Lift3D-VLA: Lifting VLA Models to 3D Geometry and Dynamics-Aware Manipulation

Jiaming Liu†, *Student Member, IEEE,* Qingpo Wuwu†, Nuowei Han†, Hao Chen†, Zhuoyang Liu, Fan Fei, Yueru Jia, Chenyang Gu, Yandong Guo, Boxin Shi, *Senior Member, IEEE,* and Shanghang Zhang

***Abstract*—Recently, Vision–Language–Action (VLA) models have demonstrated strong generalization across diverse tasks. However, effective robotic manipulation in physical environments fundamentally requires geometric understanding and spatial reasoning. While some VLA approaches attempt to incorporate 3D information, they are constrained by limited data availability and geometric information loss in current 3D encoding pipelines, and fail to jointly capture 3D geometry and temporally structured actions in dynamic environments. To address these limitations, we introduce Lift3D-VLA, a unified VLA framework that equips models with explicit 3D point cloud reasoning and enables temporally coherent action generation. First, building upon our previous work Lift3D, an enhanced 2D model-lifting strategy is proposed to geometrically align 3D points with pretrained 2D positional embeddings. This design enables direct point-cloud encoding within the VLA vision encoder while minimizing spatial information loss. Based on explicit 3D inputs, we propose Geometry-Centric Masked Autoencoding (GC-MAE), a dual-objective self-supervised framework that reconstructs the current point cloud while predicting its future geometric evolution. This formulation allows the 2D vision encoder to internalize both 3D structure and physical dynamics. To fully exploit 3D representations, we further design layer-wise temporal action modeling, which leverages multiple layers of the LLM to collaboratively predict action chunks, enabling temporally consistent predictions. Across 22 simulated tasks and 8 real-world manipulation tasks, Lift3D-VLA achieves 10.8% and 11.1% higher mean success rates on MetaWorld and RLBench than the best-performing prior VLA methods, and outperforms the strongest real-world baseline by 4 percentage points, while exhibiting stronger generalization to out-of-distribution perturbations. Project website: https://lift3dvla.github.io/.**

***Index Terms*—Robotic Manipulation, Vision–Language–Action Model, 3D Representation.**

I. Introduction

Vision–Language–Action (VLA) models have recently emerged as a promising paradigm for robotic manipulation [2]–[4], showing strong generalization across diverse tasks by combining visual perception, language conditioning, and action generation. Despite this progress, robotic manipulation fundamentally requires spatial reasoning in the physical world [5]–[9]: the robot must infer 3D structure, reason about geometric relationships (e.g., reachability, occlusion, and contact), and plan actions that remain temporally consistent as the geometry evolves. Purely 2D VLA pipelines often struggle to reliably capture these geometric constraints, particularly in cluttered or dynamic environments.

A natural direction is to explicitly inject 3D information into VLA models or manipulation policies, with existing approaches primarily falling into two paradigms, as shown in Figure 1 a). First, some methods directly encode point clouds, voxels, or multi-view observations [9]–[16]. However, unlike large-scale 2D vision–language pretraining, large robotic 3D datasets and strong 3D foundation encoders remain scarce, making it difficult to learn transferable and generalizable 3D representations. Second, other approaches rely on cross-modal transformations, such as lifting 2D features into 3D space [5], [17]–[20] or projecting 3D point clouds into multi-view images [21]–[24]. However, such transformations are inherently lossy, compromising geometric fidelity and weakening the structural correspondence between 2D pretrained representations and 3D spatial structure. This ultimately limits scalability and diminishes the benefits of large-scale 2D pretraining.

To mitigate these bottlenecks, our prior work Lift3D [1] innovatively proposes a 2D-pretraining reuse paradigm to endow 2D foundation models with 3D geometric perception for robotic manipulation. Rather than training a new 3D foundation model from scratch, Lift3D progressively augments a pretrained 2D backbone with implicit 3D representation enhancement via a task-aware masked autoencoder (MAE), and an explicit point-cloud encoding that aligns 3D points with pretrained 2D positional priors. This design largely eliminates the need for massive 3D pretraining data, reduces spatial fidelity loss introduced by cross-modal transformations, and maintains computational efficiency and scalability.

Despite its effectiveness, Lift3D still exhibits two limitations that become critical when applied to VLA models in dynamic environments. **First**, its implicit enhancement remains indirect: the masked reconstruction objective operates primarily on RGB and depth signals, which do not explicitly enforce learning of coherent 3D structure. As a result, the model struggles to acquire robust geometric understanding that generalizes across viewpoints and physical interactions. **Second**, Lift3D is not designed to jointly model evolving 3D geometry and temporally structured actions, which can lead to brittle behavior when long-horizon, time-consistent decisions are required. In dynamic manipulation scenarios, an agent must reason about how the scene geometry evolves over time while generating temporally coherent action sequences [25]–[27]. These limitations highlight the need for a framework that

Jiaming Liu, Qingpo Wuwu, Nuowei Han, Zhuoyang Liu, Yueru Jia, Chenyang Gu, Fan Fei, Boxin Shi, and Shanghang Zhang are with the State Key Laboratory of Multimedia Information Processing and National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, Beijing, China. Hao Chen is with the CUHK, Shatin, Hong Kong. Yandong Guo is with the AI^{2}Robotics, Beijing, China.

†Jiaming Liu, Qingpo Wuwu, Nuowei Han, and Hao Chen contributed equally as co-first authors. Corresponding author: Shanghang Zhang. E-mail: {jiamingliu@stu.pku.edu.cn, shanghang@pku.edu.cn}

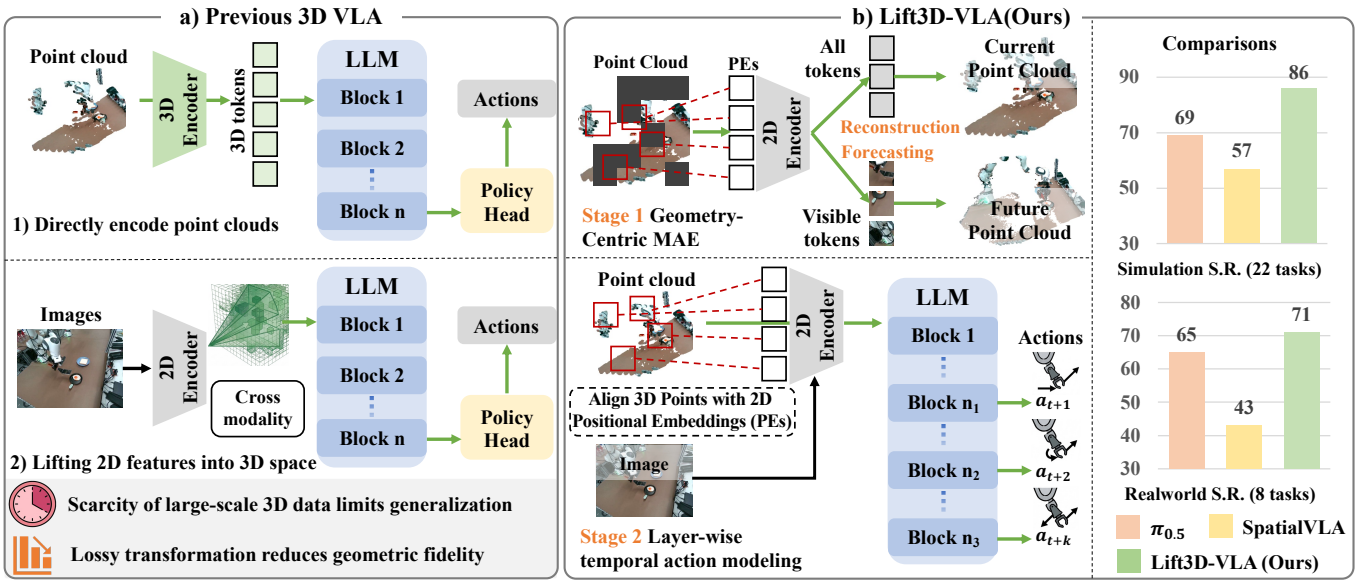

Fig. 1: **Overview.** a) Unlike previous 3D VLA methods that encode point clouds either with newly introduced 3D encoders or by projecting features between 2D and 3D spaces, b) We propose **Lift3D-VLA**, equipping 2D VLA models with explicit 3D reasoning and temporally coherent action generation. First, following our prior work Lift3D [1], we align 3D points with 2D positional embeddings to enable direct point-cloud encoding. Building on this, in Stage 1 we introduce Geometry-Centric MAE, which reconstructs the present point cloud while predicting its future geometric evolution, enhancing 3D representations. In Stage 2, we further propose layer-wise temporal action modeling, leveraging multiple layers of the LLM to produce temporally consistent actions. Across 22 simulated and 8 real-world tasks, Lift3D-VLA achieves SOTA performance.

learns rich 3D structure, models how geometry evolves, and generates temporally consistent actions.

In this paper, as shown in Figure 1 b), we propose Lift3D-VLA, a unified VLA framework that enables explicit 3D point-cloud reasoning and temporally structured action generation. Building upon the core philosophy of Lift3D, we inherit the 2D model lifting strategy to enable faithful and efficient explicit 3D encoding. Specifically, we project point cloud data onto multiple virtual planes, establishing a structured alignment between 3D geometry and 2D positional embeddings. To avoid geometric distortion, we further leverage camera parameters to anchor the front virtual plane to the camera viewpoint. This simplified design enables point-cloud tokens to be processed by a pretrained 2D VLA model while minimizing spatial information loss. To improve the 3D representations learned by the VLA vision encoder, we propose Geometry-Centric Masked Autoencoding (GC-MAE). Given explicit 3D input, GC-MAE reconstructs the corresponding point cloud under a temporally asymmetric self-supervision scheme, jointly recovering present geometry and predicting its future geometric evolution. This recovery–forecast formulation encourages the model to capture physical dynamics in addition to static structure, enabling the VLA's 2D encoder to develop a robust understanding of 3D physical properties. Building upon these enriched 3D representations, we introduce layer-wise temporal action modeling to improve action generation in dynamic environments. Instead of relying on a separate decoding head to predict action chunks [4], [28], [29], our method leverages sequence representations from intermediate to deep LLM layers to capture temporal dependencies. Multiple layers consecutively predict sequential steps within the action chunk, resulting in temporally coherent and execution-stable actions.

