is an elegant balance of efficiency and performance. More importantly, Consistency-FM can be trained to produce a robust flow model without the aid of distillation, which is valuable to robots performing unseen tasks, as it is difficult to find a robust teacher model in practical applications. While consistency-based models have shown promising results in some areas of embodied intelligence (Ding and Jin 2024; Chen, Li, and Zhao 2024; Prasad et al. 2024), its effectiveness has not been demonstrated on embodied tasks with complex 3D vision representation as conditions, especially for flow-based methods. To address this issue, we propose FlowPolicy, a real-time 3D policy generation framework based on consistency flow matching.

## Method

Our method expects a limited number of expert demonstrations to teach an agent to learn a policy  $\pi : O \implies A$ , i.e., mapping from visual observations  $o \in O$  to actions  $a \in A$ . Visual observations include the robot state and scene point clouds, and actions are usually sequences of trajectories of the robot to accomplish a specific task. Diffusion-based and flow-based methods can generate high-quality robot actions, but expensive iterative sampling makes them unable to accomplish real-time operations, and thus the application scenarios are very limited. While Consistency-FM balances efficiency and effectiveness, existing works have not explored the application of 3D condition generation based on consistent flow matching to robotic manipulation. Therefore, we propose FlowPolicy, a conditional consistency flow matching model, which guarantees the generation of high-quality actions while also accomplishing one-step inference for real-time applications. This section is organized as follows: we first review flow matching and consistent flow matching, and then introduce the pipeline of FlowPolicy. Finally, we describe the design details of each component.

### Consistency Flow Matching

We first introduce the notion of flow matching (Lipman et al. 2022), given a ground truth vector field  $u(t, x)$  that gener-

ates the probabilistic path  $p_t$  under two marginal constraints,  $p_{t=0} = p_0$  and  $p_{t=1} = p_1$ . The expectation is to find an ODE whose solution transmits the noise  $x_0 \sim p_0$  to the data  $x_1 \sim p_1$ :

$$\begin{cases} \frac{d\xi_x(t)}{dt} = \nu_\theta(t, \xi_x(t)) \\ \xi_x(0) = x \end{cases} \quad (1)$$

$\xi_x(t)$  is called a flow. The goal of flow matching is therefore to learn a vector field  $\nu_\theta(t, x_t)$ :

$$L_{\text{FM}}(\theta) = E_{t, p_t} \|\nu_\theta(t, x_t) - u(t, x_t)\|_2^2 \quad (2)$$

Flow matching is a simple and attractive objective. However, lack of a prior knowledge about  $u$  and  $p_t$ , conditional flow matching (Lipman et al. 2022) is typically employed to regress  $\nu_\theta(t, x_t)$  given a conditional vector field  $u(t, x_t|x_1)$  and a conditional probability path  $p_t(x_t|x_1)$ :

$$L_{\text{FM}^*}(\theta) = E_{t, q(x_1)} E_{t, (p_t|x_1)} \|\nu_\theta(t, x_t) - u(t, x_t|x_1)\|_2^2 \quad (3)$$

Where  $x_1 \sim q(x_1)$ . Learning straight-line flows enables faster inference efficiency. Consistency flow matching (Consistency-FM) (Yang et al. 2024) is a generalized method for efficiently learning straight-line flows without approximating the entire probabilistic path. Consistent velocity is defined as  $\nu(t, \xi_x(t)) = \nu(0, x)$ , indicating that the velocity along the solution of Equation 1 remains constant. Due to the complexity of the target distribution solution, Consistency-FM does not regress directly on the ground truth vector field, instead, it directly defines a velocity-consistent straight-line flow that departs at different times to the same endpoint. Consistency-FM can regress on the following objective to learn a velocity-consistent vector field:

$$\begin{aligned} L(\theta) = & E_{t \sim \mu} E_{x_t, x_{t+\Delta t}} \|f_\theta(t, x_t) - f_{\theta^-}(t + \Delta t, x_{t+\Delta t})\|_2^2 \\ & + \alpha \|\nu_\theta(t, x_t) - \nu_{\theta^-}(t + \Delta t, x_{t+\Delta t})\|_2^2 \\ & f_\theta(t, x_t) = x_t + (1 - t) * \nu_\theta(t, x_t) \end{aligned} \quad (4)$$

where  $\mu$  is a uniform distribution on  $[0, 1 - \Delta t]$ ,  $\alpha$  is a positive scalar, and  $\Delta t$  is the time interval, a small and positive scalar.  $\theta^-$  denotes the running average of past  $\theta$  values computed using an exponential moving average (EMA). Training  $L(\theta)$  not only regularizes the velocity but also learns the data distribution.

