Diffusion Policy: Visuomotor Policy Learning via Action Diffusion

Abstract

This paper introduces Diffusion Policy, a new way of generating robot behavior by representing a robot’ s visuomotor policy as a conditional denoising diffusion process. Diffusion Policy learns the gradient of the action-distribution score function and iteratively optimizes with respect to this gradient field during inference via a series of stochastic Langevin dynamics steps. We find that the diffusion formulation yields powerful advantages when used for robot policies, including gracefully handling multimodal action distributions, being suitable for high-dimensional action spaces, and exhibiting impressive training stability. To fully unlock the potential of diffusion models for visuomotor policy learning on physical robots, this paper presents a set of key technical contributions including the incorporation of receding horizon control, visual conditioning, and the time-series diffusion transformer. We hope this work will help motivate a new generation of policy learning techniques that are able to leverage the powerful generative modeling capabilities of diffusion models. We applied both Unet-based and Transformer-based diffusion models to Metaworld benchmark respectively. And their effectiveness in robotic tasks was successfully verified.

Keywords: Diffusion model, Imitation learning, Manipulation.

1 Introduction

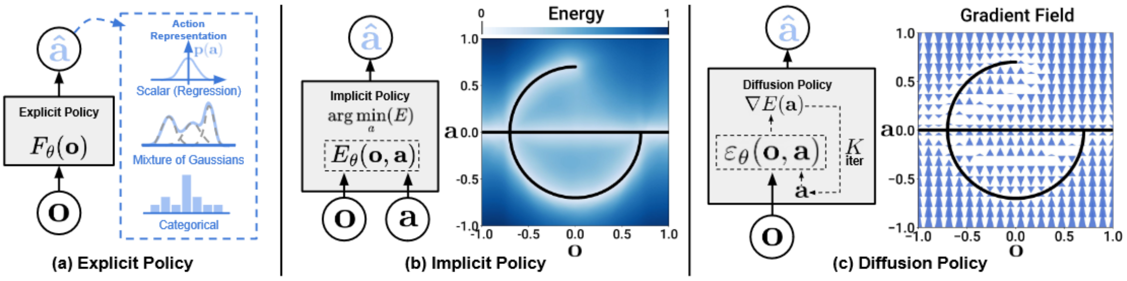
Policy learning from demonstration, in its simplest form, can be formulated as the supervised regression task of learning to map observations to actions – such as the existence of multimodal distributions, sequential correlation, and the requirement of high precision – makes this task distinct and challenging compared to other supervised learning problems.

In this work, we seek to address this chanllenge by introducing a new form of robot visuomotor policy that generates behavior via a ”conditional denoising diffusion process [6] on robot action space”, Diffusion Policy. In this formulation, instead of directly outputting an action, the policy infers the action-score gradient, conditioned on visual observations, for K denoising iterations(Fig.1 c). This formulation allows robot policies to inherit several key properties from diffusion models – significantly improving performance.

- **Expressing multimodal action distributions.** By learning the gradient of the action score function [14] and performing Stochastic Langevin Dynamics sampling on this gradient field, Diffusion policy can express arbitrary normalizable distributions [11], which includes multimodal action distributions, a well-known challenge for policy learning.

- **High-dimensional output space.** As demonstrated by their impressive image generation results, diffusion models have shown excellent scalability to highdimension output spaces. This property allows the policy to jointly infer a sequence of future actions instead of single-step actions, which is critical for encouraging temporal action consistency and avoiding myopic planning.

- **Stable training.** Training energy-based policies often requires negative sampling to estimate an intractable normalization constant, which is known to cause training instability [5]. Diffusion Policy bypasses this requirement by learing the gradient of the energy function and thereby achieves stable training while maintaining distributional expressvity.



(a) Explicit Policy

(b) Implicit Policy

(c) Diffusion Policy

Figure 1. Overview of Diffusion Policy

Our **primary contribution** is to bring the above advantages to the field of robotics and demonstrate their effectiveness on complex real-world robot manipulation tasks. To successfully employ diffusion models for visuomotor policy learning, we present the following technical contributions that enhance the performance of Diffusion Policy and unlock its full potential on physical robots:

- **Closed-loop action sequences.** We combine the policy ' s capability to predict high-dimensional action sequences with receding-horizon control to achieve robust execution. This design allows the policy to continuously re-plan its action in a closed-loop manner while maintaining temporal action consistency achieving a balance between long-horizon planning and responsiveness.

