TOWARDS FUSING POINT CLOUD AND VISUAL REPRESENTATIONS FOR IMITATION LEARNING

Atalay Donat[*†] **Xiaogang Jia** [*†] **Xi Huang**[†] **Aleksandar Taranovic**[†] **Denis Blessing**[†] **Ge Li**[†] **Hongyi Zhou**[†] **Hanyi Zhang**[‡] **Rudolf Lioutikov**[†] **Gerhard Neumann**[†]
[†] Karlsruhe Institute of Technology [‡] University of Liverpool
[*] Equal contribution, correspondence to jia266163@gmail.com

ABSTRACT

Learning for manipulation requires using policies that have access to rich sensory information such as point clouds or RGB images. Point clouds efficiently capture geometric structures, making them essential for manipulation tasks in imitation learning. In contrast, RGB images provide rich texture and semantic information that can be crucial for certain tasks. Existing approaches for fusing both modalities assign 2D image features to point clouds. However, such approaches often lose global contextual information from the original images. In this work, we propose a novel imitation learning method that effectively combines the strengths of both point cloud and RGB modalities. Our method conditions the point-cloud encoder on global and local image tokens using adaptive layer norm conditioning, leveraging the beneficial properties of both modalities. Through extensive experiments on the challenging RoboCasa benchmark, we demonstrate the limitations of relying on either modality alone and show that our method achieves state-of-the-art performance across all tasks.

1 INTRODUCTION

Imitation Learning (IL) has become a fundamental approach in robotic learning Brohan et al. (2022); Chi et al. (2023); Zhao et al. (2023); Black et al. (2024); Kim et al. (2024), allowing agents to acquire complex behaviors by mimicking expert demonstrations. IL can additionally benefit from contextual information that provides task description, therefore reducing the need for inferring task goal from the demonstrations Ding et al. (2019). A crucial aspect of IL is the choice of the used input representation, as it directly impacts the agent's ability to generalize and make informed decisions. RGB images are a common input modality because they offer rich texture and semantic information that can be critical for tasks involving object recognition and contextual reasoning Mandlekar et al. (2021); Reuss et al. (2024b); Liu et al. (2024). Additionally, they are easy to obtain and relatively cheap, making them a practical choice in many scenarios. Another input modality is a point cloud Zhu et al. (2024); Ze et al. (2024); Ke et al. (2024), which provides us with geometric information. Point cloud representations have proven highly effective for robotic manipulation due to their ability to directly encode 3D spatial structures. A further modality are language instructions. They contain relevant task context Stepputtis et al. (2020); Li et al. (2023); Reuss et al. (2024b), such as human understandable task descriptions. All these input types provide different benefits and limitations in the learning process, and we should fuse them appropriately to extract all the individual benefits, while offsetting the limitations. Therefore, fusing different modalities is a relevant but challenging problem.

In this paper, we focus on the fusion of RGB images and point clouds while also taking language instructions into account. Despite their complementary nature, integrating these RGB images and point clouds remains a significant challenge in IL. Existing approaches Gervet et al. (2023); Shridhar et al. (2023); Ze et al. (2024) primarily attempt to assign 2D visual features to point clouds, thereby incorporating RGB information into 3D representations. However, such strategies often fail to retain the global contextual information from images, leading to suboptimal performance in tasks that require both precise spatial reasoning and high-level semantic understanding. As a result, neither modality alone—nor naïve fusion techniques—achieves universally strong performance across diverse imitation learning benchmarks. Yet, more recent approaches of combining modalities such as adaptive conditioning in Layer-Norm layers Peebles & Xie (2022) has not yet been explored in the imitation learning context, even though it allows a more flexible sensor fusion scheme.

To address this limitation, we introduce **F**usion of **P**oint Cloud and **V**isual Representation **Net**work (FPV-Net), a novel imitation learning method designed to effectively align and balance the strengths of both point cloud and RGB images. Our approach leverages novel conditioning methods for sensor fusion Peebles & Xie (2022) and ensures that the geometric precision of point clouds is preserved while leveraging the global semantic richness of RGB inputs, enabling a more robust and generalizable policy learning process. For the extraction of representations from RGB images, we use a neural network based on the FiLM-ResNet architecture Perez et al. (2018). This extraction process is conditioned on the language instruction, thus effectively incorporating this modality into our method. Moreover, we make use of both local features and global features, which we show to be critical for the manipulation tasks. To extract data from point clouds, we apply Furthest Point Sampling Eldar et al. (1994) and k-Nearest Neighbors, that are then encoded into learned embeddings. For fusing the modalities, we explore 3 different approaches, and show that fusing Point Cloud and Language as main modalities while using RGB images as the conditional modality using AdaLN conditioning Peebles & Xie (2022) performs best. Figure 1 illustrates how FPV-Net extracts features from different modalities.

We evaluate FPV-Net on RoboCasa (Nasiriany et al., 2024), a challenging benchmark for robotic manipulation. We conduct extensive experiments to analyze the impact of different input modalities. Our results indicate that neither point clouds nor RGB images alone provide optimal performance across all tasks, whereas naïve fusion methods often degrade performance due to poor alignment between modalities. FPV-Net consistently outperforms state-of-the-art approaches Ke et al. (2024); Ze et al. (2024) across all tasks, establishing a new benchmark in multimodal imitation learning.

To summarize, our main contributions are threefold. First, we conduct systematic experiments on RoboCasa, showing that neither RGB images nor point clouds alone are sufficient for strong performance, as each modality excels in some tasks but performs poorly in others. Second, we introduce FPV-Net, a diffusion-based multi-modal imitation learning method that leverages point cloud inputs as the main modality and visual inputs as a conditional modality, integrated via AdaLN conditioning Peebles & Xie (2022), while also incorporating language instructions for contextual guidance. FPV-Net achieves state-of-the-art performance across most tasks, and, to our knowledge, using AdaLN to fuse point cloud and RGB modalities is a novel insight. Third, we demonstrate the critical role of local RGB features in fine-grained robotic manipulation tasks, showing that integrating both global and local features significantly enhances model performance.