Compared with Lift3D, we scale up pretraining to equip Lift3D-VLA with richer pretraining knowledge. Specifically, we perform GC-MAE self-supervised training on 140K trajectories and robotic pretraining on 400K trajectories collected from large-scale robotic datasets [30]–[32]. The downstream evaluation is also extended by introducing dual-arm tasks beyond those considered in Lift3D. Across 22 simulated tasks and 8 real-world tasks, Lift3D-VLA achieves 10.8% and 11.1% higher mean success rates on MetaWorld [33] and RLBench [34] than prior state-of-the-art VLA methods, respectively, and outperforms the strongest real-world baseline. We further validate its long-horizon manipulation capability on tasks such as repeatedly scooping eggs from a pan under continuously changing conditions, while demonstrating strong generalization to unseen objects, backgrounds, and lighting variations. In summary, our main contributions are:

- We extend our preliminary work [1] into a unified Lift3D-VLA framework that integrates explicit 3D point-cloud reasoning with temporally structured action generation, supported by large-scale robotic pretraining. The key improvements include:
- We propose Geometry-Centric Masked Autoencoding, a self-supervised scheme that reconstructs the present point cloud while forecasting its future geometric evolution, enabling the VLA's vision encoder to learn more robust 3D representations.
- We develop a layer-wise temporal action modeling strategy that leverages sequence representations from con-

secutive LLM layers to predict action steps, improving temporal coherence in dynamic physical environments.

II. Related Work

A. Representation Learning for Robotics

Recent progress in pretrained visual representations has been largely driven by self-supervised learning paradigms such as contrastive learning [35], self-distillation [36], and MAE [37]. Building upon these advances, several works aim to enhance visual representations to better support robotic perception and control. For instance, R3M [38] learns universal embodied representations from large-scale human video data using contrastive learning, VIP [39] produces dense reward functions for unseen robotic tasks, and MVP [40], VC-1 [41], and Voltron [42] explore MAE-style pretraining for robotic perception. However, these approaches are still primarily built upon 2D visual representations, which often lack the spatial fidelity required for complex manipulation. To address this limitation, recent work explores 3D-aware robotic representations. One approach learns geometry-aware representations through multi-view modeling or reconstruction, where methods such as MV-MWM [43], 3D-MVP [44], SPA [45], CL3R [46], and HyperMVP [47] learn consistent geometric features across views and align 3D structure with pretrained 2D semantic representations. Another approaches [48], [49] integrate geometric modeling more directly into policy learning frameworks, which improve spatial grounding through multi-view geometry, robot-centric reconstruction, canonical 3D coordinates, or fused RGB-D scene representations. Different from the aforementioned approaches, our previous work Lift3D [1] leverages large-scale 2D foundation models and progressively augments them with implicit 3D representations, enabling enhanced spatial modeling while avoiding learning pretrained knowledge from scratch. Building upon Lift3D, we further propose Geometry-Centric Masked Autoencoding, a dual-objective self-supervised framework that reconstructs the present point cloud while predicting its future state, which explicitly enforces the learning of coherent 3D geometric structure and physical dynamics.

B. Vision-Language-Action Models

VLA models extend pretrained vision-language models (VLMs) to robotic control, enabling robots to follow natural language instructions while benefiting from large-scale semantic priors [50], [51]. Early VLA research primarily focused on scaling robot demonstration data and adapting pretrained VLMs for policy learning [51], demonstrating improved generalization across manipulation tasks. Subsequent works further improve the expressiveness of VLA policies by introducing continuous generative policy heads, where diffusion-based approaches model complex action distributions through iterative denoising [3], [28], while flow-matching formulations provide a more efficient alternative for continuous action generation [4], [52]. Building upon these generation paradigms, another line of VLA research [53], [54] investigates the use of future state prediction to facilitate physical world modeling and improve action generation. Motivated by the need for richer and more unified representations, recent approaches attempt to integrate generative and understanding capabilities within a single model using mixture-of-transformers architectures [29], [55]–[57]. Beyond 2D representations, some works focus on enhancing spatial understanding and reasoning in VLA models. Methods such as 3DS-VLA [24], SpatialVLA [20], and PointVLA [16] introduce geometric information by aligning visual features with spatial coordinates or incorporating point cloud observations, while other approaches extend the perception space toward multisensory representations by integrating visual, depth, or tactile signals [58]. In contrast, Lift3D-VLA builds upon Lift3D's 2D model lifting strategy, allowing for the explicit incorporation of 3D point clouds, and further introduces layer-wise temporal action modeling to generate temporally coherent actions from 3D inputs.

III. Preliminaries

Task Formulation. VLA models aim to enable robots to execute diverse tasks by mapping visual observations and language instructions to action sequences. Formally, given a natural language task instruction ℓ and observation o_t at timestep t, a VLA policy π_θ predicts an action chunk $(a_t, a_{t+1}, \ldots, a_{t+H-1})$ for task execution:

$$\pi_\theta : (\ell, o_t) \rightarrow (a_t, a_{t+1}, \ldots, a_{t+H-1}), \tag{1}$$

where H denotes the action chunk size and the observation $o_t = \{I_t^1, \ldots, I_t^n, P_t, q_t\}$ comprises n camera images, point cloud P_t, and proprioceptive state q_t (e.g., joint positions). Each action $a_t \in \mathbb{R}^7$ represents the end-effector pose:

$$a_t = [\Delta x, \Delta y, \Delta z, R_r, R_p, R_y, g], \tag{2}$$

where $\Delta x, \Delta y, \Delta z$ represent relative Cartesian position offsets, R_r, R_p, R_y denote absolute Euler angles (roll, pitch, yaw) for rotation, and $g \in [0, 1]$ indicates the gripper width. For dual-arm manipulation tasks, the action space is extended to $a_t \in \mathbb{R}^{14}$ by concatenating two 7-DoF vectors corresponding to the left and right arms.

VLA Architecture. VLA models largely inherit the design of VLMs, consisting of three components: (1) a *vision encoder* that processes multi-view images into visual tokens, (2) a *large language model (LLM)* that models multimodal features, and (3) an *action expert* that generates action predictions, either using the LLM itself or an additional action head. These components are typically initialized from pre-trained VLMs to leverage internet-scale vision–language knowledge. The training objective maximizes the log-likelihood of action chunks given observations and instructions:

$$\max_\theta \mathbb{E}_{(a_{t:t+H-1}, o_t, \ell) \sim \mathcal{D}} \left[\log \pi_\theta(a_{t:t+H-1} \mid o_t, \ell) \right]. \tag{3}$$

The action outputs can be represented as either discrete tokenizations [51] or continuous formulations, such as diffusion [3], [59] and flow matching [4], [60], which provide more expressive action distributions for complex manipulation tasks.

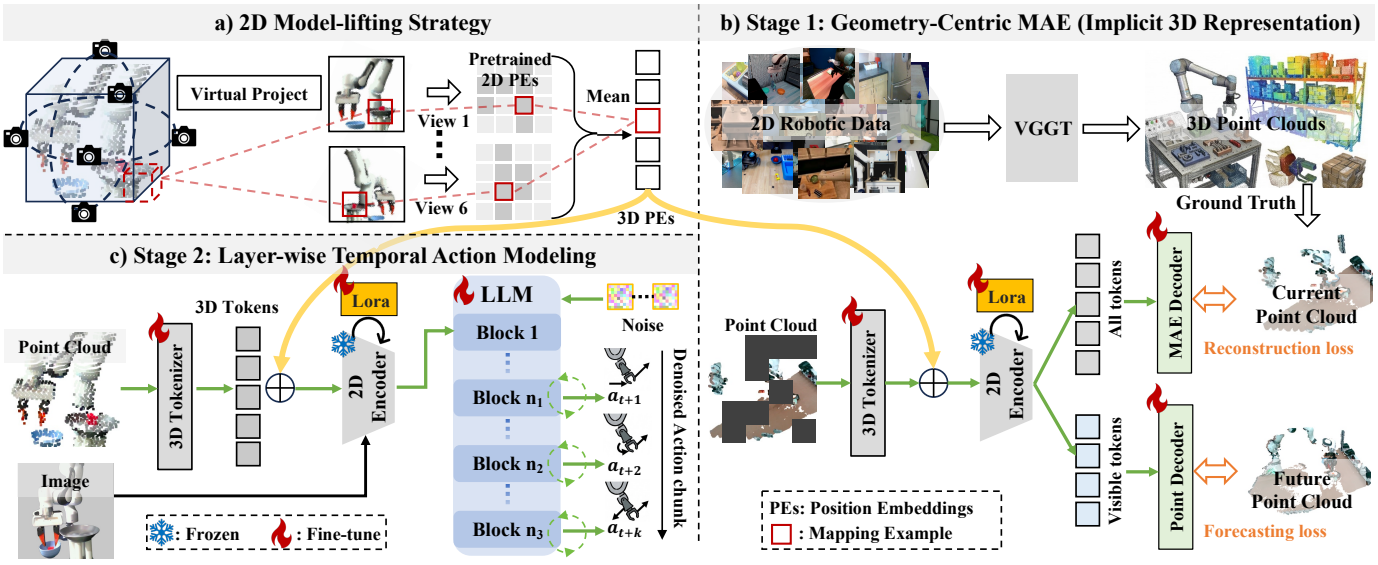

Fig. 2: **Lift3D-VLA Framework. a)** Following our previous work Lift3D, we perform virtual projection to align 3D points with pretrained 2D positional embeddings (PEs), thereby constructing geometry-aligned 3D PEs that enable the 2D vision encoder in VLA models to directly process point cloud inputs. **b)** Stage 1. To enhance 3D physical representations, we first leverage VGGT to synthesize 3D point clouds from large-scale robotic data, which serve as self-supervised training targets. We then introduce Geometry-Centric MAE, a framework that reconstructs the current point cloud while simultaneously predicting its future geometric evolution, enabling the model to capture physical dynamics in addition to static spatial structure. **c)** Stage 2. Building on the strengthened 3D representations, we further propose layer-wise temporal action modeling, which leverages the sequence modeling capabilities of intermediate and deep LLM layers to generate temporally consistent action sequences.

IV. Proposed Method (Lift3D-VLA)

Our previous work, Lift3D [1], is limited by its reliance on indirect 3D reconstruction objectives, which hinders effective geometric and physical modeling. Furthermore, it does not jointly model evolving geometry and temporally structured actions, resulting in brittle performance in long-horizon dynamic scenarios. To address these limitations, as shown in Figure 2, we propose Lift3D-VLA, a unified VLA framework that systematically equips VLA models with explicit 3D point cloud reasoning and facilitates temporally coherent action generation. We begin by introducing the overall VLA architecture in Section IV-A. We then present the 2D model lifting strategy in Section IV-B, following our previous work Lift3D, enabling VLA models to directly encode 3D point clouds. In Section IV-C, we describe our Geometry-Centric Masked Autoencoding (GC-MAE), which enhances the implicit 3D and temporal-aware encoding capabilities of the VLA's 2D encoder. Building on these representations, we introduce a novel layer-wise temporal action modeling approach in Section IV-D, which enables temporally coherent action predictions. Finally, we present the training objectives in Section IV-E.

