A Comparative Study on State-Action Spaces for Learning Viewpoint Selection and Manipulation with Diffusion Policy

Xiatao Sun[1], Francis Fan[1*], Yinxing Chen[1*], Daniel Rakita[1]

***Abstract*— Robotic manipulation tasks often rely on static cameras for perception, which can limit flexibility, particularly in scenarios like robotic surgery and cluttered environments where mounting static cameras is impractical. Ideally, robots could jointly learn a policy for dynamic viewpoint and manipulation. However, it remains unclear which state-action space is most suitable for this complex learning process. To enable manipulation with dynamic viewpoints and to better understand impacts from different state-action spaces on this policy learning process, we conduct a comparative study on the state-action spaces for policy learning and their impacts on the performance of visuomotor policies that integrate viewpoint selection with manipulation. Specifically, we examine the configuration space of the robotic system, the end-effector space with a dual-arm Inverse Kinematics (IK) solver, and the reduced end-effector space with a look-at IK solver to optimize rotation for viewpoint selection. We also assess variants with different rotation representations. Our results demonstrate that state-action spaces utilizing Euler angles with the look-at IK achieve superior task success rates compared to other spaces. Further analysis suggests that these performance differences are driven by inherent variations in the high-frequency components across different state-action spaces and rotation representations.**

I. Introduction

Robotic manipulation with fixed viewpoints has been extensively studied and deployed in real-world scenarios. Traditionally, viewpoints for manipulation tasks are either static in the workspace or rigidly attached to the manipulation arm, such as with a wrist camera. However, this fixed-viewpoint paradigm is limited in scenarios where mounting static cameras is impractical or provides limited perception. For instance, in modern robotic surgery, a robotic arm with a mounted camera needs to be controlled alongside other arms with instruments [6]. In cluttered environments, a single camera may be obstructed by surrounding objects, while multiple cameras increase computational costs due to the additional video streams.

These challenges highlight the need for simultaneous viewpoint selection and manipulation within a robotic system featuring a dynamic camera. Ideally, robots could jointly learn viewpoint selection and manipulation, allowing them to generalize patterns of what, when, and how to move, along with determining when and from where to update their understanding of the environment to complete tasks. However, integrating manipulation with viewpoint selection introduces complexity, and there is no consensus on the choice of state-action space for this joint learning process. Therefore, in this work, we explore how different state-action spaces impact the performance of visuomotor policies that combine viewpoint selection with manipulation. As illustrated in Fig. 1, we conduct a comparative study by evaluating diffusion policies learned from different state-action spaces, including the configuration space of the robotic system, the end-effector space with a dual-arm IK solver §IV-A, and the reduced end-effector space that we refer to as the look-at space §IV-B. For the look-at space, we design a look-at IK solver that optimizes the camera rotation based on a viewpoint selection objective. This reduces the dimensionality of the

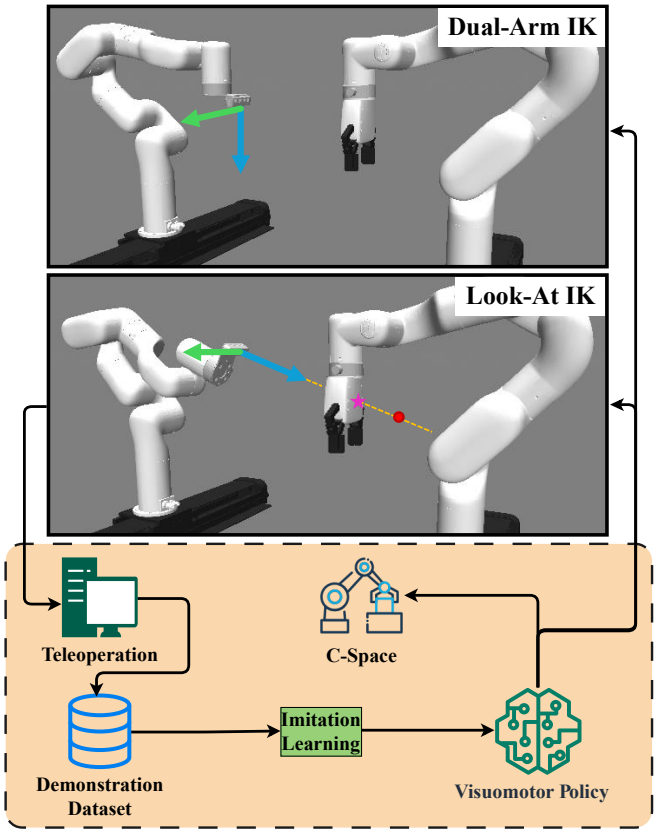

Fig. 1: Visuomotor policies control manipulation and viewpoint selection with different state-action spaces. The Look-At IK automatically determines the orientation of the viewpoint arm end-effector. Green and blue arrows represent the local y and z axes $\mathbf{w}_y$ and $\mathbf{w}_z$ of the viewpoint end-effector, respectively. The red dot indicates a distant point $\mathbf{p_f}$ along the orange line, showing the direction of $\mathbf{w}_z$. The pink star marks the visual target $\mathbf{p_t}$, set with a fixed offset from $\mathbf{p_m}$.

* Equal contribution

[1]The authors are with the Department of Computer Science, Yale University, New Haven, CT 06520, USA {xiatao.sun, francis.fan, j.y.chen, daniel.rakita}@yale.edu

This work was supported by Office of Naval Research award N00014-24-1-2124

state-action space, which enhances the efficiency of diffusion policy training and provide a more effective comparison with the standard end-effector space. We also examine variants of the end-effector space and the look-at space using different rotation representations.

