3DVLA: ENHANCING VISION-LANGUAGE-ACTION MODELS VIA 3D SPATIAL AND INSTANCE UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action models have achieved remarkable progress in robotic manipulation, yet they suffer from a critical limitation: a lack of 3D scene understanding. This deficiency manifests as three intertwined challenges: weak extraction of 3D spatial positions without enforcing multi-view consistency, inadequate 3D instance understanding, and fragile reasoning under occlusion. Although mature 3D perception methods exist, their direct integration into VLA pipelines is hindered by architectural incompatibility and by heavy reliance on costly instance-level annotations. To address the above challenges, we propose 3DVLA, a plug-and-play framework that injects robust 3D reasoning into pretrained VLAs without requiring extra manual labels or discarding VLM priors. Specifically, 3DVLA tackles the three challenges through: (1) pervasive 3D feature encoding with explicit multi-view consistency constraints across all modalities and a Spatially-Conditioned Geometry Aggregation method, (2) an instance estimation module with high-level instance tokens for 3D instance awareness, and (3) a masked self-supervised 3D encoding branch that retains its predictor for visual token completion to handle occlusions. We integrate 3DVLA with multiple VLA baselines and evaluate on LIBERO-Plus and RoboTwin 2.0. Results show consistent and significant gains in manipulation performance, validating both the effectiveness and plug-and-play compatibility of our approach.

1 INTRODUCTION

By leveraging large-scale vision-language pre-training, Vision-Language-Action Models (VLAs) Brohan et al. (2022); Zitkovich et al. (2023); Kim et al. (2025b); Team et al. (2024) have demonstrated impressive manipulation capabilities across diverse tasks and embodiments, underscoring their immense practical value for general-purpose robotics. Despite these advances, a fundamental limitation persists: mainstream VLA models operate predominantly on 2D visual inputs and lack genuine 3D scene understanding. This deficiency impedes their deployment in real-world environments where spatial reasoning is crucial and can be characterized by three tightly coupled challenges.

The first challenge is weak 3D spatial position extraction and the absence of multi-view consistency constraints. Existing VLA architectures typically encode single-view observations or loosely fuse multiple views without explicitly reconstructing 3D coordinates or enforcing consistencies across viewpoints. As a result, the policy cannot ground its decisions in a metric 3D reference frame, making it incapable of reasoning about precise 6-DoF poses.

The second challenge is insufficient 3D instance understanding. Standard VLAs reason about objects as flat image regions, without access to their volumetric extent, 3D boundaries, or fine-grained geometry. Consequently, they struggle with tasks that demand precise spatial interaction—such as grasping a specific part of an object or placing one item relative to another's 3D structure—because the model cannot reliably distinguish which pixels belong to which object instance in 3D space.

For the first and second challenges, prior 3D scene-understanding approaches Rukhovich et al. (2022); Xia et al. (2026; 2024; 2025) offer strong spatial reasoning, but they cannot be directly applied to VLA-based manipulation, which depends on 2D VLM pre-training. Integrating specialized 3D encoders forces a redesign that discards this pre-training, sacrificing the generalization and language

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

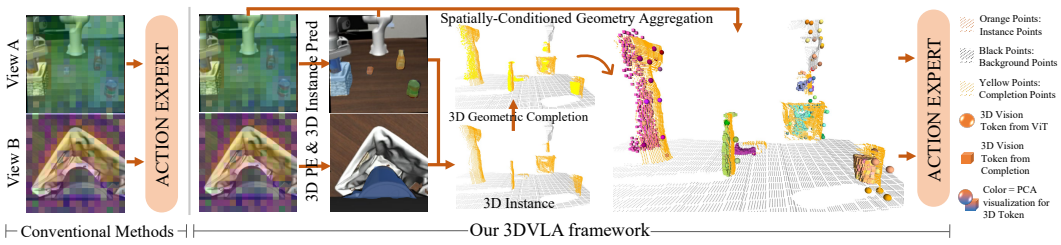

Figure 1: **Overview of 3DVLA.** Compared to conventional methods, 3DVLA extracts view-consistent 3D instances (orange) and synthesizes occluded geometries (yellow). These features are aggregated into spatially-grounded vision tokens to enable robust manipulation under complex occlusions.

grounding that make VLAs valuable. Besides, 3D perception models assume the availability of dense instance-level or semantic annotations, which are costly to obtain at scale in robotic manipulation settings. Motivated by this, we propose a 3D VLA compatible with VLM architectures that does not rely on additional manual annotations.

The third challenge is fragile reasoning under occlusion, particularly when viewpoint variations hide task-relevant object parts. A telling example is observed on the LIBERO-Plus benchmark Fei et al. (2025): slightly shifting the camera viewpoint so that the target object becomes partially occluded causes the state-of-the-art VLAs' success rates to drop dramatically. This brittleness arises because purely 2D models lack the capacity to infer the unseen geometry and maintain a coherent spatial memory, leading to action generation based on incomplete and misleading visual cues. This fragility calls for a mechanism that can explicitly infer unseen geometry from partial views. Interestingly, masked self-supervised learning offers an overlooked building block. Masked self-supervised methods Assran et al. (2023) typically discard the predictor used to predict masked tokens after self-supervised training. However, we realize that after adopting a unified 3D positional encoding, this predictor can be repurposed to complete tokens at occluded locations.

In this paper, we propose 3DVLA, a plug-and-play method that enriches VLA models with robust 3D understanding by leveraging priors from off-the-shelf foundation perception models. Crucially, 3DVLA is designed to be fully compatible with VLM pre-training and requires no extra manual annotations. Concretely, our contributions include: **(1)** To address the first challenge, we impose *Multi-View Spatial Fusion* and *Spatially-Conditioned Geometry Aggregation*, enforcing explicit multi-view consistency constraints throughout the entire model. **(2)** To address the second challenge of weak 3D instance understanding, 3DVLA incorporates an *Object-Centric 3D Instance Module* and introduces high-level 3D instance tokens that explicitly encode object identity and geometry. **(3)** For the third challenge of occlusion reasoning, we complement the architecture with a 3D encoding branch trained via masked self-supervision. Unlike conventional masked auto-encoding paradigms that discard the lightweight predictor after pre-training, we retain it and repurpose it for visual token completion, enabling the model to reconstruct occluded visual features and reason about invisible parts of objects during policy execution. **(4)** We integrate 3DVLA with multiple VLA baselines and conduct experiments on the LIBERO-Plus and RoboTwin 2.0 benchmarks. The results demonstrate that our method consistently improves manipulation performance across different architectures and tasks, validating both its effectiveness and its plug-and-play compatibility.