- **Visual conditioning.** We introduce a visionconditioned diffusion policy, where the visual observations are treated as conditioning instead of a part of the joint data distribution. In this formulation, the policy extracts the visual representation once regardless of the denoising iterations, which drastically reduces the computation and enables real-time action inference.

- **Time-series diffusion transformer.** We propose a new transformer-based diffusion network that minimizes the over-smoothing effects of typical CNN-based models and achieves state-of-the-art performance on tasks that require high-frequency action changes and velocity control.

2 Related works

Creating capable robots without requiring explicit programming of behaviors is a longstanding challenge in the field [2]; [1]; [12].While conceptually simple, behavior cloning has shown surprising promise on an array of real-world robot tasks, including manipulation [9]; [18]; [3] and autonomous driving [4]. Current behavior cloning approaches can be categorized into two groups, depending on the policy’ s structure.

2.1 Explicit Policy

The simplest form of explicit policies maps from world state or observation directly to action [19]; [15]; [4]. They can be supervised with a direct regression loss and have efficient inference time with one forward pass. Unfortunately, this type of policy is not suitable for modeling multi-modal demonstrated behavior, and struggles with high-precision tasks [5]. A popular approach to model multimodal action distributions while maintaining the simplicity of direction action mapping is convert the regression task into classification by discretizing the action space [18]; [17]; [3]. However, the number of bins needed to approximate a continuous action space grows exponentially with increasing dimensionality. Another approach is to combine Categorical and Gaussian distributions to represent continuous multimodal distributions via the use of MDNs [10] or clustering with offset prediction [13]. Nevertheless, these models tend to be sensitive to hyperparameter tuning, exhibit mode collapse, and are still limited in their ability to express high-precision behavior [5]

2.2 Implicit policy

Implicit policies [5] define distributions over actions by using Energy-Based Models (EBMs) [8]. In this setting, each action is assigned an energy value, with action prediction corresponding to the optimization problem of finding a minimal energy action. Since different actions may be assigned low energies, implicit policies naturally represent multi-modal distributions. However, existing implicit policies [5] are unstable to train due to the necessity of drawing negative samples when computing the underlying Info-NCE loss.

3 Method

3.1 Overview

The overview of diffusion policy is as shown in Fig. 2: a) General formulation. At time step t, the policy takes the lastest T_o steps of observation data O_t as input and outputs T_a steps of actions A_t. b) In the CNN-based Diffusion Policy, FiLM (Feature-wise Linear Modulation) conditioning of the observation feature O_t is applied to every convolution layer, channel-wise. Starting from A_t^K drawn from Gaussian noise, the output of noise-prediction network ϵ_θ is subtracted, repeating K times to get A_t^0, the denoised action sequence. c) In the Transformer-based Diffusion Policy, the embedding of observation O_t is passed into a multi-head cross-attention layer of each transformer decoder block. Each action embedding is constrained to only attend to itself and previous action embeddings (causal attention) using the attention mask illustrated.

