RN-VLA: REFERENCE-POINT NORMALIZED VISION-LANGUAGE-ACTION MODELS FOR SUB-CENTIMETER FINE-GRAINED MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-grained robotic manipulation—such as pressing a specific key on a dense keyboard or performing micro-adjustments on household dials—remains challenging, as moderate per-step errors compound exponentially over long-horizon sequences. Monolithic Vision-Language-Action (VLA) models struggle here due to instance-level semantic ambiguity, geometric attention dispersion, and coordinate scale mismatch between reaching and fine manipulation. We propose RN-VLA, which grounds task-specified targets via Semantic Query Grounding (SQG), suppresses background clutter through FiLM-based Cross-Modal Gated Modulation (CMGM), and re-centers the action space onto the target part via Reference Point Normalization (RPN), enabling a diffusion policy to achieve sub-centimeter precision. Experiments in Isaac Sim and on a physical xArm6 robot show RN-VLA achieves over 80% atomic success across seven household tasks, outperforming state-of-the-art VLA and 3D diffusion baselines, with strong robustness under out-of-distribution placements. By eliminating atomic-level errors, reliable single-step policies compose zero-shot into long-horizon tasks—without LLM planning or online verification—achieving up to a 68% relative improvement over the strongest baselines (up to 1.68× higher success rate).

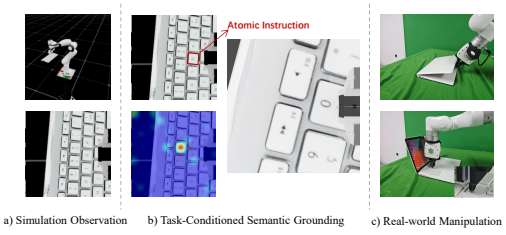

Figure 1: Overview of RN-VLA. (a) Simulation observation in Isaac Sim. (b) Task-Conditioned Semantic Grounding, where the atomic instruction (red box) is resolved via an attention heatmap that isolates the target key among visually identical neighbors. (c) Real-world manipulation on the physical xArm6 platform.

1 INTRODUCTION

Fine-grained robotic manipulation tasks, such as typing a specific key on a dense keyboard or executing micro-dial adjustments on household appliances, are ubiquitous in everyday human environments. While robotic policies have made significant strides in gross motor tasks like pick-and-place, achieving reliable sub-centimeter precision remains a persistent open challenge. This difficulty is profoundly amplified when atomic operations are chained into sequential behaviors: a policy achieving only moderate per-step accuracy sees its multi-step success probability collapse exponentially—a single 15×15 mm key among dozens of visually identical neighbors can reduce a seven-step sequence's success rate to below 1%. This mathematical reality dictates that the cornerstone of robust long-horizon manipulation is the complete eradication of execution errors at the atomic, single-step level.

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.

A dominant paradigm for robotic control is the monolithic Vision-Language-Action (VLA) model, which directly maps raw multi-view observations and language instructions to absolute motor actions within a single end-to-end network. However, for fine-grained manipulation among densely arranged targets, monolithic VLAs suffer from three deeply coupled bottlenecks. First, **instance-level semantic ambiguity**: when confronted with visually identical adjacent targets, global visual encoders struggle to identify the exact correct instance. Second, **part-level geometric attention dispersion**: even when the correct instance is embedded, the action decoder's attention often scatters across the holistic 3D scene point cloud rather than focusing on the localized geometry required for precise contact. Third, **absolute coordinate scale mismatch**: a single global coordinate frame cannot simultaneously represent long-range reaching maneuvers and close-range micro-adjustments without severely diluting control fidelity.

To break these bottlenecks and bridge the semantic-geometric gap, we present the Reference-Point Normalized Vision-Language-Action (**RN-VLA**) framework, designed to eliminate execution errors at the atomic level. RN-VLA explicitly isolates target semantics and establishes a precise geometric reference point before executing the action. Given a straightforward atomic instruction tuple (e.g., *object, part, action*), our perception frontend employs a Semantic Query Grounding (SQG) mechanism to resolve instance-level semantic ambiguity by extracting reliable 9-DOF spatial bounding boxes directly from uncurated multi-view observations. As shown in Figure 1, SQG's attention heatmap precisely isolates the instructed key among visually identical neighbors, enabling sub-centimeter execution both in simulation and on the real xArm6 platform.

