Generalizable 3D Feature Representations for Vision-Language-Action Policies in Robotic Manipulation

Dhruv Sheth
California Institute of Technology
dsheth@caltech.edu

Mentored by: Ziqi Ma
California Institute of Technology
zma2@caltech.edu

Supervised by: Prof. Georgia Gkioxari
California Institute of Technology
georgia@caltech.edu

Abstract

Robotic manipulation in real-world environments faces significant challenges due to variations in camera viewpoints, lighting, and object appearances. While recent advancements in visual-language models (VLMs) and policy architectures like diffusion models, autoregressive transformers (ACT), and flow matching methods have improved data efficiency and action generation, their reliance on 2D visual features limits robustness and generalization. This proposal investigates a unified framework that combines 3D feature extraction, memory-augmented policy architectures, and explicit language conditioning to enhance long-horizon, generalizable manipulation. By integrating and extending concepts from recent works such as 3D Diffuser Actor, RVT-2, and SAM2Act, we aim to understand and optimize the interplay between 3D visual representations and various policy architectures. The goal of this research is to develop manipulation policies that are significantly more robust to environmental variations and task novelty.

1 Introduction and Background

Recent progress in robotic manipulation has been driven by advancements in visual-language models (VLMs) and end-to-end policies trained from demonstrations [1, 2]. These policies often utilize 2D representations from models like CLIP [1] and R3M [2], sometimes projected into 3D using depth information [4, 12]. However, these 2D-to-3D lifting methods typically assume known camera parameters, making them vulnerable to calibration errors and viewpoint changes. Even small discrepancies can lead to brittle policies that fail to generalize in real-world settings.

In parallel, policy architectures for manipulation have evolved, including diffusion-based models [14, 4], autoregressive transformers (ACT) [13], and flow matching methods [15], [16]. Diffusion policies iteratively refine actions by denoising, while ACT models predict actions sequentially. Flow matching generates continuous trajectories by learning a vector field. These approaches offer different strengths: diffusion models handle multimodal action distributions, ACT captures temporal dependencies, and flow matching provides smoothness guarantees [11, 22]. However, the performance of all these architectures fundamentally depends on the robustness of the underlying visual representations.

Large-scale ablation studies, such as [11], indicate that pretrained visual representations can be robust on in-distribution tasks, but their out-of-distribution generalization is a significant challenge. In contrast, methods that explicitly construct 3D scene representations have demonstrated improved performance under camera perturbations [4, 12, 5]. This improvement is likely because the visual scene tokens and robot actions interact in a common 3D space, mitigating the need for the policy to implicitly learn a 2D-to-3D mapping. For instance, 3D Diffuser Actor [4] projects features into a 3D workspace, RVT-2 [12] uses multi-view inputs for precision with few demonstrations, and SAM2Act [5] incorporates a memory module for long-horizon tasks.

Recent work in self-supervised 3D representation learning, such as Find3D's feature matching across views [8] and Trellis' 3D latent diffusion [**?**], suggests alternative pathways for acquiring geometry-aware features without full 3D supervision.

The central question this research addresses is: *How can we design visual representations and policy architectures that, when combined, yield robust and generalizable manipulation policies in 3D space?* We hypothesize that using 3D-aware visual representations, obtained either by lifting 2D features or through explicit 3D training, and conditioning policies on these features, will significantly improve generalization and reduce sample complexity. Specifically, we aim to investigate how different policy architectures (diffusion, ACT, flow matching) benefit from various 3D representations and how the resulting distributions of visual features influence action prediction stability and accuracy. The interaction between the policy and the representation is crucial; since, the iterative refinement of diffusion policies may potentially benefit more from SE(3)-equivariant features than the stepwise predictions of ACT.

2 Objectives

The overarching goal of this project is to develop and evaluate a robust and generalizable robotic manipulation system that leverages 3D visual representations, language understanding, and memory.