2 Related Works

Visual Imitation Learning. Recent state-of-the-art imitation learning methods Chi et al. (2023); Reuss et al. (2024a); Kim et al. (2024); Liu et al. (2024); Li et al. (2025) often use 2D images as state representation due to their rich global information and ease of acquisition from raw sensory inputs. However, 2D images lack explicit 3D information such as precise 3D coordinates and object geometry Zhu et al. (2024), which are crucial for many robotic manipulation tasks. While using multiple camera views can partially mitigate this drawback, it requires significantly more training data to infer the 3D spatial information effectively Ze et al. (2024). Moreover, image-based policies struggle with occlusions and viewpoint variations Peri et al. (2024), making generalization across diverse environments challenging.

Imitation Learning with 3D Scene Representation. An alternative approach is to leverage 3D scene representations, such as point cloud Zhu et al. (2024); Ze et al. (2024); Ke et al. (2024), which provide explicit spatial structure and thus enable better spatial reasoning. However, using point clouds usually requires down-sampling Eldar et al. (1994), leading to loss of fine-grained information from the raw sensory data. Recently, several studies Shridhar et al. (2023); Gervet et al. (2023); Ke et al. (2024) have investigated how to effectively incorporate both 2D and 3D representations into imitation learning. For instance, Act3D Gervet et al. (2023) generates feature clouds using multi-view RGB images and depth information. 3D Diffuser Actor Ke et al. (2024) lifts ResNet features to 3D using the depth map. Unlike these approaches, FPV-Net introduces a novel 2D-3D fusion strategy by conditioning Transformer policy with 2D images from multiple views while processing tokenized 3D representations, enabling better generalization and spatial reasoning.

Multi-modal Sensory Fusion in Imitation Learning. Most existing research on multi-modal sensory fusion in imitation learning focuses on combining image observations with language goal conditioning. A common strategy is to treat image and language inputs as separate tokens within a Transformer and train the policy from scratch Reuss et al. (2024b); Bharadhwaj et al. (2024). Another line of research leverages large pre-trained Vision-Language Models (VLMs) and fine-tunes them with demonstrations to create Vision-Language-Action (VLA) models Cheang et al. (2024); Kim et al. (2024); Black et al. (2024). However, these methods predominantly rely on 2D image features, which limits their effectiveness when working with small datasets or tasks requiring detailed spatial reasoning. In the contrary, FPV-Net fuses 2D and 3D observations, enabling more efficient multi-modal learning.

3 Preliminaries

3.1 Problem Formulation

Imitation learning (IL) aims to train an agent to perform tasks by learning from expert demonstrations. Given a dataset of expert trajectories $\mathcal{D} = \{(\tau_i)\}_{i=1}^{N}$, where each trajectory τ_i consists of a sequence of observations and corresponding expert actions

$$\tau_i = (\mathbf{o}_1, \mathbf{a}_1, \mathbf{o}_2, \mathbf{a}_2, \ldots, \mathbf{o}_K, \mathbf{a}_K), \tag{1}$$

the goal is to learn a policy $\pi(\mathbf{a}|\mathbf{o}) : \mathcal{O} \rightarrow \mathcal{A}$ that maps observations to actions in a manner that mimics expert behavior.

3.2 Multi-Modal Imitation Learning

In a multi-modal imitation learning framework, the agent receives a multi-modal observation at each time step k consisting of:

Language instruction ($\mathbf{x}_k^L$): Provides high-level task semantics and contextual guidance, enabling the agent to generalize across diverse instructions.

RGB image ($\mathbf{x}_k^I$): Captures visual scene information, including object appearances, spatial arrangements, and environmental semantics.

Point cloud ($\mathbf{x}_k^P$): Offers a structured 3D representation of the environment, encoding geometric and spatial relationships that are crucial for manipulation.

Thus, an observation in the framework is defined as

$$\mathbf{o} = (\mathbf{x}_k^L, \mathbf{x}_k^I, \mathbf{x}_k^P) \in \mathcal{O}, \tag{2}$$

where $\mathcal{O}$ denotes the observation space. Building on the success of Action Chunking Zhao et al. (2023) in Imitation Learning, we formulate the objective as predicting a sequence of future actions

$$\mathbf{a} = (\mathbf{a}_k, \mathbf{a}_{k+1}, \ldots, \mathbf{a}_{k+H}) \in \mathcal{A}^H, \tag{3}$$

where H is the prediction horizon, and $\mathcal{A}$ denotes the action space.

3.3 Score-Based Diffusion Policies

FPV-Net adopts the continuous-time denoising diffusion model from EDM Karras et al. (2022) to represent the policy. Denoising diffusion models aim to time-reverse a stochastic noising process that transforms the data distribution into Gaussian noise Song et al. (2020), allowing for generating new samples that are distributed according to the data. In FPV-Net, a score-based diffusion model is used for the policy $\pi(\mathbf{a}|\mathbf{o})$. The denoising process is governed by a stochastic differential equation (SDE) given by

$$\mathrm{d}\mathbf{a} = \big(\beta_t \sigma_t - \dot{\sigma}_t\big)\sigma_t \nabla_{\mathbf{a}} \log p_t(\mathbf{a}|\mathbf{o})\mathrm{d}t + \sqrt{2\beta_t}\sigma_t \mathrm{d}B_t, \tag{4}$$

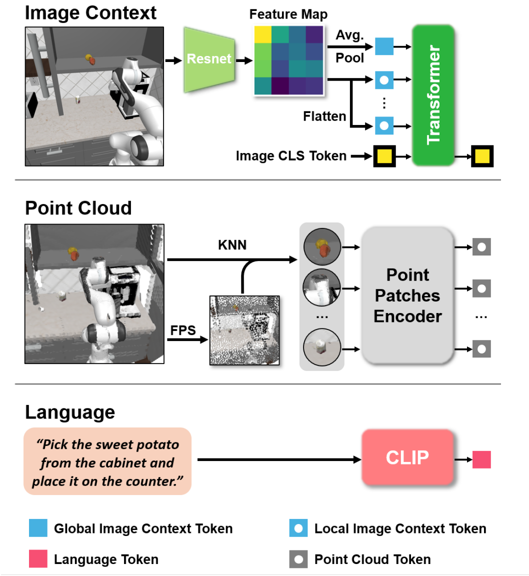


Figure 1: Processing each input modality to generate corresponding embeddings. **Top**: A FiLM-ResNet architecture is used to extract a feature map from the context image. The feature map is processed through average pooling and flattening to obtain global and local feature tokens, which are then concatenated and fed into the transformer along with a learnable CLS token, whose output is used as a condition vector for the diffusion policy (Figure 2). **Middle**: The point cloud input is processed by applying FPS to sample points, followed by KNN to group point patches using these FPS points as centers. The resulting patches are passed through a point patches encoder, which can be a lightweight MLP or the pretrained SUGAR model. **Bottom**: The CLIP model is employed to generate the language embedding for the behavior prompt.