Additionally, Consistency-FM allows for multi-segment training to enhance its ability to express the distribution of transmission through  $p_t$ . Specifically, multi-segment Consistency-FM splits time into uniform subsets, normalizes a consistent vector field  $\nu_\theta^i(t, x_t)$  within each subset, and in turn constructs segmented linear trajectories. This approach relaxes the velocity consistency requirement and makes data transmission more flexible and efficient. Given the number of segments  $K$ , the training loss of segmented Consistency-FM is as follows:

$$\begin{aligned} L(\theta^*) = & E_{t \sim \mu^i} \lambda^i E_{x_t, x_{t+\Delta t}} \|f_\theta^i(t, x_t) \\ & - f_{\theta^-}^i(t + \Delta t, x_{t+\Delta t})\|_2^2 \\ & + \alpha \|\nu_\theta^i(t, x_t) - \nu_{\theta^-}^i(t + \Delta t, x_{t+\Delta t})\|_2^2 \\ & f_\theta^i(t, x_t) = x_t + ((i + 1)/k - t) * \nu_\theta^i(t, x_t) \end{aligned} \quad (5)$$

Here,  $\mu^i$  is a uniform distribution on  $[i/K, (i + 1)/K - \Delta t]$ . The moving step of each segment is controlled by a very small positive number  $\Delta t$ .  $x$  and  $x_t$  both follow a well-established distribution pattern. The flow and vector fields of each segment are described by  $\nu_\theta^i(t, x_t)$ , which depend on the parameters  $\theta$  and time  $t$ . In addition,  $\lambda^i$  is a positive weighting factor used to balance the importance of the different sections, especially since vector fields in the middle section are usually more difficult to train.

## FlowPolicy

The overall pipeline of our proposed FlowPolicy is presented in Fig. 2. FlowPolicy takes the powerful 3D diffusion policy (DP3) as the underlying architecture. DP3 is the state-of-the-art 3D vision-based conditional diffusion policy model. We address the inefficiency of 3D policy inference by constructing conditional consistency flow matching as a policy generator while ensuring the quality of task completion. Specifically, we consider the observed 3D point cloud as a visual conditional representation, which is combined with state embedding to guide the flow model to learn the manipulation policy. In the training phase, we regress the consistency vector field and learn the straight-line flow through velocity consistency loss, which directly generates the robot action sequences. Depending on the consistency training and condition representation, FlowPolicy can generate high-quality actions in one-step inference and accomplish complex manipulation tasks.

**3D visual conditional representation** FlowPolicy applies 3D point clouds as a guide to generate execution-specific policies given its rich spatial information. Specifically, we first acquire single-view depth-observation images and then convert the depth information into a point cloud via camera intrinsics and extrinsic. The single-view image is more practically applicable and the point cloud provides a richer spatial representation. Further, a farthest point sampling (FPS) (Qi et al. 2017) is applied to downsample the point cloud to preserve spatial wholeness and reduce redundancy. Finally, we employ a lightweight MLP to encode the point cloud into a compact 3D representation as the visual conditional representation for flow-based policy. With the conditional constraints of visual features, FlowPolicy enables the efficient capture of spatial information that is critical to fine robotic manipulation tasks.

**Conditional Trajectory Generation with FlowPolicy** For FlowPolicy, given the following definitions:

$$a^{\text{src}} \sim \Gamma^{\text{src}}, a^{\text{tar}} \sim \Gamma^{\text{tar}}(\cdot|s, v), t \sim \mu[\epsilon, 1 - \epsilon] \quad (6)$$

where  $\Gamma^{\text{src}}$  is the source distribution, usually described as Gaussian noise  $\Gamma^{\text{src}} = N(0, I)$ .  $\Gamma^{\text{tar}}$  is the target distribution conditioned by the state  $s$  and visual representation  $v$ .  $a^{\text{src}}$  and  $a^{\text{tar}}$  are denoted as sequences of trajectories sampled from the source and target distributions.  $t$  obeys a uniform distribution  $t \in [\epsilon, 1 - \epsilon]$ . Define  $a_t$  to be the interpolated trajectory between the source and destination trajectories at the transmission time step  $t$ .  $F$  is the flow model conditioned by the state  $s$  and visual representation  $v$ , FlowPolicy can be