For this comparative study, we developed a simulation environment for the dual-arm robotic system with a dynamic viewpoint and performed various experiments in simulation and real-world environments. Our results suggest that the learned diffusion policy, integrated with the look-at IK solver, achieves a higher task success rate (§V-B, SV-C). Although usually high dimensionality is considered as the primary factor for affecting policy learning [10, 21, 1], further analysis indicates that performance differences among diffusion policies with various state-action spaces might be more likely attributed to differences in High Frequency Component (HFC) in these spaces with different rotation representation (§V-D) instead of the dimensionality. We discuss the implications of this analysis concerning the spatial and optimization properties of different rotation representations and findings from the machine learning community on learning from HFC in commonly used diffusion model backbones (§VI-A). Finally, we conclude with future work for searching diffusion backbones that better capture HFC (§VI-B). Our contributions are summarized as follows:

- A comparative study of diffusion policies for viewpoint selection and manipulation learned from different state-action spaces for various tasks.
- Evidence of the correlation between the HFC on different state-action spaces and the performance of the learned visuomotor policies.
- Open-source implementations of our proposed method and the simulation environment to facilitate future research on viewpoint selection and manipulation.[1]

II. Related Works

A. Viewpoint Selection and Manipulation

Viewpoint selection, or active perception [2], has been extensively studied on mobile robotic platforms using both optimization-based [7, 34] and learning-based methods [40, 3]. However, its application to robotic manipulation, especially for dual-arm systems with high DOFs, remains limited.

Research has shown that viewpoint selection can improve the success rate of manipulation tasks in cluttered and occluded environments [9, 16]. Most optimization-based planners with dynamic viewpoint select the next best view based on metrics of information gain, such as the number of revealed occluded voxels [4], or the reduction in grasp quality uncertainty [24]. Recent work has started applying learning to viewpoint selection and manipulation. For instance, Saito et al. [31] used a recurrent neural network for a 6-DOF manipulation scenario, though they decoupled viewpoint selection and manipulation into separate phases. Lv et al. [22] utilized model-based reinforcement learning to train a policy for dual 7-DOF arms, showing that learning-based viewpoint selection can enhance task execution compared to static viewpoints. However, this work is limited to short-horizon tasks without any insight on impact of different state-action spaces under such complex decision making process.

[1] https://github.com/Apollo-Lab-Yale/spaces_comparative_study

B. Robot Learning and Diffusion Models

Learning-based methods have been successfully applied to various robotic platforms, such as quadrotors [45], autonomous driving [37, 47], and robotic arms [42, 17]. In particular, research on Imitation Learning (IL) has focused on improving the training loop for better data efficiency and quality, such as incorporating self-supervision [39] or data aggregation [36]. These methods, while enhancing the IL training loop, often use simple neural network architectures with a limited number of parameters.

With the rise of large generative models, recent advances in IL have integrated more sophisticated network architectures, including transformer-based models [33, 44] and diffusion-based policies [5, 46]. Diffusion policies, inspired by generative models for images and videos that employ a denoising process [14, 27], typically utilize Convolutional Neural Networks (CNNs), such as UNet [30], or Transformers [41] as their backbone. These models have shown significant potential and have been successfully applied to various robotic manipulation tasks [32, 20]. However, diffusion policies have not yet been explored in the context of manipulation with viewpoint selection, and while some studies have examined the connection between state-action spaces and data efficiency [48, 8], there is limited analysis on how different state-action spaces influence performance from a frequency perspective. In contrast, the machine learning community has studied how common diffusion backbones interact with frequency components in datasets. For instance, residual structures in CNNs, often used in UNet backbones, can suppress learning from HFC [43], whereas Vision Transformers are more effective at capturing low-frequency components (LFC) and are less suited for HFC [25]. Building on these insights, we aim to analyze the performance of different state-action spaces in robotic tasks through the lens of frequency analysis.

III. Technical Overview

A. Problem Formulation

This work employs imitation learning to train a visuomotor policy for a 17-DOF dual-arm robotic system shown in Fig. 2. This process requires a demonstration dataset $\mathcal{D} = \{r_n\}_{n=1}^{N}$, which contains N demonstration rollouts r_n. Each rollout is a sequence of observation-action pairs $(\mathbf{O}_t, \mathbf{A}_t)$ at each timestep t. This dataset is collected by a human expert with an intrinsic policy π^*. The goal of imitation learning is to mimic π^* by learning a policy π using $\mathcal{D}$ that maps observations $\mathbf{O}_t$ to actions $\mathbf{A}_t$:

$$\pi : \mathbf{O}_t \rightarrow \mathbf{A}_t$$

In our study, $\mathbf{O}_t$ consists of $\mathbf{v}_t$, the RGB image from a camera on the viewpoint arm, and $\mathbf{s}_t$, the state of the robot in a space for policy learning.

B. State-Action Spaces

We explore different state-action spaces, including configuration space, end-effector space, and look-at space.

Configuration Space (C-Space) is a multidimensional space where each dimension corresponds to a DOF of the robot. The C-Space $\mathcal{C}$ encompasses all possible configurations of a robotic system. For our dual-arm system:

$$\mathcal{C} = \{\mathbf{c} \in \mathbb{R}^{17} \mid c_i^{\min} \leq c_i \leq c_i^{\max}, \forall i\}$$

where $\mathbf{c}$ is the configuration vector constrained by the minimum and maximum values of each motor of the 17-DOF system, as illustrated in Fig. 2.