where β_t determines how much noise is injected, B_t denotes a standard Wiener process, and $p_t(\mathbf{a}|\mathbf{o})$ is the score function of the diffusion process which moves samples towards regions of high data density. To generate new samples from noise, one trains a neural network to approximate $\nabla_{\mathbf{a}} \log p_t(\mathbf{a}|\mathbf{o})$ using Score Matching (SM) Vincent (2011). The SM objective is

$$\mathcal{L}_{D_\theta} = \mathbb{E}_{\sigma_{\mathbf{t}}, \mathbf{a}, \boldsymbol{\epsilon}} \left[\alpha(\sigma_t) \| D_\theta(\mathbf{a} + \boldsymbol{\epsilon}, \mathbf{o}, \sigma_t) - \mathbf{a} \|_2^2 \right], \tag{5}$$

where $D_\theta(\mathbf{a} + \boldsymbol{\epsilon}, \mathbf{o}, \sigma_t)$ is the trainable network. During training, noise is sampled from a predefined distribution and added to an action sequence. The network then predicts the denoised actions and computes the SM loss. Once training is complete, new action sequences can be generated by starting from random noise and approximating the reverse SDE in discrete steps using a numerical ODE solver. Specifically, one samples an initial action $\mathbf{a}_t \sim \mathcal{N}(0, \sigma_t^2 I)$ from the prior and progressively denoises it. In FPV-Net, this is accomplished via the DDIM-solver Song et al. (2020), which is an ODE solver tailored for diffusion models that can denoise actions in just a few steps. In all experiments, FPV-Net uses 4 denoising steps.

4 Method

Fusion of Point Cloud and Visual representation Network (FPV-Net) is a multi-modal transformer-based diffusion policy which leverages point cloud, image and language inputs. In this section, we introduce how we process these different modalities and propose three different fusion methods to combine point cloud features and image features. An overview of our model is shown in Figures 1 and 2.

4.1 Image Processing

To extract meaningful representations from RGB inputs, we utilize a FiLM-ResNet architecture Perez et al. (2018), which is conditioned on the language instructions. This approach allows the

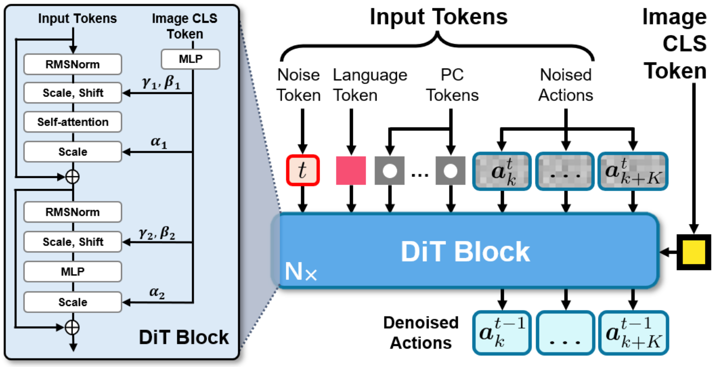


Figure 2: Conditioned on image CLS tokens, the transformer-based diffusion policy (DiT block) denoises action chunk tokens by utilizing 3D point cloud tokens and language tokens as inputs. The conditioning process is detailed within the structure of the DiT block.

model to modulate feature extraction based on linguistic context, improving the alignment between vision and language modalities. Most prior works Chi et al. (2023); Zhao et al. (2023); Reuss et al. (2024b) in imitation learning extract only a global token from ResNet He et al. (2016) feature maps, discarding fine-grained local spatial information. However, we argue that both global and local features are critical for capturing fine-grained visual details necessary for action prediction. To address this, we extract features as follows:

Global Token: We apply global average pooling over the ResNet feature map to obtain a single global representation.

Local Tokens: Instead of discarding spatial features, we flatten the feature map into a sequence of local tokens, preserving important spatial details.

Finally, we concatenate the global token with the local feature tokens, forming a comprehensive visual representation

$$\mathbf{z}_t^I = \text{Concat}(\text{AvgPool}(F_{\text{ResNet}}(\mathbf{I})), \text{Flatten}(F_{\text{ResNet}}(\mathbf{I}))), \tag{6}$$

where $F_{\text{ResNet}}(\mathbf{I})$ denotes the extracted feature map from FiLM-ResNet. This enriched representation provides the policy with a multi-scale visual understanding, ensuring that both high-level semantics and fine-grained local details contribute to decision-making. The illustration of the image processing can be found in Figure 1.

4.2 Point Cloud Processing

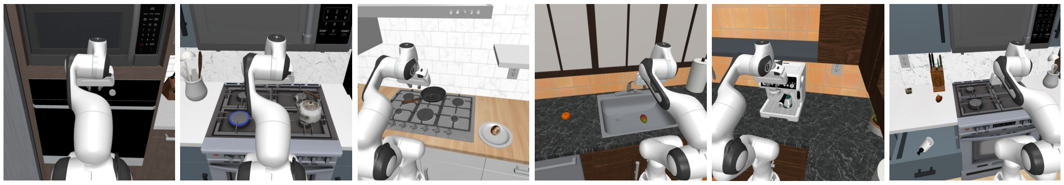
Prior approaches in 3D imitation learning, such as 3D Diffusion Policy (DP3) Ze et al. (2024) and 3D Diffuser Actor (3DA) Ke et al. (2024), suffer from key limitations. DP3's max pooling discards local geometric features, while 3DA's 2D feature lifting loses global contextual information from original images. Moreover, 3DA generates an excessive number of point tokens, leading to higher computational costs. To effectively process a point cloud $\mathbf{x}_t^P \in \mathbb{R}^{N\times 3}$ consisting of N points in 3D space, we construct a structured representation as follows:

Furthest Point Sampling (FPS) Eldar et al. (1994); Qi et al. (2017a): We sample $M = 256$ center points, ensuring a coverage of the global geometric structure.

k-Nearest Neighbors (KNN) Qi et al. (2017b): For each center point, we retrieve its $K = 32$ nearest neighbors, forming local point groups that capture fine-grained spatial structures.

