X-IL: EXPLORING THE DESIGN SPACE OF IMITATION LEARNING POLICIES

Xiaogang Jia[*1] **Atalay Donat**[1] **Xi Huang**[1] **Xuan Zhao**[2] **Denis Blessing**[1] **Hongyi Zhou**[1]
Hanyi Zhang[4] **Han A. Wang**[†3] **Qian Wang**[1] **Rudolf Lioutikov**[1] **Gerhard Neumann**[1]
[1] Karlsruhe Institute of Technology [2] Independent Researcher
[3] Reality Labs, Meta [4] University of Liverpool

ABSTRACT

Designing modern imitation learning (IL) policies requires making numerous decisions, including the selection of feature encoding, architecture, policy representation, and more. As the field rapidly advances, the range of available options continues to grow, creating a vast and largely unexplored design space for IL policies. In this work, we present **X-IL**, an accessible open-source framework designed to systematically explore this design space. The framework's modular design enables seamless swapping of policy components, such as backbones (e.g., Transformer, Mamba, xLSTM) and policy optimization techniques (e.g., Score-matching, Flow-matching). This flexibility facilitates comprehensive experimentation and has led to the discovery of novel policy configurations that outperform existing methods on recent robot learning benchmarks. Our experiments demonstrate not only significant performance gains but also provide valuable insights into the strengths and weaknesses of various design choices. This study serves as both a practical reference for practitioners and a foundation for guiding future research in imitation learning.

1 INTRODUCTION

Imitation learning (IL) Osa et al. (2018) has emerged as a powerful paradigm for teaching agents complex behaviors by mimicking demonstrations, eliminating the need for explicit reward engineering Argall et al. (2009). However, the rapid development of novel machine-learning techniques across various domains makes it challenging to assess their potential impact on imitation learning. To address this, we introduce X-IL, a novel framework designed to integrate and explore recently developed techniques within an imitation learning pipeline. Our framework decomposes the imitation learning process into four key modules: (1) observation representations, (2) backbones, (3) architectures, and (4) policy representations. Each module is interchangeable, enabling systematic exploration of the design space for imitation learning policies. This modularity facilitates rapid prototyping, benchmarking, and deployment. Figure 1 provides an overview of our framework.

More specifically, we offer various observation representations, including 2D RGB inputs and 3D point cloud representations, to accommodate diverse perception tasks. Our framework incorporates versatile encoders, such as MLPs, ResNet He et al. (2015), ViT Dosovitskiy et al. (2021), and pre-trained models, which can be tailored to different input types. Furthermore, we offer both Decoder-only and Encoder-Decoder architectures. Decoder-only is simpler with fewer parameters, while Encoder-Decoder supports additional representation learning for better generalization and scaling. We define the backbone as the core computational unit of the policy architecture, offering support for Transformer Vaswani et al. (2017), Mamba Gu & Dao (2024), and xLSTM Beck et al. (2024). Lastly, we offer several state-of-the-art policy representations including popular diffusion- and flow-based models Ho et al. (2020); Chi et al. (2023); Reuss et al. (2023); Lipman et al. (2023); Du et al. (2022).

Our contributions can be summarized as follows:

[*]Correspondence to jia266163@gmail.com
[†]This work is not related to Han A. Wang's position at Meta.

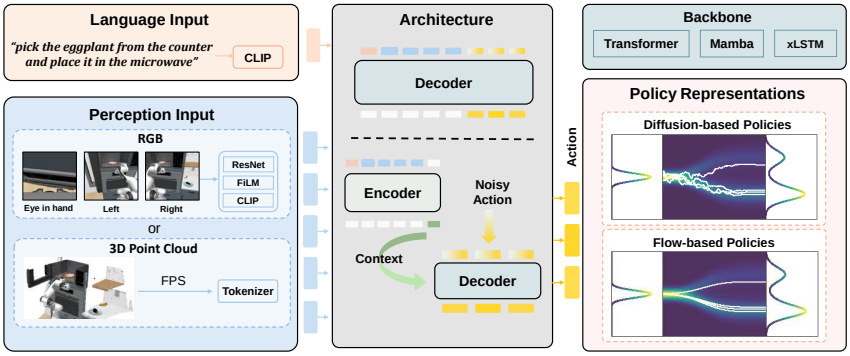


Figure 1: Overview of X-IL framework. X-IL supports multi-modal inputs (Language, RGB, and Point Cloud) and two architectures: Decoder-Only and Encoder-Decoder. Inside each architecture, the Backbone serves as the core computational unit, offering support for Transformer, Mamba, and xLSTM. For policy representations, X-IL supports Behavior Cloning (BC), Diffusion-based, and Flow-based Policies, enabling diverse learning paradigms for imitation learning. Notably, each component—input modality, architecture, backbone, and policy—can be easily swapped to efficiently explore various model configurations.

1) We introduce X-IL, a user-friendly and highly modular framework for imitation learning that supports multi-modal inputs, flexible encoders, diverse architectures and policy representations.

2) We leverage our framework to systematically explore the design space of IL policies. In doing so, we obtain novel policy designs that achieve state-of-the-art results on the LIBERO Liu et al. (2023) and RoboCasa Nasiriany et al. (2024) benchmarks.

3) Our extensive experiments yield valuable insights, such as the potential of recent sequence models as strong alternatives to Transformers, or, that fusing different input modalities can lead to significant performance improvements.

Our work is organized as follows. In Section 2, we review related work. Section 3 describes the proposed framework, detailing design choices and its modular components. In Section 4, we provide experimental evaluations across multiple benchmarks, followed by a discussion of results and observations in Section 5.

2 Related Work

Multi-modal Imitation Learning. Early imitation learning methods relied on either state Schaal (1996); Ho & Ermon (2016); Torabi et al. (2018) or images Pomerleau (1988); Lynch et al. (2020); Young et al. (2021) to describe the environment and define the goal. However, obtaining accurate state information in real-world setting is not straightforward, and state-based representation struggles to capture the complexity of unstructured environments. Conversely, images provide a rich representation for behavior learning Mandlekar et al. (2021) and can be directly acquired from raw sensory input. Despite these advantages, using images as goal conditions in imitation learning is limited by their ambiguity in goal representation and difficulty in goal specification, making them less flexible for real-world deployment. To address this, natural language has been explored as an alternative goal representation, offering a more intuitive and accessible way to specify tasks. Recent studies Shridhar et al. (2022); Reuss et al. (2024b); Bharadhwaj et al. (2024) have explored the integration of language goals with image observation, enabling more flexible policy learning. Another line of research fine-tunes Vision-Language Models (VLMs) models to obtain Vision-Language Action Models (VLAs) Kim et al. (2024); Li et al. (2023; 2024). However, purely image-based representation lack crucial 3D structural information, which is essential for many tasks. Therefore, there is a recent trend on incorporating richer 3D scene presentations, such as point clouds, to enhance policy performance Ke et al. (2024); Ze et al. (2024a).