End-Effector Space refers to the set of all positions and orientations that end-effectors can reach within the workspace. For our dual-arm system, the end-effector space $\mathcal{E} = \mathcal{E}_m \times \mathcal{E}_v$ combines the manipulation arm end-effector space $\mathcal{E}_m$ and the viewpoint arm end-effector space $\mathcal{E}_v$. The end-effector vector of single arm consists of a position vector $\mathbf{p} = [x, y, z]^T \in \mathbb{R}^3$ and a rotation representation. We consider three rotation representations: Euler angles $\mathbf{R}_{\text{E}} \in \mathbb{R}^3$, unit quaternions $\mathbf{R}_{\text{Q}} \in \mathbb{H}_1$, and axis-angle $\mathbf{R}_{\text{AA}} \in (\mathbb{R}^3, \mathbb{R})$. For Euler angles, $\mathbf{R}_{\text{E}} = [\alpha, \beta, \gamma]^T$, where α, β, and γ denote roll, pitch, and yaw angles. For unit quaternions, $\mathbf{R}_{\text{Q}} = [q_0, q_1, q_2, q_3]^T$, with q_0 as the scalar part, $[q_1, q_2, q_3]^T$ as the vector part, and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. For axis-angle, $\mathbf{R}_{\text{AA}} = (\mathbf{u}, \theta)$, where $\mathbf{u} = [u_x, u_y, u_z]^T$ is a unit vector representing the rotation axis, and θ is the rotation angle in radian.

When using end-effector space, both arms in the dual-arm system use the same rotation representation. We consider three variants of the dual-arm end-effector space, including End-Effector Euler Space $\mathcal{E}_{\text{E}} = \{[\mathbf{p}_m, \mathbf{R}_{\text{E},m}, \mathbf{p}_v, \mathbf{R}_{\text{E},v}]\}$, End-Effector Quaternion Space $\mathcal{E}_{\text{Q}} = \{[\mathbf{p}_m, \mathbf{R}_{\text{Q},m}, \mathbf{p}_v, \mathbf{R}_{\text{Q},v}]\}$, and End-Effector Axis-Angle Space $\mathcal{E}_{\text{AA}} = \{[\mathbf{p}_m, \mathbf{R}_{\text{AA},m}, \mathbf{p}_v, \mathbf{R}_{\text{AA},v}]\}$. $\mathbf{p}_m$ and $\mathbf{p}_v$ are the respective positions of the manipulation and viewpoint end-effectors, $\mathbf{R}_{\text{E},m}$ and $\mathbf{R}_{\text{E},v}$ are the respective Euler angles of the manipulation and viewpoint end-effectors, $\mathbf{R}_{\text{Q},m}$ and $\mathbf{R}_{\text{Q},v}$ are the respective quaternions of the manipulation and viewpoint end-effectors, and $\mathbf{R}_{\text{AA},m}$ and $\mathbf{R}_{\text{AA},v}$ are the respective axis-angle representations of the manipulation and viewpoint end-effectors.

During policy learning from the end-effector space, all elements are treated as real numbers to facilitate fitting the neural network. Therefore, concatenating with the gripper configuration, policy learning with $\mathcal{E}_{\text{E}}$, $\mathcal{E}_{\text{Q}}$, and $\mathcal{E}_{\text{AA}}$ is actually in $\mathbb{R}^{13}$, $\mathbb{R}^{15}$, and $\mathbb{R}^{15}$ respectively.

Look-At Space simplifies the end-effector space by excluding the orientation of the viewpoint arm, which is automatically determined (see §IV-B) [29]. Similar to the end-effector space, we consider three variants of the look-at space: Look-At Euler Space $\mathcal{L}_{\text{E}} = \{[\mathbf{p}_m, \mathbf{R}_{\text{E},m}, \mathbf{p}_v]\}$, Look-At Quaternion Space $\mathcal{L}_{\text{Q}} = \{[\mathbf{p}_m, \mathbf{R}_{\text{Q},m}, \mathbf{p}_v]\}$, and Look-At Axis-Angle Space $\mathcal{L}_{\text{AA}} = \{[\mathbf{p}_m, \mathbf{R}_{\text{AA},m}, \mathbf{p}_v]\}$ after excluding $\mathbf{R}_{\text{E},v}$, $\mathbf{R}_{\text{Q},v}$, or $\mathbf{R}_{\text{AA},v}$.

Similar to the end-effector space, policy learning with $\mathcal{L}_{\text{E}}$, $\mathcal{L}_{\text{Q}}$, $\mathcal{L}_{\text{AA}}$ is in $\mathbb{R}^{10}$, $\mathbb{R}^{11}$, and $\mathbb{R}^{11}$ respectively.

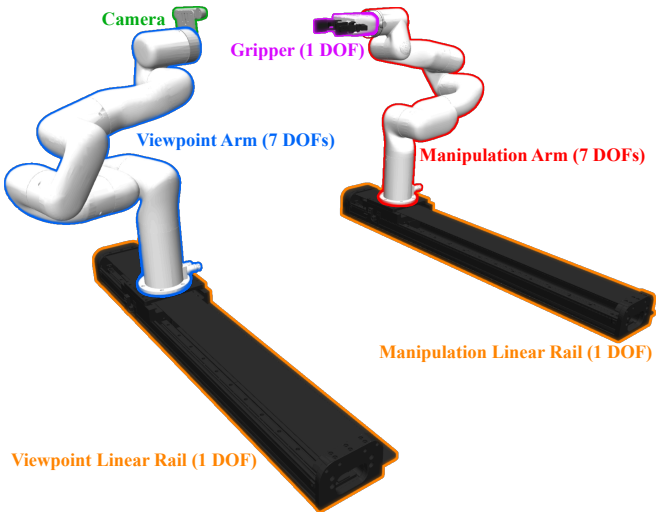

Fig. 2: Illustration of the 17-DOF dual-arm robotic system used in this paper. The system comprises two 7-DOF robotic arms, each mounted on a linear rail with 1 DOF. The viewpoint arm is equipped with a camera, while the manipulation arm features a gripper with an additional 1 DOF.

IV. Technical Details

A. Dual-Arm IK

The dual-arm IK solver is designed to convert from any end-effector space variant $\mathcal{E}_{\text{E}}, \mathcal{E}_{\text{Q}}, \mathcal{E}_{\text{AA}}$ to the C-space $\mathcal{C}$. The goal is to find the joint configurations $\mathbf{c}$ that minimize the transformation error between the desired and current positions $\mathbf{p}_{\text{m}}^*, \mathbf{p}_{\text{m}}$ and orientations $\mathbf{R}_{\text{m}}^*, \mathbf{R}_{\text{m}}$ of the manipulation end-effector, and similarly for the viewpoint end-effector.

