FOURIER FEATURES LET AGENTS LEARN HIGH PRECISION POLICIES WITH IMITATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Various 3D modalities have been proposed for high-precision imitation learning tasks to compensate for the short-comings of RGB-only policies. Modalities that explicitly represent positions in Cartesian space have an inherent advantage over purely image-based ones, since they allow policies to reason about geometry. Point clouds are a common way to represent geometric information, and have several benefits such as permutation invariance and flexible observation size. Despite their effectiveness, a number of hybrid 2D/3D architectures have been proposed in the literature, indicating that this performance can often be task-dependent. We hypothesize that this may be due to the spectral bias of neural networks towards learning low frequency functions, which especially affects models conditioned on slow-moving Cartesian features. Building on prior work that uses a parametric projection from Cartesian space into high-dimensional Fourier space to overcome the innate low-pass filtering characteristic of neural networks, we apply Fourier features to several representative point cloud encoder architectures. We validate this approach on challenging manipulation tasks from the RoboCasa and ManiSkill3 benchmarks, and find that adding Fourier feature projections provides benefits across diverse encoder architectures and tasks, with meaningful improvements seen in the vast majority of tasks. We show that Fourier features are a general-purpose tool for point cloud-based imitation learning, which consistently improves performance by enabling policies to leverage geometric details more effectively than models conditioned on Cartesian features.

1 INTRODUCTION

Diffusion-based imitation learning (IL) has emerged as a powerful framework for visuomotor control (Chi et al., 2023; Reuss et al., 2023; Wu et al., 2025; Intelligence et al., 2025) in robotics. By treating action generation as a denoising process (Ho et al., 2020), diffusion policies naturally capture multi-modal action distributions, enabling robots to represent the diverse strategies often present in human demonstrations. This capability has made diffusion policies the state-of-the-art on long-horizon and multi-task manipulation benchmarks.

Diffusion models excel at capturing the multi-modality of expert demonstrations, but require the input representations to preserve the fine-grained information that distinguishes successful strategies from failed ones. In high-precision manipulation tasks, 3D information about the scene can help the agent reason about geometry and occlusions and execute complex motions accurately. Policies that cannot perceive fine geometric information encoded in observations are unable to imitate expert demonstrations that depend on these details.

RGB images remain the most common observation space due to their semantic richness and the widespread availability of pretrained vision encoders (Ke et al., 2025; Wilcox et al., 2025; Donat et al., 2025). However, they lack explicit 3D geometry and require the policy backbone to implicitly infer a 2D-to-3D mapping, while also being sensitive to viewpoint and lighting variations. In contrast, 3D modalities such as depth maps, point clouds, or point maps that explicitly encode shape, distance, and spatial relationships, allow policies to learn behaviors in a common 3D space, enabling multi-view consistency. Yet despite the success of these 3D representations (Ze et al., 2024; Zhu et al., 2024; Ze et al., 2025), a number of hybrid 2D/3D architectures have recently been sug-

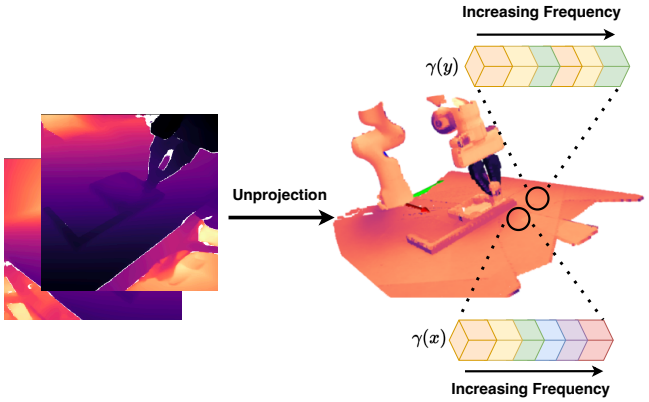

Figure 1: **Overview of our method.** Adding a Fourier feature mapping from Cartesian coordinates into a higher-dimensional feature space improves performance for any point cloud encoder used for diffusion imitation learning. For high-precision policies, the network must learn to condition on fine details in the scene geometry to e.g. device whether to insert the leg into the slot or reposition it, yet neural networks learn the high frequency components of the target function only slowly, if at all. While neighbouring points in the scene have very similar Cartesian features, the high-dimensional Fourier features allow them to easily be distinguished.

gested (Ke et al., 2025; Wilcox et al., 2025; Goyal et al., 2023), indicating that the performance of 3D representations may depend on specific tasks and dataset.

While neural networks are universal function approximators (Hornik et al., 1989), they have a *spectral bias* toward learning low-frequency components first, while high-frequency components converge slowly or may not be learned at all (Rahaman et al., 2019; Tancik et al., 2020). In the context of precise robotic manipulation tasks, such as inserting a peg into a socket, these high frequency components can make the difference between a successful trajectory and one where the peg and the socket are slightly mis-aligned. Both observations are very similar in terms of absolute Euclidean coordinates and distance, making it difficult to robustly learn to differentiate between them. In fields such as novel view synthesis, this shortcoming is remedied using a Fourier feature mapping (Mildenhall et al., 2021; Tancik et al., 2020). More recently Adapt3R (Wilcox et al., 2025) has shown that incorporating Fourier features benefits their architecture, yet a systemic study of Fourier features for other 3D representations in imitation learning is missing in the literature.

Inspired by these insights, we propose to encode the 3D representations for pointcloud-based approaches (Qi et al., 2017a; Gyenes et al., 2024) in Fourier space. By amplifying high-frequency components of these representations, we counteract spectral bias and make subtle temporal and geometric differences accessible to diffusion backbones. This simple modification allows different models acting on 3D representations to more easily understand small details in geometric observations, thus improving their performance for high-precision control tasks. Experimentally, we show that using Fourier-encoded input representations leads to consistent improvements across different point cloud architectures and benchmarks. In RoboCasa and ManiSkill3, we achieve an average success rate improvement of up to 19% and 7%, respectively. Qualitatively, policies trained with Fourier mappings exhibit smoother and more precise motions, particularly on robotic control tasks where fine-grained manipulation matters (Nasiriany et al., 2024; Tao et al., 2025).