Imitation Learning with Sequence Models. In recent years, sequence models have been increasingly applied to learning human behaviors, as human decision-making is inherently non-Markov

and requires incorporating historical observations Mandlekar et al. (2021). Early works utilized RNN-based structuresMandlekar et al. (2021). However, these models suffers from vanish gradient for handling long observation sequence and low training efficient due to the sequential processing nature. To address this limitation, Transformer-based architectures have been widely adopted Shafiullah et al. (2022a); Reuss et al. (2023); Bharadhwaj et al. (2024), offering superior scalability and sequence modeling capabilities. Most recently, State-Space Models(SSM) Gu & Dao (2024); Jia et al. (2024) have emerged as a promising alternative to Transformers, demonstrating remarkable efficiency on small datasets and the ability to learn consistent representation. Additionally, improved RNN-based architectures, such as xLSTM Beck et al. (2024), have shown potential to rival both Transformer and SSMs in natural language processing. However, their application in imitation learning remains largely unexplored. X-IL aims to bridge this gap by incorporating Transformers, Mamba, and xLSTM as modular components and empirically evaluating their performance under different tasks and representations.

Modular Imitation Learning Libraries. While numerous open-source libraries provide algorithm-specific implementation of imitation learning methods Chi et al. (2023); Lee et al. (2024); Jia et al. (2024), only a few offer modular design that spans multiple algorithms and architectures. Robomimic Mandlekar et al. (2021) implements Behavior Cloning (BC) with MLP, RNN, and Transformer-based policies, while Imitation Gleave et al. (2022) provides modular implementations of several imitation learning and inverse reinforcement learning methods. However, these libraries do not include recent diffusion-based imitation learning approaches. To address this gap, a recent work, CleanDiffuser Dong et al. (2024), introduces a modular implementation for diffusion models in decision-making, supporting policy architectures such as MLP, UNet, ResNet, and Transformer. However, its evaluation is limited to tasks with low dimensional state input and 2D image input. In contrast, X-IL expands modularity by supporting multi-modal inputs, including 2D images, point clouds, and language-conditioned goals. Additionally, X-IL integrates state-of-the-art sequence models, such as Mamba and xLSTM, broadening its applicability to more complex environments and diverse IL architectures.

3 The X-IL Framework

In this section, we introduce X-IL, a modular and open-source framework for imitation learning, based on the following design principles:

Modularity. X-IL systematically decomposes the imitation learning pipeline into different modules with different components that are easily interchangeable. This modular design enables flexible integration and evaluation of different approaches, facilitating systematic exploration of the design space of imitation learning policies.

Ease-of-use principle. Our framework is easy to use, supporting popular tools such as Hydra Yadan (2019) for configuration management and Weights & Biases (Wandb) Biewald (2020) for logging and visualization, streamlining the experimentation process.

Incorporation of Novel Techniques. X-IL integrates recent advancements such as Mamba Gu & Dao (2024) and xLSTM Beck et al. (2024) for sequence modeling and Diffusion and Flow Matching for policy learning, improving the efficiency and generalization of imitation learning policies.

To enable flexible experimentation, we decompose the imitation learning pipeline into four key modules: 1) observation representations, 2) backbones, 3) architectures, and 4) policy representations. An overview of our framework is shown in Figure 1. Below, we provide a detailed description of each module and its components.

3.1 Observation Representations

Our framework considers three primary types of representations: RGB inputs, Point Cloud, and Language. Below, we introduce these representations and how we encode them.

RGB Inputs. Visual imitation learning has received significant attention in recent research Chi et al. (2023). RGB images, captured from multiple camera viewpoints, provide essential texture and semantic information for object recognition and scene understanding. Prior works have demonstrated that ResNet is a strong encoder for manipulation tasks, making it a widely adopted choice Shafiullah

et al. (2022b); Wan et al. (2024); Zhu et al. (2023). To effectively leverage RGB data, X-IL supports various feature extractors, including ResNet, FiLM-ResNet Turkoglu et al. (2022), and CLIP Radford et al. (2021), with a modular codebase that allows easy integration of additional image encoders. Each image is encoded as a single token and passed into the backbone (see Section 3.2) for further processing, enabling flexible and efficient representation learning.

Point Cloud. Point clouds provide 3D spatial structures obtained from RGB-D cameras or LiDAR sensors, offering geometric information for manipulation tasks. Unlike RGB images, point clouds inherently encode object positions and shapes, making them ideal for tasks requiring fine-grained spatial reasoning. Prior works have emphasized the importance of preserving geometric features for effective representation learning Wan et al. (2024); Ze et al. (2024c); Gyenes et al. (2024). In X-IL, we use Furthest Point Sampling (FPS) Qi et al. (2017) to downsample points, which helps preserve the geometric structures of the 3d space. We adopt a token-based representation to capture scene geometry efficiently. Our framework supports two encoders: a lightweight MLP with Max Pooling Ze et al. (2024c) for computational efficiency and an attention-based encoder with class token for enhanced feature extraction. The implementation details can be found in Appendix A.2.

Language. Language-guided imitation learning Stepputtis et al. (2020); Lynch & Sermanet (2021); Mees et al. (2022); Yu & Mooney (2023); Reuss et al. (2024c) has gained increasing attention as it provides a high-level, abstract way to describe tasks, object attributes, and robot actions, making it a valuable modality in imitation learning. Unlike visual and geometric inputs, language offers context that enhances generalization and adaptability across diverse tasks. To process language, X-IL integrates the pre-trained language model CLIP Radford et al. (2021) to convert textual information into dense embeddings. These embeddings are then fused with visual and point cloud features, enabling a richer multimodal representation for policy learning.

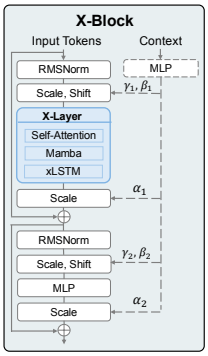


Figure 2: Network details of X-Block. X-Layer is the core part, which is used to process sequence tokens; AdaLn conditioning is used to inject the context information. Details can be found in Appendix A.1.