Methods	NFE	Adroit			Metaworld				Average
		Hammer	Door	Pen	Easy(21)	Medium(4)	Hard(4)	Very Hard(5)	
DP	10	103.6±0.6	102.5±0.4	94.9±6.7	104.9±2.2	105.3±0.6	120.3±1.4	103.3±2.7	106.1±2.4
DP3	10	146.6±0.7	146.8±2.0	144.8±2.9	145.9±2.1	144.7±2.3	148.6±3.4	143.2±2.6	145.7±2.3
Simple DP3	10	64.5±1.4	64.2±2.4	71.2±9.3	62.7±6.2	59.4±4.1	63.0±6.8	65.3±6.7	63.0±5.9
<b>FlowPolicy (Ours)</b>	1	20.4±0.5	19.3±0.9	20.6±0.2	19.9±0.1	19.8±0.3	20.1±0.2	20.0±0.1	<b>19.9±0.2</b>

Table 1: Quantitative comparison of runtime between state-of-the-art policy models (Chi et al. 2023; Ze et al. 2024) and the proposed FlowPolicy. We evaluate 37 tasks from Adroit and Metaworld across 3 random seeds and report inference time per step (ms) with standard deviation. Our FlowPolicy enables real-time robot operations with an average time of 19.9ms in one-step inference, which is  $7 \times$  faster than DP3 and  $3 \times$  faster than its lightweight version, SimpleDP3.

Methods	NFE	Adroit			Metaworld				Average
		Hammer	Door	Pen	Easy(21)	Medium(4)	Hard(4)	Very Hard(5)	
DP	10	16±10	34±11	13±2	50.7±6.1	11.0±2.5	5.25±2.5	22.0±5.0	35.2±5.3
Adaflow*	-	45±11	27±6	18±6	49.4±6.8	12.0±5.0	5.75±4.0	24.0±4.8	35.6±6.1
CP	1	45±4	31±10	13±6	69.3±4.2	21.2±6.0	17.5±3.9	30.0±4.9	50.1±4.7
DP3	10	100±0	56±5	46±10	87.3±2.2	44.5±8.7	32.7±7.7	39.4±9.0	68.7±4.7
Simple DP3	10	98±2	40±17	36±4	86.8±2.3	42.0±6.5	38.7±7.5	35.0±11.6	67.4±5.0
<b>FlowPolicy (Ours)</b>	1	100±0	58±5	53±12	90.2±2.8	47.5±7.7	37.2±7.2	36.6±7.6	<b>70.0±4.7</b>

Table 2: Comparisons on success rate between state-of-the-art 2D-based (Chi et al. 2023; Hu et al. 2024; Prasad et al. 2024) and 3D-based (Ze et al. 2024) policy models and the proposed FlowPolicy. We evaluate 37 tasks from Adroit and Metaworld across 3 random seeds and report the success rate (%) with standard deviation. ‘\*’ indicates that the NFE of Adaflow is not fixed.

simply described as:

$$\text{FlowPolicy } F : a, t, s, v \longrightarrow \Delta a \quad (7)$$

To train a high-quality one-step generator, we employ a two-segment Consistency-FM and use backpropagation to minimize the following equation:

$$\begin{aligned}
L_{\text{FlowPolicy}}(\theta) = & E_{t \sim \mu^i} \lambda^i E_{a_t, a_t + \Delta t} \|f_{\theta}^i(t, a_t, s, v) \\
& - f_{\theta}^i(t + \Delta t, a_t + \Delta t, s, v)\|_2^2 \\
& + \alpha \| \nu_{\theta}^i(t, a_t, s, v) - \nu_{\theta}^i(t + \Delta t, a_t + \Delta t, s, v) \|_2^2 \\
f_{\theta}^i(t, a_t, s, v) = & a_t + (i + 1)/k - t * \nu_{\theta}^i(t, a_t, s, v) \quad (8)
\end{aligned}$$

By training the two-segment loss above, two consistent vector fields can be learned, which in turn constructs a two-segment linear trajectory that transforms the noisy trajectory into the target trajectory. In the inference process, the noise trajectories are sampled from the source distribution  $a_0 \in a^{\text{src}}$  and then predicted by a flow model to obtain action trajectories  $a_1 \in a^{\text{tar}}$ .