To formalize, we seek to optimize an imitation learning objective. Given a visual representation ϕ and a policy π, our objective is to minimize the discrepancy between predicted actions ($\hat{a}$) and ground-truth actions (a) over a test-time dataset D:

$$\min_{\phi,\pi} \mathbb{E}_{(a,\hat{a})\sim D} \left[\|a - \hat{a}\|^2 \right]$$

This objective function is a high level description and can later be adapted for multi-task learning and different policy architectures (ACT and flow matching).

To achieve this, we have the following specific objectives:

(a) **Develop and Compare 3D Visual Representations:** Investigate and compare methods for creating 3D representations from pre-trained 2D models (e.g., CLIP, R3M, SAM2, MVP, and pre-trained VLMs). This includes exploring frozen pre-trained models and, if feasible, fine-tuning strategies. We will explore techniques like depth-based projection, multi-view geometry, and multi-resolution feature extraction. Achieving invariance to camera poses is also one of our objectives.

(b) **Evaluate Policy Architectures:** Experiment with and compare different policy architectures, including diffusion-based models, autoregressive transformer policies (ACT), and flow matching approaches. We will analyze how these policies utilize the 3D visual representations and how conditioning on these representations affects performance, particularly under distribution shifts. The interaction between representation and policy is a core focus.

(c) **Integrate Memory and Context:** Investigate the incorporation of a memory module, inspired by SAM2Act+, to store spatio-temporal features. This is intended to enable the policy to maintain context across sequential tasks, reduce error accumulation, and improve performance on complex, multi-step tasks, especially those requiring long-horizon reasoning.

(d) **Quantitatively Assess Generalization:** Define and utilize robust evaluation metrics to measure improvements in policy generalization. We aim to use the following metrics (build upon them): cross-view success rate, feature consistency using Centered Kernel Alignment (CKA), memory recall accuracy (for memory-augmented policies), KL divergence between demonstrated and predicted action distributions, and average path length similarity (DTW) for long-horizon tasks.

(e) **Refine the Imitation Learning Objective:** Adapt and extend the imitation learning objective from 3D Diffuser Actor [4] to incorporate multi-task extensions, different policy architectures, and the various 3D representation methods.

Success will be determined by demonstrating significant improvements in robustness (e.g., higher task success rates under camera shifts and novel task conditions) and sample efficiency compared to existing 2D-based methods.

3 Approach

We aim to build upon three interconnected components: 3D feature extraction and representation, policy architecture evaluation, and context integration. We will also do a systematic comparison of how different visual representations, particularly after being lifted to 3D, interact with various policy architectures.