A. Lift3D-VLA Architecture

To enable robots to reason about 3D spatial structures and generate precise manipulation actions, we design a unified VLA architecture that integrates 2D visual observations, 3D point clouds, and language instructions. Our model is built upon a VLM backbone, with parameters initialized from Prismatic VLM [50]. Notably, 2D and 3D observations share a common vision encoder, while employing modality-specific tokenizers and positional embeddings. All raw multisensory inputs are projected into a unified token sequence for the LLM, enabling multimodal reasoning and action prediction.

Vision Encoder. We utilize a dual-encoder design combining SigLIP [61] and DINOv2 [62] to capture complementary visual representations, including both global semantic context and fine-grained details. For each input RGB observation $I_t \in \mathbb{R}^{H\times W\times 3}$ (with $H = W = 224$), the shared vision encoder processes the image and extracts dense visual features. This produces feature embeddings $f^{\text{SigLIP}} \in \mathbb{R}^{N_v\times 1024}$ and $f^{\text{DINO}} \in \mathbb{R}^{N_v\times 1152}$, where $N_v = 256$ denotes the number of visual tokens. The resulting features are then concatenated along the channel dimension to form a unified visual representation for downstream processing.

3D Point Cloud Tokenizer. To augment 2D visual features with explicit 3D geometric information, we introduce a dedicated point cloud tokenizer that converts raw point clouds into structured token representations. The tokenizer consists of three stages: farthest point sampling [63] for downsampling, k-nearest neighbors aggregation for capturing local geometric structures, and a learnable linear projection for feature embedding. Specifically, given a point cloud input $PC \in \mathbb{R}^{1024\times 3}$, the tokenizer produces a compact representation $f^{\text{pc}} \in \mathbb{R}^{N\text{pc}\times d_h}$ with $N_{\text{pc}} = 256$ tokens, where d_h equals the 2D feature channel dimension. The resulting tokens are subsequently encoded with spatially aligned 3D positional embeddings (described in Section IV-B) and fed into the shared vision encoder to extract spatial features.

LLM Backbone. We adopt the 7B LLaMA2 model [64] as the LLM backbone for Lift3D-VLA. LLaMA2 follows a decoder-only transformer architecture with 32 layers, where each layer transforms the input token sequence into progressively refined high-dimensional representations. Visual tokens, geometric tokens, and linguistic tokens are projected into the LLM's word embedding space and concatenated into a unified sequence $f \in \mathbb{R}^{B\times N_t\times d_h}$. This token sequence is then processed jointly by the transformer layers through standard self-attention and feed-forward operations. Unlike prior approaches that introduce additional heads after the LLM for action prediction [4], [28] or rely on the final LLM layer for next-token prediction [3], we instead leverage sequence representations from intermediate to deep LLM layers to generate temporally coherent action chunks, as detailed in Section IV-D.

B. 2D Model-lifting Strategy

In this section, we propose a lifting mechanism that enables the 2D vision encoder in VLA models to explicitly process point cloud inputs, building upon our previous work, Lift3D [1]. Existing approaches typically either project 3D point clouds into multi-view images [21], [22] or lift 2D features into 3D representations [5], [17]. However, such modality transformations often lead to the loss of geometric structure and spatial consistency, making it challenging to effectively represent 3D data for robotic manipulation. Meanwhile, positional embeddings (PEs) in VLA architectures play a critical role in encoding spatial relationships among tokens. A straightforward solution is to design new 3D positional embeddings, but this may introduce a mismatch with pretrained 2D models and hinder knowledge transfer.

To address this, as illustrated in Figure 2 a), we project 3D tokens onto multiple virtual planes and reuse pretrained 2D positional embeddings for 3D encoding. Specifically, raw point clouds are first transformed into high-dimensional features of size $B \times 128 \times 768$ using the 3D point cloud tokenizer. We denote the spatial coordinates of these tokens as $\{C^i_{3D}\}^k_{i=1}$. Each 3D coordinate is then projected onto n virtual planes, producing corresponding 2D coordinates $\{C^{ij}_{2D}\}^n_{j=1}$. This projection is parameter-free and efficient. We adopt a cube-based projection with six faces to ensure comprehensive coverage of spatial information. In contrast to Lift3D, which relies on randomly assigned front-view virtual planes, we leverage camera extrinsic parameters to align the virtual front view with the observation camera. This alignment enforces a consistent projection viewpoint, improving geometric consistency and reducing distortion. Each virtual plane shares the pretrained 2D positional embedding grid. Based on the projected coordinates, each 3D token is associated with n positional embeddings, denoted as $\{PE_{2D}(C^{ij}_{2D})\}^n_{j=1}$. To obtain a unified positional representation, we aggregate these embeddings by averaging:

$$PE_{3D} = \frac{1}{n}\sum_{j=1}^{n} PE_{2D}(C^{ij}_{2D}). \tag{4}$$

The resulting PE_{3D} is combined with the 3D token features and fed into the 2D vision encoder. By reusing pretrained 2D positional embeddings from multiple views, this design captures diverse spatial relationships while mitigating information loss. With explicit 3D inputs available, we further propose a novel 3D self-supervised training framework to enhance the 3D representations in VLA models.

C. Geometry-Centric Masked Autoencoding

In our preliminary work, Lift3D [1], we enhance 2D foundation models with implicit 3D awareness by reconstructing depth maps and RGB from task-relevant patches. While effective for static spatial understanding, this design relies on "2.5D" depth representations and does not explicitly enforce coherent 3D geometric structure. Moreover, it treats observations independently, overlooking the temporal dynamics required for continuous manipulation. To address these limitations, as shown in Figure 2 b), we propose GC-MAE, a 3D temporal-aware self-supervised learning framework that leverages the 2D Model-lifting Strategy to directly encode point cloud inputs, and jointly models geometric structure and its temporal evolution via a scalable point cloud synthesis pipeline.

Scalable 3D Data Synthesis. We adopt a scalable 3D data synthesis pipeline to augment existing 2D robotic datasets with 3D point clouds. As most robotic datasets [31], [65], [66] provide only RGB observations without depth and camera parameters, we leverage VGGT [67], a feed-forward transformer pretrained on large-scale visual-geometry data, to generate pseudo 3D annotations. Given an RGB observation $I_t \in \mathbb{R}^{H\times W\times 3}$, VGGT directly predicts the corresponding point cloud $\hat{P}_t \in \mathbb{R}^{N\times 3}$. We apply VGGT to all observations in the pretraining dataset (Table I), producing pseudo point clouds that provide geometric supervision complementary to 2D visual features. We further empirically verify that the quality of these synthesized point clouds is sufficient to support reliable self-supervised training. Building on the synthesized 3D data, we jointly capture static geometric structure and dynamic evolution by designing a dual-branch decoding framework over shared latent representations.

Masked Point Reconstruction. To model static geometric structures, we adopt a masked autoencoding paradigm [68] over 3D point tokens. Specifically, a high proportion of input tokens are randomly masked, and the encoder processes only the visible subset $\mathcal{P}^v_t$. A lightweight transformer decoder reconstructs the 3D coordinates of masked tokens from encoded visible tokens and learnable mask tokens. This intra-frame geometric completion is supervised using the Chamfer Distance (CD):

$$\mathcal{L}_{\text{static}} = \sum_{p\in\mathcal{P}^m_t}\min_{\hat{p}\in\hat{\mathcal{P}}^m_t}\|p-\hat{p}\|^2_2 + \sum_{\hat{p}\in\hat{\mathcal{P}}^m_t}\min_{p\in\mathcal{P}^m_t}\|\hat{p}-p\|^2_2, \tag{5}$$

where $\mathcal{P}^m_t$ denotes the ground-truth point set of masked tokens, and $\hat{\mathcal{P}}^m_t$ denotes the reconstructed point set.

Future Geometric Prediction. To model temporal dynamics, we introduce a future prediction branch that captures inter-frame geometric evolution without explicit motion supervision. Given the visible 3D tokens $\mathcal{P}^v_t$ at time t, the decoder predicts their corresponding geometry in the next frame $\hat{\mathcal{P}}^v_{t+1}$. Unlike

the static reconstruction objective, this branch enforces temporal causality by learning how geometry evolves over time (e.g., object motion or end-effector trajectories). The temporal loss is defined as:

$$\mathcal{L}_{\text{dynamic}} = \sum_{p \in \mathcal{P}^v_{t+1}} \min_{\hat{p} \in \hat{\mathcal{P}}^v_{t+1}} \|p - \hat{p}\|_2^2 + \sum_{\hat{p} \in \hat{\mathcal{P}}^v_{t+1}} \min_{p \in \mathcal{P}^v_{t+1}} \|\hat{p} - p\|_2^2. \tag{6}$$

To adapt the vision encoder for 3D understanding while preserving large-scale pretrained priors, we inject Low-Rank Adaptation (LoRA) [69] into the attention layers. We freeze most backbone parameters and update only the LoRA modules and the 3D tokenizer, ensuring parameter efficiency while mitigating catastrophic forgetting.

D. Layer-wise Temporal Action Modeling

Despite learning strong 3D representations, Lift3D relies on a simple policy head on top of the 2D vision encoder for action prediction, which limits its ability to fully exploit 3D structural reasoning and constrains its generalization capability. To address this limitation, we leverage the intrinsic sequence modeling capability of Transformer-based LLMs [70], [71] to capture temporal dependencies in physical control. Moreover, recent VLA models have shown that action prediction need not be restricted to the final layer, as intermediate representations often contain rich and actionable information [72], [73]. Building on these insights, as illustrated in Figure 2 c), we propose Lift3D-VLA, which introduces a layer-wise temporal action modeling strategy to better utilize the learned spatial features in VLA models while effectively modeling temporal dependencies. Instead of relying solely on the final layer, we progressively decode action sequences across layers, where each layer predicts a corresponding action step. This layer-wise design naturally captures temporal dependencies, enabling more coherent modeling of action sequences.

Specifically, given an action chunk of horizon H starting from time step t, we assign each future step $t+k$ (where $k \in \{0, \ldots, H-1\}$) to a corresponding intermediate layer l_k of the LLM backbone. We then instantiate H action layer $\{\phi_k\}_{k=0}^{H-1}$, where each layer ϕ_k predicts the denoised action at time step $t+k$ from the hidden state $\mathbf{h}_{l_k}$:

$$\hat{\boldsymbol{\epsilon}}_k = \phi_k(\mathbf{h}_{l_k}), \quad k \in \{0, \ldots, H-1\}, \tag{7}$$

where l_k denotes the layer index assigned to step k. Ablation studies are conducted to analyze the impact of the number of layers used for action decoding and the interval between decoding layers. In practice, we empirically find that leveraging the deeper layers of the LLM and uniformly spacing the decoding layers yields the best performance. For example, with a 32-layer LLM backbone and an action chunk of $H = 4$, we uniformly select layers $20, 24, 28, 32$ from the deeper portion of the network to decode actions, assigning one action step to each selected layer. More generally, when the action horizon exceeds the number of selected layers, each layer can predict multiple consecutive action steps, enabling flexible and scalable action decoding. This hierarchical formulation naturally leverages the temporal conditioning across different layers of the LLM: deeper layers (corresponding to future time steps) inherently attend to features from shallower layers (representing current or near-term states), thereby enabling temporally consistent modeling. Furthermore, distributing supervision across network depths facilitates the learning of robust features at multiple levels of abstraction, improving closed-loop stability during dynamic interactions.