Each local point group is encoded into a latent representation using a point cloud encoder ψ_θ. The final point cloud embedding is represented as

$$\mathbf{z}_t^P = \{\psi_\theta(\mathbf{G}_m)\}_{m=1}^M, \quad \mathbf{z}_t^P \in \mathbb{R}^{M\times d}, \tag{7}$$



"open the microwave door"

"turn off the front left burner of the stove"

"pick the mushroom from the plate and place it in the pan"

"turn on the sink faucet"

"press the button on the coffee machine to serve coffee"

"close the left drawer"

Figure 3: Example scenarios from the RoboCasa benchmark Nasiriany et al. (2024) used in our experiments.

where $\mathbf{G}_m \in \mathbb{R}^{K\times 3}$ represents the K-neighbor subset for the m-th sampled center, $\psi_\theta(\cdot)$ is the point cloud encoder that extracts a per-group embedding, and $\mathbf{z}_t^P$ consists of $M = 256$ tokens, each of dimension d. By structuring the point cloud representation into a tokenized format, our approach preserves both local fine-grained features and global contextual information, ensuring a more expressive representation for 3D imitation learning. We explore two different point cloud encoding strategies:

Lightweight MLP Encoder: Inspired by 3D Diffusion Policy Ze et al. (2024), we use a multi-layer perceptron (MLP) followed by a max pooling layer to process each point group independently. This method is computationally efficient and preserves local structures.

Pretrained SUGAR Model: We leverage a pretrained point cloud encoder, SUGAR Chen et al. (2024), to extract richer and more informative features, benefiting from knowledge gained in large-scale 3D datasets.

4.3 Fusing Multi-Modal Embeddings

To effectively integrate multi-modal observations, including RGB images, point clouds, and language embeddings, we explore three different fusion strategies for combining image and point cloud features. In the following, other than the image embedding $\mathbf{z}_t^I$ and the point cloud embedding $\mathbf{z}_t^P$, we use $\mathbf{z}_t^L \in \mathbb{R}^{d_L}$ to denote language embeddings, which are obtained via the frozen CLIP model Radford et al. (2021).

4.3.1 Concatenation-Based Fusion

A straightforward approach is to directly concatenate the embeddings of these three modalities and use it as input for the transformer policy. This fused representation $\mathbf{z}_t^{\text{fusion}}$ can be written as

$$\mathbf{z}_t^{\text{fusion}} = \text{Concat}(\mathbf{z}_t^I, \mathbf{z}_t^P, \mathbf{z}_t^L). \tag{8}$$

Although this fusion retains all feature information, it lacks a structured interaction between modalities.

4.3.2 Adaptive LayerNorm Conditioning

Inspired by the use of Adaptive LayerNorm (AdaLN) layers to condition on classes in DiT models Peebles & Xie (2022), we explore using AdaLN conditioning layers not on language, but on the point cloud or the image modality. In this way, AdaLN conditioning treats one modality as conditioning input and the other modalities as main feature inputs. The conditioning inputs scale or shift main feature within the attention mechanism

$$\text{AdaLN}(\mathbf{z} \mid \mathbf{c}) = \gamma(\mathbf{c}) \odot \frac{\mathbf{z} - \mu(\mathbf{z})}{\sigma(\mathbf{z})} + \beta(\mathbf{c}),$$

where $\mathbf{z}$ is the main feature, $\mathbf{c}$ is the conditioning input, $\mu(\mathbf{z})$ and $\sigma^2(\mathbf{z})$ are the mean and variance of the main input $\mathbf{z}$, and $\gamma(\mathbf{c})$ and $\beta(\mathbf{c})$ are learnable functions that map the conditioning input to a pair of scale and shift parameters. More details about AdaLN conditioning can be found in Appendix D.

Image and Language as Main Modality In this setup, we select the image embeddings $\mathbf{z}_t^I$ and language embeddings $\mathbf{z}_t^L$ as the primary modality. The AdaLN layers take the point cloud embeddings $\mathbf{z}_t^P$ as conditions to modulate the activation of the primary modality. The fusion is formulated as

$$\mathbf{z}_t^{\text{fusion}} = \text{AdaLN}(\mathbf{z}_t^I, \mathbf{z}_t^L | \mathbf{z}_t^P). \tag{9}$$

Point Cloud and Language as Main Modality Alternatively, we consider using point cloud embedding and language embedding as primary modality and image embedding as conditions

$$\mathbf{z}_t^{\text{fusion}} = \text{AdaLN}(\mathbf{z}_t^P, \mathbf{z}_t^L | \mathbf{z}_t^I). \tag{10}$$

The observation embedding $\mathbf{z}_t^{\text{fusion}}$ will then be fed into the transformer-based diffusion policy (Figure 2).

5 Experiments

We conduct extensive experiments to answer the following questions:

Q1) Is a single modality enough to perform efficiently on challenging environments?

Q2) How does our method compare with state-of-the-art imitation learning policies?

Q3) What kinds of fusion types are most powerful?