2 Related Work

Vision-Language-Action Models With the development of large language models and large-scale robotic datasets, Vision-Language-Action (VLA) models have emerged as a crucial paradigm for general-purpose robot policy learning. OpenVLA Kim et al. (2025b) fuses DINOv2 and SigLIP, while NORA Hung et al. (2025) is built upon a lightweight VLM. WorldVLA Cen et al. (2025) unifies action and future-state modeling into an autoregressive world model, and UniVLA Bu et al. (2025) enables cross-embodiment generalization via a task-centric implicit action space. π_0 Black et al. (2024) pioneered flow matching for continuous action generation, while $\pi_{0.5}$ Intelligence et al. (2025) added open-world generalization through heterogeneous co-training and knowledge insulation. RIPT-VLA Tan et al. (2025) extends the pre-training plus supervised fine-tuning pipeline with a reinforcement-learning-based interactive post-training stage. Diffusion Policy (DP) Chi et al. (2025) formulates action generation as a denoising diffusion probabilistic process, while DP3 Ze

et al. (2024) replaces its 2D input with 3D point cloud representations. ACT Zhao et al. (2023) recasts action prediction as a conditional VAE-driven sequence-generation problem, and RDT Liu et al. (2025) employs a physically interpretable, unified action space within a scalable diffusion Transformer. Recent studies incorporate explicit 3D representations into VLAs. SpatialVLA Qu et al. (2025) introduces 3D positional encoding. PointVLA Li et al. (2026) and GeoVLA Sun et al. (2025) combine a pretrained vision-language backbone with a point backbone. Beyond incorporating 3D positional encoding, we focus on associating object instances across views and augmenting their representations under partial observability. To this end, 3DVLA couples multi-view instance modeling with coordinate-conditioned feature completion within a shared 3D representation for action prediction.

Self-Supervised Representation Learning In the construction of visual encoders, self-supervised pre-training is a commonly adopted technique. Early methods Edwards et al. (2019); Lynch et al. (2020) learn latent actions by modeling pixel variations between adjacent frames, but they are prone to degenerating into reliance on local visual correlations. DINOv2 Oquab et al. (2023) employs masked self-distillation for Vision Transformer pre-training. I-JEPA Assran et al. (2023) transitioned from pixel-level reconstruction to latent space modeling. This paradigm has further influenced the field of embodied learning; for example, VLA-JEPA Sun et al. (2026) extends it to action-relevant representation alignment tasks. However, these self-supervised methods typically discard the masked tokens predictor after training. We realize that after adopting a unified 3D positional encoding, this predictor can be repurposed to complete tokens at occluded locations.

3 METHOD

We introduce 3DVLA, a 3D framework that enhances 2D Vision-Language-Action (VLA) models without disrupting their pre-trained representations. As illustrated in Figure 2, 3DVLA consists of four interconnected components: (1) *Multi-View Spatial Fusion* lifts 2D visual tokens into a unified 3D memory bank to enforce multi-view consistency; (2) The *Object-Centric 3D Instance Module* extracts physical entities directly in 3D space; (3) A *Coordinate-Driven 3D Self-Supervised Predictor* synthesizes occluded geometric structures; (4) *Spatially-Conditioned Geometry Aggregation* explicitly embeds these 3D geometries as affordance-aware instance tokens for the downstream action expert.

3.1 MULTI-VIEW SPATIAL FUSION

To construct a unified 3D representation from multi-view feature maps $\mathcal{X}_{fmap} \in \mathbb{R}^{V\times C\times H\times W}$ without computationally expensive voxelization, we flatten the spatial dimensions into an unordered sequence $\mathcal{M}_{flat} \in \mathbb{R}^{N\times C}$ ($N = V \times H \times W$). Using the unprojected 3D coordinate grids $\mathbf{P}_{flat} \in \mathbb{R}^{N\times 3}$, we ground the visual features in absolute 3D space via a coordinate projection:

$$\mathbf{E}_{3D} = \text{LayerNorm}\left(\mathbf{W}_2\,\text{GELU}(\mathbf{W}_1\mathbf{P}_{flat} + \mathbf{b}_1) + \mathbf{b}_2\right), \tag{1}$$

where $\mathbf{W}_{(\cdot)}$ and $\mathbf{b}_{(\cdot)}$ denote learnable weights and biases. The position-aware features ($\mathcal{M}_{flat}+\mathbf{E}_{3D}$) are processed by a multi-layer Transformer Encoder. Driven by absolute spatial embeddings, this module natively establishes cross-view geometric correspondences, yielding a globally consistent memory $\mathcal{M}$. While the fusion module relies on absolute coordinates to align global context, synthesizing occluded structures (Section 3.3) strictly requires fine-grained relative distances. To provide this, we apply Continuous 3D Rotary Position Embedding (RoPE) exclusively to the attention queries and keys. For a specific axis component $p \in \{x, y, z\}$ of a 3D coordinate and its channel dimension d, the phase angle is computed continuously as $\theta_d = p \cdot 10000^{-2d/D_{axis}}$, where D_{axis} is the channel dimension allocated per spatial axis. The independent embeddings for the three axes are concatenated to impose strict Euclidean relative-distance priors on the network.

3.2 OBJECT-CENTRIC 3D INSTANCE MODULE

To overcome the limitations of 2D bounding boxes and masks, we initialize state probes directly in continuous 3D space. This unifies absolute 3D geometric anchoring with semantic feature extraction.

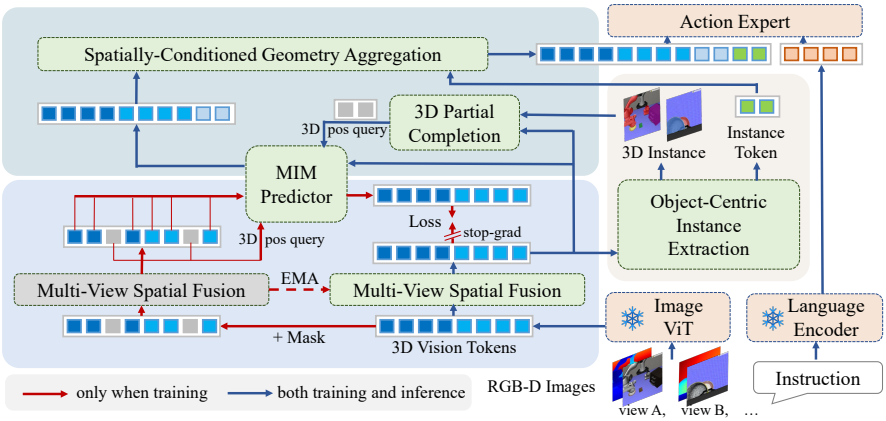

Figure 2: **Overall Architecture of 3DVLA.** The pipeline consists of four interconnected components: (1) Multi-View Spatial Fusion, (2) Object-Centric 3D Instance Module, (3) Coordinate-Driven 3D Self-Supervised Predictor, and (4) Spatially-Conditioned Geometry Aggregation.