E. Training Objectives

As shown in Figure 2 b) and c), our training pipeline consists of two stages: (1) GC-MAE for self-supervised learning, and (2) supervised fine-tuning (SFT) on downstream tasks based on the proposed layer-wise temporal action modeling. After equipping the 2D vision encoder with strong 3D representations in the first stage, the second stage further adapts the VLA model for task-specific action prediction.

GC-MAE self-supervised learning. We optimize the VLA's 2D vision encoder and dual-branch decoder using only geometric supervision, without any action labels. The objective of this stage is a weighted sum of the static reconstruction (Eq. 5) and dynamic prediction losses (Eq. 6):

$$\mathcal{L}_{\text{MAE}} = \mathcal{L}_{\text{static}} + \lambda \cdot \mathcal{L}_{\text{dynamic}}. \tag{8}$$

The weighting factor λ balances the value of the two losses. By minimizing $\mathcal{L}_{\text{MAE}}$, the encoder learns representations that jointly capture 3D structure and dynamics, thereby equipping the VLA's vision encoder with fundamental 3D understanding for downstream robotic manipulation tasks. Note that the decoders are used only during Stage 1.

Supervised Fine-tuning. During Stage 2, we freeze the encoder and fine-tune the LLM backbone along with an action projection MLP, which maps the denoised tokens to the action space, using task-specific demonstrations. For action prediction, we adopt a standard DDPM [74]. Given the layer-wise temporal action modeling strategy in Eq. 7, we supervise the denoising process simultaneously across all time steps within the action chunk. The action generation loss is defined as the average Mean Squared Error (MSE) over horizon H:

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{k,s,\epsilon}\left[\left\|\boldsymbol{\epsilon}_k - \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{z}^{(s)}_{t+k}, s, \mathbf{h}_{l_k})\right\|_2^2\right], \tag{9}$$

where $\boldsymbol{\epsilon}_k$ denotes the ground-truth noise added to action a_{t+k}, $\mathbf{z}^{(s)}_{t+k}$ is the corresponding noisy action at diffusion timestep s, and $\mathbf{h}_{l_k}$ is the hidden representation from layer l_k. During inference, we adopt DDIM [75] with 4 denoising steps, following prior methods [3], [76].

V. Experiment

In Section V-A, we provide a detailed description of the construction of the self-supervised and pretraining datasets. We then benchmark Lift3D-VLA against recent VLA models in Section V-B, evaluating manipulation performance on the simulation environments. The contributions of key components are further isolated and quantified through comprehensive ablation studies in Section V-C. Section V-D demonstrates the real-world effectiveness of Lift3D-VLA on manipulation tasks with both single-arm and dual-arm configurations. Finally, in Section V-E, we evaluate the generalization capability of our method under unseen objects, lighting conditions, and scene backgrounds.

TABLE I: **Datasets used for pre-training.** The names of selected datasets for large-scale pretraining and their sampling ratios (%)

Training Dataset Mixture			
Fractal	9.1%	Austin Sirius	1.2%
Kuka	27.8%	CMU Pickup	0.7%
Bridge	4.1%	UTAustin Mutex	1.6%
Taco Play	2.1%	Berkeley Fanuc	0.6%
Jaco Play	0.3%	CMU Stretch	0.1%
Berkeley Cable	0.2%	BC-Z	5.4%
Roboturk	1.7%	FMB Dataset	5.0%
Viola	0.7%	DobbE	1.0%
Berkeley UR5	0.9%	DROID	7.2%
Toto	1.5%	Stanford Kuka	0.1%
Language Table	3.1%	Robocook	0.1%
Stanford Hydra	3.2%	Maniskill	6.3%
Austin Buds	0.2%	Berkeley RPT	0.1%
NYU Franka	0.6%	QUT Dexterous	0.1%
Furniture Bench	1.8%	RoboSet	1.8%
UCSD Kitchen	<0.1%	BridgeData V2	4.7%
Austin Sailor	1.6%	RoboMind	5.2%

A. Pretraining Configuration

We construct a large-scale pretraining corpus by integrating diverse open-source robotic datasets, including Open-X-Embodiment [65], DROID [31], and RoboMIND [32]. Detailed statistics on data composition and proportions are provided in Table I. Prior to downstream task fine-tuning, we perform a two-stage pretraining procedure. First, we conduct large-scale robotic data pretraining to endow the VLM with VLA capabilities. Subsequently, we further enhance the vision encoder using our proposed Geometry-Centric MAE (GC-MAE) self-supervised learning framework.

For robotic data pretraining, we curate a diverse corpus of 400K trajectories (28M frames) from the aggregated dataset, enabling the model to acquire a strong foundation of motor primitives and physical commonsense reasoning. During this stage, to ensure supervision accuracy, the model takes only 2D images as input and predicts the end-effector pose as the supervision signal (Eq. 9). The implementation details in this stage are consistent with those used for real-world tasks and are described in the following section.

For self-supervised learning, we employ the 3D data synthesis pipeline described in Section IV-C to generate point cloud annotations from RGB observations, resulting in over 140K trajectories with synthesized 3D annotations. For computational efficiency, point clouds are uniformly subsampled to 1,024 points per frame. This pretraining phase is conducted for 15 epochs on the synthesized 3D dataset, using a batch size of 4096 and a learning rate of 1×10^{-4}. We adopt the AdamW optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$, along with a cosine learning rate schedule and linear warmup over the first 10% of training steps. We optimize the vision encoder of the VLA model together with the dual-branch decoder (introduced only during training) using the objective defined in Eq. 8. The decoder comprises separate static and dynamic branches, each consisting of four Transformer layers with eight attention heads per layer.

B. Simulation Experiment

1) Data Collection: We evaluate our method on two commonly used simulation environments. **MetaWorld** [33] provides a large-scale suite of simulated manipulation tasks designed to assess the robustness of robotic policies. In our experiments, we follow [1] and select 13 tasks that cover a diverse set of manipulation skills, including precise positioning, articulated object interaction, and fine-grained manipulation: 1) *Assembly*, 2) *Bin picking*, 3) *Box close*, 4) *Button press*, 5) *Dial turn*, 6) *Drawer open*, 7) *Hammer*, 8) *Hand insert*, 9) *Peg unplug side*, 10) *Push wall*, 11) *Reach*, 12) *Shelf place*, and 13) *Sweep into*. All experiments are conducted using a Sawyer robotic arm equipped with a parallel-jaw gripper. For each task, we collect 100 trajectories using scripted policies provided by the benchmark. All RGB observations are captured from a single third-view corner camera at a resolution of 224×224. **RLBench** [34] is a manipulation benchmark built on the CoppeliaSim robotics simulator. In this environment, we follow previous VLA papers [3], [57] and select 9 tasks that span a range of manipulation skills: 1) *Close box*, 2) *Close laptop*, 3) *Toilet seat down*, 4) *Sweep to dustpan*, 5) *Close fridge*, 6) *Take umbrella out*, 7) *Frame off hanger*, 8) *Wine at rack*, and 9) *Water plants*. All tasks are performed using a Franka Panda robotic arm with a single front-view camera observation. Demonstration trajectories are generated using the Open Motion Planning Library (OMPL) [80], with 100 trajectories per task, and keyframes are extracted following prior work [21], [81]. 3D point cloud annotations are uniformly subsampled to 1,024 points per frame.

2) Single-Task on MetaWorld: Since our previous work, Lift3D [1], does not incorporate language understanding, it cannot be applied to multi-task manipulation. To enable a fair comparison and better evaluate the effectiveness of our proposed method, we validate the GC-MAE self-supervised learning strategy in a single-task setting. Specifically, consistent with Lift3D, we attach an MLP head to the pretrained vision encoder to predict actions, instead of using the full LLM in Lift3D-VLA for action prediction.

Baselines. For 2D visual representation learning, we include CLIP (ViT-Base) [77], R3M [38], and VC [41], all of which are pretrained on large-scale vision datasets and adapted for robotic control. For 3D representation learning, we consider PointNet [63], PointNet++ [78], PointNeXt [79], and SPA [45], which include both generic point cloud encoders and 3D robotic pretraining methods. For 3D robotic policies, we compare with DP3 [14] and our previous work, Lift3D.

Implementation Details. We denote our method as Lift3D-VLA†, which shares identical training configurations with the prior Lift3D model. Both models are trained using the same setups, including a three-layer MLP policy head, the Adam optimizer, and a constant learning rate of 1×10^{-3}. During downstream fine-tuning, the backbone parameters are frozen, and LoRA [69] with rank $r = 2$ is applied to the attention layers. All models are trained for 100 epochs on 8 NVIDIA A800 GPUs. We conduct 25 rollout evaluations every 5 epochs and report the highest average success rate.

Quantitative Results. As shown in the single-task results in

TABLE II: **Results on the MetaWorld.** We evaluate models in both single-task and multi-task settings over 25 rollouts. Success is determined by the built-in MetaWorld evaluation module. Results report average manipulation success rates (S.R.).

Models	Assembly	Bin picking	Box close	Button press	Dial turn	Drawer open	Hammer	Hand unplug	Peg wall	Push	Reach place	Shelf into	Sweep	Mean S.R.
Single-Task Setting														
CLIP [77]	64	92	60	100	82	100	88	36	78	64	56	26	40	68.2
R3M [38]	100	60	92	92	100	100	60	66	96	60	60	36	60	75.5
VC-1 [41]	60	80	66	96	76	100	88	44	50	60	60	36	60	67.4
PointNet [63]	100	44	46	100	94	100	38	32	68	36	52	14	24	57.5
PointNet++ [78]	96	72	86	98	78	84	70	26	78	**98**	48	12	24	66.9
PointNeXt [79]	98	82	78	100	92	100	50	20	78	28	48	12	42	63.7
SPA [45]	96	92	76	100	84	96	100	36	68	55	56	16	64	72.2
DP3 [14]	100	24	48	100	92	100	100	14	98	54	40	18	22	62.3
Lift3D(Clip)	100	92	92	100	100	100	94	64	98	44	74	42	72	82.5
Lift3D(Dinov2)	100	**100**	92	100	100	100	100	**76**	96	40	80	28	80	84.0
Lift3D-VLA†(Clip)	**100**	92	**100**	100	100	100	92	60	**100**	56	88	**72**	92	**88.6**
Lift3D-VLA†(Dinov2)	96	92	100	**100**	**100**	**100**	**100**	56	96	40	**92**	68	**92**	87.1
Multi-Tasks Setting														
OpenVLA [51]	90	70	**70**	100	90	100	50	70	80	50	30	70	**90**	73.9
$\pi_{0.5}$ [4]	100	85	50	100	45	80	75	45	65	55	45	**75**	60	67.7
SpatialVLA [20]	75	65	50	90	65	65	85	45	55	35	65	40	70	61.9
3DS-VLA [24]	75	100	45	90	95	100	90	70	75	50	60	60	**90**	76.9
Lift3D-VLA	**100**	**100**	64	**100**	**100**	**100**	**96**	**88**	**100**	**84**	**72**	52	84	**87.7**

TABLE III: **Results on the RLBench.** All models are trained in the multi-task setting and evaluated over 25 rollouts.