To achieve this, we utilize Forward Kinematics (FK) to determine the current positions and orientations from $\mathbf{c}$:

$$\mathbf{p_m}, \mathbf{R_m}, \mathbf{p_v}, \mathbf{R_v} = \text{FK}(\mathbf{c})$$

In this process, we employ quaternions $\mathbf{R}_{\text{Q}}$ for rotation representation, due to their numerical stability and ability to avoid gimbal lock. The FK results, expressed in quaternion form, are then converted back to the appropriate rotation representations ($\mathbf{R}_{\text{E}}, \mathbf{R}_{\text{Q}}, \mathbf{R}_{\text{AA}}$) for the respective state-action space variants.

The transformation errors are calculated using Euclidean distance for positions and displacement-based distance for quaternions, where the quaternion distance between two quaternions $\mathbf{R}_1$ and $\mathbf{R}_2$ is defined as:

$$d(\mathbf{R}_1, \mathbf{R}_2) = \min\left(\|\ln(\mathbf{R}_1^{-1}\mathbf{R}_2)^\vee\|_2^2, \|\ln(\mathbf{R}_1^{-1}(-\mathbf{R}_2))^\vee\|_2^2\right)$$

which accounts for the antipodal equivalents of quaternions. ln represents the logarithm map that transforms a rotation from the Lie Group to its corresponding Lie Algebra, and $\vee$ is the operator that maps the Lie Algebra element to its corresponding vector in Euclidean space [28].

The objective function for a single arm is then given by:

$$J_{\mathcal{E},i}(\mathbf{p_i}, \mathbf{R_i}, \mathbf{p_i}^*, \mathbf{R_i}^*) = \|\mathbf{p_i} - \mathbf{p_i}^*\|_2^2 + d(\mathbf{R_i}, \mathbf{R_i}^*)$$

For both arms combined, the objective function is:

$$J_{\mathcal{E}}(\mathbf{c}, \mathbf{p}_{\mathrm{m}}^*, \mathbf{R}_{\mathrm{m}}^*, \mathbf{p}_{\mathrm{v}}^*, \mathbf{R}_{\mathrm{v}}^*) = J_{\mathcal{E},\mathrm{m}}(\mathbf{p_m}, \mathbf{R_m}, \mathbf{p}_{\mathrm{m}}^*, \mathbf{R}_{\mathrm{m}}^*) + J_{\mathcal{E},\mathrm{v}}(\mathbf{p_v}, \mathbf{R_v}, \mathbf{p}_{\mathrm{v}}^*, \mathbf{R}_{\mathrm{v}}^*)$$

The optimal joint configuration $\mathbf{c}^*$ is found by minimizing this combined objective function.

B. Look-At IK

Building on the dual-arm IK solver, the look-at IK solver is introduced to further enhance viewpoint selection. It incorporates an additional viewpoint-selection loss, which automatically calculates the orientation of the viewpoint end-effector. This reduces the state-action space to a look-at space variant $\mathcal{L}_{\mathrm{E}}, \mathcal{L}_{\mathrm{Q}}, \mathcal{L}_{\mathrm{AA}}$, which has lower dimensionality compared to the full end-effector space variants $\mathcal{E}_{\mathrm{E}}, \mathcal{E}_{\mathrm{Q}}, \mathcal{E}_{\mathrm{AA}}$.

The look-at IK solver continues to use the FK process for determining the current state of the end-effectors. Although the orientation $\mathbf{R_v}$ of the viewpoint is excluded from the state-action space, it is still necessary to extract the local y axis $\mathbf{w}_y$ and z axis $\mathbf{w}_z$ from $\mathbf{R_v}$ for the optimization process.

The objective function for the manipulation arm remains unchanged:

$$J_{\mathcal{L},m}(\mathbf{p_m}, \mathbf{R_m}, \mathbf{p_m}^*, \mathbf{R_m}^*) = \|\mathbf{p_m} - \mathbf{p_m}^*\|_2^2 + d(\mathbf{R_m}, \mathbf{R_m}^*)$$

For the viewpoint arm, the orientation is determined by minimizing a viewpoint selection objective function, which consists of two main components: a viewpoint orientation objective term and a viewpoint stability objective term.

The viewpoint orientation objective term measures the distance between the visual target $\mathbf{p_t}$ and its projection onto a line segment defined by the viewpoint position $\mathbf{p_v}$ and a distant point $\mathbf{p_f}$, located along the z axis direction $\mathbf{w}_z$ at a fixed distance δ:

$$\mathbf{p_f} = \mathbf{p_v} + \delta \mathbf{w}_z$$

To find this distance, the visual target $\mathbf{p_t}$ is projected onto the line segment using the clamped projection function:

$$\mathrm{proj}_{clamped}(\mathbf{v}, \mathbf{u}) = \min(\max(\mathrm{proj}_{scalar}(\mathbf{v}, \mathbf{u}), 0), 1) \cdot \mathbf{u}$$

The line segment projection of $\mathbf{p_t}$ is then:

$$\mathrm{proj}_{line}(\mathbf{p_t}, \mathbf{p_v}, \mathbf{p_f}) = \mathrm{proj}_{clamped}(\mathbf{p_t} - \mathbf{p_v}, \mathbf{p_f} - \mathbf{p_v}) + \mathbf{p_v}$$

The viewpoint orientation objective function is defined as:

$$J_{\mathcal{L},vo}(\mathbf{p_t}, \mathbf{p_v}, \mathbf{p_f}) = \|\mathrm{proj}_{line}(\mathbf{p_t}, \mathbf{p_v}, \mathbf{p_f}) - \mathbf{p_t}\|_2^2$$