5.1 Simulations

RoboCasa Nasiriany et al. (2024): RoboCasa is a large-scale simulation framework designed to train generalist robots in diverse and realistic household environments, with a particular emphasis on complex kitchen tasks. It features 120 meticulously crafted kitchen scenes, over 2,500 high-quality 3D objects across 150 categories, and 100 tasks divided into foundational atomic tasks and intricate composite tasks. Leveraging generative AI tools, RoboCasa achieves unparalleled diversity, realism, and scalability in robotic learning. This benchmark is characterized by its exceptional difficulty, stemming from the highly diverse scenarios it presents. Each scenario is accompanied by only one demonstration, significantly increasing the challenge for learning algorithms. For instance, in pick-and-place tasks, the object to be manipulated varies across scenarios, with just one demonstration per case. Furthermore, the training and evaluation datasets feature completely distinct scenes, further testing a model's ability to generalize and adapt robot behaviors to novel scenarios. With its extensive task set, environmental variability, and high-fidelity simulations, RoboCasa establishes itself as a new standard for evaluating robotic learning methodologies, pushing the boundaries of generalization and adaptability in robot learning.

Training and Evaluation: We train each method for 100 epochs and rollout the models for 50 times for all tasks in RoboCasa. We group similar tasks together as shown in Table 4 and train the models for each of the groups.

5.2 Baselines

BC Nasiriany et al. (2024): We inherit the result reported in RoboCasa. RoboCasa uses the BC-Transformer implemented by RoboMimic. The BC policy uses a CLIP model to encode the goal instruction and a ResNet-18 with FilM layers to encode the image-based observations.

3D Diffusion Policy (DP3) Ze et al. (2024): DP3 extracts point-wise features from single-view points clouds with a MLP-based encoder and forms a compact 3D visual representation. Robot actions are then generated by a convolutional network-based architecture, conditioned on this representation and the current robot states.

3D Diffuser Actor (3DA) Ke et al. (2024): 3DA is a diffusion-based policy conditioned on 3D scene features and language instructions. The 3D scene features are extracted and aggregated from single

CATEGORY	TASK	BC	DP3	3DA	PC-ONLY	RGB-ONLY	PC+RGB	FPV-MLP	FPV-SUGAR
PICK AND PLACE	PNPCABTOCOUNTER	0.02	0.04	0.00	0.02	0.00	0.04	**0.16**	0.10
	PNPCOUNTERTOCAB	0.06	0.02	0.00	0.00	0.00	0.08	0.08	**0.14**
	PNPCOUNTERTOMICROWAVE	0.02	0.06	0.00	0.00	0.02	0.10	**0.26**	0.10
	PNPCOUNTERTOSINK	0.02	0.00	0.00	0.00	0.02	0.04	0.06	**0.08**
	PNPCOUNTERTOSTOVE	0.02	0.00	0.00	0.00	0.00	0.02	**0.06**	0.04
	PNPMICROWAVETOCOUNTER	0.02	0.00	0.00	0.02	0.00	0.04	0.08	**0.12**
	PNPSINKTOCOUNTER	0.08	0.00	0.00	0.00	0.00	0.18	0.22	**0.30**
	PNPSTOVETOCOUNTER	0.06	0.00	0.00	0.02	0.02	0.06	0.20	**0.26**
OPEN/CLOSE DOORS	OPENSINGLEDOOR	0.46	0.24	0.00	0.44	0.38	0.72	0.68	**0.74**
	OPENDOUBLEDOOR	0.28	0.20	0.00	0.38	0.50	0.86	**0.94**	0.92
	CLOSEDOUBLEDOOR	0.28	0.56	0.00	0.50	0.50	0.76	**0.82**	0.78
	CLOSESINGLEDOOR	0.56	0.62	0.14	0.76	0.82	0.80	**0.86**	0.84
OPEN/CLOSE DRAWERS	OPENDRAWER	0.42	0.36	0.00	0.36	0.34	0.56	0.62	**0.72**
	CLOSEDRAWER	0.80	0.48	0.00	0.90	0.94	**0.96**	0.90	0.94
TWISTING KNOBS	TURNONSTOVE	0.32	0.24	0.10	0.48	0.30	0.50	0.46	**0.66**
	TURNOFFSTOVE	0.04	0.06	0.02	0.12	0.10	0.16	0.12	**0.20**
TURNING LEVERS	TURNONSINKFAUCET	0.38	0.32	0.06	0.40	0.38	0.24	0.68	**0.70**
	TURNOFFSINKFAUCET	0.50	0.42	0.28	0.58	0.42	0.34	**0.82**	0.78
	TURNSINKSPOUT	0.54	0.54	0.26	**0.70**	0.48	0.40	0.54	0.52
PRESSING BUTTONS	COFFEEPRESSBUTTON	0.48	0.16	0.08	0.08	0.76	0.86	0.86	**0.90**
	TURNONMICROWAVE	0.62	0.38	0.06	0.24	0.32	0.64	**0.74**	0.68
	TURNOFFMICROWAVE	0.70	0.54	0.32	0.56	0.66	0.82	0.86	**0.96**
AVERAGE SUCCESS RATE		0.2880	0.2275	0.0550	0.2800	0.3000	0.4042	0.4942	**0.5050**

Table 1: Results for each task in RoboCasa. The models were trained for 100 epochs with 50 human demonstrations per task and evaluated with 50 episodes for each task. The bold numbers highlight the best achieved success rate for that task among all the models.

or multi-view images and depth maps. The policy denoises rotation and translation of the robot's end-effector as action.

5.3 FPV-NET

We systematically evaluate how the FPV-Net deals with different modalities while maintaining a consistent architecture and diffusion policy configuration across all experiments. This setup allows us to directly compare the effectiveness of different representations.

PC-only: We first group the point cloud by selecting 256 centers via Furthest Point Sampling (FPS), then retrieve 32 nearest neighbors using K-Nearest Neighbors (KNN) to form 256 point groups. Each group is passed through a lightweight MLP encoder, obtaining an embedding per group. These embeddings, along with a language embedding from CLIP, a timestep embedding, and the noisy action, are provided to a transformer-based diffusion policy.

RGB-only: In this model, each camera view is processed by a ResNet-18 model, which is pretrained and then finetuned separately for each view. FiLM layers condition the network on the CLIP-encoded language instruction. The resulting embeddings from all camera views are subsequently given to the same transformer-based diffusion policy employed in the PC-only model.