Our contributions are the following: **1)** we introduce a framework for incorporating Fourier feature mappings into various point cloud encoders; **2)** we instantiate this framework with representative point cloud architectures commonly used for imitation learning; and **3)** through extensive experiments on the RoboCasa and ManiSkill task suites, we demonstrate consistent improvements over baselines without Fourier feature mappings.

2 Related Work

Imitation Learning in Robotics. Recent progress in IL has been driven by incorporating diffusion (Chi et al., 2023; Reuss et al., 2023) or flow matching (Lipman et al., 2023), which enable policies to learn multi-modal action distributions, and by training policies on large-scale datasets (Black et al., 2024; Intelligence et al., 2025; Brohan et al., 2022; Zitkovich et al., 2023; Zhu et al., 2025) to significantly increase generalization and performance. However, these approaches are primarily conditioned on RGB images. This choice allow leveraging powerful pretrained visual encoders and provides strong semantic features, but RGB inputs lack explicit 3D geometry and are sensitive to viewpoint and lighting variations (Zhu et al., 2024; Ze et al., 2024; Wilcox et al., 2025). To address these shortcomings, several works incorporate 3D information by either entirely using 3D inputs or by combining them with RGB (Ze et al., 2024; Gervet et al., 2023; Wilcox et al., 2025; Goyal et al., 2023). In our work, we focus specifically on using 3D inputs for high-precision manipulation tasks and show that their performance is not only inherent to the modality itself, but is affect by the spectral bias of neural networks, which can be mitigated through Fourier mappings.

3D Visual Representations for Imitation Learning. 3D inputs can be leveraged in different ways, as stand-alone modalities (e.g., point clouds or point maps) or in combination with RGB. On a number of challenging tasks, imitation learning with lightweight point cloud-based policies consistently outperforms RGB and RGB-D modalities while requiring significantly less data (Ze et al., 2024; Zhu et al., 2024). Additionally, Zhu et al. (2024) observe that training on point maps yields no advantage over training on point clouds. RVT (Goyal et al., 2023) re-renders virtual viewpoints as seven-channel point maps containing RGB, depth, and global coordinate information to decouple the current observation from the input used for downstream decision-making, using point clouds only as a intermediate representation. A variety of hybrid 2D/3D approaches (Ke et al., 2025; Gervet et al., 2023; Wilcox et al., 2025) lift 2D features from pre-trained image encoders into 3D space by concatenating them with their 3D positions reconstructed from the original depth maps. This allows models to combine the benefits of pre-trained visual encoders with the explicit 3D representation of point clouds. While these works emphasize architectural design or multi-view fusion, our method focuses on 3D representations, introducing non-parametric Fourier mappings that can be combined with any 3D encoder to make fine geometric details more accessible.

Deep Learning with Fourier Features. Fourier features (Tancik et al., 2020; Mildenhall et al., 2021) mitigate the spectral bias of neural networks (Rahaman et al., 2018; 2019), i.e. their tendency to learn low-frequency components faster than high-frequency ones. This bias is amplified by the data manifold geometry where variations that look high-frequency along it may correspond to low-frequency modes in the ambient space (Rahaman et al., 2019), causing fine details to be suppressed. These effects explain why many architectures struggle to condition on fine geometric details, and motivate our use of Fourier feature mappings. Fourier mappings address this by lifting low-dimensional inputs such as Cartesian coordinates into sinusoidal embeddings with multiple frequencies, which can be fixed or learnable (Gao et al., 2023; Sun et al., 2024). Neural radiance fields (Mildenhall et al., 2021; Barron et al., 2022) use this technique to be able to learn detailed 3D scenes with high fidelity, and they are only able to learn burry, oversmoothed representations without it Tancik et al. (2020). Adapt3R (Wilcox et al., 2025) propose a novel observation encoder that outperforms other architectures on novel viewpoints unseen during training. While they show the benefit of Fourier features for their architecture, they do not investigate their effect in other contexts. In contrast, we apply Fourier mappings systematically across 3D modalities in diffusion-based IL, providing a frequency-domain perspective that complements architectural approaches.

3 Method

3.1 Problem Formulation

Imitation Learning aims to learn a policy from expert demonstrations. We are given a dataset containing N expert trajectories $\mathcal{D} = \{\tau_i\}_{i=1}^{N}$, where each trajectory $\tau_i = (\mathbf{g}_i, (o_1, a_1), (o_2, a_2), \ldots, (o_K, a_K))$, where K is the trajectory length and $\mathbf{g}_i$ is the language description for the trajectory. The objective is to learn a policy $\pi(\bar{a}|o, \mathbf{g})$ that maps observations o and embedded goal $\mathbf{g}$ to a sequence of actions $\bar{a} = (a_k, a_{k+1}, \ldots, a_{k+H})$ for the agent to execute in the environment. Predicting sequences of actions, i.e. action chunking, results

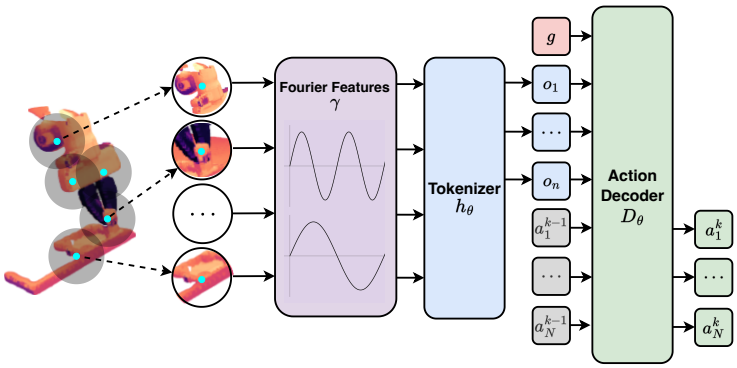

Figure 2: **Overview of our framework** Given a pointcloud, we first map each point i and its neighbourhood $\mathcal{N}(i)$ (indicated by the encircled patches) and map them to Fourier feature space. This amplifies subtle geometric differences in each neighborhood. The tokenizer extracts and aggregates features for each neighborhood to produce a set of tokens which are then forwarded to a goal-conditioned diffusion policy D_θ to denoise the next chunk of actions.

in more temporally correlated trajectories compared to predicting individual actions (Zhao et al., 2023). Each observation o contains depth images from M cameras. In combination with the camera intrinsic and extrinsic parameters from calibration, we can construct any desired 3D observation representation from these depth images. We use a frozen CLIP (Radford et al., 2021) encoder to generate language embeddings from text descriptions of the goal.