3.2 Backbones: X-Block

The choice of backbone architectures is critical for learning effective policies in imitation learning, as it determines how different input modalities are processed and how sequential dependencies are captured. We define backbones as the core components responsible for modeling sequential information. Previous works Chi et al. (2023); Jang et al. (2022); Chen et al. (2021); Rouxel et al. (2024); Shaier et al. (2022); Bertasius et al. (2021) have predominantly utilized convolutional 1D architectures, such as U-Net Ronneberger et al. (2015), or Transformer-based models for sequence modeling in imitation learning. However, recent advancements in sequence modeling, such as Mamba and xLSTM, remain largely unexplored in this domain. We provide a comparative analysis of these architectures:

Transformer Vaswani et al. (2017). A widely used attention-based model that has demonstrated strong performance in imitation learning due to its ability to handle non-Markovian behavior in human demonstrations. Most imitation learning models including Vision-Language Action Models (VLAs) use transformers as the backbone.

Mamba Gu & Dao (2024). A structured state-space model (SSM) that significantly improves the efficiency of SSMs while rivaling Transformers in performance. Unlike Transformers, Mamba maintains linear computational complexity. Mamba Imitation Learning (MaIL) Jia et al. (2024) has shown that Mamba-based policies outperform Transformer-based policies with small datasets.

xLSTM Beck et al. (2024). A variant of LSTM that is designed to enhance long-term dependency modeling while maintaining computational efficiency. Unlike standard LSTMs, which struggle with long-range dependencies, xLSTM incorporates architectural improvements to mitigate vanishing gradient issues. While recurrent models generally lack the expressiveness of self-attention, xLSTM offers a balance between efficiency and performance, making it a potential alternative for imitation learning tasks where computational constraints are a concern.

In X-IL, we aim to investigate the potential of these architectures for policy learning. Inspired by the DiT-Block Peebles & Xie (2023) structure, our framework introduces X-Block, as illustrated in Figure 2. The core component of X-Block is the X-Layer, which is responsible for processing temporal information. We provide three backbone options: Transformer, Mamba, and xLSTM, allowing flexibility in sequential modeling. Additionally, AdaLN conditioning Peebles & Xie (2023) is incorporated—not only for conditioning time embeddings in diffusion models but also for integrating representation features. Our findings indicate that using representations as conditioning signals enhances performance, further improving the effectiveness of policy learning.

3.3 Architectures

The architecture of an imitation learning model defines how input representations are processed and how action outputs are generated. X-IL supports two architectures: Decoder-Only and Encoder-Decoder. Prior works such as ACT Zhao et al. (2023) and MDT Reuss et al. (2024c) adopt an encoder-decoder design, whereas PearceTransformer Pearce et al. (2023) and MoDE Reuss et al. (2024a) follow a decoder-only approach. Below, we introduce these architectures and explain their integration within our framework. The illustrations of them are given in Figure 1.

Decoder-only Models. In X-IL, the Decoder-Only architecture is implemented by stacking multiple X-Blocks, where both observations and actions are jointly processed within the decoder. The model outputs only the action tokens, which are then used to train the policy representations.

Encoder-Decoder Models. The Encoder-Decoder architecture in X-IL follows a two-stage approach: the Encoder first encodes multi-modal inputs into a latent representation, and the decoder then generates actions based on this structured embedding. Prior works primarily utilize cross-attention to connect the encoder's output with the decoder's input. However, Mamba and xLSTM lack a built-in mechanism to handle variable-length sequences in this manner. Instead, we find that AdaLN conditioning provides an efficient and flexible alternative for constructing the Encoder-Decoder architecture, enabling effective integration of encoded representations into the decoding process.

3.4 Policy Representations

Besides naive behavior cloning approaches, our framework offers a variety of state-of-the-art policy representations, which can be broadly categorized as diffusion-based and flow-based models.

Behavior Cloning Behavior cloning (BC) assumes a Gaussian distribution as policy representation and maximizes the likelihood of predicted actions in the given ground truth distributions.

Diffusion-Based Policies Denoising diffusion probabilistic models (DDPM) Ho et al. (2020) captures the score function field and iteratively optimizes the action. BESO Reuss et al. (2023) is based on a continuous-time diffusion framework. BESO allows for varying diffusion steps, as well as diverse sampling techniques. Our framework supports both DDPM-style and continuous-time BESO-style policies.

Flow-Based Policies Continuous-time normalizing flows trained via flow matching Lipman et al. (2022) have recently gained a lot of attention and are also suitable as policy representations. These methods, often referred to as rectified flows (RF) Liu et al. (2022) are fully supported in our framework.

More detailed description of the policy representations see Appendix A.4.

4 EXPERIMENTS

To explore the design space of Imitation Learning, we conduct extensive experiments on two robot learning benchmarks: LIBERO and RoboCasa. Our study systematically examines various backbones, architectures, and policy designs for both visual and point cloud-based imitation learning.

4.1 SIMULATION BENCHMARK

LIBERO Liu et al. (2023) We evaluate our modular framework with various model architectures and policy heads using RGB inputs on the LIBERO benchmark, which comprises four distinct task suites: *LIBERO-Spatial*, *LIBERO-Object*, *LIBERO-Goal*, and *LIBERO-Long*. These task suites are specifically designed to evaluate different aspects of robotic learning and manipulation capabilities.

To thoroughly compare the performance of each architecture, we conduct evaluations using both 10 trajectories (20% of the available demonstrations) and 50 trajectories (the full dataset). All models were trained for 100 epochs in LIBERO task suites, and we used the last checkpoint for evaluation. Following the official LIBERO benchmark settings, we simulated 50 rollouts for each sub-task, totaling 500 simulations per task suite. We report the average success rate for each task suite over 3 seeds.

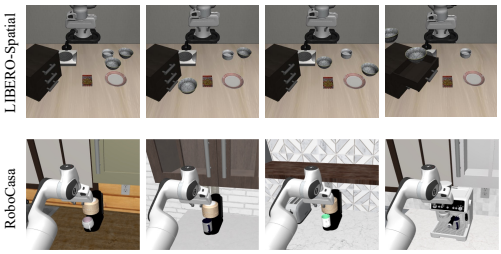


Figure 3: Illustration of LIBERO and RoboCasa. While LIBERO demonstrates minimal variations in the same task, e.g. LIBERO-Spatial, RoboCasa provides diversities in different aspects. CoffeeServeMug is shown in the figure.

RoboCasa Nasiriany et al. (2024) is a large-scale simulation framework, which provides various tasks in everyday scenarios. Besides the large amounts of tasks, there are extensive intra-task variations in RoboCasa. The variations include scenes, objects, and initial positions of the robot and the objects, while LIBERO does not provide this kind of diversity. As shown in Figure 3, in the CoffeeServeMug task, the coffee mugs, the coffee machine, and their surroundings are different and indicate different constraints, i.e. the robot can not grasp the mug from left in the left third variation due to the toast machine. This high level of diversity demands strong generalization from the model, which makes this benchmark very challenging.