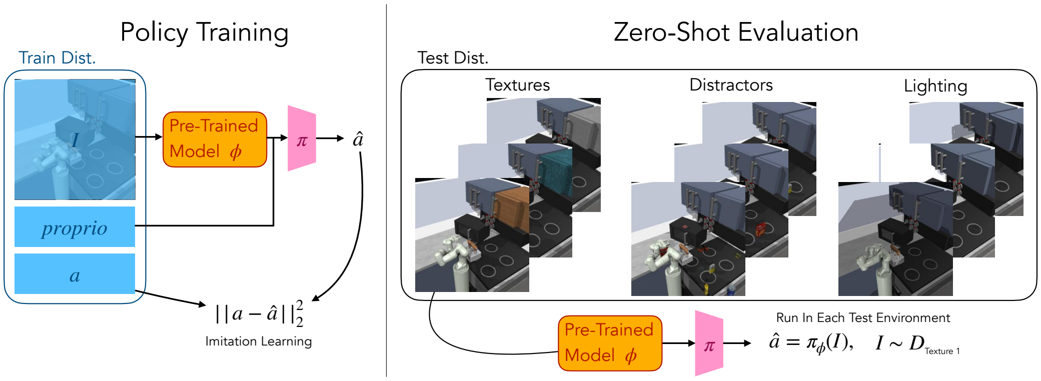


Figure 1: **Training and Zero-Shot Evaluation Pipeline.** We aim to train our policy, π, using imitation learning on a training distribution of observations (images I and proprioceptive information proprio) and actions (a). The policy conditions on a pre-trained visual representation, ϕ, which is kept frozen during training. The goal is to minimize the L2 distance between the predicted action $\hat{a}$ and the ground truth action a. At test time, we evaluate the pre-trained model ϕ and the trained policy π in a zero-shot setting across various test distributions with different textures, distractors, and lighting conditions. The policy takes an image I from the test distribution and outputs a predicted action, $\hat{a} = \pi_\phi(I)$. In this research, we aim to study multiple approaches, depth-based projection, multi-view fusion and multi-resolution features, to obtain robust 3D-representations from 2D pre-trained models. The image shown is an example and has been adapted from Wang et al. [11].

3.1 3D Feature Extraction and Representation Comparison

A key objective is to identify the most effective pre-trained visual representation for robotic manipulation and the optimal method for integrating it into a 3D context. We will go beyond basic 2D-to-3D lifting using only CLIP features, exploring a variety of approaches.

3.1.1 Pre-trained Representation Candidates

We will investigate a range of pre-trained models including the following:

- *CLIP* [1]: Serves as a strong baseline, providing robust visual-language understanding due to its training on a massive dataset of image-text pairs.
- *R3M* [2]: Offers representations specifically trained for robotic manipulation tasks, potentially leading to improved data efficiency and task-specific performance.
- *SAM2* (as used in SAM2Act [5]): Provides multi-resolution features, which may be particularly valuable for capturing both coarse and fine-grained spatial details, crucial for precise manipulation.

- *DINOv2* [7]: Leverages self-supervised pretraining to learn robust, general-purpose visual features without finetuning, showing promising performance across diverse image distributions.
- *Pre-trained VLMs* (used in Pi0 [16]): These models leverage internet-scale pre-training, potentially providing broad visual and semantic understanding that can generalize well to diverse robotic environments and tasks. We will consider models like PaliGemma, used in Pi0, due to their demonstrated capabilities in visual reasoning and language understanding.
- *MVP* [3]: Trained via masked visual pre-training, specifically for robot manipulation, offering an alternative to R3M.

3.1.2 3D Lifting and Representation Methods

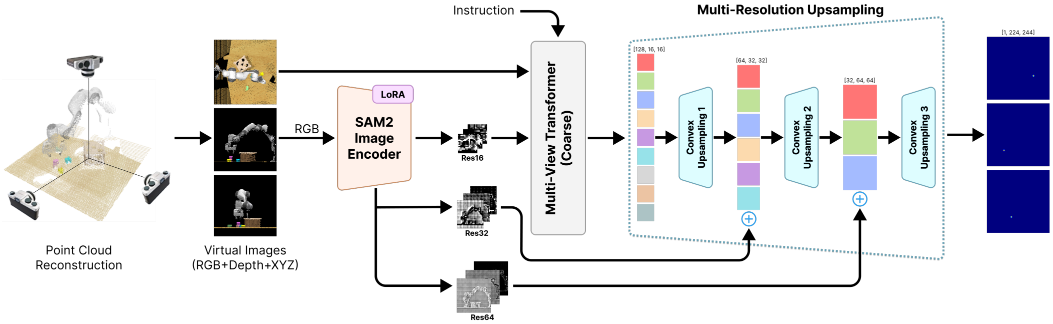


Figure 2: **Multi-Resolution Upsampling (adapted from SAM2Act [5])**. This figure illustrates the multi-resolution upsampling technique used in SAM2Act. Feature maps are processed by a cascade of convex upsamplers, with multi-resolution embeddings from the SAM2 encoder integrated at each stage. This allows the network to combine both high-level semantic information and fine-grained spatial details for accurate feature representation. Our work explores using this and other methods to create robust 3D feature representations.