3.2 Score-Based Diffusion

To learn policies from expert demonstrations, we use the typical EDM framework (Karras et al., 2022; Reuss et al., 2023) for score-based action diffusion conditioned on observations of the scene. Diffusion models are generative models that learn to generate new samples through learning to reverse a Gaussian perturbation process. The policy $\pi_\theta(\bar{a}|o)$ is formulated as a score-based diffusion model that can be used to successively denoise actions generated from Gaussian noise back to the data manifold. This perturbation and its inverse process can be expressed with the following Stochastic Differential Equation

$$\mathrm{d}\bar{a} = \big(\beta_t\sigma_t - \dot{\sigma}_t\big)\sigma_t\nabla_a \log p_t(\bar{a}|o,\mathbf{g})dt + \sqrt{2\beta_t}\sigma_t d\omega_t, \tag{1}$$

where β_t determines the noise injection rate at diffusion time step t, $d\omega_t$ represents infinitesimal Gaussian noise, and $p_t(\bar{a}|o,\mathbf{g})$ denotes the score function of the diffusion process. During policy sampling (i.e. the reverse process), action samples are guided towards high-density regions of the data distribution. To learn this score, we train a neural network D_θ via score matching (Vincent, 2011):

$$\mathcal{L}_{\text{SM}} = \mathbb{E}_{\sigma,\bar{a},\boldsymbol{\epsilon}}\big[\alpha(\sigma_t)|D_\theta(\bar{a}+\boldsymbol{\epsilon},o,\mathbf{g},\sigma_t) - \bar{a}|_2^2\big], \tag{2}$$

where $D_\theta(\bar{a}+\boldsymbol{\epsilon},o,\sigma_t)$ represents our trainable neural architecture.

After training, we can generate new action sequences beginning with Gaussian noise by iteratively denoising the action sequence with a numerical ODE solver. Our approach utilizes the DDIM-solver, a specialized numerical ODE-solver for diffusion models (Song et al., 2021) that enables efficient action denoising in a minimal number of steps.

3.3 Point Clouds

Given a set of depth images from M cameras $D^{(0)},\dots,D^{(M)} \in \mathbb{R}^{W\times H}$ as well as the intrinsic matrices $K_{\text{int}}^{(0)},\dots,K_{\text{int}}^{(M)} \in \mathbb{R}^{3\times 3}$, we first construct point clouds $X^{(j)} \in \mathbb{R}^{WH\times 3}$ in each camera's local coordinate frame via unprojection:

$$X_{iW+j}^{(m)} = (K_{\text{int}}^{(m)})^{-1}\left(i\cdot D_{i,j}^{(m)}, j\cdot D_{i,j}^{(m)}, D_{i,j}^{(m)}\right)^T \tag{3}$$

By multiplying each point cloud with its corresponding extrinsic matrix, we can transform it from the camera coordinate frame to the world frame. The final point cloud $\mathbf{X}$ is obtained by concatenating point clouds from all M views.

We treat point clouds as graphs, where the coordinates XYZ are the node features $\mathbf{x}^0$. This allows us to formulate the point cloud encoder as a message-passing Graph Neural Network (GNN) (Scarselli et al., 2009), a flexible framework that encompasses numerous well-known architectures. Each step l computes new node features

$$\mathbf{x}_i^l = h_\theta^l(\mathbf{x}_i^{l-1}, \bigoplus_{j \in \mathcal{N}(i)} h_\phi^l(\mathbf{x}_i^{l-1}, \mathbf{x}_j^{l-1})), \tag{4}$$

where $\mathcal{N}(i)$ is the neighborhood of point i, e.g., i's k nearest neighbors or, in the case of a complete graph, all other nodes. The permutation invariant aggregation $\bigoplus$ can be instantiated as a sum, max, or mean, and h_θ^l and h_ϕ^l denote learnable parametrized functions. After the final step, the tokenized embedding of the observed point cloud $\{\mathbf{T}_i\} \in \mathbb{R}^{n \times d}$ is a subset of the node features $\mathbf{T}_i = \mathbf{x}_S^L$, where $S = f_{\text{sampling}}(\cdot) \subseteq [n]$ are indices selected by some sampling function. Each token may optionally be augmented with a positional encoding $\mathbf{T}_i \leftarrow \mathbf{T}_i + \text{PE}_\psi(\mathbf{x}_i^0)$ based on the cartesian coordinates of the corresponding point, where PE_ψ represents some (potentially parametric) function.

3.4 Fourier Feature Mapping

Despite the fact that neural networks are universal function approximators (Hornik et al., 1989), they are biased toward learning low-frequency components first, while high-frequency components converge slowly or may not be learned at all (Rahaman et al., 2019; Tancik et al., 2020). However, an imitation learning policy parametrized by a neural network may need to learn a high frequency function to represent a sharp decision boundary, such as whether to reposition a grasped object or insert it. For a diffusion denoising model, this would allow the network to represent a score function that is a high-frequency function of the scene geometry, though not necessarily of the actions. In 3D point clouds, a Fourier feature mapping allows the network to better distinguish nearby points, which have extremely similar features in Cartesian space.