We evaluate 5 tasks in RoboCasa with 50 human demonstrations for each task. The 5 tasks contain different behaviors, CloseSingleDoor, OpenDrawer, TurnOnStove, CoffeePressButton, and CoffeeServeMug. For training, we train each model for 200 epochs and rollout the models for 50 episodes for each task. We report the success rate over 3 seeds.

4.2 EXPERIMENTAL SETUP IN X-IL

To ensure a fair comparison, we match the model sizes of Transformer, Mamba, and xLSTM. For both the diffusion policy and flow matching policy, we set the number of sampling steps to 4 in the main experiments. In the LIBERO Benchmark, all models use ResNet-18 for image processing, whereas in the RoboCasa Benchmark, we employ FiLM-ResNet18 for image encoding and an attention-based encoder for point cloud inputs.

		LIBERO-SPATIAL		LIBERO-OBJECT		LIBERO-GOAL		LIBERO-LONG		AVG.	
EXISTING MODELS	BACKBONES	20%	100%	20%	100%	20%	100%	20%	100%	20%	100%
DIFFUSION POLICY	TRANSFORMER	-	78.3±1.1	-	92.5±0.7	-	68.3±1.2	-	50.5±1.3	-	72.4
OCTO	TRANSFORMER	-	78.9±1.0	-	85.7±0.9	-	84.6±0.9	-	51.1±1.3	-	75.1
OPENVLA	LLAMA	-	84.7±0.9	-	88.4±0.8	-	79.2±1.0	-	53.7±1.3	-	76.5
MDT	TRANSFORMER	-	78.5±1.5	-	87.5±0.9	-	73.5±2.0	-	64.8±0.3	-	76.1
MAIL	MAMBA	57.6±2.2	74.3±4.3	78.1±4.5	90.1±2.0	56.5±2.3	81.8±1.6	49.9±4.8	78.6±3.0	60.5	83.5
ATM	TRANSFORMER	**79.0±3.7**	-	81.0±2.4	-	58.6±4.6	-	44.0±6.38	-	65.6	-
ENERVERSE	TRANSFORMER	-	91.2	-	97.7	-	85.0	-	80.0	-	88.5
X-BC (OURS)											
DEC	TRANSFORMER	63.6±1.8	91.7±2.7	89.7±1.6	97.4±1.0	57.4±2.0	87.1±4.0	24.3±5.9	79.1±1.5	58.8	88.8
	MAMBA	66.2±10.7	84.1±2.7	72.8±25.0	97.9±0.4	72.5±2.4	88.4±2.3	34.3±1.8	72.1±2.9	61.5	85.6
	XLSTM	72.8±3.3	89.8±1.2	93.5±1.2	96.7±0.9	72.7±3.2	91.7±1.7	47.6±2.7	78.6±1.1	71.6	89.2
X-BESO (OURS)											
DEC	TRANSFORMER	66.5±4.1	89.5±1.2	90.6±1.2	98.6±0.2	59.8±2.8	85.8±0.5	40.5±2.2	79.3±0.9	64.4	88.3
	MAMBA	73.3±5.4	92.0±1.4	**96.6±0.3**	99.1±0.2	69.4±5.6	**94.5±0.8**	46.3±3.4	**85.2±1.9**	71.4	**92.7**
	XLSTM	74.9±3.0	**93.5±0.5**	93.5±0.5	98.6±0.9	77.9±3.5	92.9±0.6	**51.8±3.6**	84.1±2.6	74.5	92.3
X-RF (OURS)											
DEC	TRANSFORMER	46.3±1.1	87.6±2.7	94.3±2.0	98.6±0.9	32.6±1.2	82.4±2.2	34.4±0.4	80.5±1.1	51.9	87.3
	MAMBA	63.9±4.9	92.6±2.0	91.9±1.0	**99.7±0.1**	55.4±2.4	93.7±1.8	42.7±1.5	84.5±0.6	63.5	92.6
	XLSTM	76.9±3.8	92.9±0.1	92.9±1.1	98.8±0.4	**79.2±4.5**	91.9±0.2	50.5±0.4	84.3±0.6	**74.9**	92.0

Table 1: Results on LIBERO benchmark with 20% and 100% demonstrations, averaged across three seeds. The best overall results are highlighted in bold, with category-specific best results underlined. DEC refers to the Decoder-only architecture.

4.3 BASELINES

We additionally report the performance of the following baselines:

BC-Transfromer BC-Transformer is used in RoboCasa Nasiriany et al. (2024). It uses a CLIP model and a ResNet-18 with FiLM layers to encode goal instructions and the the image-based observations, respectively.

Diffusion Policy Chi et al. (2023) is a visuomotor policy that optimizes the action distribution iteratively using a conditional denoising diffusion process on a learned gradient field.

Octo Octo Model Team et al. (2024) is an open-source vision-language-action (VLA) model trained on a large-scale dataset. It uses a transformer-based diffusion policy that supports both language and goal image as task input.

OpenVLA Kim et al. (2024) is an vision-language-action model based on a much larger model Llama 2 7B.

MDT Reuss et al. (2024c) is a diffusion-based framework that is able to learn versatile behavior from multimodal goal specification including images and languages.

MaIL Jia et al. (2024) uses MAMBA to replace transformer-based backbones in the imitation learning. It demonstrates superior performance compared to transformer-based architectures, especially in the case of small datasets.

ATM Wen et al. (2024) Any-point Trajectory Modeling (ATM) is a framework learning from video demonstrations. ATM predicts the trajectories of arbitrary points in a video frame using images and language instructions as input.

EnerVerse Huang et al. (2025) is a framework designed for future space generation regarding robotic manipulation tasks. A policy head is added to the video generator in order to predict the corresponding action simultaneously.

3D Diffusion Policy (DP3) Ze et al. (2024b). DP3 extracts point-wise features from single-view points clouds. Robot actions are generated conditioned on these features and the current robot states.

More detailed description of the policy representations see Appendix C.1.