3.2.1 3D Probe Decoding and Multi-Scale Sampling

To extract entities from the globally consistent memory $\mathcal{M}$ (Section 3.1), we instantiate N_q state probes $\mathcal{Q} = \{(\mathbf{c}_j, \mathbf{p}_j)\}_{j=1}^{N_q}$, where $\mathbf{c}_j \in \mathbb{R}^C$ encapsulates the semantic state and $\mathbf{p}_j \in \mathbb{R}^3$ denotes its 3D reference point. Specifically, $\mathcal{M}$ is decoded via transposed convolutions into a multi-scale semantic pyramid $\mathcal{M}_{ms}$ (strides 8, 16, 32) and a 1/4-scale map $\mathcal{M}_{mask}^{(4)}$ for high-resolution mask supervision.

The 3D point $\mathbf{p}_j$ is projected to the v-th camera view via intrinsic $\mathbf{K}_v$ and extrinsic $[\mathbf{R}_v|\mathbf{t}_v]$ matrices. To prevent numerical instability near the camera plane, we introduce a depth clamping mechanism:

$$\mathbf{u}_{j,v} = \Phi_{norm}\left(\frac{\mathbf{K}_v(\mathbf{R}_v\mathbf{p}_j + \mathbf{t}_v)}{\max(Z_c, \epsilon)}\right) \in [0,1]^2, \tag{2}$$

where Z_c is the projected depth, $\epsilon = 0.01$ serves as a numerical stabilizer, and Φ_{norm} normalizes the coordinates. Treating $\mathbf{u}_{j,v}$ as the sampling pivot, the semantic vector $\mathbf{c}_j$ predicts 2D spatial offsets $\Delta\mathbf{u}$ and attention weights A. The probe then extracts textures from $\mathcal{M}_{ms}$ via Deformable Cross-Attention:

$$\mathbf{h}_{j,v} = \sum_{m=1}^{M} \mathbf{W}_m \sum_{k=1}^{K}\sum_{s=1}^{S} A_{j,v,m,k,s} \cdot \mathbf{W}'\mathcal{M}_{ms}^{(s)}\left(\mathbf{u}_{j,v} + \Delta\mathbf{u}_{j,v,m,k,s}\right), \tag{3}$$

where M, K, S denote the number of attention heads, sampled keys, and feature scales, respectively, and the predicted attention weights A are normalized to sum to 1.

Crucially, a single 3D physical entity naturally projects to drastically different 2D shapes across multiple cameras. Forcing a shared 3D semantic state $\mathbf{c}_j$ to directly predict these varying 2D masks would lead to conflicting gradient updates. To resolve this, we strictly anchor predictions to $\mathbf{p}_j$, but condition the regression of view-dependent bounding boxes and masks on a view-specific positional encoding $\mathrm{PE}(\mathbf{u}_{j,v})$. This enables the shared probe $\mathbf{c}_j$ to handle perspective variations without compromising its 3D topological consistency.

3.2.2 Occlusion-Aware Gating and Bounded Coordinate Refinement

To filter geometric noise from occluded or out-of-view perspectives, an adaptive gating mechanism evaluates view validity: $g_{j,v} = \mathrm{sigmoid}(\mathrm{MLP}(\mathbf{h}_{j,v})/\tau_{view})$, where τ_{view} is a learnable temperature scalar that controls the sharpness of the gating. At each decoder layer l, the semantic state is residually updated via $\mathbf{c}_j^{(l+1)} = \mathbf{c}_j^{(l)} + \sum_v \tilde{g}_{j,v}\mathbf{h}_{j,v}^{(l)}$, where $\tilde{g}_{j,v}$ is the normalized gating weight, naturally suppressing unclear observations.

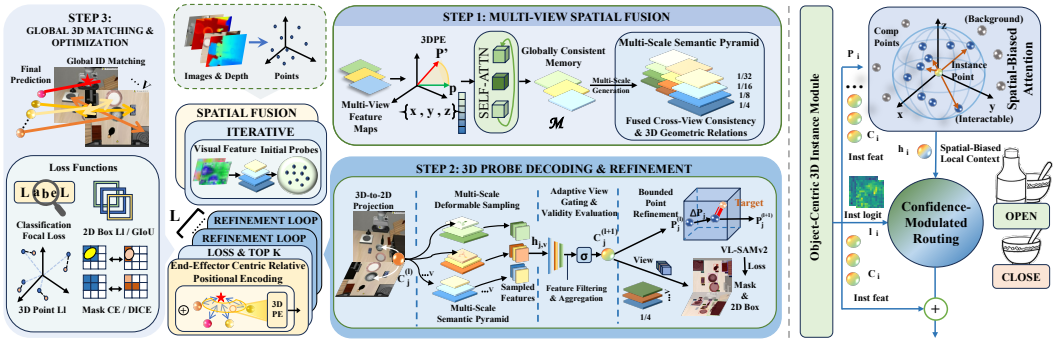

Figure 3: **Left**: Object-centric 3D instance modeling. **Right**: Spatially-conditioned geometry aggregation with uncertainty-guided routing.

Concurrently, the reference point $\mathbf{p}_j$ must be calibrated to correct initial depth inaccuracies. To prevent spatial divergence during training, we propose a bounded coordinate refinement:

$$\mathbf{p}_j^{(l+1)} = \mathbf{p}_j^{(l)} + \alpha \cdot \tanh\left(\mathrm{MLP}_{coord}(\mathbf{c}_j^{(l+1)})\right). \tag{4}$$

The $\tanh$ activation, scaled by a predefined physical factor α, limits the maximum spatial displacement per layer, guaranteeing stable 3D centroid convergence.

3.2.3 Global 3D Matching and Joint Optimization

We shift label assignment exclusively into the 3D space to eliminate cross-view ID conflicts. Pseudo ground-truth 2D bounding boxes and masks are generated by an off-the-shelf frozen perception model Lin & Wang (2025), and their 3D centroids are derived by unprojecting 2D centers using aligned depth maps. The global matching cost $\mathcal{C}_{global}(j,k)$ between probe j and target k is averaged across all valid views $\mathcal{V}_k$:

$$\mathcal{C}_{global}(j,k) = \frac{1}{\max(|\mathcal{V}_k|,1)} \sum_{v \in \mathcal{V}_k} \mathcal{C}_v(j,k), \tag{5}$$

where $\mathcal{C}_v(j,k)$ measures the semantic and geometric consistency between prediction j and target k in view v, including classification, bounding-box, mask, and 3D centroid distance terms. We apply the Hungarian algorithm once to $\mathcal{C}_{global}$ to obtain the optimal assignment.