Models	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Umbrella out	Frame off hanger	Place wine at rack	Water plants	Mean S.R.
OpenVLA [51]	60	35	75	55	85	30	15	20	5	42.2
$\pi_{0.5}$ [4]	90	95	85	75	**100**	10	**80**	75	35	71.7
SpatialVLA [20]	80	70	85	20	80	25	40	15	30	49.4
3DS-VLA [24]	85	**95**	**95**	15	90	**80**	50	85	35	70.0
Lift3D-VLA	**95**	80	**95**	**95**	90	65	**80**	**95**	**50**	**82.8**

Table II, incorporating our GC-MAE pretraining consistently improves performance across different visual backbones. Compared with prior baselines, our model significantly outperforms VC-1 (67.4%), SPA (72.2%), and DP3 (62.3%), demonstrating the effectiveness of combining large-scale 2D pretrained knowledge with robust 3D representations. Notably, Lift3D-VLA† consistently surpasses Lift3D under both CLIP and DINOv2 initialization, achieving 88.6% vs. 82.5% and 87.1% vs. 84.0%, respectively. The gains are particularly evident in tasks requiring sustained spatial precision: shelf-place (72% vs. 42%), sweep-into (92% vs. 72%), and reach (92% vs. 80%). These results highlight not only the effectiveness of directly reconstructing 3D point clouds, but also the importance of jointly modeling their future geometric evolution for manipulation tasks.

3) Multi-Task on MetaWorld and RLBench: In this section, we conduct multi-task experiments.

Baselines. Since our previous work, Lift3D, does not support understanding task instructions, it cannot be applied to multi-task settings. Therefore, we compare Lift3D-VLA with recent state-of-the-art (SOTA) VLA models. For 2D VLA baselines, OpenVLA [51] leverages a large pretrained vision-language model to perform end-to-end autoregressive action prediction. $\pi_{0.5}$ [4] is trained on a large-scale, self-collected robotic dataset, enabling strong generalization. For 3D VLA baselines, SpatialVLA [20] augments VLA models with explicit spatial reasoning modules to capture geometric structure, while 3DS-VLA [24] incorporates 3D structural constraints to further enhance spatial understanding.

Implementation Details. All baselines are initialized from their officially released pretrained checkpoints and trained using their original full fine-tuning configurations. For Lift3D-VLA, we also adopt full fine-tuning and jointly train across all tasks for 300 epochs with a learning rate of 1×10^{-4}. All remaining settings are consistent with the single-task setup. Evaluation follows the protocol of prior work [82]. We perform 20 rollouts per task using the final checkpoint, repeat the evaluation with three different random seeds, and report the average success rate.

Quantitative Results. In multi-task experiments, Lift3D-VLA achieves 87.7% average success across 13 MetaWorld tasks, substantially outperforming OpenVLA, $\pi_{0.5}$, SpatialVLA, and 3DS-VLA by 13.8%, 20.0%, 25.8%, and 10.8%, respectively. Notably, our method surpasses our previous work, even though Lift3D is trained in a single-task setting, demonstrating both stronger spatial reasoning and improved model capacity. On RLBench (Table III), Lift3D-VLA achieves 82.8% average success across 9 tasks, consistently outperforming prior VLA methods. In particular, it attains near-perfect success rates on tasks requiring precise spatial reasoning, such as *close box* (95%), *toilet seat down* (95%), and *place wine at rack* (95%). These results highlight the

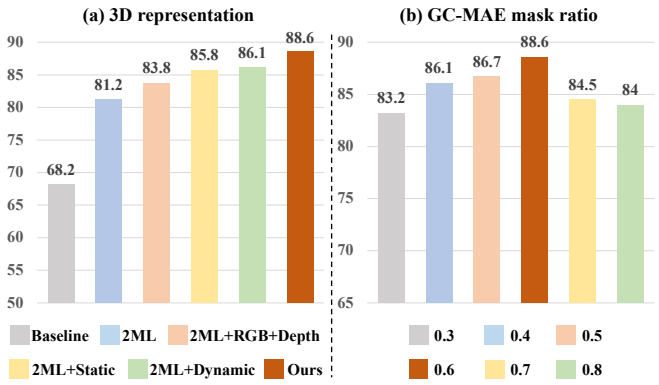

Fig. 3: **Ablation Studies. (a)** Impact of the 2D model-lifting strategy and GC-MAE on 3D representation. **(b)** Effect of the mask ratio in GC-MAE. All experiments are conducted in the MetaWorld single-task setting.

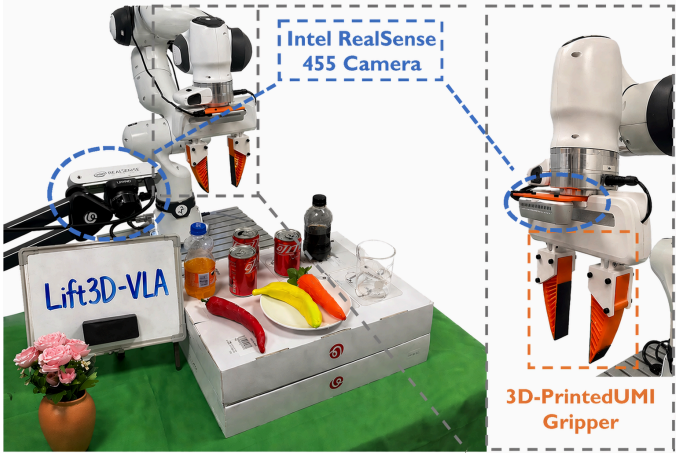

Fig. 4: **Real-world experimental robot platform.** We employ a Franka Research 3 arm equipped with an Intel RealSense D455 RGB-D camera.

TABLE IV: **Ablation of layer-wise temporal action modeling.** 'Layers' indicates the number of Transformer layers used for parallel action prediction, and 'Stride' denotes the interval between them.

Metric	1 Layer	2 Layers	4 Layers
	Stride 1	Stride 1 / 2 / 4	Stride 1 / 2 / 4
Mean S.R.	82.5	84.3 / 85.1 / 85.8	84.0 / 86.3 / **87.7**

effectiveness of combining explicit 3D spatial grounding with temporally coherent action generation.

C. Ablation Study

We conduct comprehensive ablation studies to validate the effectiveness of each component in Lift3D-VLA.

1) 3D Representation Analysis: As shown in Figure 3(a), we systematically ablate the 2D model-lifting strategy and GC-MAE to evaluate their impact on the 3D representation of Lift3D-VLA in the single-task MetaWorld setting (using MLP action head). Without any 3D modeling, the RGB *baseline* achieves only 68.2%, highlighting the necessity of geometric reasoning. Applying our enhanced 2D model-lifting strategy (*2ML*), which leverages camera parameters to define a virtual front view, improves performance to 81.2%, indicating that better geometric alignment between 3D points and 2D positional embeddings provides more effective spatial grounding. Building on this explicit 3D representation, incorporating the implicit *RGB+Depth* reconstruction objective from our previous Lift3D further improves performance to 83.8%. We then replace the reconstruction objective with point cloud self-supervised learning. Using only the static branch for masked point reconstruction (*2ML+Static*) achieves 85.8%, while using only the dynamic branch for future geometric prediction (*2ML+Dynamic*) yields 86.1%. The full GC-MAE model (Ours) achieves 88.6%, demonstrating the complementary roles of the two branches in enabling robust 3D understanding of physical dynamics.

2) GC-MAE Hyperparameter Analysis: We examine three critical hyperparameters in GC-MAE pretraining: the mask ratio, decoder depth, and the scale of the pretraining data. **Mask ratio.** As shown in Figure 3 (b), a mask ratio of 0.6 yields the best performance. This is because our framework jointly performs masked point cloud reconstruction and future prediction on visible regions, requiring a moderate mask ratio to balance the two objectives. **Decoder Depth.** We vary the depth of the MAE decoder by setting it to 1, 2, 4, and 16 layers, achieving 84.3%, 86.1%, 86.6%, and 87.2%, respectively. Both 4-layer and 16-layer decoders yield strong performance, indicating that a relatively lightweight decoder is sufficient for GC-MAE self-supervised learning. **Pretraining data.** We vary the size of the pretraining corpus to analyze its impact on downstream manipulation performance. Using 20K and 140K samples for GC-MAE pretraining yields 85.6% and 88.6% success rates, respectively, demonstrating the scalability of our method with increased data.

3) Layer-wise Temporal Action Modeling Analysis: We further investigate the design of the layer-wise temporal action modeling strategy on the MetaWorld multi-task setting, providing insights into the formulation of action chunk generation. As shown in Table IV, *Layers* denotes the number of layers used for action prediction, while *Stride* indicates the interval between selected layers. For example, *Layers*=1 and *Stride*=1 correspond to predicting all action chunks from the final layer, whereas *Layers*=4 and *Stride*=4 (for a 32-layer LLM) use layers 20, 24, 28, and 32, with each layer predicting a subset of consecutive action steps. For clarity, all experiments use a fixed action chunk size of four. Using a single layer yields a baseline success rate of 82.5%. Incorporating additional intermediate LLM layers (e.g., two or four layers) for action chunk prediction consistently improves performance, indicating that leveraging intermediate representations enhances robustness. Moreover, given the same number of layers, larger stride intervals between LLM output layers lead to further performance gains, suggesting the benefit of capturing more temporally diverse features. These results validate our layer-wise action modeling design, where parallel predictions from intermediate LLM layers enable richer temporal modeling of

TABLE V: **Comparison across real-world manipulation tasks.** We report success rates (S.R.) for standard single-arm and dual-arm tasks (Franka). Mean S.R. denotes the average success rate.