Crucially, to suppress part-level geometric attention dispersion, we introduce a **Cross-Modal Gated Modulation (CMGM)** module. This allows dense 2D visual semantics to dynamically gate 3D point cloud features, effectively filtering background clutter. Finally, to resolve the scale mismatch, we propose **Reference Point Normalization (RPN)**, a strategy that dynamically re-centers the 3D policy's coordinate frame onto the target part's centroid. A low-level diffusion policy then operates strictly within this local reference frame, enabling highly consistent sub-centimeter control regardless of the object's absolute distance from the robot base.

We validate our framework through extensive simulations in Nvidia Isaac Sim and real-world deployments on a physical 6-DoF robotic arm. Our method demonstrates unprecedented sub-centimeter precision and robust spatial generalization under out-of-distribution (OOD) object placement shifts. Furthermore, we demonstrate that by solving the atomic precision bottleneck, our framework inherently unlocks zero-shot long-horizon composability: the highly reliable atomic policies can be sequentially concatenated to complete complex long-horizon tasks—without any LLM-based planning or online state verification.

In summary, this paper makes the following contributions:

- We identify and formalize the three coupled bottlenecks of *semantic ambiguity*, *geometric attention dispersion*, and *absolute coordinate scale mismatch* that inherently limit the spatial precision of monolithic VLA models in dense environments.
- We propose the RN-VLA framework, featuring Semantic Query Grounding (SQG) to resolve instance-level ambiguity, Cross-Modal Gated Modulation (CMGM) to suppress geometric attention dispersion, and Reference Point Normalization (RPN) to decouple global reaching from local micro-manipulation.
- Extensive evaluations demonstrate that our framework achieves state-of-the-art sub-centimeter precision and OOD spatial generalization on single-step tasks, and enables zero-shot long-horizon task completion through simple sequential composition of atomic policies, without any long-horizon-specific training.

2 Related Work

2.1 Vision-Language-Action Models and Robot Learning

Generalist robot policies trained on large-scale datasets have driven the emergence of Vision-Language-Action (VLA) models Brohan et al. (2022); Zitkovich et al. (2023); Kim et al. (2024); Team et al. (2024); O'Neill et al. (2024); Black et al. (2024). Multimodal foundation models such as PaLM-E Driess et al. (2023) and CogACT Li et al. (2024) enable VLAs to interpret multimodal

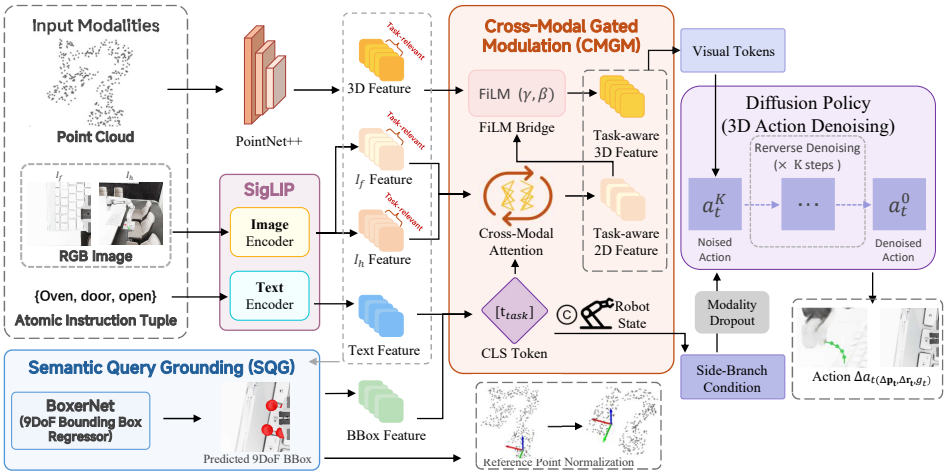

Figure 2: Overview of the RN-VLA framework. Given multi-view RGB images, a point cloud, and an atomic instruction tuple, the Semantic Query Grounding (SQG) module regresses a 9-DoF bounding box to resolve instance-level ambiguity. Cross-Modal Gated Modulation (CMGM) fuses SigLIP text and visual features with 3D point features via cross-modal attention and a FiLM bridge, yielding task-aware 2D and 3D representations. Reference Point Normalization re-centers the point cloud and robot state onto the target centroid. The fused visual tokens and side-branch condition then guide a diffusion policy that iteratively denoises the 7-DoF action.