The network is optimized by 6 joint loss components:

$$\mathcal{L} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{box}\mathcal{L}_{L1}^{2D} + \lambda_{giou}\mathcal{L}_{GIoU} + \lambda_{mask}\mathcal{L}_{CE}^{mask} + \lambda_{dice}\mathcal{L}_{DICE}^{mask} + \lambda_{3d}\mathcal{L}_{L1}^{3D}. \tag{6}$$

Specifically, $\mathcal{L}_{cls}$ is the focal loss for classification. $\mathcal{L}_{L1}^{2D}$ and $\mathcal{L}_{GIoU}$ optimize 2D boxes. 2D masks are optimized via cross-entropy ($\mathcal{L}_{CE}^{mask}$) and DICE losses ($\mathcal{L}_{DICE}^{mask}$) using the spatial dot product between $\mathbf{c}_j$ and $\mathcal{M}_{mask}^{(4)}$. Lastly, $\mathcal{L}_{L1}^{3D}$ constrains absolute 3D positions against unprojected pseudo ground-truth centroids, firmly anchoring 2D features to physical 3D space.

3.3 Coordinate-Driven 3D Self-Supervised Predictor

To address fragile reasoning under occlusion, we propose a **Coordinate-Driven 3D Self-Supervised Predictor**. We repurpose the masked self-supervised predictor to formulate 3D completion as a coordinate-conditioned feature-regression process for synthesizing unobserved structures.

3.3.1 Target Coordinate Formulation and Geometric Completion

We define Mask Coordinates ($\mathbf{P}_{mask}$) for self-supervised distillation and Completion Coordinates ($\mathbf{P}_{comp}$) for unobservable geometries. During training, asymmetric multi-view masking forces the network to leverage cross-view geometric priors instead of trivial spatial interpolation to infer missing structures, with unmasked regions forming the visible context $\mathcal{X}_{vis}$ anchored at $\mathbf{P}_{vis}$. To generate $\mathbf{P}_{comp}$, we back-project 2D masks into a raw point cloud, denoise it via dynamic density-based clustering, and use a pre-trained completion network Wang et al. (2024) to predict a dense shape. We

isolate unobservable regions by sampling the farthest points from the observed geometry and then apply Farthest Point Sampling to obtain a uniform distribution. At inference, masking is disabled, and the predictor queries the continuous space directly with $\mathbf{P}_{comp}$.

3.3.2 Position-View Conditioned Hybrid Attention

Inferring unobserved features requires encoding both absolute coordinates and specific camera perspectives. We propose a Hybrid 3D Attention Mechanism that decouples identity injection from relative distance encoding. We inject a continuous Fourier 3D positional encoding $\Phi_{pos}(\cdot)$ and discrete camera view embedding $\mathbf{E}_{view}(\cdot)$ into the context $\mathcal{X}_{vis}$ and target queries $\mathcal{Z}_{tgt}$:

$$\tilde{\mathcal{X}}_{vis} = \mathcal{X}_{vis} + \Phi_{pos}(\mathbf{P}_{vis}) + \mathbf{E}_{view}(\mathbf{V}_{vis}), \tag{7}$$

$$\tilde{\mathcal{Z}}_{tgt} = \mathcal{Z}_{tgt} + \Phi_{pos}(\mathbf{P}_{tgt}) + \mathbf{E}_{view}(\mathbf{V}_{tgt}), \tag{8}$$

where $\mathbf{V}_{vis}$ and $\mathbf{V}_{tgt}$ denote the respective view identifier vectors for the context and target coordinates, $\mathbf{P}_{tgt} = \mathbf{P}_{mask} \cup \mathbf{P}_{comp}$. Continuous 3D RoPE is applied exclusively to the Query ($\mathbf{Q}$) and Key ($\mathbf{K}$) matrices to inject Euclidean distance priors. The core Hybrid Attention (H-Attn) operator is defined as:

$$\mathbf{Q} = \mathrm{RoPE}_{3D}(\mathcal{X}_q\mathbf{W}_Q, \mathbf{P}_q), \quad \mathbf{K} = \mathrm{RoPE}_{3D}(\mathcal{X}_{kv}\mathbf{W}_K, \mathbf{P}_{kv}), \tag{9}$$

$$\text{H-Attn}(\mathcal{X}_q, \mathcal{X}_{kv}) = \mathrm{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\Big)(\mathcal{X}_{kv}\mathbf{W}_V), \tag{10}$$

where d_k is the channel dimension per head. Each predictor block sequentially employs H-Attn for cross-attention ($\mathcal{X}_q = \tilde{\mathcal{Z}}_{tgt}, \mathcal{X}_{kv} = \tilde{\mathcal{X}}_{vis}$) to extract visual features from the unmasked context, and for self-attention ($\mathcal{X}_q = \mathcal{X}_{kv} = \tilde{\mathcal{Z}}_{tgt}$) to exchange spatial information. This sequential integration aligns structural consistencies in the final representations $\hat{\mathcal{Z}}_{tgt} = [\hat{\mathcal{Z}}_{mask}, \hat{\mathcal{Z}}_{comp}]$.

3.3.3 Distillation Objective and Anti-Collapse Regularization

To stabilize continuous feature regression, representations are distilled from an EMA teacher network ($\theta_{\mathcal{T}} \leftarrow m\theta_{\mathcal{T}} + (1-m)\theta_{\mathcal{S}}$). Continuous feature prediction is highly susceptible to minimizing reconstruction error by simply outputting an averaged vector. To prevent this, our objective governs the magnitude, direction, and variance of features:

$$\mathcal{L}_{ss} = \lambda_{recon}\mathcal{L}_{recon} + \lambda_{cos}\mathcal{L}_{cos} + \lambda_{var}\mathcal{L}_{var}. \tag{11}$$

Here, $\mathcal{L}_{recon}$ is a Smooth-L1 loss for magnitude alignment. $\mathcal{L}_{cos}$ is a cosine distance penalty aligning semantic angles independent of their norms:

$$\mathcal{L}_{cos} = 1 - \frac{1}{N_{mask}} \sum_{i=1}^{N_{mask}} \frac{\hat{\mathbf{z}}_i \cdot \mathbf{y}_i}{\|\hat{\mathbf{z}}_i\|_2 \|\mathbf{y}_i\|_2}, \tag{12}$$

where $\mathbf{y}_i \in \mathcal{Y}_{tea}$ is the corresponding teacher target token. Finally, $\mathcal{L}_{var}$ is a variance regularization loss to avoid spatial homogeneity:

$$\mathcal{L}_{var} = \frac{1}{D} \sum_{d=1}^{D} \mathrm{ReLU}\Big(\mathrm{std}_d(\mathcal{Y}_{tea}) - \mathrm{std}_d(\hat{\mathcal{Z}}_{mask})\Big), \tag{13}$$

where $\mathrm{std}_d(\cdot)$ computes the standard deviation for the d-th channel.