Models	Wipe whiteboard	Place dish on rack	Place egg on bread	Pick and place banana	Pour water into cup	Stack cola cans	Scoop popcorn	Open pot pick corn	Mean S.R.	Place egg on bread Step1	Step2	Step3
SpatialVLA [20]	60	33	20	40	87	33	27	40	43	20	7	0
$\pi_{0.5}$ [4]	60	60	47	**87**	87	**66**	**53**	60	65	47	20	7
CoT-VLA [53]	53	**66**	33	53	47	33	33	53	46	33	13	7
Lift3D-VLA	**66**	**66**	**66**	**87**	**93**	**66**	47	**73**	**71**	**66**	**33**	**20**

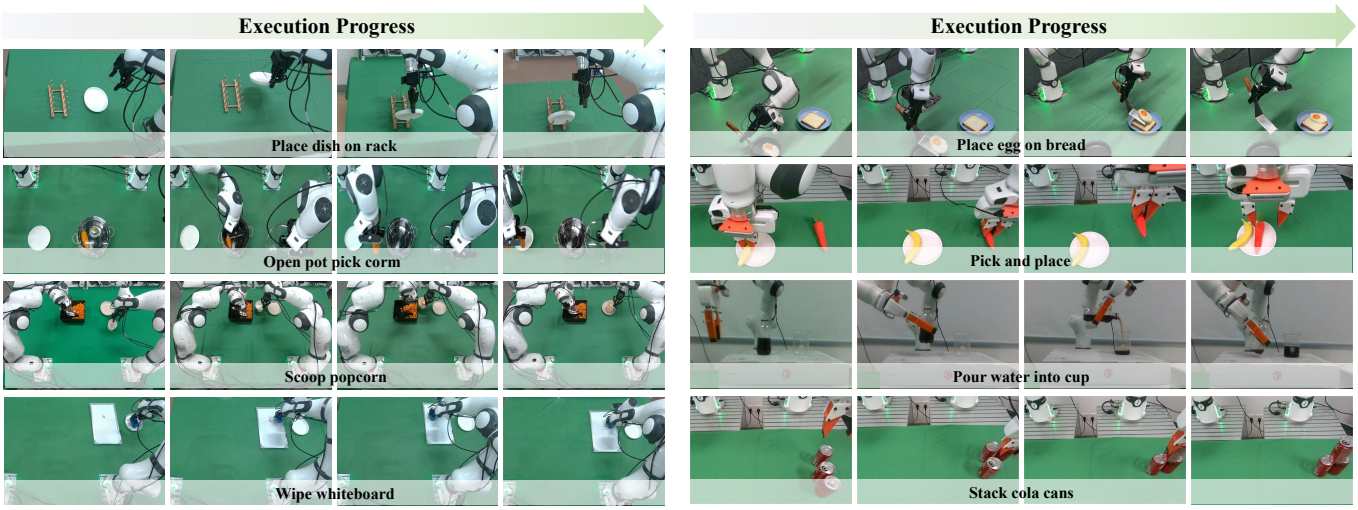

Fig. 5: **Visualization.** We illustrate the execution progress of all real-world task execution.

actions and improve downstream manipulation performance.

D. Real-World Experiment

To evaluate the practical applicability of Lift3D-VLA, we instantiate our framework on real-world robot platforms using both single-arm and dual-arm Franka Research 3 setups.

1) Data Collection: As shown in Figure 4, we illustrate all task assets and real-world experimental setups. In the single-arm configuration, we curate a dataset of 200 demonstrations per task across six manipulation skills: (1) *whiteboard cleaning with an eraser*, (2) *precision placement of a dish onto a rack*, (3) *placing an egg on bread using a spatula*, (4) *picking up a banana and placing it on a white plate*, (5) *pouring water into a cup*, and (6) *stacking three cola bottles*. Meanwhile, we evaluate a long-horizon *placing egg* task, where the robot repeats the task three times under continuously changing object positions. The system is equipped with two Intel RealSense cameras, including one third-person view and one wrist view. To evaluate dual-arm coordination, we design two collaborative tasks: (1) *scooping popcorn into a bowl*, where one arm stabilizes the bowl while the other performs the scooping motion, and (2) *opening a pot lid and picking corn*, which requires sequential coordination between both arms. In this setting, we use three Intel RealSense cameras, including one third-person view and two wrist views. Point clouds are obtained by projecting third-person depth maps using camera parameters. All trajectories are recorded at 30 FPS via human teleoperation using the 3D space mouse.

2) Training and Evaluation Protocol: We train Lift3D-VLA following the protocol described in Section V-B3, with the primary difference being the use of multi-view visual inputs: two camera perspectives for single-arm tasks and three perspectives for dual-arm tasks. We compare our method against three representative baselines: (i) $\pi_{0.5}$ [4], a state-of-the-art 2D VLA model; (ii) SpatialVLA [20], a 3D-aware VLA framework; and (iii) CoT-VLA [53], which incorporates explicit chain-of-thought reasoning for action generation. For fair comparison, all methods are initialized from official pretrained checkpoints, trained with full fine-tuning, and evaluated over 15 rollouts per task under varying object positions.

3) Quantitative and Qualitative Results: Table V summarizes the real-world manipulation performance of Lift3D-VLA and competitive baselines. Our framework achieves an average success rate of 71% across all tasks, outperforming π_0 (65%), SpatialVLA (43%), and CoT-VLA (46%). In the *placing egg on bread* task, which involves tool use and complex contact dynamics, Lift3D-VLA achieves a success rate of 66%, largely attributed to our recover–forecast 3D representation, which enables the robot to reason about geometric interactions between objects. We further evaluate Lift3D-VLA on a long-horizon manipulation task requiring one, two, and three consecutive successful executions within a single rollout. Lift3D-VLA maintains markedly higher success rates across all stages (66% $\rightarrow$ 33% $\rightarrow$ 20%) compared to $\pi_{0.5}$ (47% $\rightarrow$ 20% $\rightarrow$ 7%), with the performance gap widening as the horizon increases. This demonstrates that our proposed layer-wise temporal action modeling enables more coherent action generation in long-horizon tasks. For dual-arm tasks,

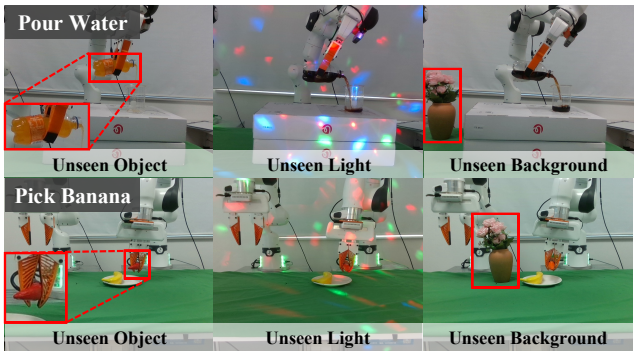

Fig. 6: **Generalization.** We evaluate the model under three OOD conditions: *Unseen Object*, *Unseen Light*, and *Unseen Background*. Red boxes highlight the changed conditions.

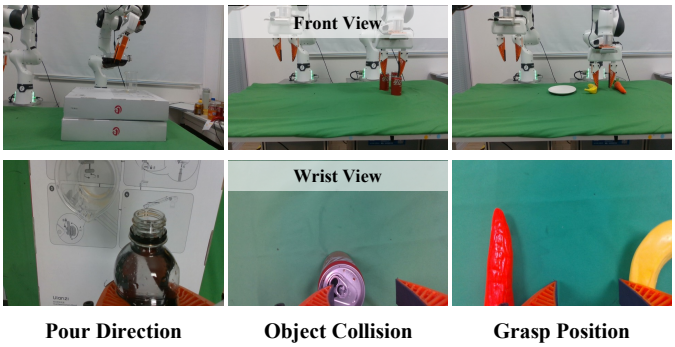

Fig. 7: **Failure case visualization in real-world tasks.** The top row shows the Front View, and the bottom row shows the corresponding Wrist View.

TABLE VI: **Generalization experiments.** We evaluate the robustness of **Lift3D-VLA** against $\pi_{0.5}$ under three unseen scenarios. The percentages in brackets denote the performance degradation relative to the *Original* setup.

Scenario	Pick Banana		Pour Water	
	$\pi_{0.5}$	Lift3D-VLA	$\pi_{0.5}$	Lift3D-VLA
Original	87	87	87	93
Unseen Object	80 (-8%)	87 (-0%)	80 (-8%)	87 (-6%)
Unseen Lighting	73 (-16%)	80 (-8%)	67 (-23%)	80 (-14%)
Unseen Background	47 (-46%)	80 (-8%)	60 (-31%)	87 (-6%)
Average Drop	64(-26%)	82(-6%)	69(-21%)	85(-9%)

Lift3D-VLA achieves the highest average success rate of 60%. For example, in the *opening a pot lid and picking corn* task, our method achieves a 73% success rate compared to 60% for $\pi_{0.5}$, demonstrating that improved 3D structural understanding enables more effective temporally coordinated dual-arm actions. As shown in Figure 5, we visualize the execution progress across several tasks in both single-arm and dual-arm configurations. The results demonstrate that our method can execute tasks accurately and smoothly across a wide range of scenarios, including basic manipulation, contact-rich interactions, tool use, and structured stacking. Additional execution videos are provided on our project website.

E. Generalization Experiment

To evaluate the robustness of Lift3D-VLA, we conduct experiments under three out-of-domain (OOD) settings: (1) *Unseen Object*, where target objects (e.g., the bottle or banana) are replaced with different instances; (2) *Unseen Lighting*, where multi-colored lighting perturbations are introduced; and (3) *Unseen Background*, where novel distractor objects (e.g., a flower vase) are added to the scene. Example scenarios are shown in Figure 6. We compare against the strongest baseline in real-world experiments, $\pi_{0.5}$. As shown in Table VI, $\pi_{0.5}$ exhibits performance degradation across all OOD settings, particularly under *Unseen Background*, where performance drops by 46% on *Pick Banana* (87→47). This sharp decline suggests a strong reliance on 2D appearance and sensitivity to background clutter. In contrast, Lift3D-VLA maintains stable performance across all scenarios, with performance drops consistently bounded within 6%–8%. These results demonstrate that incorporating robust 3D representations significantly improves the model's understanding of object relationships and enhances generalization. Under *Unseen Object*, Lift3D-VLA shows minimal degradation (0% on *Pick Banana*), indicating strong invariance to object appearance. Under *Unseen Lighting*, the performance gap further widens, suggesting that GC-MAE pretraining encourages the model to focus on geometric structure rather than being affected by pixel-level perturbations. Overall, the consistently smaller performance drops across tasks validate that combining explicit 3D reasoning with temporal action modeling leads to stronger generalization in dynamic manipulation scenarios.

F. Failure Case Analysis

Despite the strong performance of Lift3D-VLA, we observe several failure cases during real-world execution, as illustrated in Figure 7.