The viewpoint stability objective term is designed to keep the viewpoint upright, preventing rolling and ensuring consistent orientation. This is calculated as:

$$J_{\mathcal{L},vs}(\mathbf{w}_y) = (\mathbf{w}_y \cdot [0, 0, 1]^T)^2$$

Combining these components, the complete viewpoint selection objective function is:

$$J_{\mathcal{L},v}(\mathbf{w}_y, \mathbf{w}_z, \mathbf{p_v}, \mathbf{p_v}^*) = \|\mathbf{p_v} - \mathbf{p_v}^*\|_2^2 + J_{\mathcal{L},vo}(\mathbf{p_t}, \mathbf{p_v}, \mathbf{p_f}) + J_{\mathcal{L},vs}(\mathbf{w}_y)$$

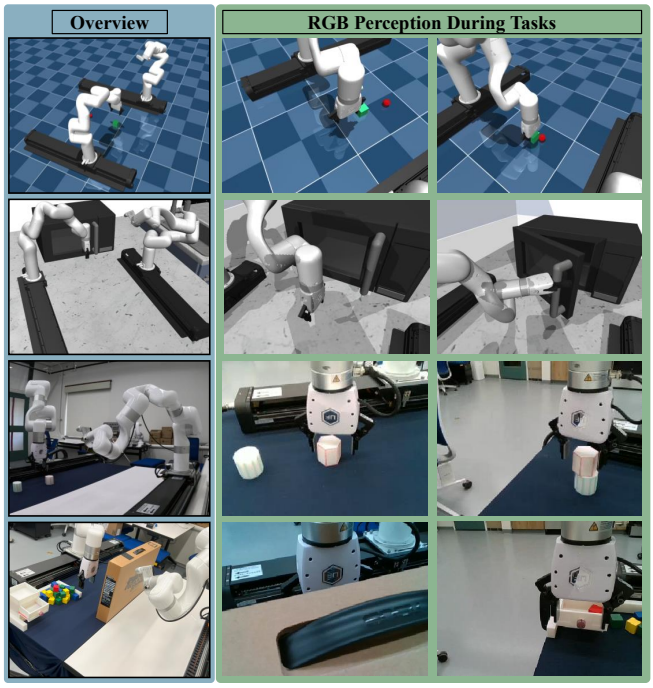

Fig. 3: Illustration of all simulated and real-world tasks, including an overview and the ego RGB perception of the system for each task. The first row depicts the simulated pick-and-place scenario. The second row shows the simulated microwave scenario. The third row presents the real-world block stacking scenario. The fourth row illustrates the real-world drawer interaction scenario.

In this framework, the visual target $\mathbf{p_t}$ is set as a fixed offset relative to the manipulation arm's end-effector position $\mathbf{p_m}$. Finally, the entire objective function for the look-at IK solver becomes:

$$J_{\mathcal{L}}(\mathbf{c}, \mathbf{p}_{\mathrm{m}}^*, \mathbf{R}_{\mathrm{m}}^*, \mathbf{p}_{\mathrm{v}}^*) = J_{\mathcal{L},m}(\mathbf{p_m}, \mathbf{R_m}, \mathbf{p_m}^*, \mathbf{R_m}^*) + J_{\mathcal{L},v}(\mathbf{w}_y, \mathbf{w}_z, \mathbf{p_v}, \mathbf{p_v}^*)$$

where $\mathbf{c}$ is optimized to minimize this combined objective function.

C. Diffusion Policy

In this work, we use the diffusion policy [5] as the visuomotor policy for imitation learning. We use U-Net [30] as the backbone for the diffusion policy. The diffusion policy employs receding horizon control and models the conditional distribution $p(\mathbf{A}_t \mid \mathbf{O}_t)$. During inference, the diffusion policy starts from Gaussian noise $\mathbf{A}_t^K$, where K is the number of steps for denoising. The denoising process repeatedly utilizes the following equation:

$$\mathbf{A}_t^{k-1} = \alpha \left(\mathbf{A}_t^k - \gamma \epsilon_\theta(\mathbf{O}_t, \mathbf{A}_t^k, k) + \mathcal{N}(0, \sigma^2, \mathbf{I}) \right)$$

until the process reaches the output $\mathbf{A}_t^0$ of the policy.

V. Evaluation

A. Implementation

We compared diffusion policies trained with different state-action spaces in various simulated and real-world tasks

with demonstrations collected from teleoperation and the same hyperparameters as [5]. For the look-at IK, we set δ to 999 and adjusted p_t to $p_t - [0.0, 0.0, -0.15]^T$ with m as the unit. We used PyTorch [26] as the machine learning framework for training the policy. All IK optimizations are performed using SLSQP [18] in NLopt [15]. Both training and inference were performed using an AMD PRO 5975WX CPU, dual 4090 GPUs, and 128GB RAM. The primary metrics throughout all evaluations were task success rate and the mean and standard deviation of task completion duration.