3.4 Spatially-Conditioned Geometry Aggregation

3.4.1 End-Effector Centric Relative Positional Encoding

In downstream manipulation, actions are inherently executed relative to the robot's gripper. Providing only absolute world coordinates forces the policy to implicitly learn complex kinematic transformations. To bypass this, we explicitly ground the geometric representations in end-effector-centric spatial offsets: $\Delta\mathbf{p} = \mathbf{p}_{obj} - \mathbf{p}_{ee}$. To capture high-frequency spatial variations, $\Delta\mathbf{p}$ is expanded using sinusoidal functions across F frequencies:

$$\Phi_{freq}(\Delta\mathbf{p}) = [\Delta\mathbf{p}, \sin(2^0\pi\Delta\mathbf{p}), \cos(2^0\pi\Delta\mathbf{p}), \ldots, \cos(2^{F-1}\pi\Delta\mathbf{p})]. \tag{14}$$

Table 1: **Main Results on LIBERO-Plus.** We report success rates (%) across 7 perturbation categories. **Ours** achieves the highest average success rate, significantly outperforming state-of-the-art baselines under severe visual and spatial variations.

Methods	Guidance	Camera	Robot	Language	Light	Background	Noise	Layout	Avg.
WorldVLA Cen et al. (2025)	Visual	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
OpenVLA Kim et al. (2025b)	Linguistics	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
NORA Hung et al. (2025)	Linguistics	2.2	37.0	65.1	45.7	58.6	12.8	62.1	39.0
UniVLA Bu et al. (2025)	Linguistics	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
π_0-Fast Pertsch et al. (2025)	Linguistics	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
RIPT-VLA Tan et al. (2025)	Linguistics	55.2	31.2	77.6	88.4	91.6	73.5	74.2	68.4
OpenVLA-OFT Kim et al. (2025a)	Linguistics	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
π_0 Black et al. (2024)	Linguistics	61.0	40.8	63.5	89.3	84.1	80.1	76.4	69.4
$\pi_{0.5}$ Intelligence et al. (2025)	Linguistics	71.2	77.1	**89.9**	94.7	94.0	84.2	84.3	84.2
$\pi_{0.5}$+3DVLA	Linguistics	**75.6**	**77.4**	88.6	**97.4**	**97.7**	**85.3**	**86.6**	**86.0**

Table 2: **Main Results on RoboTwin 2.0.**

Method	Easy	Hard
DP Chi et al. (2025)	28.0	0.6
ACT Zhao et al. (2023)	29.7	1.7
RDT Liu et al. (2025)	34.5	13.7
DP3 Ze et al. (2024)	55.2	5.0
π_0 Black et al. (2024)	46.4	16.3
π_0**+3DVLA**	54.5	23.2
X-VLA Zheng et al. (2025)	70.0	39.0
X-VLA+3DVLA	**80.6**	**48.2**

Table 3: **Real-world Evaluation on Quantum-2 Robot.** We report the mean and standard deviation of success rates (%) over 5 independent training runs. Avg. denotes the average success rate across tasks.

Method	Avg.	Pick-and-Place	Fold Clothes	Press Button
$\pi_0.5$	62.5	68.4 ± 7.6	32.8 ± 6.4	86.2 ± 2.6
3DVLA	70.3	$\mathbf{74.0 \pm 4.6}$	$\mathbf{50.4 \pm 5.1}$	$\mathbf{86.6 \pm 2.5}$

This continuous vector is projected via an MLP to match the token dimension. The resulting affordance embeddings are added to the corresponding instance and completion tokens. Finally, these spatially grounded tokens are concatenated with the visual tokens and fed into the action expert, thereby perfectly fulfilling the plug-and-play design without altering the base VLA architecture.

3.4.2 Uncertainty-Guided Instance Geometric Routing

Fusing dense 3D completion features directly into the semantic states of instance probes ($\mathbf{c}_i$) risks corrupting highly confident visual representations. We propose Uncertainty-Guided Instance Geometric Routing to adaptively inject 3D geometry based on physical proximity and explicit detection uncertainty.

Spatial-Biased Local Context. Using 3D Euclidean distance, we retrieve the K-nearest completion points $\mathcal{N}(i)$ for the instance centered at $\mathbf{p}_i$. The local geometric context $\mathbf{h}_i$ is aggregated via cross-attention, where relative physical offsets dictate the attention bias:

$$\mathbf{h}_i = \sum_{j \in \mathcal{N}(i)} \text{softmax}_j \left(\frac{\mathbf{q}_i \mathbf{k}_j^T}{\sqrt{d_k}} + \text{MLP}_{bias}(\mathbf{p}_j - \mathbf{p}_i) \right) \mathbf{v}_j \mathbf{W}_{out}, \tag{15}$$

where $\mathbf{q}_i$ is linearly projected from the instance semantic state $\mathbf{c}_i$, while $\mathbf{k}_j$ and $\mathbf{v}_j$ are projected from the local 3D completion feature at point $\mathbf{p}_j$. Crucially, the dot product $\mathbf{q}_i \mathbf{k}_j^T$ captures semantic consistency, while the penalty term $\text{MLP}_{bias}(\mathbf{p}_j - \mathbf{p}_i)$ encodes spatial constraints. This joint formulation ensures that the instance only attends to geometries that are both semantically matched and physically adjacent.

Confidence-Modulated Routing. To ensure the network absorbs 3D context only when necessary, we quantify the ambiguity of the 2D observation. Using the classification logit l_i generated by the instance module, we define an uncertainty score $u_i = 1 - \sigma(l_i) \in [0, 1]$, where σ denotes the sigmoid function. We formulate a dynamic routing gate $\mathbf{g}_i \in \mathbb{R}^D$ conditioned on the instance feature, local 3D context, and uncertainty score:

$$\mathbf{g}_i = \sigma\Big(\tau \cdot \text{MLP}_{gate}\left([\mathbf{c}_i \parallel \mathbf{h}_i \parallel u_i]\right)\Big), \tag{16}$$

where $[\cdot \parallel \cdot]$ denotes channel concatenation. Crucially, the final linear layer of MLP_{gate} is zero-initialized with a constant bias of -3.0. This guarantees an identity mapping ($\mathbf{g}_i \approx \mathbf{0}$) at the onset of training, protecting the pre-trained 2D features from gradient chaos. The instance representation is ultimately updated via $\hat{\mathbf{c}}_i = \mathbf{c}_i + \mathbf{g}_i \odot \text{LayerNorm}(\mathbf{h}_i)$.