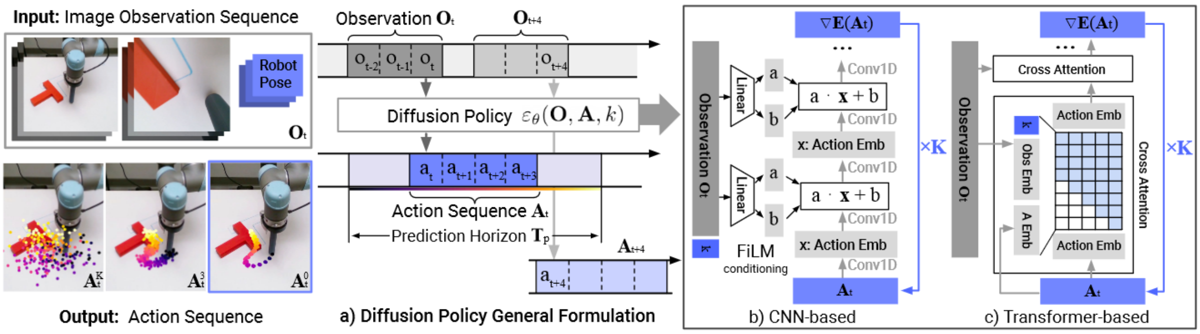

Input: Image Observation Sequence
Robot Pose
$\mathbf{O}_t$
$\mathbf{A}_t^K$
$\mathbf{A}_t^3$
$\mathbf{A}_t^0$
Output: Action Sequence
Observation $\mathbf{O}_t$
$\mathbf{O}_{t+4}$
o_{t-2} o_{t-1} o_t
o_{t+4}
Diffusion Policy $\varepsilon_\theta(\mathbf{O}, \mathbf{A}, k)$
a_t a_{t+1} a_{t+2} a_{t+3}
Action Sequence $\mathbf{A}_t$
Prediction Horizon $\mathbf{T}_p$
$\mathbf{A}_{t+4}$
a_{t+4}
a) Diffusion Policy General Formulation
$\nabla\mathbf{E}(\mathbf{A}_t)$
Observation $\mathbf{O}_t$
Linear
a
b
Conv1D
$a \cdot \mathbf{x} + b$
x: Action Emb
FiLM conditioning
×K
$\mathbf{A}_t$
b) CNN-based
$\nabla\mathbf{E}(\mathbf{A}_t)$
Cross Attention
Action Emb
Obs Emb
A Emb
Cross Attention
×K
$\mathbf{A}_t$
c) Transformer-based

Figure 2. Overview of Diffusion Policy

3.2 Denoising Diffusion Probabilistic Models

We formulate visuomotor robot policies as Denoising Diffusion Probabilistic Models (DDPMs) [6]. Crucially, Diffusion policies are able to express complex multimodal action distributions and possess stable training behavior –requiring little task-specific hyperparameter tuning. The following sections describe DDPMs in more detail and explain how they may be adapted to represent visuomotor policies.

DDPMs are a class of generative model where the output generation is modeled as a denoising process, often called Stochastic Langevin Dynamics [16].

Starting from x^K sampled from Gaussian noise, the DDPM performs K iterations of denoising to produce a series of intermediate actions with decreasing levels of noise, $x^k, x^{k-1}...x^0$, until a desired noise-free output x^0 is formed. The process follows the equation

$$\mathbf{x}^{k-1} = \alpha(\mathbf{x}^k - \gamma\epsilon_\theta(\mathbf{x}^k, k) + \mathcal{N}(0, \sigma^2\mathbf{I})), \tag{1}$$

where ϵ_θ is the noise prediction network with parameters θ that will be optimized through learning and $\mathcal{N}(0, \sigma^2\mathbf{I})$ is Gaussian noise added at each iteration.

The above equation 1 may also be interpreted as a single noisy gradient descent step:

$$x' = x - \gamma\nabla E(x) \tag{2}$$

where the noise prediction network $\epsilon_\theta(x, k)$ effectively predicts the gradient field $\nabla E(x)$, and γ is the learning rate.

The choice of α, γ, σ as functions of iteration step k, also called noise schedule, can be interpreted as learning rate scheduling in gradient decent process. An α slightly smaller than 1 has been shown to improve stability [6].

3.3 Loss

The training process starts by randomly drawing unmodified examples, x^0, from the dataset. For each sample, we randomly select a denoising iteration k and then sample a random noise ϵ^k with appropriate variance for iteration k. The noise prediction network is asked to predict the noise from the data sample with noise added.

$$\mathcal{L} = MSE(\varepsilon^k, \varepsilon_0(\mathbf{x}^0 + \varepsilon^k, k)) \tag{3}$$

As shown in [6], minimizing the loss function in Eq 3 also minimizes the variational lower bound of the KL-divergence between the data distribution $p(x^0)$ and the distribution of samples drawn from the DDPM $q(x^0)$ using Eq 1.

3.4 Diffusion for Visuomotor Policy Learning

While DDPMs are typically used for image generation (x is an image), we use a DDPM to learn robot visuomotor policies. This requires two major modifications in the formulation: 1. changing the output x to represent robot actions. 2. making the denoising processes conditioned on input observation O_t. The following paragraphs discuss each of the modifications, and Fig.2 shows an overview.

Closed-loop action-sequence prediction: An effective action formulation should encourage temporal consistency and smoothness in long-horizon planning while allowing prompt reactions to unexpected observations. To accomplish this goal, we commit to the action-sequence prediction produced by a diffusion model for a fixed duration before replanning. Concretely, at time step t the policy takes the lastest T_o steps of observation data O_t as input and predicts T_p steps of actions, of which T_a steps of actions are executed on the robot without re-planning. Here, we define T_o as the observation horizon, T_p as the action prediction horizon and T_a as the action execution horizon. This encourages temporal action consistency while remaining responsive.