In contrast to previous work that adds Fourier features to specific, novel architectures (Wilcox et al., 2025), we hypothesize that applying a Fourier feature mapping to Cartesian points feature benefits essentially *any* point cloud-based policy. We adopt a NeRF-style, axis-aligned Fourier feature mapping (Mildenhall et al., 2021). Let $\mathbf{p} \in \mathbb{R}^3$ define a Cartesian point. The encoding function $\gamma : \mathbb{R} \to \mathbb{R}^{2L}$ applied to the three coordinate values in $\mathbf{p}$ is defined as

$$\gamma_k(\mathbf{p}) \;=\; \big(\sin(\lambda_k \pi \mathbf{p}), \cos(\lambda_k \pi \mathbf{p})\big), \qquad \lambda_k \;=\; \lambda_{\min} \left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{\frac{k-1}{L-1}}, \qquad k = 1, \ldots, L. \tag{5}$$

The point cloud must be bounded by the interval $[-\lambda_{\max}/2, \lambda_{\max}/2]$, as the Fourier feature mapping is periodic, and points outside this range no longer have unique features.

3.5 Data Augmentation

As shown in (Tancik et al., 2020), the choice of wavelengths is essential, as too short wavelengths can cause the network to overfit on the data, while too long wavelengths do not resolve the spectral bias. Instead of carefully tuning the wavelengths to each task, we choose a consistent set of wavelengths and use data augmentation to train the network to ignore frequencies that do not contain useful information. To achieve this, we apply VariableJitter (Gyenes et al., 2025), which avoids the difficulty of tuning the amplitude of typical Gaussian jitter. While Gaussian jitter applies noise $\epsilon \sim \mathcal{N}(0, \sigma_{\max})$ to each point drawn from the same distribution, VariableJitter samples a $\sigma_{\max}$ for each point cloud from a uniform distribution. This achieves a trade off between augmenting the data to reduce overfitting and ensuring there is no gap between training and testing data.

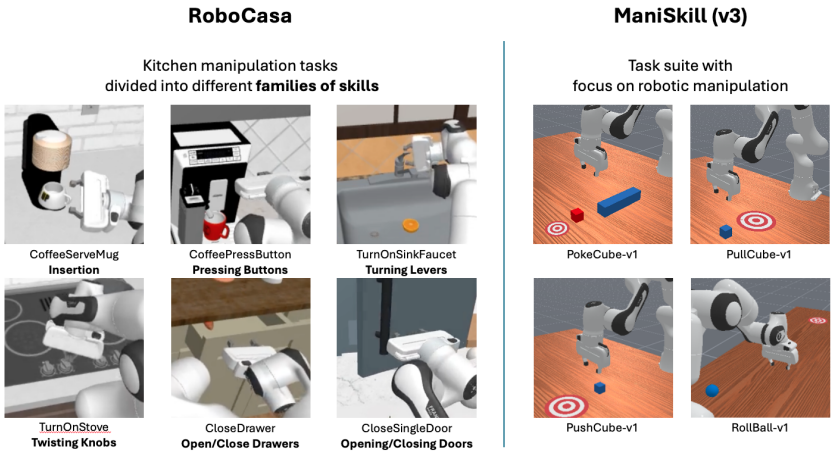

Figure 3: **Overview of all evalution tasks from RoboCasa and ManiSkill3.** For Robocasa, all evaluation tasks are ordered in six groups, resulting in overall 16 unique tasks, shown on the left column. For Maniskill, all evaluated tasks are shown on the right column.

4 Experiments

4.1 Benchmarks and Datasets

We evaluate our approach on two widely used simulation benchmarks, RoboCasa (Nasiriany et al., 2024) and ManiSkill3 (Tao et al., 2025). Figure 3 provides a visualization of our selection of tasks. All models are trained in a multi-task setting, where semantically similar tasks are grouped together in a category. The different task groups are shown in Tables 3 and 7 for RoboCasa and ManiSkill3 respectively.

RoboCasa (Nasiriany et al., 2024) includes high-precision manipulation tasks in visually rich, long-horizon household settings (kitchen scenes). In our study, we focus on 16 tasks that stress fine geometric alignment and contact, which is where spectral bias is most detrimental. For each task we use 50 human-collected demonstrations provided by RoboCasa. Each scene contains two statically-mounted cameras and a gripper camera, and some scenes features varying goal descriptions based on the randomly sampled target object in the scene. More details on RoboCasa can be found in Appendix A.1.

We further test on ManiSkill3 (Tao et al., 2025) to demonstrate our approach on object-centric manipulation with diverse objects. We evaluate on four tasks covering grasping and tool usage under varying viewpoints. Each task has one statically-mounted camera and a fixed goal description. Since the majority of tasks use color information to indicate some aspect of the target, we map the target's Cartesian coordinates to Fourier features and pass this as an additional observation token. We train on 500 (RL-generated) demonstrations from each task in one multi-task dataset. Full details on ManiSkill3 are summarized in Appendix A.1.

Implementation details. We use a fixed log-spaced set of $L = 8$ Fourier bands with wavelengths between $\lambda_{\max} = 4.0$ and $\lambda_{\min} = 0.06$, corresponding to coarse global variation down to fine detail. This results in $D = 3 \times (2L) = 48$ Fourier features for each Cartesian point. Pointclouds in ManiSkill are cropped to include only the relevant part of the scene, which removes the table surface from the observation. We apply voxel downsampling with a voxel size of 0.006 for ManiSkill3 and 0.01 for RoboCasa. We sample a $\sigma_{\max}$ for VariableJitter up to 0.002, which still allows crucial geometric details to be preserved. Camera observations are resized to 128×128 for point cloud-based policies and 224×224 for point map-based policies. In order to highlight the effect of Fourier features on 3D representations, we do not include color features in observations.