We will compare the geometric consistency of features from explicit 3D learners (Find3D, Trellis) against lifted 2D representations (CLIP, DINOV2) through SE(3) equivariance tests and cross-view feature similarity metrics.

1. Depth-Based Projection (Baseline): We extract features $\phi_{\text{CLIP}}(I) \in \mathbb{R}^d$ (e.g., from CLIP) and project them into 3D using depth maps (D) and camera intrinsics (K). For a pixel (u, v) with depth d, the 3D point $p = K^{-1} \left[u\ v\ 1\right]^T d$ is calculated. This forms a basic 3D feature volume $\mathcal{F} \in \mathbb{R}^{X \times Y \times Z \times d}$.
2. Multi-View Fusion (Inspired by RVT-2 [12]): Instead of a single RGB-D view, we render the scene from multiple virtual viewpoints. We then extract features from each rendered view and fuse them. This approach, drawing inspiration from RVT-2, aims to enhance robustness against occlusions and viewpoint variations. The specific fusion mechanism (e.g., averaging, concatenation, or attention-based methods) will be a key area of our investigation.
3. Multi-Resolution Features (Inspired by SAM2Act [5], shown in Figure 2): We leverage models like SAM2, which provide features at multiple resolutions. This may involve a multi-resolution upsampling technique, similar to that used in SAM2Act, to combine features extracted from different scales. This approach is hypothesized to enhance fine-grained spatial reasoning. The upsampling is performed as:

$$X^{l+1} = \text{LayerNorm}(U(X^l) \oplus E^l)$$

where X^l is the feature map at stage l, E^l is the corresponding multi-resolution embedding from the visual encoder, and U is the upsampling operator.
4. Explicit 3D Representations: Explore recent self-supervised 3D representation learners like *Find3D* [8] (geometric reconstruction via feature matching) and *Trellis* [9] (3D-aware latent diffusion). These methods could provide more geometrically consistent features than 2D-lifted approaches.

We aim to maintain $SE(3)$-equivariance in our 3D representations whenever feasible. This property is vital for generalization across viewpoints. We will assess the degree to which each representation and lifting method achieves $\Phi_{3D}(R \cdot o + t) = R\,\Phi_{3D}(o) + t$.

3.2 Policy Architectures for Manipulation

This section remains largely consistent with the earlier version but places greater emphasis on comparing how different policy architectures interact with different 3D representations.

We will explore the following policy architectures:

- Diffusion Policies: Employing a conditional denoising process, similar to the approach in 3D Diffuser Actor. The noisy trajectory τ_i is iteratively refined based on the observation and language instruction $a_t^{(k-1)} = \sqrt{\alpha_k}\left(a_t^{(k)} - \sqrt{1-\alpha_k}\,\epsilon_\theta\left(a_t^{(k)}, \mathcal{F}, l\right)\right)$ where ϵ_θ is the noise prediction network and α_k is the noise schedule. We will adapt a 3D relative transformer, similar to that used in 3D Diffuser Actor
$\epsilon_\theta(\tau^i, i, o) = \text{Transformer}(\Phi_{3D}(o), \text{RotaryEmbed}(\tau^i))$. Figure 3 demonstrates the policy architecture for 3D Diffuser Actor.
- Autoregressive Transformer Policies (ACT): We will also be testing other autoregressive transformer policies as a policy architecture given by $a_t = \text{Transformer}(\mathcal{F}, a_{1:t-1}, l)$. Since we incorporate 3D positional embeddings, the action is given by $a_t = \text{Decoder}(\Phi_{3D}(o_{1:t}), a_{1:t-1})$
- Flow Matching Approaches: These methods, less explored in robotics for direct manipulation control, offer an alternative to modeling high-dimensional action spaces. Instead of directly predicting actions (like diffusion or ACT), flow matching models learn a continuous vector field that guides the actions over time. Inspired by works like [15], [16], we will investigate adapting flow matching to our 3D-aware framework. This will involve extending a 3D-conditioned Vector Neural Network (VNN) to learn the vector field:

$$v_\theta(x, t) = \text{VNN}(x, \Phi_{3D}(o))$$

  where $v_\theta(x, t)$ represents the velocity of the action at state x and time t.