prompts Jiang et al. (2023) and generate continuous actions, while flow matching Lipman et al. (2022) and multi-robot diffusion models Liu et al. (2025) provide expressive alternatives to categorical action tokenization. For long-horizon tasks, LLM-based methods further decompose high-level instructions into executable sub-goals Achiam et al. (2023); Lynch et al. (2023); Ahn et al. (2022); Huang et al. (2022); Singh et al. (2022); Dalal et al. (2024); Zhang et al. (2025); Shentu et al. (2024); Ouyang et al. (2025); Fan et al. (2025).

However, semantic sequencing alone does not resolve the spatial errors critical to sub-centimeter manipulation. Recent work therefore emphasizes visual grounding and task-relevant representations. CoT-VLA Zhao et al. (2025) introduces visual reasoning to improve semantic alignment, while studies highlight the importance of task-dynamic visual representations Tian et al. (2024). Earlier approaches use object-centric representations to reduce scene complexity and improve generalization Devin et al. (2018), whereas RoboInter Li et al. (2026) provides comprehensive intermediate spatial-semantic representations for manipulation. In contrast, RN-VLA uses a single task-conditioned spatial query, without recursive LLM planning or complex intermediate representations, to resolve instance ambiguity for precise atomic manipulation.

2.2 Geometric-Aware Continuous Control

While 2D-centric VLAs excel at semantic reasoning, sub-centimeter manipulation requires robust 3D representations. Building on Diffusion Policy Chi et al. (2025), methods such as PerAct Shridhar et al. (2023), DP3 Ze et al. (2024), 3D Diffuser Actor Ke et al. (2024), Act3D Gervet et al. (2023), and RVT Goyal et al. (2023) incorporate voxels, point clouds, or multi-view features for spatially grounded action generation. Self-supervised spatial correspondence learning Florence et al. (2019) and 4D visual pre-training Hou et al. (2025) further demonstrate the importance of localized spatiotemporal features, while SpatialVLA Qu et al. (2025), GeoVLA Sun et al. (2025), and 3D-VLA Zhen et al. (2024) inject spatial and 3D information into 2D foundation backbones.

For finer control, TAX-Pose Pan et al. (2023) and CoPA Huang et al. (2024) exploit part-level relative poses, while VoxPoser Huang et al. (2023) constructs 3D affordance and constraint maps. Coarse-to-fine imitation learning Johns (2021), transferable latent skill embeddings Hausman et al. (2018), and continuous motion representations Tanwani et al. (2020) further emphasize localized geometric grounding and trajectory regularization. However, most methods still operate on holistic 3D representations, where complete point clouds or voxel grids introduce substantial background clutter, causing geometric attention dispersion and limiting sub-centimeter spatial precision.

2.3 Task-Conditioned Grounding and Cross-Modal Modulation

To bridge semantic grounding and geometric execution, RN-VLA builds on query-based grounding, feature modulation, and operational-space representations. BoxerNet DeTone et al. (2026) regresses robust 3D bounding boxes from multi-view inputs, but uses class-agnostic visual encoders. In contrast, our Semantic Query Grounding (SQG) leverages pre-trained vision-language representations such as SigLIP Zhai et al. (2023) to selectively ground task-specified targets, reducing instance ambiguity.

To mitigate geometric attention dispersion, we employ cross-modal feature modulation. FiLM Perez et al. (2018) applies language-conditioned affine transformations to visual features. RN-VLA repurposes this mechanism as a Vision-to-Geometry bridge, using dense 2D semantics to gate 3D point-cloud features from PointNet++ Qi et al. (2017), thereby suppressing irrelevant geometric clutter before action prediction.

For coordinate representation, 3D diffusion controllers such as DP3 Ze et al. (2024) and 3D Diffuser Actor Ke et al. (2024) typically operate in the robot base frame, unlike task-centered operational-space formulations Khatib (1987), thus mixing global reaching with fine-grained manipulation. TAX-Pose Pan et al. (2023) and CoPA Huang et al. (2024) exploit part-level poses for spatial correspondence but do not re-center action generation. Our Reference Point Normalization instead re-centers both 3D observations and actions on the target's local frame, decoupling gross reaching from fine-grained manipulation at the representation level.