TABLE I: Evaluation Results with Different State-Action Spaces

Task (Training Epochs)	Space	Mean Duration	Std Duration	Success Rate
Simulated Pick and Place (200 Epochs)	$\mathcal{C}$	N/A	N/A	0.00
	$\mathcal{E}_\mathrm{E}$	340.52 stp	93.69 stp	0.84
	$\mathcal{E}_\mathrm{Q}$	410.76 stp	96.06 stp	0.46
	$\mathcal{E}_\mathrm{AA}$	480.18 stp	52.47 stp	0.13
	$\mathcal{L}_\mathrm{E}$	**303.81 stp**	69.38 stp	**0.92**
	$\mathcal{L}_\mathrm{Q}$	381.18 stp	95.24 stp	0.56
	$\mathcal{L}_\mathrm{AA}$	498.55 stp	**13.47 stp**	0.05
Simulated Microwave (10 Epochs)	$\mathcal{C}$	446.98 stp	93.83 stp	0.30
	$\mathcal{E}_\mathrm{E}$	360.41 stp	84.62 stp	0.81
	$\mathcal{E}_\mathrm{Q}$	364.19 stp	103.48 stp	0.68
	$\mathcal{E}_\mathrm{AA}$	417.00 stp	115.69 stp	0.41
	$\mathcal{L}_\mathrm{E}$	**300.19 stp**	**29.58 stp**	**1.00**
	$\mathcal{L}_\mathrm{Q}$	327.55 stp	71.82 stp	0.91
	$\mathcal{L}_\mathrm{AA}$	364.95 stp	81.11 stp	0.81
Real-World Block Stacking (300 Epochs)	$\mathcal{C}$	N/A	N/A	0.00
	$\mathcal{E}_\mathrm{E}$	28.75 sec	4.10 sec	0.80
	$\mathcal{E}_\mathrm{Q}$	31.04 sec	3.74 sec	0.73
	$\mathcal{E}_\mathrm{AA}$	35.21 sec	2.85 sec	0.23
	$\mathcal{L}_\mathrm{E}$	**25.65 sec**	3.18 sec	**1.00**
	$\mathcal{L}_\mathrm{Q}$	29.12 sec	**2.78 sec**	0.83
	$\mathcal{L}_\mathrm{AA}$	34.90 sec	4.03 sec	0.47
Real-World Drawer Interaction (300 Epochs)	$\mathcal{C}$	N/A	N/A	0.00
	$\mathcal{E}_\mathrm{E}$	58.99 sec	9.48 sec	0.80
	$\mathcal{E}_\mathrm{Q}$	61.37 sec	10.37 sec	0.77
	$\mathcal{E}_\mathrm{AA}$	59.09 sec	10.06 sec	0.70
	$\mathcal{L}_\mathrm{E}$	56.86 sec	**6.63 sec**	0.80
	$\mathcal{L}_\mathrm{Q}$	**52.34 sec**	8.97 sec	**0.87**
	$\mathcal{L}_\mathrm{AA}$	56.80 sec	7.58 sec	0.83

B. Simulation Experiments

The simulation environments, developed using MuJoCo [38], include a pick-and-place scenario and a microwave scenario. In the pick-and-place scenario, the agent must pick up a green cube and place it on a red goal position, with the cube randomly generated within a 15 cm radius around the goal. This scenario is shown in the first row of Fig. 3. The microwave scenario, depicted in the second row of Fig. 3, requires the agent to open the microwave door by at least 0.5 radians. Results for all simulation experiments were calculated from 100 inference rollouts. The pick-and-place policy was trained with 282 rollouts, and the microwave policy with 151 rollouts.

The first and second rows in Tab. I present the results for the pick-and-place and microwave scenarios, respectively, measured in steps (stp) for task duration. Overall, state-action spaces using Euler angles performed better compared to other rotation representations. Using look-at IK improved success rates for all tasks, likely due to reduced dimensionality. Policies with look-at IK also had lower mean and standard deviation in task duration, indicating less uncertainty. However, differences in dimensionality cannot explain why $\mathcal{L}_\mathrm{AA}$ that has less amount of elements compared to $\mathcal{E}_\mathrm{E}$ and $\mathcal{E}_\mathrm{Q}$ performs worse in the pick and place scenario.

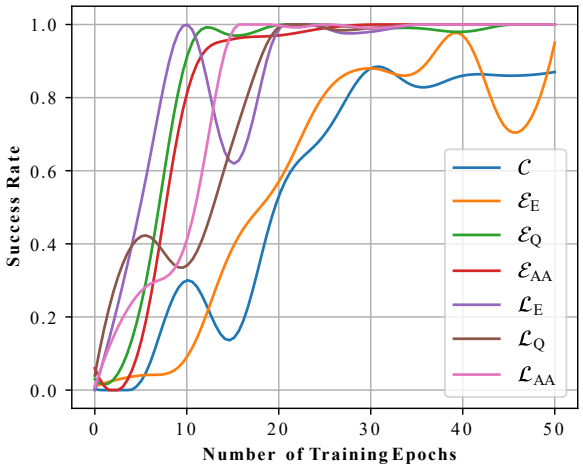

Fig. 4: The success rates of policies for the microwave task with different state-action spaces over 50 training epochs.

The microwave task was easier to learn than the pick-and-place task. For example, the policy trained with $\mathcal{L}_\mathrm{E}$ reached a success rate of 1.00 after 10 training epochs, while the pick-and-place task reached 0.92 after 200 epochs despite having a larger demonstration dataset. Given the ease of the microwave task, we explored whether additional training epochs could help policies from different state-action spaces achieve comparable performance. As shown in Fig. 4, increasing training epochs led to success rates of policies from different state-action spaces becoming more comparable, indicating that performance differences are likely due to the ability of the network to capture patterns in the corresponding space.

C. Real-World Experiments

We conducted real-world experiments in two scenarios: block stacking using 3D-printed objects from [19] and drawer interaction with a 3D-printed drawer from [13]. The block stacking task involved non-cuboid objects, requiring precise gripper orientation for successful grasping, while the drawer interaction task required the agent to get rid of visual occlusion from a cardboard box, place a red target cube into the drawer, and close it. The block stacking policy was trained with 201 rollouts, and the drawer interaction policy with 67 rollouts. Each task was evaluated with 30 inference rollouts. As shown in the third and fourth rows of Tab. I, the use of the look-at IK solver further improved both success rate and performance stability, and state-action spaces using Euler angles or quaternions for rotation representation generally achieved higher success rates, further indicating

that dimensionality is not the primary factor for affecting learning performance given that quaternions and axis-angle representations have the same amount of elements.