Since our aim is to optimize the imitation learning objective dependent on the visual representation ϕ and the policy π, we aim to iterate between all such representations and policies to effectively optimize the objective. This allows a comparison of the suitability of each representation for learning with each policy and the robustness of each combination to distribution shifts (viewpoint changes, object variations).

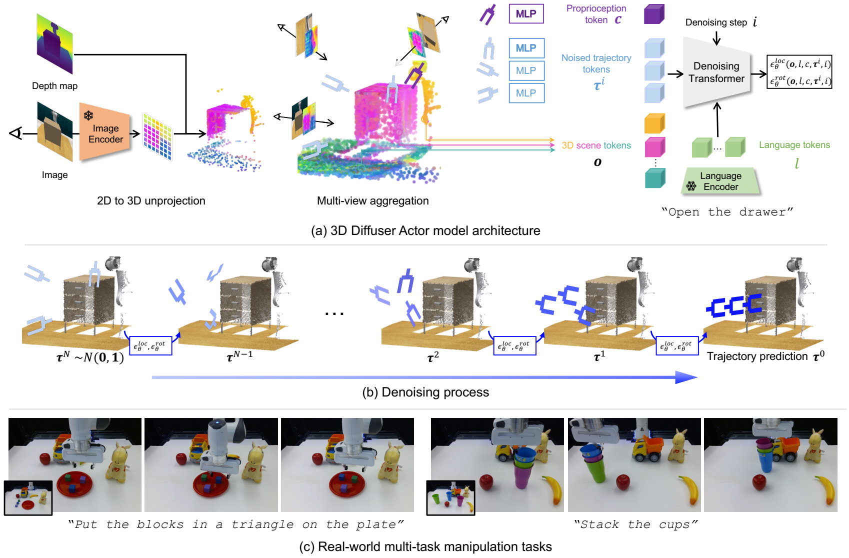
3.3 Refined Imitation Learning Objective

The core of our training procedure will be based on imitation learning. We will adapt and refine the imitation learning objective from 3D Diffuser Actor [4], making it suitable for our broader framework, which includes multi-task learning and different policy architectures (diffusion, ACT, and potentially flow matching).

The 3D Diffuser Actor objective function is:

$$\mathcal{L}_\theta = w_1 \|\epsilon_\theta^{loc}(o, l, c, \tau^i, i) - \epsilon^{loc}\| + w_2 \|\epsilon_\theta^{rot}(o, l, c, \tau^i, i) - \epsilon^{rot}\| + BCE(f_\theta^{open}(o, l, c, \tau^i, i), a_{1:T}^{open})$$

where w_1 and w_2 are hyperparameters, BCE represents binary cross-entropy loss for the gripper state, ϵ^{loc} and ϵ^{rot} represent the noise added to the 3D locations and rotations, respectively and f^{open} represents end-effector opening. This objective function will serve as a starting point, but we will adapt it as necessary. For example, we might incorporate multi-task extensions, adjust the weighting of different loss terms, or develop alternative objective formulations that are better suited for the ACT or flow matching policies. The precise details of the objective function will depend on the specific policy architecture and the characteristics of the 3D representation being used. We may adapt the trajectory matching and gripper state objective proposed in the second proposal to accomodate the other architectures.