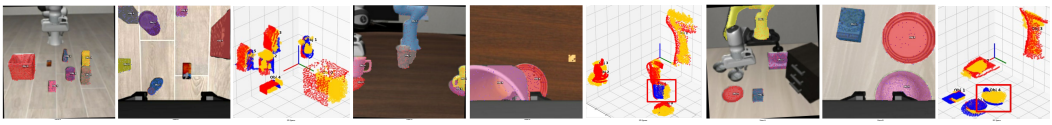

Figure 4: **Multi-view instance consistency and 3D completion. Left & Center:** Globally consistent masks and projected points generated from instance tokens across different views. **Right:** 3D geometry completion results in world space. Red and blue dots are visible points unprojected from respective views; yellow stars represent the predicted full geometry completed by our model.

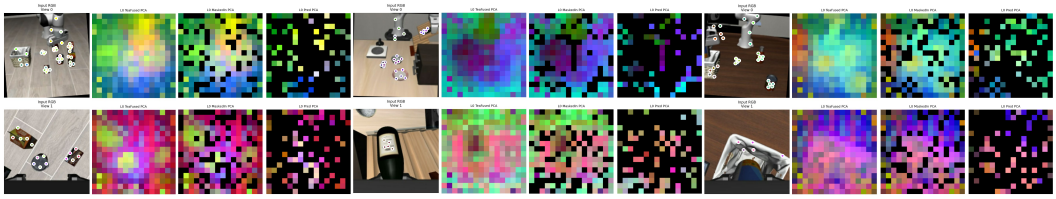

Figure 5: **Coordinate-Driven 3D Self-Supervised Predictor.** Illustration of the predictor's geometry synthesis capabilities under severe occlusion and visual perturbations.

4 Experiments

4.1 Benchmarks and Metrics

We conduct experiments on two highly challenging simulation benchmarks. We strictly follow their official training splits and evaluation protocols. **LIBERO-Plus** Fei et al. (2025) is a robotic benchmark built on LIBERO platform. It introduces seven distinct perturbation dimensions (e.g., Camera, Layout, Background, Light). **RoboTwin 2.0** Chen et al. (2025) emphasizes physics-rich, contact-heavy manipulation requiring precise 3D geometric perception. It features diverse multi-view setups and severe self-occlusions across articulated, deformable, and long-horizon tasks. These benchmarks use success rate as the evaluation metric.

4.2 Main Results

Zero-Shot Transfer on LIBERO-Plus. Under zero-shot transfer from LIBERO to LIBERO-Plus, 3DVLA achieves a new state-of-the-art average success rate of 86.0% (Table 1). It is highly resilient to spatial shifts (e.g., *Camera*, *Layout*) and visual perturbations (e.g., *Light*, *Background*). We attribute these gains to anchoring features in continuous 3D space for multi-view consistency and to the coordinate-driven predictor that reconstructs occluded task-relevant geometry. We dissect these mechanisms in Section 4.3.

Comprehensive Manipulation on RoboTwin 2.0. We further benchmark 3DVLA on RoboTwin 2.0 to assess physics-rich interaction. Demonstrating plug-and-play compatibility, we integrate 3DVLA into π_0 and X-VLA, and yield consistent and substantial improvements on both Easy and Hard settings (Table 2). This confirms the effectiveness of our proposed method, while 3DVLA adds only $\sim$0.086B parameters and 8.6% (21.22 ms on an A100) computational overhead.

Real-world Evaluation. To validate sim-to-real transfer, we evaluate 3DVLA on the Quantum-2 robot. For each task, we collect 100 teleoperated demonstrations and train five independent IL policies with different random seeds. Table 3 reports the mean success rate and standard deviation over 50 test episodes. Comparisons of baseline methods and 3DVLA, together with a real-world demonstration, are provided in Appendix B.4 and Supplementary Video.

4.3 Ablation Study

We examine the contribution of each proposed module via systematic ablation experiments. To provide a comprehensive view, we analyze the progressive macro-integration across both RoboTwin 2.0 (Table 6) and LIBERO-Plus (Table 4). We then delve into micro-ablations detailing specific architectural choices (Table 5).

Table 4: **Overall Ablation Studies on LIBERO-Plus.** *Inst.*: 3D Instance Module; *Pred.*: 3D Self-Supervised & Predictor; *Rout.*: Uncertainty-Guided Routing for the Aggregation.

Model	Inst.	Pred.	Rout.	Camera	Robot	Language	Light	Background	Noise	Layout	Avg.
Baseline				71.2	77.1	**89.9**	94.7	94.0	84.2	84.3	84.2
A	✓			71.4	78.9	87.9	96.0	94.4	86.5	86.3	85.0
B	✓	✓		73.6	**79.4**	88.2	97.1	96.5	85.2	86.4	85.7
C (Ours)	✓	✓	✓	**75.6**	77.4	88.6	**97.4**	**97.7**	**85.3**	**86.6**	**86.0**

Table 5: **Ablation of the 3D Instance Module.**

Variant	Layout	Avg.
w/o Spatial Fusion	84.6	84.6
2D Matching (vs. Global 3D)	85.4	84.5
w/o End-Effector Relative PE	86.0	84.8
Full Instance Module (A)	**86.3**	**85.0**

Table 6: **Comprehensive Ablation Studies on RoboTwin 2.0.**

Model	Inst.	Pred.	Rout.	Easy	Hard
π_0				46.4	16.3
A	✓			50.1	18.9
B	✓	✓		53.4	21.8
C (Ours)	✓	✓	✓	**54.5**	**23.2**
X-VLA				70.0	39.0
A	✓			72.3	41.7
B	✓	✓		78.5	46.5
C (Ours)	✓	✓	✓	**80.6**	**48.2**

3D Instance Module & Spatial Fusion. The 3D Instance Module (A) raises the average success rate on both LIBERO-Plus and RoboTwin 2.0, confirming that direct 3D entity extraction overcomes flat 2D region reasoning. 1) Removing voxel-free spatial fusion drops performance from 85.0% to 84.6%, showing that pre-decoding multi-view consistency is necessary to align views into coherent memory (Figure 4, Left & Center). 2) Replacing our global 3D bipartite matching with heuristic 2D matching drops the average to 84.5%, as 2D matching inevitably reintroduces cross-view ID conflicts when cameras shift. 3) Omitting the end-effector relative positional encoding degrades performance (84.8%). Logically, action execution is intrinsically relative to the robot's gripper. Feeding absolute world coordinates forces the policy to implicitly learn complex kinematic transformations, whereas our relative encoding provides a direct, actionable spatial offset.