3 Methodology

We propose RN-VLA to bridge the semantic-geometric gap in fine-grained manipulation. Our framework operates as a two-stage pipeline: Sec. 3.2 instantiates the perception stage via Semantic Query Grounding (SQG), while Sec. 3.3–3.4 instantiate the reference-normalized policy stage via Cross-Modal Gated Modulation (CMGM) and a low-level Diffusion Policy. Figure 2 illustrates the full RN-VLA pipeline: SQG first grounds the atomic instruction to a 9-DoF target bounding box, whose centroid drives Reference Point Normalization; CMGM then gates task-aware 2D and 3D features via cross-modal attention and FiLM, which together with the side-branch condition guide the diffusion policy's reverse denoising process to produce the final action.

3.1 Problem Formulation and Reference Point Normalization

We formulate fine-grained robotic manipulation as a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma)$, where we do not explicitly define a reward function $\mathcal{R}$ as the policy is trained via imitation learning from expert demonstrations.

At each timestep t, the observation $o_t \in \mathcal{S}$ is a quintuple: $o_t = \{I_{f,t}, I_{h,t}, P_t, \mathcal{B}_t, s_t\}$. This includes multi-view vision $I_{f,t}, I_{h,t} \in \mathbb{R}^{224\times224\times3}$, a point cloud $P_t \in \mathbb{R}^{1024\times3}$, a 9-DOF spatial prior $\mathcal{B}_t = [c_{obj}, l, r] \in \mathbb{R}^9$ (centroid, extent, orientation), and proprioception $s_t = [p_{ee,t}, q_{ee,t}, g_t]$.

Reference Point Normalization (RPN): We use the target centroid c_{obj} from $\mathcal{B}_t$ as a dynamic reference point, transforming geometric inputs relative to it:

$$P'_t = P_t - c_{obj}, \quad p'_{ee,t} = p_{ee,t} - c_{obj} \tag{1}$$

This re-centering ensures the normalized centroid is always $\tilde{c}_{obj} = \mathbf{0}$, allowing the policy to learn relative trajectory manifolds that transfer to unseen OOD object poses, as validated in Sec. 4.

Action Space: We define a 7-dimensional continuous action $\Delta a_t = [\Delta p_t, \Delta r_t, g_t] \in \mathbb{R}^7$. The rotation delta Δr_t is derived from the relative quaternion $q_{diff} = q_t^{-1} \otimes q_{t+1}$, mapped to a bounded axis-angle representation to avoid singularities. Actions are standardized via dataset-driven Z-score normalization to preserve the micro-scale trajectory resolution.

3.2 Semantic Query Grounding (SQG)

To obtain the 9-DOF spatial prior $\mathcal{B}_t$ dynamically, we introduce the Semantic Query Grounding (SQG) module. SQG adapts the BoxerNet regression architecture by swapping its generic visual

encoder with a pre-trained SigLIP encoder. Given an atomic instruction tuple (e.g., *laptop, lid, open*), SigLIP extracts a semantic text embedding ϕ_{text}. This embedding acts as the query in a cross-attention mechanism within the box-regression head, conditioning it to attend only to the visual and point cloud features corresponding to the specified part. This effectively resolves instance-level ambiguity, outputting $\mathcal{B}_t$.

To prevent upstream perception errors from destabilizing the policy, we apply Domain Randomization during training. We inject independent Gaussian noise ($\sigma_c \approx 0.01$m) into the predicted centroid, extent, and orientation to produce a noisy estimate $\hat{c}_{obj} \sim \mathcal{N}(c_{obj}, \sigma_c^2\mathbf{I})$. We also apply spatial jittering ($\sigma_{jitter} = 0.003$) and random point dropout (up to 10%) to the normalized point cloud P'_t.