PC+RGB: This variant simply concatenates the point group embeddings from PC-only with the RGB embeddings from RGB-only, and feeds the combined representation into the transformer-based diffusion policy.

FPV-MLP: Here, the point cloud is processed as before, but we additionally exploit local RGB features. Specifically, we use the 8x8 feature map produced by the third ResNet layer for each image. This feature map is flattened and concatenated with the global ResNet embedding, producing 65 tokens per view. Tokens from all views, along with a learnable class token, are passed to a transformer. The output of the class token serves as the condition vector for AdaLN, while the point group embeddings enter the diffusion policy in the usual way.

FPV-SUGAR: In this model, we use the point cloud encoder of the pretrained 3D visual representation model SUGAR Chen et al. (2024), which also partitions points into 256 groups of 32 via FPS and KNN, but subsequently also employs a 12-layer transformer. We use the model pretrained on multi-object scenes using objects from the Objaverse Deitke et al. (2022) dataset. To reduce computational cost, we freeze the first 10 layers and finetune only the last 2. The RGB images are

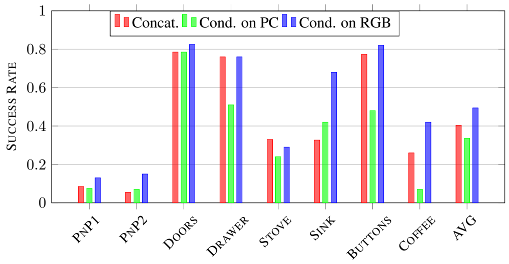


Figure 4: Success rates using different fusion types for point cloud and RGB images.

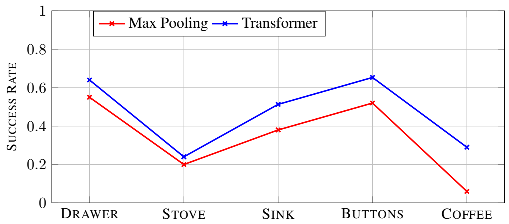


Figure 5: Success rates using max pool or transformer to obtain global feature vector of RGB images to use in AdaLN conditioning.

processed similarly to FPV-MLP, except that we use the 4x4 feature map from the fourth ResNet layer. Finally, the conditioned transformer-based diffusion policy is applied as before.

5.4 Main Results

Table 1 shows that models utilizing both modalities outperform those using a single modality, which addresses Q1. Simply concatenating point cloud and RGB features leads to a 10% improvement, illustrating the complementary nature of spatial and semantic information: each modality contributes unique advantages that are not fully captured by the other. Notably, pick-and-place and insertion tasks benefit most from having both modalities, suggesting that both spatial and semantic cues are crucial for manipulating objects unseen during training. In one particular task the PC-only method performs noticeably better than the other models, namely the TurnSinkSpout task, which requires further investigation.

Our PC-only approach outperforms 3D Diffusion Policy by a margin of 5.25%, answering Q2. A likely explanation is that the max-pooling step discards spatial information critical to the diffusion policy. By contrast, our approach retains more of the point cloud's geometric structure. Furthermore, grouping points instead of handling each point separately like DP3 allows our PC-only model to better capture local spatial features.

FPV-MLP and FPV-SUGAR, conditioning on RGB features, offer further gains, yielding an average success rate of around 50%, higher than the simple concatenation of modalities. This suggests the diffusion policy exploits the rich texture and semantic details from RGB data when using AdaLN for conditioning more effectively than taking these features purely as an additional input. Another possible reason is that the transformer-based diffusion policy can better separate the two modalities, focusing on spatial relations through self-attention over point groups while annotating each group with semantic features via AdaLN conditioning.

3DA exhibits a very low success rate on RoboCasa in our experiments. This may be attributed to our decision to train each model for 100 epochs to ensure a fair comparison. However, as a relatively more complex model, 3DA likely requires a longer training duration to achieve optimal performance.

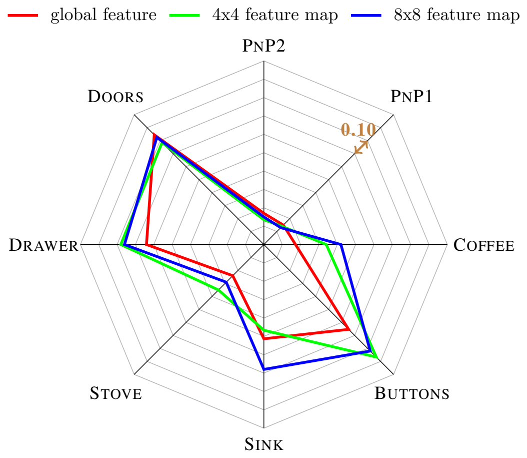


Figure 6: Success rates of conditioning on ResNet features with different granularity level. Each level in the chart corresponds to a 10% difference in success rate.

5.5 Ablation on different fusion

We compared the performance of different fusion strategies for integrating point cloud and RGB embeddings within the transformer architecture. Concat. refers to a straightforward concatenation of both embeddings. Cond. on PC denotes using RGB features as the main modality while conditioning on point cloud features through AdaLN conditioning. Conversely, Cond. on RGB treats point cloud features as the primary modality, with RGB features providing the conditioning signal via AdaLN. As shown in Figure 4, conditioning the RGB-based transformer on point cloud features underperforms compared to simple concatenation. This could be due to compressing the entire point cloud into a single vector, which may discard crucial spatial details, particularly for tasks like COFFEE, where precise grasping of a mug is required. In contrast, conditioning on RGB features yields the best performance across most tasks, effectively addressing Q3.

5.6 Ablation on obtaining condition vector

AdaLN does not directly support sequences as input, so a single token must be extracted to condition on point clouds or RGB features. We compare two methods: (1) a simple max-pooling layer, and (2) a transformer whose learnable class token serves as the global representation. Figure 5 indicates that the transformer-based approach consistently outperforms max pooling in all tested tasks.

5.7 Ablation on RGB features

In order to identify the influence of global tokens and local tokens from ResNet feature map, we evaluate FPV-Net with different feature granularity: global features versus 4x4 or 8x8 feature maps. The results are presented in Figure 6, which show that by adding local features from ResNet would gain performance significantly on most tasks such as BUTTONS and DRAWERS, whereas the DOORS task show less sensitivity. This contrast could be due to the smaller size of buttons and drawer handles, which require finer-grained feature maps for accurate manipulation.