Coordinate-Driven 3D Self-Supervised Predictor. Introducing the 3D Self-Supervised Predictor (B) directly targets fragile reasoning under occlusion. This addition increases the LIBERO-Plus average to 85.7% and yields a substantial jump in RoboTwin 2.0's *Hard* tasks (18.9% $\rightarrow$ 21.8%). Specifically, as visualized in Figure 5, the significant gain in the *Camera* split (71.4% $\rightarrow$ 73.6%) confirms that actively inferring unseen topologies prevents the policy from failing when task-relevant parts are hidden.

Uncertainty-Guided Instance Geometric Routing. Finally, we evaluate the Uncertainty-Guided Instance Geometric Routing (C). Model B feeds completion and instance tokens separately to the action expert. Model C instead integrates local 3D geometry into instance representations based on observation uncertainty. This adaptive fusion improves LIBERO-Plus performance by +2.0% under *Camera* variations and achieves the best results on both LIBERO-Plus (86.0% average) and RoboTwin 2.0 (48.2% Hard). The improvement comes from two aspects: 3D geometry fusion reduces the burden of learning spatial-semantic alignment, while uncertainty-aware routing prevents unreliable synthesized features from contaminating confident visual representations.

5 Conclusion

In this paper, we propose 3DVLA, a plug-and-play framework that enriches VLAs with robust 3D reasoning while remaining fully compatible with VLM pre-training and requiring no additional manual annotations. Our method integrates pervasive multi-view consistency constraints for viewpoint-invariant representations, instance tokens for 3D instance awareness, and a masked, self-supervised 3D-encoding branch with a retained predictor for visual token completion. Experiments on the LIBERO-Plus and RoboTwin 2.0 benchmarks demonstrate that 3DVLA consistently improves manipulation performance over multiple baselines, validating both its effectiveness and its practical plug-and-play compatibility.

AI USE STATEMENT

We used generative AI tools to polish the text and check grammar. All AI-assisted outputs were reviewed and verified by the authors. The authors take responsibility for the final content of this paper.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method, training procedures, and experimental settings in the main paper and supplementary materials. We will release code and necessary resources to facilitate reproduction after publication.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, 2023. 2, 3

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 2, 7

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

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025. 2, 7

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025. 2, 7

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025. 8

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *IJRR*, 2025. 2, 7

Ashley Edwards, Himanshu Sahni, Yannick Schroecker, and Charles Isbell. Imitating latent policies from observation. In *ICML*, 2019. 3

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025. 2, 8

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U Tan, Navonil Majumder, Soujanya Poria, et al. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*, 2025. 2, 7

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. 2, 7

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025a. 7

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *CoRL*, 2025b. 1, 2, 7

Chengmeng Li, Junjie Wen, Yaxin Peng, Yan Peng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. *IEEE Robotics Autom. Lett.*, 2026. 3

Zhiwei Lin and Yongtao Wang. Vl-sam-v2: Open-world object detection with general and specific query fusion. *arXiv preprint arXiv:2505.18986*, 2025. 5

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *ICLR*, 2025. 3, 7

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *CoRL*, 2020. 3

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2023. 3

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025. 7

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025. 3

Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *WACV*, 2022. 1

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026. 3

Lin Sun, Bin Xie, Yingfei Liu, Hao Shi, Tiancai Wang, and Jiale Cao. Geovla: Empowering 3d representations in vision-language-action models. *arXiv preprint arXiv:2508.09071*, 2025. 3

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025. 2, 7

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. In *RSS*, 2024. 1

Jun Wang, Ying Cui, Dongyan Guo, Junxia Li, Qingshan Liu, and Chunhua Shen. Pointattn: You only need attention for point cloud completion. In *AAAI*, 2024. 5

Zhongyu Xia, Jishuo Li, Zhiwei Lin, Xinhao Wang, Yongtao Wang, and Ming-Hsuan Yang. Openad: Open-world autonomous driving benchmark for 3d object detection. *arXiv preprint arXiv:2411.17761*, 2024. 1

Zhongyu Xia, Zhiwei Lin, Yongtao Wang, and Ming-Hsuan Yang. Henet++: Hybrid encoding and multi-task learning for 3d perception and end-to-end autonomous driving. *arXiv preprint arXiv:2511.07106*, 2025. 1

Zhongyu Xia, Yousen Tang, Yongtao Wang, Zhifeng Wang, and Weijun Qin. R4det: 4d radar-camera fusion for high-performance 3d object detection. In *CVPR*, 2026. 1

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *RSS*, 2024. 2, 7

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *RSS*, 2023. 3, 7

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025. 7

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023. 1

A Implementation Details

To facilitate future research and ensure full reproducibility, we will *open-source* our complete codebase, pre-trained model weights, and evaluation scripts upon publication. Below, we detail the architectural hyperparameters, optimization strategies, and computational costs associated with 3DVLA.

A.1 Architectural Hyper-parameters

A core advantage of 3DVLA is its parameter efficiency. During training, the large underlying Vision-Language Model (VLM) backbone is kept entirely frozen, while only the newly introduced 3D perception modules and the downstream action expert are actively updated. For the *Object-Centric 3D Instance Module*, we initialize $N_q = 300$ state probes to ensure comprehensive scene coverage. Through matching and uncertainty filtering, these probes are adaptively distilled into a maximum of 16 highly confident instance tokens per scene. For the *Coordinate-Driven 3D Predictor*, to balance computational efficiency and geometric fidelity, the network processes at most 6 occluded object instances per observation, with each instance uniformly sampled to ensure it contains exactly 5 completion coordinate queries.

A.2 Two-Stage Optimization Procedure

We emphasize that this optimization is performed during the post-training stage after loading publicly available pretrained VLA checkpoints. To ensure stable optimization and prevent continuous feature regression from dominating or collapsing the early learning phase, we decouple 3DVLA's training into a progressive two-stage pipeline.

Stage 1: Spatial Fusion and Instance Extraction. In the first stage, we focus exclusively on learning view-consistent physical entities. We jointly train the Multi-View Spatial Fusion module, the 3D Instance Module, and the action expert. This stage is trained for 5,000 steps with an effective batch size of 64.

Stage 2: 3D Completion and Geometry Aggregation. In the second stage, we freeze the weights of the modules trained in Stage 1 to preserve the stable 2D-to-3D geometric anchoring. We then exclusively train the 3D Self-Supervised Predictor, the Geometry Aggregation module, and continue updating the action expert. This phase is also trained for 5,000 steps with a batch size of 64.

A.3 Optimization and Loss Coefficients

The network is optimized using the AdamW optimizer with a gradient clipping norm of 0.5 and a weight decay of 1×10^{-4}. We employ a cosine learning rate decay schedule, featuring 1,200 warmup steps to reach a peak learning rate of 3×10^{-6}, followed by 3,800 decay steps down to 1×10^{-8}.