FlowPolicy provides a framework for generating high-quality actions in a one-step process that promotes a trade-off between efficiency and effectiveness.

## Experiments

### Dataset and Implementation Details

**Simulation Benchmarks** We choose two preeminent environmental simulators, Adroit (Rajeswaran et al. 2018) and Metaworld (Yu et al. 2021), as our benchmarks for a comprehensive set of 37 tasks. The Adroit benchmark presents three challenges of dexterous manipulation tasks, where the execution of complex tasks is facilitated by the intricate control of a high-dimensional, articulated hand. In contrast, the

Metaworld benchmark offers a diverse array of tasks that span the spectrum of difficulty levels, from easy to very hard, typically surmounted with the aid of a two-fingered robotic arm. For the procurement of expert demonstrations, we have harnessed a well-trained heuristic strategy, designed to yield datasets of exceptional quality.

**Baselines** The main objective of this work is to leverage 3D point clouds as visual conditional representations, aiming to elevate the robot manipulation to the real-time level via a consistency flow matching strategy. Therefore, we focus on comparing DP3, a state-of-the-art 3D conditional diffusion-based strategy model. In addition, its lightweight version, simple DP3, is also compared. We also compared state-of-the-art 2D-based approaches, including diffusion policy (DP) (Chi et al. 2023), Adaflow (Hu et al. 2024), and Consistency Policy (CP) (Prasad et al. 2024) and collected the same viewpoint images for them.

**Evaluation Metrics** We evaluate 20 episodes per 200 training epochs and record the mean of the five highest success rates. Each task is run repeatedly under three different random seeds, and their means and variances as well as inference times are calculated. In addition, the Number of Function Evaluations (NFE) is evaluated as a metric of inference efficiency. The higher the success rate, the better, and the lower the NFE and inference time, the better.

**Implementation Details** For each task, we generate a compact conditional embedding consisting of a 3D point cloud and the robot’s state with a dimension of 64. Additionally, our model employs an observation horizon of two steps, signifying that it leverages the point clouds from the two most recent time frames as conditional inputs for policy generation. We assign ten expert demonstrations to each

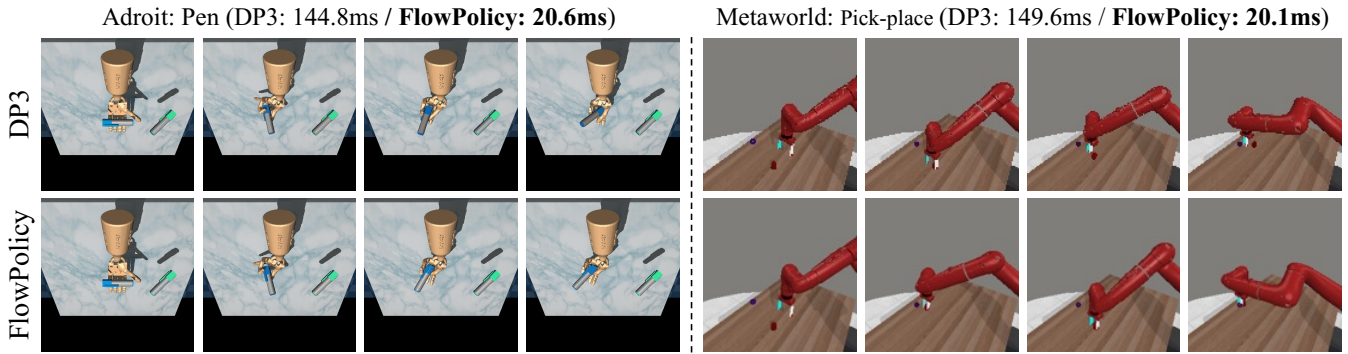


Figure 3: Qualitative Comparison of FlowPolicy and DP3 (Ze et al. 2024) on two challenging manipulation tasks from Adroit and Metaworld. Our method successfully generates high-quality actions at real-time speeds, completing these tasks effectively, whereas DP3 either produces lower-quality actions (left) or fails to complete the task (right).

task. The image size is cropped to a resolution of  $84 \times 84$  pixels, and the point cloud is downsampled to 512 points for Adroit or 1024 points for Metaworld using FPS. During the training phase, the weights are updated using the AdamW optimizer with a learning rate of  $1e-4$  and a batch size of 128. The training process is iterated for 3,000 epochs. EMA decay rate is set to the default value of 0.95. Additionally, both action and state inputs are normalized to the interval  $[-1, 1]$  to facilitate the training process, with actions being unnormalized before execution. Our model and all compared methods are developed based on the PyTorch framework and evaluated on a single NVIDIA RTX 2080Ti GPU.