Visual observation conditioning: We used a DDPM to approximate the conditional distribution $p(A_t|O_t)$ instead of the joint distribution $p(A_t, O_t)$ used in [7] for planning. This formulation allows the model to predict actions conditioned on observations without the cost of inferring future states, speeding up the diffusion process and improving the accuracy of generated actions. To capture the conditional distribution $p(A_t|O_t)$, we modify Eq 1 to:

$$\Lambda_t^{k-1} = \alpha(\Lambda_t^k - \gamma\epsilon_\theta(\mathcal{O}_t, \Lambda_t^k, k) + \mathcal{N}(0, \sigma^2 I)) \tag{4}$$

The training loss is modified from Eq 3 to:

$$\mathcal{L} = MSE(\varepsilon^k, \varepsilon_\theta(\mathcal{O}_t, \mathcal{A}_t^0 + \varepsilon^k, k)) \tag{5}$$

The exclusion of observation features O_t from the output of the denoising process significantly improves inference speed and better accommodates real-time control. It also helps to make end-to-end training of the vision encoder feasible.

4 Implementation details

4.1 Comparing with the released source codes

We followed best practices from the original paper by employing both a visual-conditional U-Net-based DDPM and a visual-conditional Transformer-based DDPM as action policy models. We used the expert policy which is explicit programmed in metaworld codebase to generate demonstrative trajectories. To adapt the trajectories data in metaworld benchmark, we amended the code of dataloader as well as the interaction with simulator.

4.2 Experimental environment setup

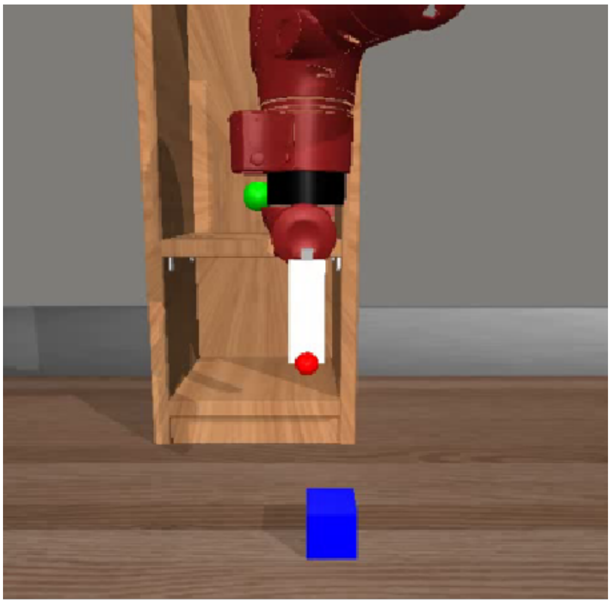


Figure 3. arm camera

Data collection. For each task, we collected 250 demonstration trajectories from a robotic arm's camera view situated below the arm(Fig. 3). Each trajectory includes observations of images with a resolution of 224*224 pixels and control signals that dictate the three-dimensional coordinates of the end-effector as well as normalized torque controls for the gripper, which are scaled between -1 and 1.

Data Preprocessing. During the training phase, the observation horizon (T_o) was set to 2, meaning that two consecutive frames were used as input to the model. Actions were predicted up to a horizon of 8 steps (T_a). If the trajectory length exceeded the specified horizon, padding was applied to ensure uniformity in the input data structure. A ResNet-18 architecture was initialized from scratch to serve as the vision encoder for processing the image inputs.

Training Configuration. The training process utilized an AdamW optimizer with a learning rate (Lr) of $1 * 10^{-4}$ and weight decay (WDecay) of $1 * 10^{-6}$. To accommodate the visual input, the batch size was set to 64. The diffusion network trained for 100 epochs.

Inference Configuration. At inference time, actions were executed based on predictions made over a horizon of 4 steps (T_p). The number of denoising iterations was kept consistent with the training phase at 100 (D-Iters Eval), ensuring a stable and predictable generation process. This setting facilitated real-time action execution while maintaining the quality of generated actions.

This setup closely adheres to the original Diffusion Policy specifications, aiming to reproduce its performance accurately across various robotic manipulation tasks. By using these parameters, we sought to validate the effectiveness of the Diffusion Policy framework in our experimental setting.

5 Results and analysis

As shown in Fig. 4, testing in 50 test tasks, Unet-based model reaches a success rate of 92%, Transformer-based model reaches a success rate of 76%.

In practice, we found the CNN-based backbone to work well on most tasks out of the box without the need for much hyperparameter tuning. However, it performs poorly when the desired action sequence changes quickly and sharply through time (such as velocity command action space), likely due to the inductive bias of temporal convolutions to prefer low-frequency signals.

In our state-based experiments, most of the bestperforming policies are achieved with the transformer backbone, especially when the task complexity and rate of action change are high. However, we found the transformer to be more sensitive to hyperparameters. The difficulty of transformer training is not unique to Diffusion Policy and could potentially be resolved in the future with improved transformer training techniques or increased data scale.