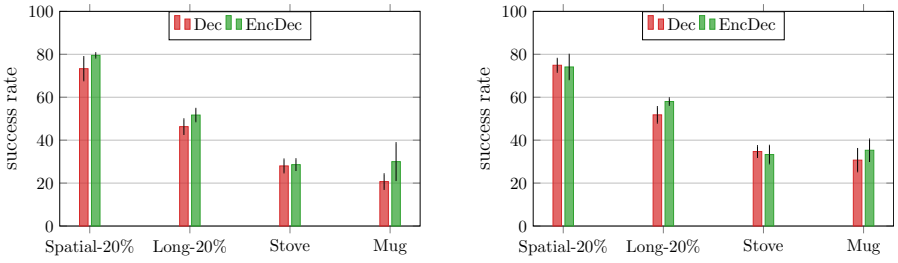
Input Types	Methods	Backbones	CloseSingleDoor	OpenDrawer	TurnOnStove	CoffeePressButton	CoffeeServeMug	Average
RGB	BC-Transformer	Transformer	56.0	42.0	32.0	48.0	22.0	40.0
	X-BESO (ours)	Dec-Transformer	$72.0_{\pm 1.6}$	$56.7_{\pm 2.5}$	$30.7_{\pm 3.8}$	$58.0_{\pm 1.6}$	$18.0_{\pm 4.3}$	47.1
		Dec-Mamba	$\underline{73.3_{\pm 1.9}}$	$\underline{\mathbf{68.0_{\pm 2.8}}}$	$28.0_{\pm 3.3}$	$\underline{72.0_{\pm 4.3}}$	$20.7_{\pm 3.4}$	52.4
		Dec-xLSTM	$70.0_{\pm 1.6}$	$68.0_{\pm 9.9}$	$\underline{34.7_{\pm 2.5}}$	$64.7_{\pm 0.9}$	$\underline{30.7_{\pm 5.2}}$	$\underline{53.6}$
Point Cloud	3D Diffusion Policy	UNet	$62.0_{\pm 2.8}$	$35.8_{\pm 3.2}$	$29.3_{\pm 0.9}$	$\underline{15.3_{\pm 5.1}}$	$\underline{16.7_{\pm 2.5}}$	31.8
	X-BESO (ours)	Dec-Transformer	$62.0_{\pm 4.9}$	$24.0_{\pm 1.6}$	$46.7_{\pm 6.2}$	$5.3_{\pm 3.8}$	$5.3_{\pm 3.4}$	28.7
		Dec-Mamba	$62.0_{\pm 3.3}$	$34.7_{\pm 11.6}$	$\underline{\mathbf{54.0_{\pm 2.8}}}$	$9.3_{\pm 3.4}$	$4.7_{\pm 4.1}$	$\underline{32.9}$
		Dec-xLSTM	$\underline{66.0_{\pm 4.3}}$	$\underline{36.0_{\pm 7.1}}$	$49.3_{\pm 0.9}$	$5.3_{\pm 3.4}$	$7.3_{\pm 5.0}$	32.8
Point Cloud + RGB	X-BESO (ours)	Dec-Transformer	$72.0_{\pm 1.6}$	$45.3_{\pm 3.4}$	$37.3_{\pm 5.0}$	$76.0_{\pm 1.6}$	$38.0_{\pm 5.7}$	53.7
		Dec-Mamba	$72.7_{\pm 1.9}$	$\underline{53.3_{\pm 0.9}}$	$\underline{44.0_{\pm 1.6}}$	$79.3_{\pm 3.8}$	$44.0_{\pm 4.3}$	58.7
		Dec-xLSTM	$\underline{\mathbf{74.7_{\pm 2.5}}}$	$50.7_{\pm 6.6}$	$42.7_{\pm 8.2}$	$\underline{\mathbf{88.0_{\pm 5.9}}}$	$\underline{\mathbf{48.7_{\pm 1.9}}}$	$\underline{\mathbf{60.9}}$

Table 2: Results for RoboCasa using different input types with 50 human demonstrations, averaged across three seeds. The best overall results are highlighted in bold, with category-specific best results underlined. DEC refers to the Decoder-only architecture.

4.4 Evaluation on Visual Inputs

LIBERO. We report the main results in Table 1. To evaluate our framework on LIBERO, we tested BC, BESO, and RF policies using Decoder-only architectures across Transformer, Mamba, and xLSTM backbones. Our results demonstrate that X-IL achieves state-of-the-art performance, surpassing publicly available models. Specifically, xLSTM shows great potential in both 20% and 100% data settings, where it achieves 74.5% average success and 92.3% average success respectively.

RoboCasa. We report the main results in Table 2. Compared to LIBERO, RoboCasa presents a more challenging benchmark due to its dynamically changing background scenes and object variations across demonstrations and evaluations. We tested X-BESO on five tasks within RoboCasa and observed that our approach outperforms the results reported in the original paper. Specifically, using xLSTM-based models, we achieved a higher average success rate of 53.6%, compared to 40.0% of BC-Transformer, demonstrating the effectiveness of our method in handling complex and dynamic environments. Additionally, we observe that Mamba and xLSTM outperform Transformer-based backbones, which is consistent with our findings from LIBERO. This result further highlights the potential of leveraging new sequential models in imitation learning, suggesting that alternative architectures beyond Transformers can offer improved efficiency and performance in complex robotic tasks.



(a) Mamba (b) xLSTM

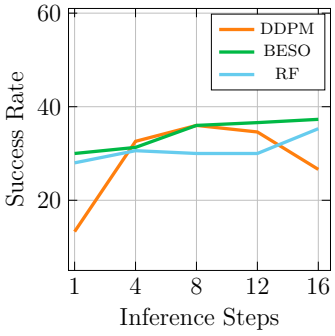
Figure 4: Comparison on different architectures. Dec refers to the Decoder-only model, while EncDec refers to the Encoder-Decoder model.

4.5 Evaluation on Point Cloud Inputs

We report the main results in Table 2. We evaluate X-BESO using point cloud inputs on RoboCasa and achieve superior results compared to 3D Diffusion Policy. An interesting observation from our results is that point cloud-based inputs do not necessarily outperform RGB-based inputs.

Our analysis suggests that this is due to the complexity of RoboCasa's scenarios, where point clouds are captured from diverse sources, leading to significant information loss during sampling—especially in tasks involving small objects. In such cases, only a sparse set of points remains, limiting the effectiveness of point cloud representations. This highlights the potential benefits of object-centric approaches that focus on preserving critical task-relevant details.

Additionally, we evaluate the performance of combining Point Cloud and RGB inputs. A compact representation is first extracted from the point cloud and then concatenated with the RGB features. Experimental results demonstrate that incorporating both modalities significantly enhances performance, particularly for the xLSTM-based model, which achieves a 60.9% success rate—compared to 53.6% with RGB alone and 32.8% with Point Cloud alone. This highlights the importance of exploring more effective multi-modal fusion strategies to fully leverage the strengths of each modality.



Steps	DDPM [ms]	BESO [ms]	RF [ms]
1	$4.91_{\pm 0.05}$	$4.98_{\pm 0.09}$	$5.02_{\pm 0.03}$
4	$14.62_{\pm 0.28}$	$14.91_{\pm 0.29}$	$14.63_{\pm 0.34}$
8	$27.39_{\pm 0.48}$	$28.00_{\pm 0.68}$	$27.65_{\pm 0.66}$
12	$40.40_{\pm 0.80}$	$40.43_{\pm 0.75}$	$40.11_{\pm 0.74}$
16	$53.39_{\pm 1.21}$	$53.78_{\pm 1.17}$	$52.96_{\pm 1.08}$

Figure 5: Comparison of different inference steps for DDPM, BESO, and RF. Left: success rate; Right: inference time.