## Comparison with State-of-the-art Methods

**Quantitative Comparisons on Runtime** Table 1 reports the average inference time of each model for 3 dexterity tasks from Adroit and 34 different difficulty tasks from Metaworld. Our FlowPolicy demonstrates remarkable results by reducing the average inference time to 19.9ms, improving by more than  $7 \times$  (19.9ms vs. 145.7ms) compared to the DP3 model. Even for the lightweight version of Simple DP3, FlowPolicy enhances the inference efficiency by over  $3 \times$  (19.9ms vs. 63.0ms). The consistent velocity field provides a more direct linear path, enabling faster and more efficient action generation. Notably, our approach exhibits less variation in inference time. Specifically, FlowPolicy has less fluctuation in the average runtime for both individual and overall tasks, with a potential explanation attributed to the deterministic sampling policy of consistent flow matching. The results show that faster ODE trajectories can be determined and the inference efficiency can be improved by directly defining a straight-line flow, which is crucial for realizing real-time robot operation.

**Quantitative Comparisons on Success Rate** Further, Table 2 shows the average success rate of each model on 37 tasks in Adroit and Metaworld. 3D-based baselines generally outperform 2D-based ones, owing to the richer spatial representation offered by point clouds. Among 3D-based baselines, our approach achieves the best success rate with the shortest inference time. Specifically, FlowPolicy’s over-

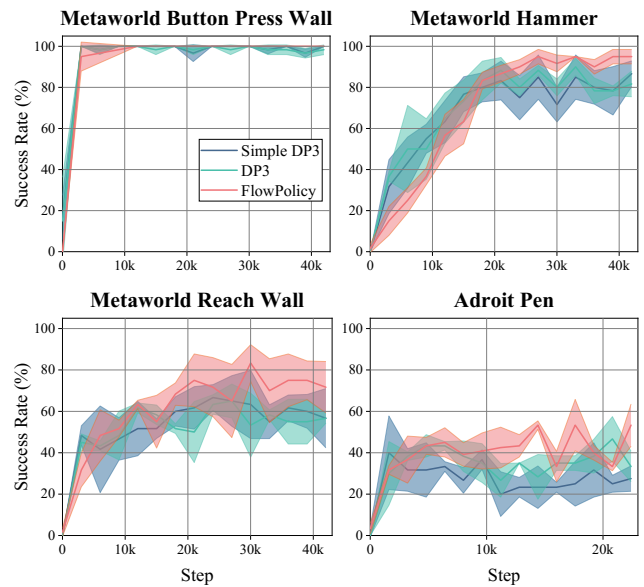


Figure 4: Illustrations of the learning curves. Compared to Simple DP3 and DP3, FlowPolicy demonstrates higher stability, learning efficiency, and success rates.

all success rate is 1.3% higher than DP3 and 2.6% higher than Simple DP3 with the same expert demonstration. The results show that FlowPolicy can achieve competitive results with state-of-the-art methods in just one step of inference. This is attributed to conditional trajectory generation based on consistent flow matching. On the one hand, the compact 3D representation leads to a richer spatial representation of the flow model. On the other hand, directly defining the straight-line flow motivates faster and simpler ODE trajectories to the extent that they can be generated in one step. Finally, multi-segment training ensures a trade-off between generation quality and efficiency. As a result, FlowPolicy decodes high-quality robot actions in just one step of inference, which demonstrates the effectiveness of consistency flow matching in robot manipulation tasks.