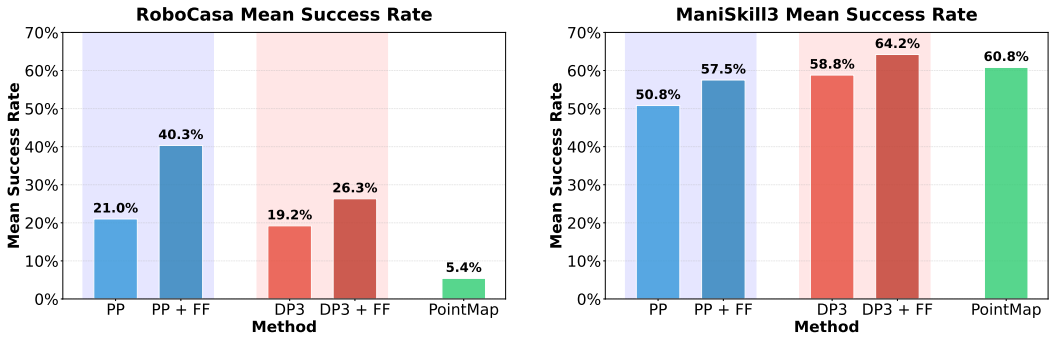

Figure 4: Mean success rate across all tasks of 3D encoders with and without Fourier features on RoboCasa (left) and ManiSkill3 (right).

4.2 Baselines

PointPatch We instantiate the point cloud encoder described in Equation 4 with two concrete architectures. The first is the common "point patch" (PP) architecture (Pang et al., 2022; Yu et al., 2022; Gyenes et al., 2024). From a pointcloud with N points, we wish to construct $c = \frac{N \cdot r}{k}$ patches $\mathbf{P}$ with k points each, where r is the oversampling ratio. We sample c centroids $\mathbf{C} \in \mathbb{R}^{c \times 3}$ from the point cloud $\mathbf{X}$ using Farthest Point Sampling (FPS) (Qi et al., 2017b), which ensures broad coverage and an even spatial distribution. For each centroid, we then identify its k nearest neighbors via a kNN search, yielding point patches $\mathbf{P} \in \mathbb{R}^{c \times k \times 3}$. Each patch is normalized by subtracting its centroid coordinate. Finally, a lightweight PointNet (Qi et al., 2017b) encoder, composed of two MLP layers and two max-pooling layers, transforms the patches into tokens $\mathbf{T} \in \mathbb{R}^{c \times D}$ of dimension D.

DP3 Encoder. Secondly, we evaluate the DP3 encoder proposed by Ze et al. (2024). Unlike patch-based methods, DP3 creates a single token that embeds information from the entire point cloud. Point features are passed through a multi-layer perceptron, followed by a max-pooling operation to obtain order-invariant global feature. A final projection head maps the embedding to the token dimension, resulting in $\mathbf{T} \in \mathbb{R}^{1 \times D}$. Although this architecture is simple, it is quite data efficient due to its small number of parameters.

Pointmap Encoder. To compare against other 3D representations, we also evaluate point maps, which contain the same information as point clouds but are arranged in a 2D grid. Given depth images from multiple cameras and their intrinsics and extrinsics parameters, we unproject each pixel into 3D and transform it into the world frame, resulting in a dense point map $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ for each camera. The resulting 3D representation can be processed directly with convolutional backbones such as ConvNextV2 (Woo et al., 2023) or ResNet (He et al., 2015).

Experiment Setup. Each model is trained for 100 epochs with 3 random seeds, and test performance after the 60th, 80th, and 100th epochs. We measure the average success rate across 20 rollouts and select the best-performing checkpoint for each seed.

5 Results

5.1 Quantitative Results

The results are summarized in Figure 4, which shows the average success rate over all tasks in each task suite. Tables 1 and 2 provide detailed results for RoboCasa and ManiSkill3, respectively.

Our experiments are designed to answer the following research questions:

Do Fourier feature mappings improve the overall performance for 3D inputs? Across both RoboCasa (Table 1) and ManiSkill (Table 2) benchmarks, we observe that Fourier feature mappings boost the success rate on a large majority of individual tasks. In RoboCasa, success rates on individual tasks jumped by as much as 35%. For example CloseDrawer improves from 33.3% to 70.0%,

Table 1: Average success rates on different Robocasa tasks across task categories. Fourier features generally lead to significant improvements for both PointPath and DP3 architectures. In contrast, the image-based PointMap struggles on these tasks, likely due to task complexity and data sparsity.

Category	**Task**	PointPatch	PointPatch + FF	DP3	DP3 + FF	PointMap
Insertion	CoffeeServeMug	0.0±0.0	**5.0±8.7**	3.3±2.9	3.3±2.9	0.0±0.0
	CoffeeSetupMug	0.0±0.0	**3.3±5.8**	0.0±0.0	0.0±0.0	0.0±0.0
Pressing Buttons	CoffeePressButton	18.3±2.9	**38.3±5.8**	18.3±2.9	**28.3±10.4**	8.3±7.6
	TurnOnMicrowave	10.0±5.0	**43.3±23.6**	26.7±12.6	**43.3±12.6**	0.0±0.0
	TurnOffMicrowave	15.0±5.0	**38.3±5.8**	28.3±7.6	**38.3±2.9**	11.6±2.9
Turning Levers	TurnOnSinkFaucet	30.0±13.2	**35.0±8.7**	16.7±7.6	**23.3±5.8**	20.0±10.0
	TurnOffSinkFaucet	41.7±5.8	**66.7±2.9**	43.3±5.8	**61.7±12.6**	13.3±10.4
	TurnSinkSpout	41.7±5.8	**73.3±5.7**	**46.7±2.9**	41.7±2.9	31.7±11.5
Twisting Knobs	TurnOnStove	18.3±10.4	**36.7±2.9**	21.7±5.8	**31.7±10.4**	3.3±2.9
	TurnOffStove	5.0±0.0	**11.7±2.9**	16.7±5.8	16.7±7.6	6.7±2.9
Open/Close Drawers	OpenDrawer	3.3±2.9	**18.3±2.9**	3.3±2.9	**10.0±5.0**	0.0±0.0
	CloseDrawer	33.3±7.6	**70.0±0.0**	40.0±17.3	**53.3±7.6**	3.3±2.9
Open/Close Doors	OpenSingleDoor	25.0*	**40.0±7.0**	6.7±5.8	**11.7±5.8**	0.0±0.0
	CloseSingleDoor	**70.0***	65.0±7.0	23.3±2.9	**41.7±7.6**	0.0±0.0
	OpenDoubleDoor	5.0*	**30.0±14.1**	0.0±0.0	**1.6±2.9**	0.0±0.0
	CloseDoubleDoor	20.0*	**65.0±7.1**	10.0±5.0	**15.0±0.0**	0.0±0.0
Average Success Rate		21.0	**40.0**	19.1	**26.3**	6.1