1) *Pour Direction Misalignment*: In the water-pouring task, the model occasionally fails to precisely align the bottle opening with the beaker. This is primarily due to the narrow tolerance of the target container and the limitations of depth sensing on transparent objects, which result in incomplete point cloud observations.

2) *Object Collision*: During the descent phase in the cola-stacking task, the gripper may prematurely collide with the side of the can. This suggests that, despite strong geometric priors from our method, single-view point clouds remain insufficiently robust under certain viewpoints.

3) *Inaccurate Grasp Position*: In the fruit-picking task, the gripper sometimes moves toward the midpoint between two adjacent fruits rather than accurately localizing a single target. This indicates insufficient instance-level discrimination when multiple similar objects are closely positioned.

Overall, these failure cases highlight that, while our method provides robust 3D representations and temporally coherent action modeling, depth perception remains limited in scenarios involving transparent or highly reflective objects. Furthermore, although single-view point clouds are efficient and practical, they lack complete 3D information. In future work, we plan

to explore depth completion techniques and multi-view fusion to further improve perception robustness.

VI. Conclusion and Limitation

In this paper, we introduced Lift3D-VLA, a unified framework that integrates explicit 3D point-cloud reasoning with temporally structured action generation for robotic manipulation. Building upon the 2D model lifting paradigm, we proposed a geometry-centric self-supervised learning scheme, GC-MAE, which jointly reconstructs present 3D structure and forecasts its future evolution, enabling the vision encoder to capture both static geometry and physical dynamics. Furthermore, we developed a layer-wise temporal action modeling strategy that leverages intermediate and deep LLM representations to produce temporally coherent action sequences. Extensive experiments across simulation and real-world benchmarks demonstrate that Lift3D-VLA significantly outperforms prior VLA methods, achieving strong performance in multi-task, long-horizon, and dual-arm manipulation scenarios. Our method also exhibits robust generalization under diverse out-of-domain conditions, including unseen objects, lighting variations, and background clutter. These results highlight the effectiveness of combining explicit 3D reasoning with temporally aware action generation in dynamic environments. Despite these advances, several challenges remain. In particular, perception under transparent or reflective objects remains limited due to the inherent constraints of depth sensing. Future work will explore integrating depth completion and multi-view fusion to improve perceptual robustness, as well as developing closed-loop control mechanisms for finer-grained interaction in contact-rich settings.

References

[1] Y. Jia, J. Liu, S. Chen, and et al., "Lift3D policy: Lifting 2D foundation models for robust 3D robotic manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2025, pp. 17 347–17 358. 1, 2, 3, 4, 5, 7

[2] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Proc. Conf. Robot Learn.*, 2023. 1

[3] J. Liu, H. Chen, P. An, Z. Liu, R. Zhang, C. Gu, X. Li, Z. Guo, S. Chen, M. Liu *et al.*, "Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model," *Proc. Int. Conf. Learn. Represent.*, 2025. 1, 3, 5, 6, 7

[4] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn *et al.*, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," 2025. [Online]. Available: https://arxiv.org/abs/2504.16054 1, 2, 3, 5, 8, 10

[5] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Proc. Conf. Robot Learn.* PMLR, 2022, pp. 894–906. 1, 5

[6] H. Zhu, Y. Wang, D. Huang, W. Ye, W. Ouyang, and T. He, "Point cloud matters: Rethinking the impact of different observation spaces on robot learning," *Proc. Adv. Neural Inform. Process. Syst.*, vol. 37, pp. 77 799–77 830, 2024. 1

[7] B. Eisner*, H. Zhang*, and D. Held, "Flowbot3d: Learning 3d articulation flow to manipulate articulated objects," in *Proc. Robot.: Sci. Syst.*, 2022. 1

[8] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Trans. Robot.*, 2023. 1

[9] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Proc. Conf. Robot Learn.*, 2023, pp. 785–799. 1

[10] S. Chen, R. Garcia, C. Schmid, and I. Laptev, "Polarnet: 3d point clouds for language-guided robotic manipulation," *Proc. Conf. Robot Learn.*, 2023. 1

[11] M. Liu, X. Li, Z. Ling, Y. Li, and H. Su, "Frame mining: a free lunch for learning robotic manipulation from 3d point clouds," *Proc. Conf. Robot Learn.*, 2022. 1

[12] T. Zhang, Y. Hu, J. You, and Y. Gao, "Leveraging locality to boost sample efficiency in robotic manipulation," *Proc. Conf. Robot Learn.*, 2024. 1

[13] C. Wang, H. Fang, H.-S. Fang, and C. Lu, "Rise: 3d perception makes real-world robot imitation simple and effective," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024. 1

[14] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," in *Proc. Robot.: Sci. Syst.*, 2024. 1, 7, 8

[15] S. James and P. Abbeel, "Coarse-to-fine q-attention with learned path ranking," *arXiv preprint arXiv:2204.01571*, 2022. 1

[16] C. Li, J. Wen, Y. Peng, Y. Peng, and Y. Zhu, "Pointvla: Injecting the 3d world into vision-language-action models," *IEEE Robot. Automat. Lett.*, vol. 11, no. 3, pp. 2506–2513, 2026. 1, 3

[17] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki, "Act3d: 3d feature field transformers for multi-task robotic manipulation," in *Proc. Conf. Robot Learn.*, 2023. 1, 5

[18] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," *arXiv preprint arXiv:2402.10885*, 2024. 1

[19] Z. Xian, N. Gkanatsios, T. Gervet, T.-W. Ke, and K. Fragkiadaki, "Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation," in *Proc. Conf. Robot Learn.*, 2023. 1

[20] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang *et al.*, "Spatialvla: Exploring spatial representations for visual-language-action model," *arXiv preprint arXiv:2501.15830*, 2025. 1, 3, 8, 10

[21] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox, "RVT: Robotic view transformer for 3D object manipulation," in *Proc. Conf. Robot Learn.*, 2023, pp. 694–710. 1, 5, 7

[22] W. Wang, Y. Lei, S. Jin, G. D. Hager, and L. Zhang, "Vihe: Virtual in-hand eye transformer for 3d robotic manipulation," *arXiv preprint arXiv:2403.11461*, 2024. 1, 5

[23] J. Zhang, C. Bai, H. He, Z. Wang, B. Zhao, X. Li, and X. Li, "Sam-e: Leveraging visual foundation model with sequence imitation for embodied manipulation," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2024, pp. 58 579–58 598. 1

[24] X. Li, L. Heng, J. Liu, Y. Shen, C. Gu, Z. Liu, H. Chen, N. Han, R. Zhang, H. Tang *et al.*, "3ds-vla: A 3d spatial-aware vision language action model for robust multi-task manipulation," in *Proc. Conf. Robot Learn.*, 2025. 1, 3, 8

[25] S. Chen, J. Liu, S. Qian, H. Jiang, Z. Liu, C. Gu, X. Li, C. Hou, P. Wang, Z. Wang *et al.*, "Ac-dit: Adaptive coordination diffusion transformer for mobile manipulation," in *Proc. Adv. Neural Inform. Process. Syst.* 1

[26] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024. 1

[27] C.-P. Huang, Y.-H. Wu, M.-H. Chen, Y.-C. F. Wang, and F.-E. Yang, "Thinkact: Vision-language-action reasoning via reinforced visual latent planning," *arXiv preprint arXiv:2507.16815*, 2025. 1

[28] J. Wen, M. Zhu, Y. Zhu, Z. Tang, J. Li, Z. Zhou, C. Li, X. Liu, Y. Peng, C. Shen *et al.*, "Diffusion-vla: Scaling robot foundation models via unified diffusion and autoregression," *arXiv preprint arXiv:2412.03293*, 2024. 2, 3, 5

[29] H. Bi, H. Tan, S. Xie, Z. Wang, S. Huang, H. Liu, R. Zhao, Y. Feng, C. Xiang, Y. Rong *et al.*, "Motus: A unified latent action world model," *arXiv preprint arXiv:2512.13030*, 2025. 2, 3

[30] Open X-Embodiment Collaboration, A. Padalkar, A. Pooley *et al.*, "Open X-Embodiment: Robotic learning datasets and RT-X models," https://arxiv.org/abs/2310.08864, 2023. 2

[31] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *Proc. Robot.: Sci. Syst.*, 2024. 2, 5, 7

[32] K. Wu, C. Hou, J. Liu, Z. Che, X. Ju *et al.*, "Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation," in *Proc. Robot.: Sci. Syst.* Robotics: Science and Systems Foundation, 2025. 2, 7

[33] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta

reinforcement learning," in *Proc. Conf. Robot Learn.* PMLR, 2020, pp. 1094–1100. 2, 7

[34] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 3019–3026, 2020. 2, 7

[35] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 574–591. 3

[36] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Proc. Adv. Neural Inform. Process. Syst.*, vol. 33, pp. 9912–9924, 2020. 3

[37] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 16 000–16 009. 3

[38] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022. 3, 7, 8

[39] Z. Li, L. Ren, J. Yang, Y. Zhao, X. Wu, Z. Xu, X. Bai, and H. Zhao, "Vip: Vision instructed pre-training for robotic manipulation," 2025. [Online]. Available: https://arxiv.org/abs/2410.07169 3

[40] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, "Masked visual pre-training for motor control," *arXiv preprint arXiv:2203.06173*, 2022. 3

[41] A. Majumdar, K. Yadav, S. Arnaud, J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil *et al.*, "Where are we in the search for an artificial visual cortex for embodied intelligence?" *Proc. Adv. Neural Inform. Process. Syst.*, vol. 36, pp. 655–677, 2023. 3, 7, 8

[42] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang, "Language-driven representation learning for robotics," 2023. [Online]. Available: https://arxiv.org/abs/2302.12766 3

[43] Y. Seo, J. Kim, S. James, K. Lee, J. Shin, and P. Abbeel, "Multi-view masked world models for visual robotic manipulation," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2023, pp. 30 613–30 632. 3

[44] S. Qian, K. Mo, V. Blukis, D. F. Fouhey, D. Fox, and A. Goyal, "3d-mvp: 3d multiview pretraining for manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2025, pp. 22 530–22 539. 3

[45] H. Zhu, H. Yang, Y. Wang, J. Yang, L. Wang, and T. He, "Spa: 3d spatial-awareness enables effective embodied representation," *arXiv preprint arXiv:2410.08208*, 2024. 3, 7, 8

[46] W. Cui, C. Zhao, Y. Chen, and et al., "CL3R: 3d reconstruction and contrastive learning for enhanced robotic manipulation representations," *arXiv preprint arXiv:2507.08262*, 2025. 3

[47] J. Yang, P. Wei, and Y. Chen, "Hyperbolic multiview pretraining for robotic manipulation," *arXiv preprint arXiv:2603.04848*, 2026. 3