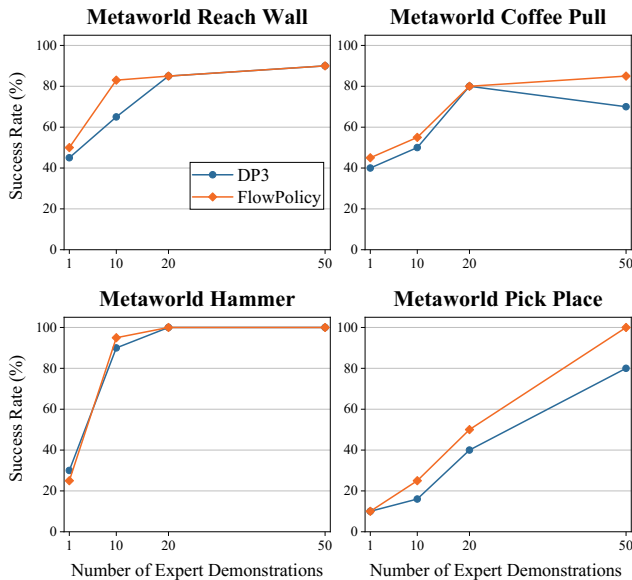


Figure 5: Ablation on the number of expert demonstrations. We choose four typical tasks to explore the impact of different numbers of demonstrations on FlowPolicy and DP3. Both generally improve the accuracy with more demonstrations, but FlowPolicy typically has a higher success rate and avoids the performance bottleneck as presented in DP3.

**Qualitative Comparisons on Manipulation Tasks** As illustrated in Figure 3, we selected two typical qualitative examples in Adroit and Metaworld, generated by DP3 and FlowPolicy, respectively. From left to right, this sequence delineates the holistic progression of the task’s execution, with each frame of the sequence corresponding to a distinct phase in the operation, as performed by the agent. On the left is a ‘Pen-turning’ task involving complex dexterity manipulation, which requires turning the blue-capped pen in the dexterous hand to a specific position (i.e., the green-capped pen serves as the reference angle). Although DP3 accomplishes the dexterity task, the diffusion policy generated based on DP3 fails to ensure consistency with the target pen in 3D space compared to FlowPolicy. Specifically, DP3 turns one end of the pen to a state so high that it is darker in the camera view. In contrast, FlowPolicy generates higher-quality action sequences to turn the pen to the most appropriate angle. More importantly, FlowPolicy gets closer to the target position faster at the same frame. The right case is a difficult ‘Pick-Place’ task in Metaworld, which requires the use of parallel gripping clips to carry a red cube to a blue target location. DP3 unsuccessfully picks up the red cube and fails the task. Instead, FlowPolicy correctly locates the position of the red cube and successfully carries it to the target location. The above findings show that our FlowPolicy not only accomplishes the tasks that DP3 cannot accomplish well, but also achieves a higher quality and speed of accomplishment than DP3. This further demonstrates that our proposed FlowPolicy has a more delicate understanding and generative ability in robot manipulation tasks.

**Analysis of Learning Curves** Similarly, we selected four representative tasks in Adroit and Metaworld and visualized their learning curves, as illustrated in Figure 4. Although we defined 3000 epochs for each task, most of the tasks converged and leveled off at 500–1000 epochs. For the simple-level task ‘Button Press Wall’, all three models learned the correct policy easily and converged with a small number of iterations. Although Simple DP3 and DP3 have high success rates, their success rates fluctuate as training progresses. On the contrary, our FlowPolicy learns the button policy well and maintains it, demonstrating the stability of FlowPolicy. For another simple level task ‘Reach Wall’ and a medium difficulty task ‘Hammer’, all three models learn faster, but Simple DP3 and DP3 reach a performance bottleneck, while our method achieves a higher task success rate. The learning curves for the dexterity task in Adroit show similar conclusions. The above results show that FlowPolicy can learn policies faster and better, and demonstrates certain stability.

### Ablation Studies on Expert Demonstrations

The success rate of the agent in accomplishing tasks depends on the number and quality of expert demonstrations, where the quality of demonstrations is ensured by a heuristic algorithm. We further conduct ablation studies to verify the influence of the number of expert demonstrations. As illustrated in Figure 5, the success rate variation curves for typical tasks with different number of demonstrations. For non-hard tasks (i.e., ‘Reach Wall’, ‘Hammer’), FlowPolicy can outperform DP3 with a limited number of presentations. For hard-level tasks (i.e., ‘Pick-Place’), the success rate of the task can be significantly improved by increasing the number of expert presentations, as shown in both FlowPolicy and DP3. However, DP3 reaches performance saturation more easily, which even leads to a decrease in the success rate when the number of demonstrations is sufficient (i.e., ‘Coffee Pull’), while FlowPolicy performs better. The experiments reveal a better trade-off between the number of expert demonstrations and the success rate of the proposed FlowPolicy, i.e., the ability to learn effective data distributions from a small amount of data and to break through performance bottlenecks when the amount of data is sufficient.