4.6 Comparison on different architectures

We conduct experiments on four tasks—Spatial (20%) and Long (20%) from LIBERO, as well as TurnOnStove and CoffeeServeMug from RoboCasa—to compare the performance of Decoder-only and Encoder-Decoder architectures. The results, presented in Figure 4, show that the AdaLN-conditioned Encoder-Decoder architecture achieves superior performance on most tasks, highlighting its effectiveness. Furthermore, by processing observations and actions separately, this design offers more flexibility in choosing different layers for the encoder and decoder, making it more scalable to larger models.

4.7 Comparison of Diffusion Models Across Varying Inference Steps

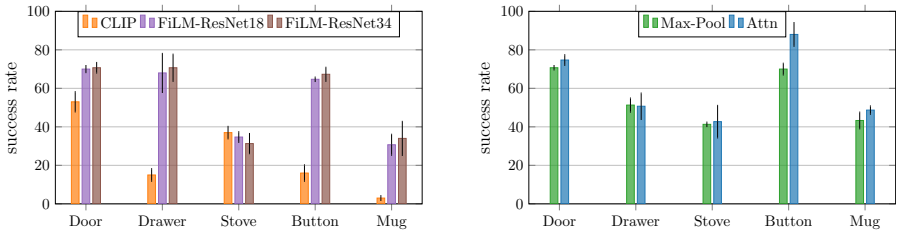
We evaluate Decoder-only xLSTM with DDPM, BESO, and RF on the challenging TurnOnStove task in RoboCasa, comparing performance and inference speed across 1, 4, 8, 12, and 16 inference steps (Figure 5). DDPM struggles with a single step, while BESO and RF perform well and improve with more steps. Their inference times are similar, and the speed advantage of flow matching is less noticeable due to the lower action dimension.

4.8 Comparison on different encoders

We evaluate different image encoders on the RoboCasa dataset using Dec-xLSTM BESO, comparing FiLM-ResNet18, FiLM-ResNet34, and CLIP (frozen) to assess their impact on performance. We also compare the Max-Pooling and Attention-based Point Cloud encoders. The results are presented in Figure 6.

Our findings reveal that despite RoboCasa's requirement for generalization to new scenes and objects, frozen CLIP encoders perform poorly on most tasks. In contrast, fine-tuned ResNet18 and ResNet34 demonstrate strong performance, suggesting that domain adaptation plays a crucial role in achieving effective visual representations for imitation learning. A potential reason for CLIP's underperformance is the significant domain gap between its pretraining data and robot manipulation datasets. This indicates that pretraining on robotics-specific datasets could further improve visual representations for imitation learning.

Compared to the Max-Pooling encoder, the Attention-based encoder shows better performance on most tasks, which indicates that attention can better capture the geometric structures of Point Cloud.



(a) RGB Encoders

(b) PC Encoders

Figure 6: Comparison of different image encoders and Point Cloud (PC) encoders.

5 Discussion

Here, we list several general observations **O1)-O4)** that are based on the experiments from Section 4.

O1) Given similar model sizes, Mamba and xLSTM-based policies outperform Transformer-based policies in both RGB inputs and Point Cloud inputs across LIBERO and RoboCasa, demonstrating their potential as viable alternatives to Transformers in imitation learning.

O2) AdaLN conditioning is a strong method to build Encoder-Decoder architectures in imitation learning and is suitable for all kinds of sequential models. Injecting observation representations to the action decoder through adaLN conditioning could further improve the model's performance compared to the Decoder-only structure.

O3) Point cloud representations do not necessarily outperform RGB-based representations in imitation learning. While previous works have demonstrated strong performance using point cloud inputs, their evaluations were typically conducted on simpler tasks, where the sampled points were highly task-relevant.

For more complex tasks, however, the Furthest Point Sampling (FPS) method distributes points evenly across the scene, often leading to significant information loss—especially for tasks requiring fine-grained object interactions. This suggests that standard point cloud sampling techniques may not always be optimal, and developing task-aware sampling or object-centric approaches could further improve the effectiveness of point cloud representations in imitation learning.

O4) Better strategies for combining point cloud and image representations need further exploration. Our experiments show that simply concatenating point cloud and RGB inputs to the policy improves performance, but the gains are not significant. This suggests that a more structured fusion mechanism is required to fully leverage the complementary nature of these modalities. Finding the right balance between point cloud and image features remains an open challenge.

O5) Our experiments highlight the necessity of robot-specific trained encoders. Comparing fine-tuned FiLM-ResNet with frozen CLIP, we observe that CLIP performs poorly on most tasks. We attribute this to the fact that manipulation tasks require task-specific features, which are not well captured by CLIP's broad, vision-language pretraining.

6 Conclusion

We introduced X-IL, a modular and user-friendly framework for imitation learning, enabling systematic exploration of various policy design choices such as backbones, architectures, and policy representations across multiple modalities. Our framework supports state-of-the-art encoders, efficient sequential models, and multi-modal fusion, providing a unified platform for researchers and practitioners. Through extensive evaluations on LIBERO and RoboCasa, we demonstrate superior performance, improved data efficiency, and better representation learning strategies. Our findings highlight the potential of alternative sequence models, task-adapted encoders, and optimized multi-modal fusion in imitation learning. We hope X-IL serves as a valuable resource for advancing scalable and generalizable policies.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xlstm: Extended long short-term memory, 2024. URL https://arxiv.org/abs/2405.04517.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding?, 2021. URL https://arxiv.org/abs/2102.05095.

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.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling, 2021. URL https://arxiv.org/abs/2106.01345.

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.

Zibin Dong, Yifu Yuan, Jianye Hao, Fei Ni, Yi Ma, Pengyi Li, and Yan Zheng. Cleandiffuser: An easy-to-use modularized library for diffusion models in decision making. *arXiv preprint arXiv:2406.09509*, 2024. URL https://arxiv.org/abs/2406.09509.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Shian Du, Yihong Luo, Wei Chen, Jian Xu, and Delu Zeng. To-flow: Efficient continuous normalizing flows with temporal optimization adjoint with moving speed, 2022. URL https://arxiv.org/abs/2203.10335.