[48] S. Yang, L. Xu, H. Li, J. Mu, J. Zeng, D. Lin, and J. Pang, "Robo3r: Enhancing robotic manipulation with accurate feed-forward 3d reconstruction," *arXiv preprint arXiv:2602.10101*, 2026. 3

[49] Z. Zhang, Z. Xu, J. N. Lakamsani, and Y. She, "Canonical policy: Learning canonical 3d representation for se (3)-equivariant policy," *arXiv preprint arXiv:2505.18474*, 2025. 3

[50] S. Karamcheti, S. Nair, A. Balakrishna, P. Liang, T. Kollar, and D. Sadigh, "Prismatic vlms: Investigating the design space of visually-conditioned language models," in *Proc. Int. Conf. Mach. Learn.*, 2024. 3, 4

[51] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024. 3, 8

[52] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn *et al.*, "pi0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024. 3

[53] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn *et al.*, "Cot-vla: Visual chain-of-thought reasoning for vision-language-action models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2025, pp. 1702–1713. 3, 10

[54] J. Zhang, Y. Guo, Y. Hu, X. Chen, X. Zhu, and J. Chen, "Up-vla: A unified understanding and prediction model for embodied agent," 2025. 3

[55] M. J. Kim, Y. Gao, T.-Y. Lin, Y.-C. Lin, Y. Ge, G. Lam, P. Liang, S. Song, M.-Y. Liu, C. Finn *et al.*, "Cosmos policy: Fine-tuning video models for visuomotor control and planning," *arXiv preprint arXiv:2601.16163*, 2026. 3

[56] C. Gu, J. Liu, H. Chen, R. Huang, Q. Wuwu, Z. Liu, X. Li, Y. Li, R. Zhang, P. Jia *et al.*, "Manualvla: A unified vla model for chain-of-thought manual generation and robotic manipulation," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2025. 3

[57] Z. Liu, J. Liu, H. Chen, and et al., "LaST$_0$: Latent spatio-temporal chain-of-thought for robotic vision-language-action model," *arXiv preprint arXiv:2601.05248*, 2026. 3, 7

[58] Z. Liu, J. Liu, J. Xu, N. Han, C. Gu, H. Chen, K. Zhou, R. Zhang, K. C. Hsieh, K. Wu *et al.*, "Mla: A multisensory language-action model for multimodal understanding and forecasting in robotic manipulation," *arXiv preprint arXiv:2509.26642*, 2025. 3

[59] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, "Rdt-1b: a diffusion foundation model for bimanual manipulation," *arXiv preprint arXiv:2410.07864*, 2024. 3

[60] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang *et al.*, "Gr00t n1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025. 3

[61] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proc. Int. Conf. Comput. Vis.*, 2023. 4

[62] M. Oquab, T. Darcet, T. Moutakanni, and et al., "DINOv2: Learning robust visual features without supervision," *Trans. Mach. Learn. Res.*, 2024. 4

[63] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 652–660. 4, 7, 8

[64] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023. 5

[65] A. O'Neill, A. Rehman, A. Maddukuri, and et al., "Open x-embodiment: Robotic learning datasets and rt-x models," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 6892–6903. 5, 7

[66] C. Hou, K. Wu, J. Liu, Z. Che, D. Wu, F. Liao, G. Li, J. He, Q. Feng, Z. Jin *et al.*, "Robomind 2.0: A multimodal, bimanual mobile manipulation dataset for generalizable embodied intelligence," *arXiv preprint arXiv:2512.24653*, 2025. 5

[67] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2025, pp. 5294–5306. 5

[68] Y. Pang, E. H. F. Tay, L. Yuan, and Z. Chen, "Masked autoencoders for 3d point cloud self-supervised learning," *World Scientific Annual Review of Artificial Intelligence*, vol. 1, p. 2440001, 2023. 5

[69] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models," 2022. 6, 7

[70] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in neural information processing systems*, vol. 34, pp. 15 084–15 097, 2021. 6

[71] M. Wen, R. Lin, H. Wang, Y. Yang, Y. Wen, L. Mai, J. Wang, H. Zhang, and W. Zhang, "Large sequence models for sequential decision-making: a survey," *Frontiers of Computer Science*, vol. 17, no. 6, p. 176349, 2023. 6

[72] Y. Yue, Y. Wang, B. Kang, Y. Han, S. Wang, S. Song, J. Feng, and G. Huang, "Deer-vla: Dynamic inference of multimodal large language models for efficient robot execution," *arXiv preprint arXiv:2411.02359*, 2024. 6

[73] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang *et al.*, "Gr00t n1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025. 6

[74] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Proc. Adv. Neural Inform. Process. Syst.*, vol. 33, pp. 6840–6851, 2020. 6

[75] S. E. Jiaming Song, Chenlin Meng, "Denoising diffusion implicit models," in *Proc. Int. Conf. Learn. Represent.*, 2021. 6

[76] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu *et al.*, "Rdt-1b: a diffusion foundation model for bimanual manipulation," in *Proc. Int. Conf. Learn. Represent.* 6

[77] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2021, pp. 8748–8763. 7, 8

[78] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Proc. Adv. Neural Inform. Process. Syst.*, vol. 30, 2017. 7, 8

[79] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, and B. Ghanem, "Pointnext: Revisiting pointnet++ with improved training and scaling strategies," *Proc. Adv. Neural Inform. Process. Syst.*, vol. 35, pp. 23 192–23 204, 2022. 7, 8

[80] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012. 7
[81] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Proc. Conf. Robot Learn.*, 2022. 7
[82] H. Chen, J. Liu, C. Gu, Z. Liu, R. Zhang, X. Li, X. He, Y. Guo, C.-W. Fu, S. Zhang *et al.*, "Fast-in-slow: A dual-system foundation model unifying fast manipulation within slow reasoning," *arXiv preprint arXiv:2506.01953*, 2025. 8

Jiaming Liu is a Ph.D. student supervised by Assistant Professor Zhang Shanghang. His research focuses on end-to-end embodied models for open-world reasoning and robotic manipulation. He has published 22 papers as first or co-first author at top-tier conferences, including NeurIPS, CVPR, ICLR, and ICRA, with over 3,000 citations on Google Scholar. He is a recipient of the 2025 ByteDance Scholarship and the National Scholarship. He has also been awarded funding from the National Natural Science Foundation of China (NSFC) Youth Program.

Qingpo Wuwu is a Ph.D. student at Peking University, supervised by Prof. Shanghang Zhang and Professor Huizhu Jia. He received his B.Sc. in Physics from Imperial College London. His research spans from 3D reconstruction and perception to embodied AI, leveraging geometric understanding to enhance world models and vision-language-action models for robotic manipulation. He has published papers at top-tier venues including ICCV, ICML, CVPR, and Nature Computational Science. He was awarded the NSFC Research Fund for Ph.D. Students in 2026.

Nuowei Han is a graduate student at Beijing University of Posts and Telecommunications. He is currently conducting a research internship in the School of Computer Science at Peking University, supervised by Assistant Professor Zhang Shanghang. His research interests focus on embodied artificial intelligence and robotic manipulation.

Hao Chen is a second-year Ph.D. student at The Chinese University of Hong Kong, jointly supervised by Prof. Pheng Ann Heng and Prof. Chi Wing Fu. His primary research interests include generalizable robotic manipulation methods and multimodal large language models. He has published seven papers as first or co-first author at top-tier international conferences such as NeurIPS, CVPR, ICCV.

Zhuoyang Liu is an undergraduate student in Yuanpei College, Peking University. He is conducting a research internship at the HMI Lab at Peking University, supervised by Assistant Professor Zhang Shanghang. His research interests focus on multimodal perception and reasoning towards general embodied large models. He has published 3 papers as first or co-first author and 6 papers in total at top-tier conferences, including NeurIPS, ICLR, and ICRA.

Fan Fei received a B.S. degree summa cum laude from Turing Class, Peking University in 2022. He is currently working towards a Ph.D degree with the School of Computer Science, Peking University, advised by Prof. Boxin Shi. His research interests include 3D reconstruction and generation, physically-based rendering, and robotic perception and manipulation.

Yueru Jia is a Master's student at Peking University's School of Computer Science, advised by Prof. Shanghang Zhang. She earned her Bachelor's degree in AI as part of the inaugural "Tong Class" at Yuanpei College and was a recipient of the National Scholarship. Her research resides at the intersection of Robotics and Generative AI, with a focus on developing scalable learning algorithms to achieve stable and reliable robotic manipulation.

Chenyang Gu is a senior student at Peking University, supervised by Assistant Professor Zhang Shanghang. His research focuses on general policy for robotic manipulation. He has published 3 papers as first or co-first author at top-tier conferences, including NeurIPS 2026, CVPR 2026 and CVPR 2025, with over 450 citations on Google Scholar.

Yandong Guo is the Founder and CEO of AI^{2}Robotics. He received his Ph.D. from Purdue University in 2013 under the supervision of Jan Allebach and Charles Bouman, both members of the U.S. National Academy of Engineering. In 2025, he was appointed Adjunct Professor at The Hong Kong University of Science and Technology (Guangzhou). Dr. Guo is an expert in AI and intelligent hardware, and has previously served as Chief Scientist at OPPO, Researcher at Microsoft Seattle, and Chief Scientist at XPeng Motors. The intelligent systems he led have been deployed in smart vehicles, consumer electronic devices, and robots. In 2021, he received the First Prize of Technological Invention from the China Society of Image and Graphics. He has published over 100 papers with nearly 10,000 citations and holds hundreds of patents worldwide.

Boxin Shi received a B.E. degree from Beijing University of Posts and Telecommunications in 2007, an M.E. degree from Peking University in 2010, and a Ph.D. degree from the University of Tokyo in 2013. He is currently a Boya Young Fellow Associate Professor (with tenure) and Research Professor at Peking University, where he leads the Camera Intelligence Lab. Before joining PKU in 2017, he was a researcher at MIT Media Lab, Singapore University of Technology and Design, Nanyang Technological University, and the National Institute of Advanced Industrial Science and Technology. He received Best Paper and Runner-Up awards at CVPR 2024 and ICCP 2015. He is an associate editor of TPAMI/IJCV and an area chair of CVPR/ICCV/ECCV.

Shanghang Zhang received her Ph.D. from Carnegie Mellon University in 2018 and conducted postdoctoral research at the University of California, Berkeley. She is an Assistant Professor at the School of Computer Science, Peking University. She has published over 100 papers in top artificial intelligence journals and conferences. Her works have been cited more than 20,000 times on Google Scholar. She was honored with the Best Paper Award at the AAAI'2021. In 2018, she was recognized as an "EECS Rising Star" in the United States. She has organized workshops at top international conferences such as NeurIPS and ICML and served as a senior program committee member for AAAI 2022, 2023, and 2024.