### Conclusion

In this work, we have introduced FlowPolicy, a novel 3D flow-based visual imitation learning algorithm. It can be summarized as a policy-generating model based on 3D point cloud and consistency flow matching. For the first time, we have explored the challenge of conditional generation within consistency flow matching, particularly under complex 3D visual representations. FlowPolicy accelerates the transfer of data from the noise to the action space by defining straight-line flows, thereby improving the inference efficiency of robotic manipulation tasks at the real-time level. Evaluations across 37 tasks of diverse challenges on Adroit and Metaworld platforms demonstrate that FlowPolicy achieves competitive success rates with real-time inference speed. We hope our work will advance 3D visual-based imitation learning towards practical applications.

## Acknowledgments

This work was supported in part by National Natural Science Foundation of China (NSFC) under Grant Nos. 62372091, 62071097 and in part by Sichuan Science and Technology Program under Grant Nos. 2023NSFSC0462, 2023NSFSC0458, 2023NSFSC1972.

## References

- Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5): 469–483.
- Atkeson, C. G.; and Schaal, S. 1997. Robot learning from demonstration. In *Proc. ICML*, volume 97, 12–20.
- Batzolis, G.; Stanczuk, J.; Schönlieb, C.-B.; and Etmann, C. 2021. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*.
- Carvalho, J.; Le, A. T.; Baiert, M.; Koert, D.; and Peters, J. 2023. Motion planning diffusion: Learning and planning of robot motions with diffusion models. In *Proc. IROS*, 1916–1923.
- Chai, W.; Guo, X.; Wang, G.; and Lu, Y. 2023. Stablevideo: Text-driven consistency-aware diffusion video editing. In *Proc. ICCV*, 23040–23050.
- Chang, W.-D.; Higuera, J. C. G.; Fujimoto, S.; Meger, D.; and Dudek, G. 2022. Il-flow: Imitation learning from observation using normalizing flows. *arXiv preprint arXiv:2205.09251*.
- Chen, Y.; Li, H.; and Zhao, D. 2024. Boosting Continuous Control with Consistency Policy. *arXiv preprint arXiv:2310.06343*.
- Chi, C.; Feng, S.; Du, Y.; Xu, Z.; Cousineau, E.; Burchfiel, B.; and Song, S. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*.
- Ding, Z.; and Jin, C. 2024. Consistency Models as a Rich and Efficient Policy Class for Reinforcement Learning. *arXiv preprint arXiv:2309.16984*.
- Epstein, D.; Jabri, A.; Poole, B.; Efros, A.; and Holynski, A. 2023. Diffusion self-guidance for controllable image generation. In *Proc. NeurIPS*, volume 36, 16222–16239.
- Feng, J.; Lee, J.; Geisler, S.; Gunnemann, S.; and Triebel, R. 2023. Topology-Matching Normalizing Flows for Out-of-Distribution Detection in Robot Learning. *arXiv preprint arXiv:2311.06481*.
- Florence, P.; Lynch, C.; Zeng, A.; Ramirez, O. A.; Wahid, A.; Downs, L.; Wong, A.; Lee, J.; Mordatch, I.; and Tompson, J. 2022. Implicit behavioral cloning. In *Conference on Robot Learning*, 158–168. PMLR.
- Florence, P.; Manuelli, L.; and Tedrake, R. 2019. Self-supervised correspondence in visuomotor policy learning. *IEEE Robotics and Automation Letters*, 5(2): 492–499.
- Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. In *Proc. NeurIPS*, volume 33, 6840–6851.
- Hu, X.; Liu, B.; Liu, X.; and Liu, Q. 2024. AdaFlow: Imitation Learning with Variance-Adaptive Flow-Based Policies. *arXiv preprint arXiv:2402.04292*.
- Jarrett, D.; Bica, I.; and van der Schaar, M. 2020. Strictly batch imitation learning by energy-based distribution matching. In *Proc. NeurIPS*, volume 33, 7354–7365.
- Li, S.; Wang, X.; Zuo, R.; Sun, K.; Cui, L.; Ding, J.; Liu, P.; and Ma, Z. 2024. Robust Visual Imitation Learning with Inverse Dynamics Representations. In *Proc. AAAI*, 13609–13618.
- Lipman, Y.; Chen, R. T.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2022. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.
- Lu, G.; Gao, Z.; Chen, T.; Dai, W.; Wang, Z.; and Tang, Y. 2024. ManiCM: Real-time 3D Diffusion Policy via Consistency Model for Robotic Manipulation. *arXiv preprint arXiv:2406.01586*.
- Luo, S.; Tan, Y.; Huang, L.; Li, J.; and Zhao, H. 2023. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*.
- Mozaffari, M.; Saad, W.; Bennis, M.; and Debbah, M. 2016. Optimal transport theory for power-efficient deployment of unmanned aerial vehicles. In *Proc. ICC*, 1–6. IEEE.
- Pearce, T.; Rashid, T.; Kanervisto, A.; Bignell, D.; Sun, M.; Georgescu, R.; Macua, S. V.; Tan, S. Z.; Momennejad, I.; Hofmann, K.; et al. 2023. Imitating human behaviour with diffusion models. *arXiv preprint arXiv:2301.10677*.
- Prasad, A.; Lin, K.; Wu, J.; Zhou, L.; and Bohg, J. 2024. Consistency Policy: Accelerated Visuomotor Policies via Consistency Distillation. *arXiv preprint arXiv:2405.07503*.
- Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *arXiv preprint arXiv:1706.02413*.
- Rahmatizadeh, R.; Abolghasemi, P.; Bölöni, L.; and Levine, S. 2018. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. In *Proc. ICRA*, 3758–3765.
- Rajeswaran, A.; Kumar, V.; Gupta, A.; Vezzani, G.; Schulman, J.; Todorov, E.; and Levine, S. 2018. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. *arXiv preprint arXiv:1709.10087*.
- Reuss, M.; Li, M.; Jia, X.; and Lioutikov, R. 2023. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*.
- Rouxel, Q.; Ferrari, A.; Ivaldi, S.; and Mouret, J.-B. 2024. Flow Matching Imitation Learning for Multi-Support Manipulation. *arXiv preprint arXiv:2407.12381*.
- Römer, R.; Brunke, L.; Schuck, M.; and Schoellig, A. P. 2024. Safe Offline Reinforcement Learning using Trajectory-Level Diffusion Models. In *Proc. ICRA Workshop*.
- Seo, J.; Jang, W.; Kwak, M.-S.; Kim, H.; Ko, J.; Kim, J.; Kim, J.-H.; Lee, J.; and Kim, S. 2023. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. *arXiv preprint arXiv:2303.07937*.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