Adam Gleave, Mohammad Taufeeque, Juan Rocamonde, Erik Jenner, Steven H. Wang, Sam Toyer, Maximilian Ernestus, Nora Belrose, Scott Emmons, and Stuart Russell. imitation: Clean imitation learning implementations. arXiv:2211.11972v1 [cs.LG], 2022. URL https://arxiv.org/abs/2211.11972.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Balázs Gyenes, Nikolai Franke, Philipp Becker, and Gerhard Neumann. Pointpatchrl – masked reconstruction improves reinforcement learning on point clouds, 2024. URL https://arxiv.org/abs/2410.18800.

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 and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Siyuan Huang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Zhengkai Jiang, Yue Hu, Peng Gao, Hongsheng Li, Maoqing Yao, and Guanghui Ren. Enerverse: Envisioning embodied future space for robotics manipulation, 2025. URL https://arxiv.org/abs/2501.01895.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning, 2022. URL https://arxiv.org/abs/2202.02005.

Xiaogang Jia, Qian Wang, Atalay Donat, Bowen Xing, Ge Li, Hongyi Zhou, Onur Celik, Denis Blessing, Rudolf Lioutikov, and Gerhard Neumann. Mail: Improving imitation learning with mamba, 2024. URL https://arxiv.org/abs/2406.08234.

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, 2024. URL https://arxiv.org/abs/2406.09246.

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024.

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

Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL https://arxiv.org/abs/2306.03310.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data, 2021. URL https://arxiv.org/abs/2005.07648.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pp. 1113–1132. PMLR, 2020.

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. In *Conference on Robot Learning (CoRL)*, 2021.

Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data, 2022. URL https://arxiv.org/abs/2204.06252.

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, 2024. URL https://arxiv.org/abs/2406.02523.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2): 1–179, 2018.

Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

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, 2017.

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, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal conditioned imitation learning using score-based diffusion policies. In *Robotics: Science and Systems*, 2023.

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 *Robotics: Science and Systems*, 2024b.

Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals, 2024c. URL https://arxiv.org/abs/2407.05996.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Quentin Rouxel, Andrea Ferrari, Serena Ivaldi, and Jean-Baptiste Mouret. Flow matching imitation learning for multi-support manipulation, 2024. URL https://arxiv.org/abs/2407.12381.

Stefan Schaal. Learning from demonstration. *Advances in neural information processing systems*, 9, 1996.

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

Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone, 2022b. URL https://arxiv.org/abs/2206.11251.

Sagi Shaier, Maziar Raissi, and Padmanabhan Seshaiyer. Data-driven approaches for predicting spread of infectious diseases through dinns: Disease informed neural networks, 2022. URL https://arxiv.org/abs/2110.05445.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on robot learning*, pp. 894–906. PMLR, 2022.

Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks, 2020. URL https://arxiv.org/abs/2010.12083.

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

Mehmet Ozgur Turkoglu, Alexander Becker, Hüseyin Anil Gündüz, Mina Rezaei, Bernd Bischl, Rodrigo Caye Daudt, Stefano D'Aronco, Jan Dirk Wegner, and Konrad Schindler. Film-ensemble: Probabilistic deep learning via feature-wise linear modulation, 2022. URL https://arxiv.org/abs/2206.00050.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Weikang Wan, Yifeng Zhu, Rutav Shah, and Yuke Zhu. Lotus: Continual imitation learning for robot manipulation through unsupervised skill discovery, 2024. URL https://arxiv.org/abs/2311.02058.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning, 2024. URL https://arxiv.org/abs/2401.00025.

Omry Yadan. Hydra - a framework for elegantly configuring complex applications. Github, 2019. URL https://github.com/facebookresearch/hydra.

Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, Abhinav Gupta, Pieter Abbeel, and Lerrel Pinto. Visual imitation made easy. In *Conference on Robot Learning*, pp. 1992–2005. PMLR, 2021.

Albert Yu and Raymond J. Mooney. Using both demonstrations and language instructions to efficiently learn robotic tasks, 2023. URL https://arxiv.org/abs/2210.04476.

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)*, 2024a.

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

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations, 2024c. URL https://arxiv.org/abs/2403.03954.

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.

Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors, 2023. URL https://arxiv.org/abs/2210.11339.

A X-IL DETAILS

A.1 X-BLOCK

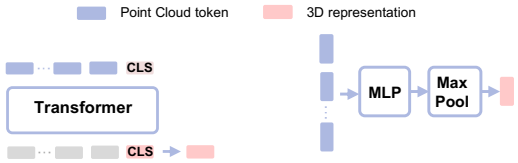
An X-Block is similar to Diffusion Transformer (DiT) block, but generalized. The key difference is the X-Layer which is capable to plug-in different backbones, including Transformer, Mamba and xLSTM. The Adaptive LayerNorm (AdaLN) is used to make the input tokens conditionally activated on the context. A MLP maps the given context to factors α, γ and β. These factors are then applied to scaling and shifting operations in order to manipuate the latent embeddings.

A.2 REPRESENTATION ENCODERS

A.2.1 RGB

- **ResNet** ResNet-18 with a latent dimension of 512 is used in this paper.
- **FiLM-ResNet** FilM introduces a general conditioning layer for visual reasoning tasks. ResNet-18 with a latent dimension of 512 is used for FilM-ResNet. The FilM has a condition dimension of 512.

A.2.2 POINT CLOUD



(a) Attention

(b) MLP with max pooling

Figure 7: Point cloud encoding.

- **Attention** We use a 4-layer self-attention Transformer to encode the point cloud tokens. A CLS token is used to capture the whole geometric representation. See Figure 7a.
- **MLP with MaxPooling** We use the same point cloud encoder in 3D Diffusion Policy Ze et al. (2024a), with a 3-layer MLP and Max Pooling to get the compact 3d representations. See Figure 7b.

A.2.3 LANGUAGE

We used pre-trained CLIP to encode language. CLIP is a vision-language model that aligns the latent embeddings of visual and language inputs. Pre-trained model ViT-B/32 is used as language encoder. We freeze the weights of both models during the training.

A.3 BACKBONE DETAILS

Hyperparameters of three different backbones, i.e., Transformer, Mamba, and xLSTM, used in different policies are shown in the Table 3, 4 and 5. These hyperparameters are not tuned for a specific task but are used for all tasks reported in this paper.