6 Conclusion

In this paper, we introduce the Fusion of Point Cloud and Visual representation Network, a novel approach that integrates RGB and point cloud features using AdaLN conditioning within a transformer. By fusing features at each residual connection, our method effectively captures complementary information from both modalities. Extensive experiments on the RoboCasa benchmark demonstrate significant performance gains over baselines, highlighting the importance of thoughtful cross-modal integration. These results open new avenues for exploring advanced fusion strategies to further enhance robotic perception and understanding of complex environments.

References

Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4788–4795. IEEE, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_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.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Shizhe Chen, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Sugar: Pre-training 3d visual representations for robotics. In *CVPR*, 2024.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. *arXiv preprint arXiv:2212.08051*, 2022.

Yiming Ding, Carlos Florensa, Pieter Abbeel, and Mariano Phielipp. Goal-conditioned imitation learning. *Advances in neural information processing systems*, 32, 2019.

Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. The farthest point strategy for progressive image sampling. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 2-Conference B: Computer Vision & Image Processing.(Cat. No. 94CH3440-5)*, pp. 93–97. IEEE, 1994.

Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. In *7th Annual Conference on Robot Learning*, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Peiyan Li, Hongtao Wu, Yan Huang, Chilam Cheang, Liang Wang, and Tao Kong. Gr-mg: Leveraging partially-annotated data via multi-modal goal-conditioned policy. *IEEE Robotics and Automation Letters*, 2025.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 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.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Skand Peri, Iain Lee, Chanho Kim, Li Fuxin, Tucker Hermans, and Stefan Lee. Point cloud models improve visual robustness in robotic learners. *arXiv preprint arXiv:2404.18926*, 2024.

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 Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Moritz Reuss, Jyothish Pari, Pulkit Agrawal, and Rudolf Lioutikov. Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. *arXiv preprint arXiv:2412.12953*, 2024a.

Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024b. URL https://openreview.net/forum?id=Pt6fLfXMRW.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pp. 785–799. PMLR, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, October 2020. URL https://arxiv.org/abs/2010.02502.

Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

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

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

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. *arXiv preprint arXiv:2402.02500*, 2024.

A Experiment Settings

RoboCasa is a state-of-the-art simulation framework developed to advance the training of generalist robots in diverse and realistic household settings, particularly in kitchen environments. It comprises 120 meticulously modeled kitchen layouts, over 2,500 high-quality 3D objects spanning 150 categories, and 25 foundational atomic tasks that are the building blocks for robot learning. These atomic tasks encompass essential sensorimotor skills, including pick-and-place, opening and closing doors or drawers, twisting knobs, turning levers, pressing buttons, performing insertions, and navigating kitchen spaces. In our work, we evaluated our model in 24 of these tasks, except for the navigation. A list of these tasks evaluated in our work is given in Table 2.

The benchmark is particularly challenging due to its unparalleled diversity and realism. Each scenario includes unique configurations and employs just a single demonstration, significantly raising the bar for generalization. For example, in pick-and-place tasks, the objects vary extensively between scenarios, with no repetitions, forcing models to adapt to new instances without direct prior exposure. Furthermore, the training and evaluation environments are entirely distinct, compelling robotic agents to exhibit robust transfer learning capabilities across unseen kitchens and objects.

These features create a demanding benchmark, testing models on their ability to understand and generalize robotic behavior in highly diverse, real-world-inspired scenarios. RoboCasa's emphasis on realistic physics, photorealistic rendering, and the integration of generative AI tools for diverse asset creation ensures it sets a new standard for evaluating robotic learning methodologies. Its extensive task variability and high fidelity make it one of the most rigorous and comprehensive platforms for advancing generalist robot capabilities in everyday household environments.

B Hyperparameters

C Further Experiments

We conduct further experiments trying out different hyperparameters in the models which conditioned on local ResNet features. The results can be seen in Figure 6. The models used are as follows:

MLP uses the MLP point encoder and 4x4 feature map from ResNet. The diffusion policy uses an embedding dimension of 128.

MLP256 is similar to MLP but the diffusion policy has an embedding dimension of 256.

SUGAR uses the point cloud encoder from the SUGAR pretrained model and 4x4 feature map from ResNet. The point cloud encoder is frozen. The diffusion policy uses an embedding dimension of 128.

SUGAR-FT2 is similar to SUGAR but the last two layers are finetuned while keeping the other layers frozen.

SUGAR256-FT2 is similar to SUGAR-FT2 but the diffusion policy uses an embedding dimension of 256.

MLP8x8 uses the MLP point encoder and 4x4 feature map from ResNet. The transformer used to get the condition vector from the ResNet features has an embedding dimension of 256. The diffusion policy uses an embedding dimension of 128.

MLP8x8-L512 is similar to MLP8x8 but the transformer used to get the condition vector from the ResNet features has an embedding dimension of 512.

D Adaptive LayerNorm conditioning

A visualization of the adaptive layer norm is given in Figure 2. We use the point cloud and language as primary modality in this visualization. In a Diffusion Transformer (DiT) block visualized in Figure 2, the most significant difference to a vanilla transformer block is scaling and shifting operations conditioned on the image CLS token. The scaling factors α, γ and the shifting factor β are applied to self-attention and feed-forward part of the DiT block. The expression $\mathrm{AdaLN}(z_t^P, z_t^L | z_t^I)$ indicates that image embedding is used as condition and mapped to factors α, γ and β, while the point cloud and language embeddings go through the self-attention and feed-forward blocks with additional scaling and shifting operations by these factors.