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

Toyer, S.; Shah, R.; Critch, A.; and Russell, S. 2020. The magical benchmark for robust imitation. In *Proc. NeurIPS*, volume 33, 18284–18295.

Urain, J.; Funk, N.; Peters, J.; and Chalvatzaki, G. 2023. Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *Proc. ICRA*, 5923–5930.

Wang, B.; Wu, G.; Pang, T.; Zhang, Y.; and Yin, Y. 2024a. DiffAIL: Diffusion Adversarial Imitation Learning. In *Proc. AAAI*, 15447–15455.

Wang, C.; Fang, H.; Fang, H.-S.; and Lu, C. 2024b. RISE: 3D Perception Makes Real-World Robot Imitation Simple and Effective. *arXiv preprint arXiv:2404.12281*.

Wang, Z.; Hunt, J. J.; and Zhou, M. 2022. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*.

Yang, L.; Zhang, Z.; Zhang, Z.; Liu, X.; Xu, M.; Zhang, W.; Meng, C.; Ermon, S.; and Cui, B. 2024. Consistency Flow Matching: Defining Straight Flows with Velocity Consistency. *arXiv preprint arXiv:2407.02398*.

Yu, T.; Quillen, D.; He, Z.; Julian, R.; Narayan, A.; Shively, H.; Bellathur, A.; Hausman, K.; Finn, C.; and Levine, S. 2021. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. *arXiv preprint arXiv:1910.10897*.

Zare, M.; Kebria, P. M.; Khosravi, A.; and Nahavandi, S. 2024. A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Trans. on Cybernetics*.

Ze, Y.; Zhang, G.; Zhang, K.; Hu, C.; Wang, M.; and Xu, H. 2024. 3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations. In *Proc. RSS*.

Zhang, T.; McCarthy, Z.; Jow, O.; Lee, D.; Chen, X.; Goldberg, K.; and Abbeel, P. 2018. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *Proc. ICRA*, 5628–5635.