D. Frequency Analysis

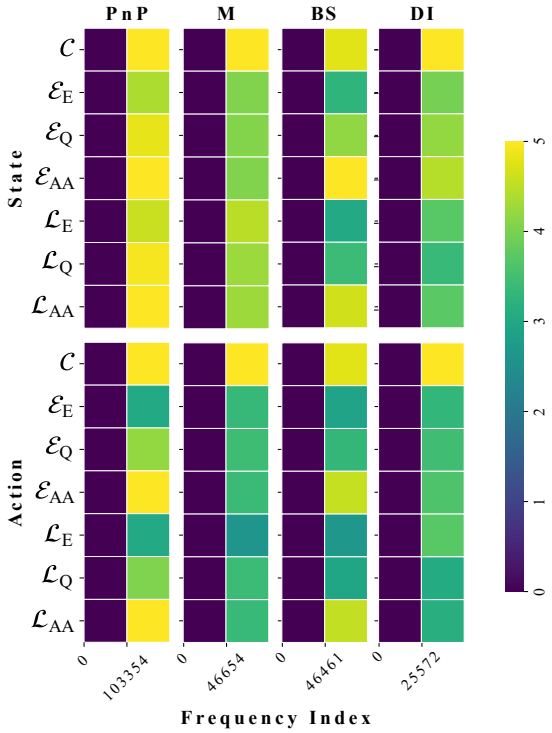

Fig. 5: Frequency heatmaps of concatenated state and action arrays from demonstration datasets across different evaluation scenarios (PnP for pick-and-place, M for microwave, BS for block stacking, DI for drawer interaction). The horizontal axes indicate the lowest and highest frequency indices, while the vertical axes correspond to different state and action spaces. Color intensity denotes the logarithmically scaled magnitude of the frequency components.

To further investigate the performance differences among state-action spaces, we conducted a frequency analysis inspired by recent work on the interplay between frequency and CNNs [43]. We performed a Fast Fourier Transform to convert the time-domain signals into the frequency domain and calculated the logarithm of the magnitudes of the Fourier coefficients:

$$\mathbf{X}_k = \frac{1}{M} \sum_{m=1}^{M} \log \left(1 + \left| \sum_{n=0}^{N-1} \mathbf{x}_{m,n} \cdot e^{-i \frac{2\pi k n}{N}} \right| \right)$$

where $\mathbf{X}_k$ is the k_{th} component of the frequency-domain representation, $\mathbf{x}_{m,n}$ is the n_{th} sample of the m_{th} element in corresponding space with dimensionality of M in a dataset with length of N.

The results, shown in Fig. 5, reveal a stronger correlation between success rates and the magnitude of HFC compared to dimensionality. For example, the relative ease of learning with $\mathcal{L}_{\mathrm{E}}$ may be due to the low magnitude of HFC, whereas the difficulty when learning with $\mathcal{E}_{\mathrm{AA}}$ may be attributed to the high magnitude of HFC despite having the same dimensionality as $\mathcal{E}_{\mathrm{Q}}$.

VI. Discussion

A. Implications

Learning from state-action spaces with Euler angles as the rotation representation yielded the best performance in most of our evaluation tasks. However, Euler angles are known to have several disadvantages in both spatial representation and optimization, such as gimbal lock due to singularities [35] and a non-smooth optimization landscape due to discontinuities [11]. Despite these drawbacks, the HFC present in other rotation representations and configuration spaces pose challenges for current diffusion policy backbones, especially when dealing with high-dimensional state-action spaces and long task horizons.

Previous work has shown that while CNNs are effective at capturing HFC, the use of residual structures can suppress HFC learning [43]. Removing these residuals from the UNet architecture, which is commonly used in diffusion policies, is not a viable solution as they are critical for deep network training as a mitigation for vanishing gradients [12]. Although Transformers [44] are another popular choice for generative diffusion models [27], they are better suited for capturing LFC rather than HFC [25], making them less ideal for diffusion policies with state-action spaces that contain significant HFC.

B. Limitations and Future Work

Despite the growing interest in the impact of different state-action spaces and rotation representations on learning performance and data efficiency, this direction remains underexplored, especially concerning the potential connection between frequency and model architecture. Although our work proposes the hypothesis and provides empirical evidence that HFC affects learning performance for current diffusion policies, more thorough theoretical analysis is needed to further explore the interplay between frequency and model architecture.

Given the lack of suitable network architectures for learning from state-action spaces with significant HFC, and considering the optimization and spatial representation benefits of these spaces, another future direction is to propose new diffusion backbones capable of capturing HFC from these spaces, potentially improving sample efficiency and generalizability. Incorporating more geometrical inductive biases into the design of network architectures may be necessary. Besides improving on architecture, another future direction would be projecting datasets with intense HFC to higher dimensional space, similar to how [23] dealing with HFC for better capturing details in learning radiance field. Additionally, this work sets the visual target $\mathbf{p_t}$ as a fixed offset from the end-effector position of the manipulation arm $\mathbf{p_m}$. Future work will also explore leveraging semantic information from the environment to set $\mathbf{p_t}$ based on detected semantic objects in the environment.

References

[1] Fares J Abu-Dakka, Yanlong Huang, João Silvério, and Ville Kyrki. A probabilistic framework for learning geometry-based robot manipulation skills. *Robotics and Autonomous Systems*, 141:103761, 2021.

[2] Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[3] Luca Bartolomei, Lucas Teixeira, and Margarita Chli. Semantic-aware active perception for uavs using deep reinforcement learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3101–3108. IEEE, 2021.

[4] Michel Breyer, Lionel Ott, Roland Siegwart, and Jen Jen Chung. Closed-loop next-best-view planning for target-driven grasping. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1411–1416. IEEE, 2022.

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

[6] Claudia D'Ettorre, Andrea Mariani, Agostino Stilli, Ferdinando Rodriguez y Baena, Pietro Valdastri, Anton Deguet, Peter Kazanzides, Russell H. Taylor, Gregory S. Fischer, Simon P. DiMaio, Arianna Menciassi, and Danail Stoyanov. Accelerating surgical robotics research: A review of 10 years with the da vinci research kit. *IEEE Robotics & Automation Magazine*, 28(4):56–78, 2021. doi: 10.1109/MRA.2021.3101646.