Task	**Skill Family**	**Description**
PickPlaceCounterToCabinet	Pick and place	Pick an object from the counter and place it inside the cabinet. The cabinet is already open.
PickPlaceCabinetToCounter	Pick and place	Pick an object from the cabinet and place it on the counter. The cabinet is already open.
PickPlaceCounterToSink	Pick and place	Pick an object from the counter and place it in the sink.
PickPlaceSinkToCounter	Pick and place	Pick an object from the sink and place it on the counter area next to the sink.
PickPlaceCounterToMicrowave	Pick and place	Pick an object from the counter and place it inside the microwave. The microwave door is already open.
PickPlaceMicrowaveToCounter	Pick and place	Pick an object from inside the microwave and place it on the counter. The microwave door is already open.
PickPlaceCounterToStove	Pick and place	Pick an object from the counter and place it in a pan or pot on the stove.
PickPlaceStoveToCounter	Pick and place	Pick an object from the stove (via a pan or pot) and place it on (the plate on) the counter.
OpenSingleDoor	Opening and closing doors	Open a microwave door or a cabinet with a single door.
CloseSingleDoor	Opening and closing doors	Close a microwave door or a cabinet with a single door.
OpenDoubleDoor	Opening and closing doors	Open a cabinet with two opposite-facing doors.
CloseDoubleDoor	Opening and closing doors	Close a cabinet with two opposite-facing doors.
OpenDrawer	Opening and closing drawers	Open a drawer.
CloseDrawer	Opening and closing drawers	Close a drawer.
TurnOnStove	Twisting knobs	Turn on a specified stove burner by twisting the respective stove knob.
TurnOffStove	Twisting knobs	Turn off a specified stove burner by twisting the respective stove knob.
TurnOnSinkFaucet	Turning levers	Turn on the sink faucet to begin the flow of water.
TurnOffSinkFaucet	Turning levers	Turn off the sink faucet to stop the flow of water.
TurnSinkSpout	Turning levers	Turn the sink spout.
CoffeePressButton	Pressing buttons	Press the button on the coffee machine to pour coffee into the mug.
TurnOnMicrowave	Pressing buttons	Turn on the microwave by pressing the start button.
TurnOffMicrowave	Pressing buttons	Turn off the microwave by pressing the stop button.

PnP1	PnP2	Doors	Drawer
PnPCounterToCab	PnPCounterToMicrowave	OpenSingleDoor	CloseDrawer
PnPCabToCounter	PnPMicrowaveToCounter	CloseSingleDoor	OpenDrawer
PnPCounterToSink	PnPStoveToCounter	OpenDoubleDoor	
PnPSinkToCounter	PnPCounterToStove	CloseDoubleDoor	
Stove	**Sink**	**Buttons**	**Coffee**
TurnOnStove	TurnOnSinkFaucet	CoffeePressButton	CoffeeSetupMug
TurnOffStove	TurnOffSinkFaucet	TurnOffMicrowave	CoffeeServeMug
	TurnSinkSpout	TurnOnMicrowave	

Table 4: Task groups used for training the models.

Hyper-params.	PC Only	RGB Only	PC + RGB	PC Cond. on RGB	RGB Cond. on PC	PC Cond. on local RGB feat.
Epoch	100	100	100	100	100	100
Batch size	256	256	256	256	256	256
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Learning Rate	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$
Weight Decay	$5e^{-2}$	$5e^{-2}$	$5e^{-2}$	$5e^{-2}$	$5e^{-2}$	$5e^{-2}$
Clip Grad						
Point Sampling	FPS	-	FPS	FPS	FPS	FPS
# Points	4096	-	4096	4096	4096	4096
# Point Groups	256	-	256	256	256	256
Size of Point Group	32	-	32	32	32	32
Latent Dim.	512					
Embedding Dim.	128	256	128	128	256	128

Table 5: Hyperparameters of the design choices discussed in this paper

Task	MLP	MLP256	SUGAR	SUGAR-FT2	SUGAR256-FT2	MLP8x8	MLP8x8-L512
PnPCabToCounter	0.16	0.10	0.04	0.08	0.10	0.10	0.16
PnPCounterToCab	0.08	0.08	0.04	0.02	0.14	0.22	0.08
PnPCounterToMicrowave	0.22	0.20	0.04	0.08	0.10	0.18	0.26
PnPCounterToSink	0.08	0.08	0.00	0.00	0.08	0.06	0.06
PnPCounterToStove	0.02	0.06	0.00	0.02	0.04	0.04	0.06
PnPMicrowaveToCounter	0.04	0.08	0.02	0.06	0.12	0.10	0.08
PnPSinkToCounter	0.24	0.26	0.08	0.08	0.30	0.20	0.22
PnPStoveToCounter	0.26	0.28	0.02	0.04	0.26	0.18	0.20
OpenSingleDoor	0.62	0.58	0.52	0.44	0.74	0.64	0.68
OpenDoubleDoor	0.88	0.94	0.74	0.70	0.92	0.90	0.94
CloseDoubleDoor	0.84	0.82	0.56	0.76	0.78	0.70	0.82
CloseSingleDoor	0.80	0.84	0.68	0.84	0.84	0.86	0.86
OpenDrawer	0.66	0.68	0.76	0.84	0.72	0.60	0.62
CloseDrawer	0.90	0.96	0.96	0.96	0.94	0.96	0.90
TurnOnStove	0.56	0.46	0.62	0.54	0.66	0.48	0.46
TurnOffStove	0.14	0.16	0.22	0.14	0.20	0.12	0.12
TurnOnSinkFaucet	0.40	0.60	0.68	0.58	0.70	0.68	0.68
TurnOffSinkFaucet	0.50	0.80	0.68	0.82	0.78	0.76	0.82
TurnSinkSpout	0.50	0.52	0.58	0.60	0.52	0.60	0.54
CoffeePressButton	0.92	0.90	0.84	0.92	0.90	0.84	0.86
TurnOnMicrowave	0.76	0.26	0.62	0.68	0.68	0.60	0.74
TurnOffMicrowave	0.92	0.68	0.90	0.82	0.96	0.82	0.86
CoffeeServeMug	0.50	0.56	0.56	0.60	0.48	0.56	0.62
CoffeeSetupMug	0.18	0.14	0.14	0.14	0.16	0.20	0.22
Average Success Rate	0.4658	0.4600	0.4292	0.4483	0.5050	0.4750	0.4942

Table 6: Further results for RoboCasa with 50 Human Demonstrations conditioning on local ResNet features