Depth map
Image
Image Encoder
2D to 3D unprojection
Multi-view aggregation
MLP
MLP
MLP
MLP
Proprioception token c
Noised trajectory tokens τ^i
3D scene tokens o
Denoising step i
Denoising Transformer
$\epsilon_\theta^{loc}(o, l, c, \tau^i, i)$
$\epsilon_\theta^{rot}(o, l, c, \tau^i, i)$
Language tokens l
Language Encoder
"Open the drawer"
$\tau^N \sim N(\mathbf{0}, \mathbf{1})$
$\epsilon_\theta^{loc}, \epsilon_\theta^{rot}$
τ^{N-1}
...
τ^2
τ^1
Trajectory prediction τ^0
"Put the blocks in a triangle on the plate"
"Stack the cups"

(a) 3D Diffuser Actor model architecture

(b) Denoising process

(c) Real-world multi-task manipulation tasks

Figure 3: **3D Diffuser Actor Architecture (adapted from [4])**. This figure illustrates the architecture of a diffusion-based policy, specifically 3D Diffuser Actor. Visual observations (o_t), proprioception (c_t), and a noised trajectory estimate (τ_i) are converted into 3D tokens. A 3D relative denoising transformer fuses these tokens, along with language instruction tokens (l), to predict the noise added to the trajectory. Our work explores similar diffusion-based policies, but with a focus on comparing different 3D feature representations as input.

By systematically varying and evaluating these components, we aim to establish the optimal configurations for creating robust and generalizable manipulation policies. This experimental design will provide insight into which representations and lifting methods lead to greater stability and accuracy.

3.4 Language Conditioning

Language instructions (l), representing task goals or commands (e.g., "insert the plug into the socket"), play a crucial role in guiding the robot's behavior. These instructions will be processed using a pre-trained language encoder (e.g., from CLIP, or potentially a stronger VLM like the one used in Pi0). The resulting language embeddings will be used to condition both the 3D feature extraction process and the policy. For instance, in diffusion policies, the noise prediction network ϵ_θ would become:

$$\epsilon_\theta = \epsilon_\theta(a_t^{(k)}, \mathcal{F}, M_t, l).$$

This integrated conditioning ensures that the semantic information contained in the language instruction directly influences both the way visual features are extracted and interpreted, and the subsequent action predictions made by the policy. The language embeddings effectively provide a task-specific context that guides the entire perception-action pipeline.

3.5 Memory-Augmented Learning

To address long-horizon tasks and mitigate error accumulation, especially in autoregressive models like ACT, we aim to incorporate a memory module inspired by the approach in SAM2Act+ [5]. This module will enable the policy to maintain context over extended sequences of actions and observations, providing a form of episodic memory.

The SAM2Act+ memory system consists of three key components: a Memory Bank, a Memory Encoder, and a Memory Attention mechanism. The Memory Bank (Q) is a set of FIFO queues (one per view in a multi-view setup) storing past "memories." These memories are derived from the predicted translation heatmaps and raw embeddings from the Multi-View Transformer (MVT).

This mechanism allows the model to selectively attend to relevant past information. These memory-conditioned embeddings, E_{mem} are then used by policy, instead of directly using raw MVT embeddings. We can then enable our policy (e.g., the diffusion policy's noise prediction network) to be conditioned on these memory-augmented features:

$$\epsilon_\theta = \epsilon_\theta(a_t^{(k)}, \mathcal{F}, E_{mem}, l).$$

Here, $\mathcal{F}$ represents the 3D features, l the language embedding, and E_{mem} the memory-conditioned embeddings. By conditioning on E_{mem}, the policy gains access to a history of past observations and actions, not just the current state.

We hypothesize that this will be particularly crucial for tasks that require remembering previous states, actions, or object locations, enabling the robot to perform actions that depend on information that is no longer directly observable. We want to explore if incorporating a memory module allows generalization to different environments and operation under environmental perturbations and to what extent.

4 Work Plan

pre SURF: Configure Franka setup to collect demonstration data, run overfitting experiments on baselines (SAM2Act, 3D Diffuser Actor, RVT-2) using collected data to replicate performance.

Week 1: Integrate pretrained visual encoders (CLIP, R3M, SAM2) to extract semantic features from RGB-D data, initiate depth-based projection and multi-view geometry for 3D feature volume creation, and document initial encoder outputs.