Policy	DDPM		BESO		FM	
Transformer	Dec	Enc-Dec	Dec	Enc-Dec	Dec	Enc-Dec
Layers	6	4 - 6	6	4 - 6	6	4 - 6
Heads	8	8	8	8	8	8
Embeddings	512	512	512	512	512	512
Batch Size	256	256	256	256	256	256
Learning rate	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
Optimizer	Adam	Adam	AdamW	AdamW	AdamW	AdamW

Table 3: Hyperparameters of Transformer

Policy	DDPM		BESO		FM	
Mamba	Dec	Enc-Dec	Dec	Enc-Dec	Dec	Enc-Dec
Layers	8	4 - 8	8	8 - 8	8	4 - 8
Embeddings	512	512	512	512	512	512
Batch Size	256	256	256	256	256	256
Learning rate	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
Optimizer	Adam	Adam	AdamW	AdamW	AdamW	AdamW

Table 4: Hyperparameters of Mamba

A.4 Policies

Behavior Cloning Behavior cloning (BC) assumes a Gaussian distribution as policy representation and has therefore limited model capacity. Maximizing the likelihood of the policy results in a mean squared error (MSE) minimization between ground truth and predicted actions. Due to its simplicity, BC is often used as a default naive baseline for comparing imitation learning policies.

Diffusion-Based Policies Denoising diffusion probabilistic models (DDPM) Ho et al. (2020) are a popular choice for policy representation due to their simplicity and minimal design choices compared to advanced models like BESO Reuss et al. (2023). BESO, based on a continuous-time diffusion framework, allows for varying diffusion steps during training and inference, as well as diverse sampling techniques such as DDIM. Despite these differences, both DDPM and BESO rely on regression losses, either learning a score function or a denoising model. Our framework supports both DDPM-style policies and continuous-time BESO-style policies.

Flow-Based Policies Continuous-time normalizing flows trained via flow matching Lipman et al. (2022) have recently gained a lot of attention and are also suitable as policy representation. These methods, often referred to as rectified flows (RF) Liu et al. (2022) or stochastic interpolants Albergo & Vanden-Eijnden (2022), are fully supported in our framework.

B Additional Task Details

B.1 LIBERO Benchmark

The LIBERO Benchmark comprises five distinct task suites: *LIBERO-Spatial*, *LIBERO-Object*, *LIBERO-Goal*, *LIBERO-Long*, and *LIBERO-90*. Each suite is designed to evaluate distinct dimensions of robotic learning and manipulation capabilities:

- *LIBERO-Spatial* evaluates spatial reasoning precision through tasks requiring differentiation of identical objects (e.g., bowls) based solely on their relational positioning (e.g., placement relative to plates or other objects).
- *LIBERO-Object* tests object-centric manipulation by requiring precise recognition and interaction with unique objects (e.g., pick-and-place tasks) in each trial, emphasizing perceptual discrimination.
- *LIBERO-Goal* assesses goal-conditioned adaptability by defining distinct objectives (e.g., placing objects in varying sequences or configurations) within fixed object-layout environments, necessitating dynamic behavioral adjustments.

Policy	DDPM		BESO		FM	
xLSTM	Dec	Enc-Dec	Dec	Enc-Dec	Dec	Enc-Dec
Blocks	8	4 - 8	8	8 - 8	8	4 - 8
Embeddings	512	512	512	512	512	512
Batch Size	256	256	256	256	256	256
Learning rate	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
Optimizer	Adam	Adam	AdamW	AdamW	AdamW	AdamW

Table 5: Hyperparameters of xLSTM

- *LIBERO-Long* challenges long-term planning and endurance with multi-stage tasks demanding sustained execution, error mitigation, and coordination across extended timelines.
- *LIBERO-90* serves as a generalization benchmark, aggregating 90 short-horizon tasks across heterogeneous settings to evaluate robustness to variability in objects, layouts, and objectives.

Figure 8: LIBERO benchmark suites with total 130 tasks in five different scenes

With the exception of *LIBERO-90*, which includes a diverse collection of 90 tasks, the remaining suites each consist of 10 tasks accompanied by 50 demonstrations per task. For the purposes of training and evaluation efficiency, our study focuses on the first four suites (LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long).

B.2 RoboCasa Benchmark

RoboCasa is a large-scale simulation framework developed to train generalist robots in realistic and diverse home environments, with a particular focus on kitchen scenarios. The benchmark comprises 100 tasks, including 25 atomic tasks with 50 human demonstrations and 75 composite tasks with auto-generated demonstrations. These tasks are centered around eight fundamental robotic skills relevant to real-world home environments: (1) pick-and-place, (2) opening and closing doors, (3) opening and closing drawers, (4) twisting knobs, (5) turning levers, (6) pressing buttons, (7) insertion, and (8) navigation.

To comprehensively evaluate our method, we selected five tasks from the atomic tasks, each representing a distinct skill:

- *Close Single Door* : Opening and closing doors
- *Open Drawer* : Opening and closing drawers
- *Turn on Stove* : Twisting knobs
- *Coffee Press Button* : Pressing buttons
- *Coffee Serve Mug* : Insertion

One of the key advantages of RoboCasa is its provision of both image-based (RGB) and 3D observations, enabling evaluations across different sensing modalities. Accordingly, we assessed our method using both RGB and 3D observations.

C Additional Baselines Details

C.1 Baselines

Diffusion Policy Chi et al. (2023) is a visuomotor policy that optimizes the action distribution iteratively using a conditional denoising diffusion process on a learned gradient field. It demonstrates the capability to capture the multi-modal action distributions. Incorporating techniques such as receding horizon control and visual conditioning, the learned visuomotor policy can be deployed to real-world embodiments.

Octo Octo Model Team et al. (2024) is an open-source vision-language-action (VLA) model. It uses a transformer-based diffusion policy that supports both language and goal image as task input. The policy is trained on a large-scale dataset and can be deployed to various embodiments.

OpenVLA Kim et al. (2024)is another open-source vision-language-action model. Different from Octo, OpenVLA is based on a much larger model Llama 2 7B. It uses 256 reserved tokens for action values, providing higher resolution for robot control signals.

MDT Reuss et al. (2024c) is a diffusion-based framework that is able to learn versatile behavior from multimodal goal specification including images and languages. It introduces latent goal representations by aligning the latent embeddings of the goal image and the language instructions of the same tasks. This alignment is especially beneficial when there are few language annotations in the dataset.

MaIL Jia et al. (2024) uses MAMBA to replace transformer-based backbones in the imitation learning. It demonstrates superior performance compared to transformer-based architectures, especially in the case of small datasets.

ATM Wen et al. (2024) Any-point Trajectory Modeling (ATM) is a framework learning from video demonstrations. ATM predicts the trajectories of arbitrary points in a video frame using images and language instructions as input. The robot action is derived from trajectory predictions and the current video frame using MLP.

EnerVerse Huang et al. (2025) is a framework designed for future space generation regarding robotic manipulation tasks. This framework uses Free Anchor View (FAV) and 4D Gaussian Splatting (4DGS) to generate the next frames of the videos in manipulation scenarios. A policy head is added to the video generator in order to predict the corresponding action simultaneously. EnverVerse shows improved performance in long-horizon manipulation tasks.