For the joint geometric alignment loss (Equation 6), the balancing coefficients are empirically set to precisely match the scale of heterogeneous geometric properties: $\lambda_{cls} = 2.0$, $\lambda_{box} = 5.0$, $\lambda_{giou} = 2.0$, $\lambda_{mask} = 5.0$, $\lambda_{dice} = 5.0$, and $\lambda_{3d} = 5.0$. For the self-supervised distillation objective (Equation 11), to strictly prevent feature homogeneity, we strongly penalize variance collapse while maintaining directional alignment: $\lambda_{recon} = 1.0$, $\lambda_{cos} = 0.1$, and $\lambda_{var} = 10.0$. The Exponential Moving Average (EMA) momentum for updating the teacher network is set to $m = 0.999$.

A.4 Computational Cost

The entire two-stage training process is highly efficient and is conducted on a single compute node equipped with 4 NVIDIA A800 GPUs. Stage 1 takes approximately 5 hours, and Stage 2 takes roughly 9 hours, bounding the total training pipeline to merely 14 hours.

During evaluation, due to the physics-rich nature of the environments and the requirement for comprehensive validation across diverse perturbation setups, the testing phase is heavily bottlenecked by simulation rendering and physical stepping rather than model inference. We scale the evaluation across 8 NVIDIA A800 GPUs, which take approximately 14 hours to complete the extensive simulation benchmark suites.

B Additional Experimental Analysis

B.1 Ablation on Self-Supervised Masking Ratio

As shown In Table 7, we evaluate the masking ratio during self-supervised training. A lower ratio of 25% yields sub-optimal results (85.5%) because the network bypasses 3D reconstruction by simply interpolating features

from adjacent visible 2D pixels. Conversely, our 50% asymmetric masking explicitly forces the predictor to perform true cross-view spatial reasoning, yielding the highest structural fidelity.

Table 7: Ablation on the masking ratio for the self-supervised training on LIBERO-Plus.

Masking Ratio	Camera	Noise	Avg.
25% Masking	73.1	85.0	85.5
50% Masking (B)	**73.6**	**85.2**	**85.7**

B.2 Robustness Analysis

Existing VLA models have rarely explored 3D understanding, largely because manual 3D annotation is prohibitively expensive and it is difficult to effectively integrate such 3D information into the decision-making process. We believe that one of the key advantages of 3DVLA is its ability to fully leverage the prior information from existing models. The modules we added are not a simple stack. Instead, they are coupled with each other and follow an inherent internal logic.

Admittedly, 3DVLA introduces additional sequential modules. Therefore, we supplement a series of robustness experiments to evaluate the cumulative error, as shown in Table 8. We conduct controlled inference-time perturbation experiments on three components: depth observations, instance predictions, and completion coordinates. Depth Noise and Completion Noise are implemented by adding Gaussian noise to the depth maps or predicted completion coordinates, respectively, where $a = 1$ means the signal is entirely noise. Instance Noise is implemented by randomly replacing a proportion a of the predicted instances with random instances sampled from other scenes.

For the composite noise setting, we adopt $a = 0.1$ for Depth Noise, $a = 0.3$ for Instance Noise, and $a = 0.3$ for Completion Noise. In fact, even without manually injected noise, the model's own intermediate predictions are not perfect: we manually evaluate 100 test scenes and observe that the instance prediction recall is approximately 95%. The experimental results demonstrate that 3DVLA consistently achieves a higher success rate than the baseline under varying degrees of noise, proving that our proposed 3D modules are robust against imperfect intermediate geometric representations.

Table 8: Robustness evaluation under different noise perturbations. All perturbations are applied only during inference, where a denotes the artificially added noise ratio.

	Depth Noise				Instance Noise				Completion Noise				Composite Noise
a	0	0.1	0.3	0.5	0	0.3	0.5	1	0	0.3	0.5	1	
LIBERO-Plus													
Baseline Pi0.5	84.2	83.7	70.5	56.1	84.2	-	-	-	84.2	-	-	-	83.7
Pi0.5 + 3DVLA	**86.0**	**84.9**	**75.8**	**60.2**	**86.0**	**85.4**	**85.7**	84.1	**86.0**	**85.6**	**85.1**	**85.0**	**84.7**
RoboTwin 2.0													
Baseline Pi0	46.4	44.1	38.8	**23.2**	46.4	-	-	-	46.4	-	-	-	44.1
Pi0 + 3DVLA	**54.5**	**52.0**	**43.9**	21.2	**54.5**	**53.3**	**50.8**	**46.9**	**54.5**	**53.9**	**51.6**	**50.1**	**51.7**

B.3 Multi-seed Evaluation

To rigorously validate our performance deltas and rule out run-to-run variance, we conduct expanded evaluations using five independent random seeds for both the LIBERO-Plus and RoboTwin 2.0 benchmarks.

As summarized in Table 9 and Table 10, the results from repeated experiments with multiple random seeds continue to support our original conclusions. Moreover, we observe that 3DVLA can reduce variance to a certain extent. This suggests that injecting explicit 3D structural priors helps alleviate behavioral uncertainty under complex visual and geometric variations.

B.4 Qualitative Video Comparisons

We provide a qualitative demonstration of 3DVLA on diverse manipulation scenarios, including LIBERO-Plus, RoboTwin 2.0, and a real-world cloth folding task. The video shows that baseline VLA models often fail under

Table 9: Multi-seed evaluation on LIBERO-Plus. Results are reported as Mean ± Standard Deviation (%) over five independent runs.

Model	Camera	Robot	Language	Light	Background	Noise	Layout	Avg
Baseline Pi0.5	70.9±2.1	76.8±1.8	89.5±1.5	94.4±1.2	93.7±1.6	84.5±2.0	83.9±2.2	83.9±1.2
Pi0.5 + 3DVLA	**75.3±1.4**	**77.8±1.5**	88.9±1.6	**97.2±0.9**	**96.9±0.8**	**85.7±1.7**	**87.0±1.5**	**86.2±0.8**

Table 10: Multi-seed evaluation on RoboTwin 2.0. Results are reported as Mean ± Standard Deviation (%) over five independent runs.

Model	Easy	Hard
Baseline Pi0	46.2±1.7	16.6±2.9
Pi0 + 3DVLA	**54.8±1.4**	**23.1±2.2**
Baseline X-VLA	69.7±0.9	39.0±3.1
X-VLA + 3DVLA	**80.6±1.2**	**48.2±2.4**

challenging spatial conditions and severe occlusions, while 3DVLA achieves robust manipulation by leveraging 3D-aware representations and geometry reasoning.