Week 2: Implement and validate 3D lifting procedures by applying controlled SE(3)-equivariance tests, refining the feature extraction pipeline, and preparing data for subsequent processing.

Week 3: Work on integrating memory module with GRU updates in developed architecture, setting up a simulation environment, and linking the module to the existing 3D feature pipeline.

Week 4: Develop and implement the memory module, conduct unit tests on GRU updates, and integrate the module into the policy framework for early evaluation.

Week 5: Perform ablation studies on the memory module, refine its integration via end-to-end data flow tests.

Week 6: Launch training of diffusion-based, ACT, and flow matching policies by incorporating 3D features and memory outputs, and set up the training environment to monitor convergence metrics.

Week 7: Integrate language conditioning using triplet loss and cross-attention fusion of CLIP/R3M features into the policy architecture, validate cross-modal alignment, and resolve integration challenges.

Week 8: Refine policy models through continued training, adjust hyperparameters based on performance metrics, and iterate on fusion strategies to enhance language-conditioned outcomes.

Week 9: Execute comprehensive evaluations on synthetic benchmarks (RLBench, CALVIN and real-world) by introducing perturbations and measuring task success, sample efficiency, and other key metrics.

Week 10: Finalize the comprehensive project report, and prepare final documentation and an open-source code release.

5 Conclusion

This proposal outlines a research plan to develop robust 3D feature representations for generalizable vision-language-action policies in robotic manipulation. By combining 3D lifting of pretrained features, memory-augmented policy architectures, and integrated language conditioning, our approach aims to addresses the limitations of current 2D-based approaches. The methodology aims to improve performance under controlled conditions and enhance robustness to real-world variations and task novelty. Through theoretical analysis and rigorous experimentation, we expect to demonstrate significant improvements in generalization and sample efficiency, bridging the gap between simulation and real-world robotic manipulation.

References

[1] A. Radford et al., “Learning Transferable Visual Models From Natural Language Supervision,” *arXiv:2103.00020*, 2021.

[2] S. Nair et al., “R3M: A Universal Visual Representation for Robot Manipulation,” *arXiv:2203.12601*, 2022.

[3] T. Xiao et al., “MVP: Masked Visual Pre-training for Robot Manipulation,” *arXiv:2303.12073*, 2023.

[4] P. Mandikal et al., “3D Diffuser Actor: Policy Diffusion with 3D Scene Representations,” presented at CoRL, 2023.

[5] Y. Zeng et al., “SAM2Act: Learning to Segment and Act in the World,” *arXiv:2501.18564*, 2024.

[6] X. Li et al., “Act3D: Integrating NeRF-based 3D Reconstruction with Robotic Manipulation,” *arXiv:2306.17817*, 2023.

[7] A. Caron et al., “DINOv2: Learning Robust Visual Features without Supervision,” *arXiv:2304.07193*, 2022.

[8] Z. Ma et al., “Find Any Part in 3D,” *arXiv:2411.13550*, 2024.

[9] J. Xiang et al., “Structured 3D Latents for Scalable and Versatile 3D Generation,” *arXiv:2412.01506*, 2024.

[10] T. S. Cohen et al., “SE(3)-Equivariant Neural Networks,” in *ICML*, 2020.

[11] C. Wang et al., “What Makes Pre-Trained Visual Representations Successful for Robust Manipulation?,” *arXiv:2312.12444*, 2023.

[12] A. Goyal et al., “RVT-2: Learning Precise Manipulation from Few Demonstrations,” *arXiv:2306.14896*, 2023.

[13] T. Zhao et al., “ACT: Action-Conditioned Transformer for Robotic Manipulation,” ICLR 2023.

[14] C. Chi et al., “Diffusion Policy: Visuomotor Policy Learning via Action Diffusion,” RSS 2023.

[15] Y. Lipman et al., “Flow Matching for Generative Modeling,” ICLR 2023.

[16] K. Black et al., “π0: A Vision-Language-Action Flow Model for General Robot Control,” Physical Intelligence.