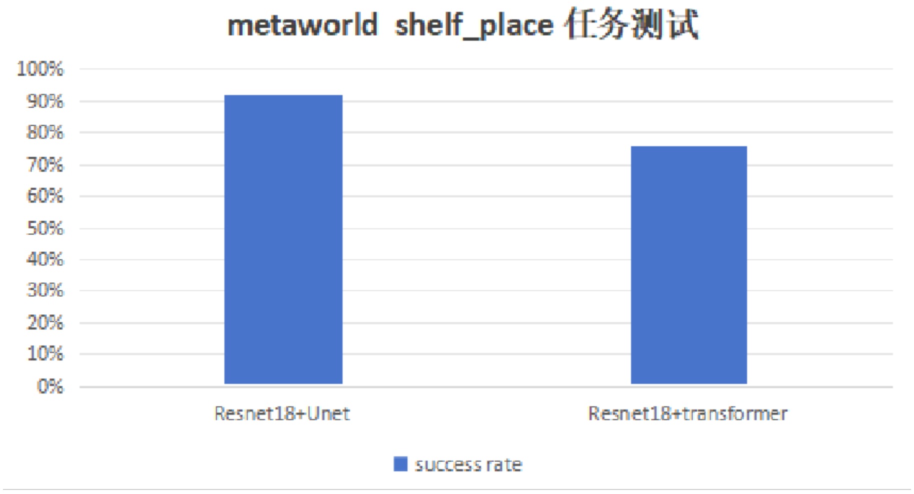


Figure 4. Experimental results

6 Conclusion and future work

In this work, we implement the Diffusion Policy in the MetaWorld benchmark, proving its effectiveness in robotic tasks. However, the vanilla Diffusion Policy only conditions on vision, which leads to poor task generalization and necessitates training a separate model for each specific task. In future work, it will be important to explore action policy models that can follow text instructions alongside visual input.

参考文献

[1] Brenna D. Argall, Sonia Chernova, Manuela M. Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics Auton. Syst.*, 57(5):469–483, 2009.

[2] Christopher G. Atkeson and Stefan Schaal. Robot learning from demonstration. In Douglas H. Fisher, editor, *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997*, pages 12–20. Morgan Kaufmann, 1997.

[3] Yahav Avigal, Lars Berscheid, Tamim Asfour, Torsten Kröger, and Ken Goldberg. Speedfolding: Learning efficient bimanual folding of garments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2022, Kyoto, Japan, October 23-27, 2022*, pages 1–8. IEEE, 2022.

[4] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016.

[5] Pete Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. *CoRR*, abs/2109.00137, 2021.

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[7] Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis, 2022.

[8] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[9] Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Silvio Savarese, and Li Fei-Fei. GTI: learning to generalize across long-horizon tasks from human demonstrations. In Marc Toussaint, Antonio Bicchi, and Tucker Hermans, editors, *Robotics: Science and Systems XVI, Virtual Event / Corvalis, Oregon, USA, July 12-16, 2020*, 2020.

[10] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *CoRR*, abs/2108.03298, 2021.

[11] Radford M Neal. Mcmc using hamiltonian dynamics. In Steve Brooks, Andrew Gelman, Galin L Jones, and Xiao-Li Meng, editors, *Handbook of Markov Chain Monte Carlo*, pages 113–162. CRC Press, 2011.

[12] Harish Ravichandar, Athanasios S. Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annu. Rev. Control. Robotics Auton. Syst.*, 3:297–330, 2020.

[13] Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone, 2022.

[14] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11895–11907, 2019.

[15] Sam Toyer, Rohin Shah, Andrew Critch, and Stuart Russell. The MAGICAL benchmark for robust imitation. In *Advances in Neural Information Processing Systems*, 2020.

[16] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

[17] Jimmy Wu, Xingyuan Sun, Andy Zeng, Shuran Song, Johnny Lee, Szymon Rusinkiewicz, and Thomas A. Funkhouser. Spatial action maps for mobile manipulation. In Marc Toussaint, Antonio Bicchi, and Tucker Hermans, editors, *Robotics: Science and Systems XVI, Virtual Event / Corvalis, Oregon, USA, July 12-16, 2020*, 2020.

[18] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, and Johnny Lee. Transporter networks: Rearranging the visual world for robotic manipulation. In Jens Kober, Fabio Ramos, and Claire J. Tomlin, editors, *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, volume 155 of *Proceedings of Machine Learning Research*, pages 726–747. PMLR, 2020.

[19] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. *CoRR*, abs/1710.04615, 2017.