[7] Davide Falanga, Philipp Foehn, Peng Lu, and Davide Scaramuzza. Pampc: Perception-aware model predictive control for quadrotors. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[8] Andreas René Geist, Jonas Frey, Mikel Zhobro, Anna Levina, and Georg Martius. Learning with 3d rotations, a hitchhiker's guide to so (3). In *Forty-first International Conference on Machine Learning*.

[9] Marcus Gualtieri and Robert Platt. Viewpoint selection for grasp detection. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 258–264. IEEE, 2017.

[10] Masashi Hamaya, Robert Lee, Kazutoshi Tanaka, Felix von Drigalski, Chisato Nakashima, Yoshiya Shibata, and Yoshihisa Ijiri. Learning robotic assembly tasks with lower dimensional systems by leveraging physical softness and environmental constraints. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7747–7753. IEEE, 2020.

[11] Hashim A Hashim. Special orthogonal group so (3), euler angles, angle-axis, rodriguez vector and unit-quaternion: Overview, mapping and challenges. *arXiv preprint arXiv:1909.06669*, 2019.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.

[13] Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023.

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

[15] Steven G. Johnson. The NLopt nonlinear-optimization package. `https://github.com/stevengj/nlopt`, 2007.

[16] Gregory Kahn, Peter Sujan, Sachin Patil, Shaunak Bopardikar, Julian Ryde, Ken Goldberg, and Pieter Abbeel. Active exploration using trajectory optimization for robotic grasping in the presence of occlusions. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4783–4790. IEEE, 2015.

[17] Heecheol Kim, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Transformer-based deep imitation learning for dual-arm robot manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8965–8972. IEEE, 2021.

[18] D. Kraft. *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988. URL `https://books.google.com/books?id=4rKaGwAACAAJ`.

[19] Alex X Lee, Coline Manon Devin, Yuxiang Zhou, Thomas Lampe, Konstantinos Bousmalis, Jost Tobias Springenberg, Arunkumar Byravan, Abbas Abdolmaleki, Nimrod Gileadi, David Khosid, et al. Beyond pick-and-place: Tackling robotic stacking of diverse shapes. In *5th Annual Conference on Robot Learning*, 2021.

[20] Hang Li, Qian Feng, Zhi Zheng, Jianxiang Feng, and Alois Knoll. Generalizable robotic manipulation: Object-centric diffusion policy with language guidance. In *Workshop on Embodiment-Aware Robot Learning*.

[21] Zvezdan Lončarević, Andrej Gams, Aleš Ude, et al. Robot skill learning in latent space of a deep autoencoder neural network. *Robotics and Autonomous Systems*, 135:103690, 2021.

[22] Jun Lv, Yunhai Feng, Cheng Zhang, Shuang Zhao, Lin Shao, and Cewu Lu. Sam-rl: Sensing-aware model-based reinforcement learning via differentiable physics-based simulation and rendering. In *Robotics science and systems*, 2023.

[23] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106, 2021.

[24] Douglas Morrison, Peter Corke, and Jürgen Leitner. Multi-view picking: Next-best-view reaching for im-

proved grasping in clutter. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8762–8768. IEEE, 2019.

[25] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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

[28] Daniel Rakita. *3D Spatial Modeling and Computing*. 2024. URL `https://tinyurl.com/3c4ysbb6`.

[29] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. Remote telemanipulation with adapting viewpoints in visually complex environments. *Robotics: Science and Systems XV*, 2019.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[31] Namiko Saito, Tetsuya Ogata, Satoshi Funabashi, Hiroki Mori, and Shigeki Sugano. How to select and use tools?: Active perception of target objects using multimodal deep learning. *IEEE Robotics and Automation Letters*, 6(2):2517–2524, 2021.

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

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

[34] Igor Spasojevic, Varun Murali, and Sertac Karaman. Perception-aware time optimal path parameterization for quadrotors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3213–3219. IEEE, 2020.

[35] Mark W Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*. John Wiley & Sons, 2020.

[36] Xiatao Sun, Shuo Yang, and Rahul Mangharam. Megadagger: Imitation learning with multiple imperfect experts. *arXiv preprint arXiv:2303.00638*, 2023.

[37] Xiatao Sun, Mingyan Zhou, Zhijun Zhuang, Shuo Yang, Johannes Betz, and Rahul Mangharam. A benchmark comparison of imitation learning-based control policies for autonomous racing. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–5. IEEE, 2023.

[38] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

[39] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

[40] Jesus Tordesillas and Jonathan P How. Deep-panther: Learning-based perception-aware trajectory planner in dynamic environments. *IEEE Robotics and Automation Letters*, 8(3):1399–1406, 2023.

[41] Ashish Vaswani. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[42] Gaotian Wang, Kejia Ren, and Kaiyu Hang. Uno push: Unified nonprehensile object pushing via non-parametric estimation and model predictive control. *arXiv preprint arXiv:2403.13274*, 2024.

[43] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P. Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[44] Xudong Wang, Chao Wei, Hanqing Tian, Weida Wang, and Jibin Hu. Implicit predictive behavior cloning for autonomous driving decision-making in urban traffic. *IEEE Transactions on Intelligent Vehicles*, 2024.

[45] Yuwei Wu, Xiatao Sun, Igor Spasojevic, and Vijay Kumar. Deep learning for optimization of trajectories for quadrotors. *IEEE Robotics and Automation Letters*, 2024.

[46] Xiaoyu Zhang, Matthew Chang, Pranav Kumar, and Saurabh Gupta. Diffusion meets dagger: Supercharging eye-in-hand imitation learning. *arXiv preprint arXiv:2402.17768*, 2024.

[47] Mingyan Zhou, Biao Wang, and Xiatao Sun. Developing trajectory planning with behavioral cloning and proximal policy optimization for path-tracking and static obstacle nudging. *arXiv preprint arXiv:2409.05289*, 2024.

[48] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.