* Method evaluated on 1 seed

and CloseDoubleDoor improves from 20.0% to 65.0%, while the overall average increases from 21.0% to 40.0%. The results confirm that Fourier mappings help preventing the spectral bias and expose high-frequency geometric cues, making them a robust enhancement across tasks.

Does the effectiveness of Fourier features transfer across different encoders and benchmarks? As shown in RoboCasa (Table 1) and ManiSkill (Table 2) the improvements are not tied to a specific architecture. For PointPatch, Fourier features lead to substantial gains on nearly every task, and the benefit is observed for tasks of any difficulty. For example, the challenging CoffeeServeMug task improves from no success at all to 5.0%, a modest but noticeable gain. DP3 shows a similar trend, although its architecture is substantially different from PointPatch. While it starts with a lower base performance, Fourier features still result in significant improvements for 12 of 16 tasks in RoboCasa and 3 of 4 tasks in ManiSkill.

In comparison, point maps do not perform competitively on RoboCasa, while they only slightly underperform DP3 on the simpler ManiSkill tasks. This may indicate a lower data efficiency of the point map representations trained with convolutional architectures, since we train on 500 demonstrations for each ManiSkill task and only 50 for each RoboCasa task. This is backed up by prior work (Zhu et al., 2024) which shows that simple point cloud encoders outperform point maps.

5.2 Qualitative Results

Qualitatively, we also notice that policies trained on Fourier feature mappings move faster and more decisively, and more closely imitate the demonstration data. Policies trained without Fourier features tend to hesitate before making contact with objects, or behave as if they cannot perceive the scene. In Figure 5, we show representative rollouts from the TurnOnSinkFaucet task. The agent trained with Fourier features makes contact with the faucet at the correct position, requiring accurate perception of the scene, which the agent trained without Fourier features fails the task altogether.

6 Conclusion

In this work we apply the well-known Fourier feature mapping introduced in NeRF (Mildenhall et al., 2021) to a variety of point cloud-based imitation learning methods and test them on high-

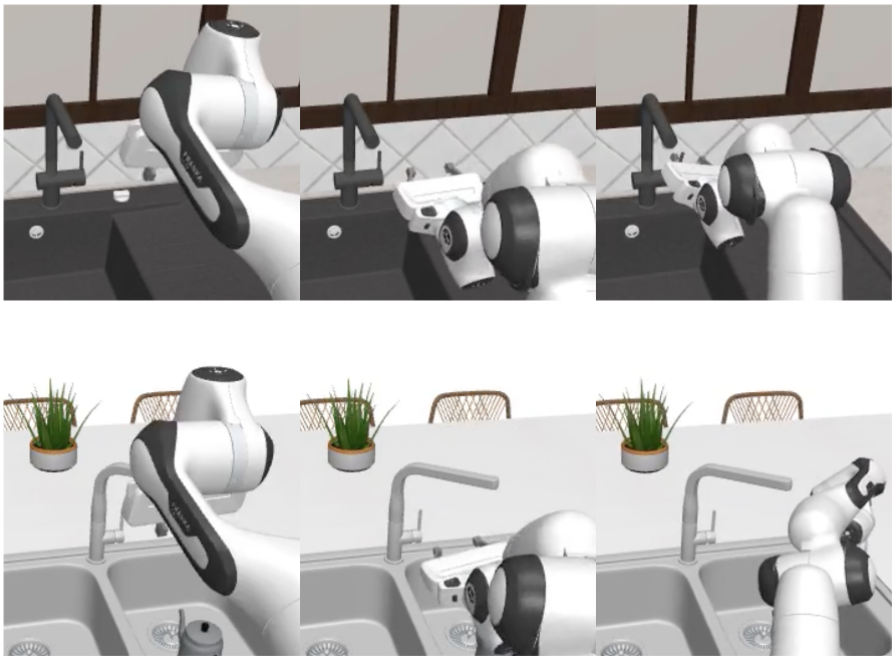

Figure 5: Snapshots over time from policy rollouts on the TurnOnSinkFaucet task in RoboCasa. Top: policy trained with Fourier features executes smooth and accurate rotations. Bottom: baseline policy without Fourier features struggles to complete the motion.

Table 2: Average success rates on different Maniskill tasks for the high-precision Table-Top 2 Finger Gripper category. Similar to the Robocasa results in 1, Fourier features improve the performance of point-cloud based architectures, likely because they enable better differentiation of fine-grained details. For Maniskill, PointMaps are competitive with approaches enhanced with Fourier features, presumably due to larger training datasets.

Category	**Task**	PointPatch	PointPatch + FF	DP3	DP3 + FF	PointMap
Table-Top 2 Finger Gripper	PullCube-v1	56.7±7.6	**63.3±5.8**	**91.7±5.8**	80.0±13.2	63.3±7.6
	PushCube-v1	68.3±7.6	**75.0±5.0**	51.7±16.1	**76.7±16.1**	78.3±7.6
	PokeCube-v1	56.7±12.6	**68.3±5.8**	61.7±2.9	**63.3±2.9**	71.7±10.4
	RollBall-v1	21.7±7.6	23.3±7.6	30.0±13.2	**36.7±2.9**	30.0±5.0
Average Success Rate		50.8	**57.5**	58.8	**64.2**	60.8

precision manipulation tasks. Neural networks are biased towards learning low-frequency functions of their inputs, while ignoring the high-frequency information that is essential for high-precision manipulation, such as insertion tasks or grasping. We demonstrate that Fourier feature mappings provide significant performance benefits for all encoders tested across the vast majority of tasks, for both simpler and more challenging tasks.

Through experiments on RoboCasa and ManiSkill3, we demonstrate that Fourier features consistently improve performance across different 3D input modalities and encoders. On ManiSkill3 tasks, this modification brings point clouds on par with alternative 3D representations such as point maps, while on RoboCasa, they far exceed them. For this reason, we argue that Fourier features should be used almost any point cloud encoder architecture rather than Cartesian point features. Future work may investigate gradient-based learning of the optimal wavelengths or additional regularization to improve scalability.

Ethics Statement

This work introduces Fourier feature projections to enhance 3D modalities for high-precision imitation learning. While our evaluation emphasizes robotic manipulation, the approach is broadly relevant to other domains where spatial reasoning is required. As with many advances in robot learning, the outcomes depend on the context of deployment: increased accuracy and robustness can yield positive impacts in assistive and industrial settings, but also raise risks if applied irresponsibly. We emphasize that the governance of powerful robotic technologies must extend beyond the research community and individual organizations, requiring oversight by public institutions and democratic processes.

Reproducibility Statement

To ensure the reproducibility of our research, we provide a detailed description of our implementation is given in Chapter 4.1, and include a list of hyperparameters in Appendix A.2. The used RoboCasa and ManiSkill datasets are the public available versions which can be found on their corresponding websites. Further information on the datasets used in our experiments can be found in Chapter 4 and in Appendix A.1. Our source code will be released with the final version of the paper.

On LLM Usage

Large language models were employed to refine individual phrases during the writing of the paper, to assist with literature search and exploration, and to aid in code implementation. All outputs from large language models were checked verified by the authors at every stage of the project, including text, literature, and code. We also used them in limited ways for generating illustrative visualizations, but used our own images and material as the basis for these visualizations.

References

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5470–5479, 2022.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Lingdong Chi, Aravind Rajeswaran Gupta, Silvio Savarese, Li Fei-Fei, and Nicholas Rhinehart. Diffusion policy: Visuomotor policy learning via action diffusion. In *International Conference on Learning Representations (ICLR)*, 2023.

Atalay Donat, Xiaogang Jia, Xi Huang, Aleksandar Taranovic, Denis Blessing, Ge Li, Hongyi Zhou, Hanyi Zhang, Rudolf Lioutikov, and Gerhard Neumann. Towards fusing point cloud and visual representations for imitation learning. In *7th Robot Learning Workshop: Towards Robots with Human-Level Abilities*, 2025. URL https://openreview.net/forum?id=5cG7ilWX1V.

Zelin Gao, Weichen Dai, and Yu Zhang. Adaptive positional encoding for bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3284–3294, 2023.

Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023.

Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pp. 694–710. PMLR, 2023.

Balazs Gyenes, Nikolai Franke, Philipp Becker, and Gerhard Neumann. PointpatchRL - masked reconstruction improves reinforcement learning on point clouds. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=3jNEz3kUSl.

Balázs Gyenes, Nikolai Franke, Paul Maria Scheikl, Pit Henrich, Rayan Younis, Gerhard Neumann, Martin Wagner, and Franziska Mathis-Ullrich. Point cloud segmentation for autonomous clip positioning in laparoscopic cholecystectomy on a phantom. *IEEE Robotics and Automation Letters*, 10(8):8522–8529, 2025. doi: 10.1109/LRA.2025.3585357.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

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

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(89)90020-8. URL https://www.sciencedirect.com/science/article/pii/0893608089900208.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=k7FuTOWMOc7.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 1949–1974. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/ke25a.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

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.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European Conference on Computer Vision*, pp. 604–621. Springer, 2022.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017a.

Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017b.

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

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Dräxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron C. Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:53012119.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks, 2019. URL https://openreview.net/forum?id=r1gR2sC9FX.

Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal conditioned imitation learning using score-based diffusion policies. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Chuanhao Sun, Zhihang Yuan, Kai Xu, Luo Mai, N Siddharth, Shuo Chen, and Mahesh K Marina. Learning high-frequency functions made easy with sinusoidal positional encoding. *arXiv preprint arXiv:2407.09370*, 2024.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-Kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh N, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: GPU parallelized robot simulation and rendering for generalizable embodied AI. In *7th Robot Learning Workshop: Towards Robots with Human-Level Abilities*, 2025. URL https://openreview.net/forum?id=GgTxudXaU8.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

Albert Wilcox, Mohamed Ghanem, Masoud Moghani, Pierre Barroso, Benjamin Joffe, and Animesh Garg. Adapt3r: Adaptive 3d scene representation for domain transfer in imitation learning. *CoRR*, abs/2503.04877, March 2025. URL https://doi.org/10.48550/arXiv.2503.04877.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. *arXiv preprint arXiv:2301.00808*, 2023.

Runzhe Wu, Yiding Chen, Gokul Swamy, Kianté Brantley, and Wen Sun. Diffusing states and matching scores: A new framework for imitation learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=kWRKNDU6uN.

Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19313–19322, 2022.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

Yanjie Ze, Zixuan Chen, Wenhao Wang, Tianyi Chen, Xialin He, Ying Yuan, Xue Bin Peng, and Jiajun Wu. Generalizable humanoid manipulation with 3d diffusion policies, 2025. URL https://arxiv.org/abs/2410.10803.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016.

Haoyi Zhu, Yating Wang, Di Huang, Weicai Ye, Wanli Ouyang, and Tong He. Point cloud matters: Rethinking the impact of different observation spaces on robot learning. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=zgSnSZ0Re6.

Minjie Zhu, Yichen Zhu, Jinming Li, Junjie Wen, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, et al. Scaling diffusion policy in transformer to 1 billion parameters for robotic manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10838–10845. IEEE, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

A Appendix

A.1 Experiment Details

Figure 6: **Overview of RoboCasa Simulation Environments.** Example kitchen scenes and tasks illustrating the diversity of household manipulation settings provided by RoboCasa.

RoboCasa benchmark. RoboCasa (Nasiriany et al., 2024) is a large-scale simulation benchmark designed for training generalist robots in realistic household settings, with an emphasis on kitchen environments. It provides 100 tasks in total: 25 atomic tasks with 50 human demonstrations each, and 75 composite tasks with automatically generated demonstrations. The task set covers eight fundamental skills that are essential for home robotics: (1) pick-and-place, (2) door opening and closing, (3) drawer opening and closing, (4) knob turning, (5) lever manipulation, (6) button pressing, (7) insertion, and (8) navigation. To evaluate our method, we selected 16 tasks from the atomic tasks described in Table 3, each representing a different skill. The joint action space is 7-dimensional, including end-effector translation, rotation, and gripper control.

Table 3: RoboCasa evaluation tasks.

Category	Task	Description
Insertion	CoffeeServeMug	Remove the mug from the holder and place it on the counter.
	CoffeeSetupMug	Place the mug into the coffee machine's mug holder.
Pressing Buttons	CoffeePressButton	Press the button to pour coffee into the mug.
	TurnOnMicrowave	Start the microwave by pressing the start button.
	TurnOffMicrowave	Stop the microwave by pressing the stop button.
Turning Levers	TurnOnSinkFaucet	Turn on the sink faucet to start water flow.
	TurnOffSinkFaucet	Turn off the sink faucet to stop water flow.
	TurnSinkSpout	Rotate the sink spout.
Twisting Knobs	TurnOnStove	Turn on a specific stove burner by twisting its knob.
	TurnOffStove	Turn off a specific stove burner by twisting its knob.
Open/Close Drawers	OpenDrawer	Open a drawer.
	CloseDrawer	Close a drawer.
Opening and Closing Doors	OpenSingleDoor	Open a microwave door or a cabinet with a single door.
	CloseSingleDoor	Close a microwave door or a cabinet with a single door.
	OpenDoubleDoor	Open a cabinet with two opposite-facing doors.
	CloseDoubleDoor	Close a cabinet with two opposite-facing doors.

ManiSkill3 ManiSkill3 (Tao et al., 2025) is a large-scale GPU-parallelized simulation benchmark designed for scalable training of embodied agents. It offers diverse object-centric manipulation tasks such as grasping, assembling, and tool use, with support for both imitation and reinforcement learning. Unlike RoboCasa, which emphasizes long-horizon household tasks in visually rich kitchen

environments, ManiSkill3 provides highly parallelized simulation and rendering of physics-based interactions, enabling efficient large-scale experimentation and evaluation of manipulation policies.

A summary of all ManiSkill3 tasks can be found in Table 4, each representing a distinct skill.

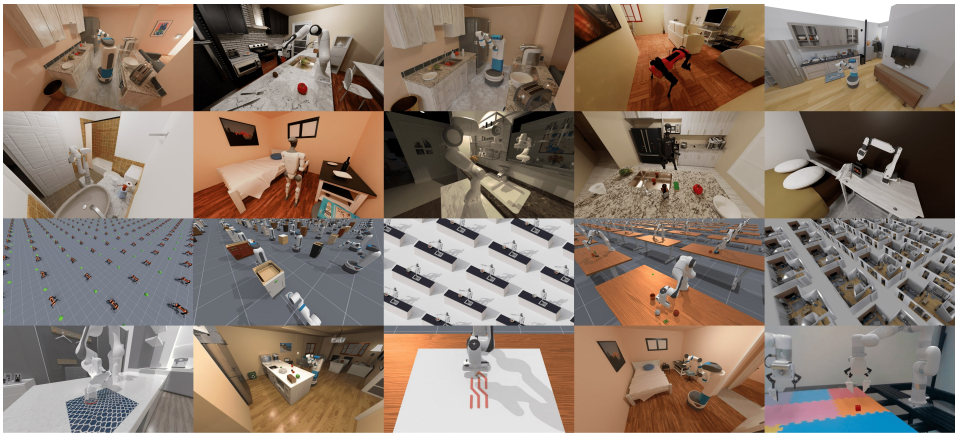

Figure 7: **Overview of ManiSkill3 Simulation Environments.** Example object-centric manipulation tasks illustrating the diversity of interactions supported by ManiSkill3.

Table 4: ManiSkill3 evaluation tasks.

Category	Task	Description
Table-Top 2 Finger Gripper	PullCube-v1	A task where the objective is to pull a cube onto a target.
	PushCube-v1	A task where the objective is to push and move a cube to a goal region in front of it.
	PokeCube-v1	A task where the objective is to poke a red cube with a peg and push it to a target goal position.
	RollBall-v1	A task where the objective is to push and roll a ball to a goal region at the other end of the table.

A.2 HYPERPARAMETERS

Table 5: Summary of the Hyperparameters for all of our experiments.

Hyperparameter	**ManiSkill**	**RoboCasa**
Number of Attention Blocks	4	4
Attention Heads	4	4
Action Chunk Size	10	20
History Length	1	1
Embedding Dimension	256	256
Goal Lang Encoder	CLIP Resnet-50	CLIP Resnet-50
Attention Dropout	0.3	0.3
Residual Dropout	0.1	0.1
MLP Dropout	0.1	0.1
Optimizer	AdamW	AdamW
Betas	[0.9, 0.9]	[0.9, 0.9]
Learning Rate	1e-4	1e-4
Weight Decay	0.05	0.05
$\sigma_{\max}$	80	80
$\sigma_{\min}$	0.001	0.001
σ_t	0.5	0.5
EMA decay	0.995	0.995
Time steps	Exponential	Exponential
Sampler	DDIM	DDIM
Denoising Steps	10	10