3.3 Cross-Modal Gated Modulation (CMGM)

To suppress geometric attention dispersion, we construct a task-guiding token t_{task} by fusing the text embedding ϕ_{text} and the 9-DOF spatial prior $\mathcal{B}_t$:

$$z_{task} = \text{MLP}_{task}([W_{text}\phi_{text}; W_{bbox}\mathcal{B}_t]) \tag{2}$$

$$t_{task} = \text{MLP}_{fuse}(z_{task}) \tag{3}$$

This task token is prepended to the multi-view image patches T_f, T_h and processed through a 2-layer Transformer encoder. The output slot corresponding to t_{task} yields the task-aware 2D visual representation f_{2D}.

Instead of naive concatenation, f_{2D} is used to dynamically gate the 3D features via Feature-wise Linear Modulation (FiLM). The RPN-normalized point cloud P'_t is processed by a PointNet++ backbone to yield point features F_{geo}. The visual representation f_{2D} predicts affine parameters $[\gamma, \beta] = W_{film}f_{2D} + b_{film}$. The geometric features are modulated element-wise:

$$F'_{geo} = \gamma \odot F_{geo} + \beta \tag{4}$$

3.4 Multimodal Diffusion Policy

The modulated 3D features F'_{geo} are max-pooled into $f_{geo} \in \mathbb{R}^D$ and stacked with f_{2D} to form a two-token unified multimodal sequence $F_{unified} = [f_{geo}; f_{2D}] \in \mathbb{R}^{2\times D}$. To ensure robustness under partial observability, we employ Modality Dropout during training: the entire $F_{unified}$ tensor is zeroed out with probability $p = 0.1$. This forces the UNet to reconstruct trajectories based solely on the Side-Branch Condition.

The Side-Branch Condition $C_{side} = [s'_{prop}; \phi_{text}; \mathcal{B}_t]$ is formed by concatenating the normalized robot proprioceptive state s'_{prop} and the raw task inputs, providing a high-fidelity physical anchor to regularize the trajectory generation. The fully assembled external condition is defined as:

$$\begin{aligned} g_{ext} &= [F_{unified} \cdot m_{drop}; C_{side}], \\ m_{drop} &\sim \text{Bernoulli}(1-p) \end{aligned} \tag{5}$$

The external condition g_{ext} is passed to the Conditional 1D-UNet. Inside the UNet, the diffusion timestep k is encoded into a step embedding $\psi(k)$. The full conditioning vector $g = [\psi(k); g_{ext}]$ is injected into every Convolutional Residual Block via FiLM. The network ϵ_θ is trained via the DDPM objective $\mathcal{L}_{DDPM} = \mathbb{E}_{A,\epsilon,k}[||\epsilon - \epsilon_\theta(A_k, g)||^2]$. During inference, the policy operates via Model Predictive Control (MPC), executing only the first $n_a = 8$ steps before replanning.

Table 1: Quantified Task Definitions and Atomic Success Criteria

Task	Instances	Atomic Train Traj.	Quantitative Success Threshold (Single-Step)
Keyboard	25	2,000	Depth $\geq$ 5 mm on the correct target key
Remote	20	1,600	Button depression depth $\geq$ 4 mm
Laptop	15	1,200	Screen angle $\geq 90^\circ$ or PIN key depth $\geq$ 4 mm
Phone	13	1,040	Numeric/directional button depth $\geq$ 3 mm
Oven	15	1,200	Door opened $\geq 80^\circ$ or fully latched
Safe	13	1,040	Dial angle within $\pm 2^\circ$ of target
Dishwasher	15	1,200	Door open $\geq 70^\circ$, Rack travel $\geq 90\%$



Figure 3: Qualitative results of RN-VLA across representative atomic tasks in Isaac Sim. For each task (keyboard, remote, phone, and oven), we show the initial and final end-effector poses alongside a close-up of the target region, illustrating precise, contact-accurate execution despite densely arranged, visually similar targets.

Table 2: Atomic Single-Step Success Rates with 95% Confidence Intervals

Policy Model	Keyboard	Remote	Laptop	Phone	Oven	Safe	Dishwasher
RT-2	$41.9\% \pm 3.2\%$	$44.5\% \pm 3.1\%$	$41.1\% \pm 3.4\%$	$37.9\% \pm 2.9\%$	$29.5\% \pm 3.4\%$	$24.2\% \pm 2.6\%$	$31.7\% \pm 2.8\%$
CoT-VLA	$44.6\% \pm 2.8\%$	$49.3\% \pm 3.3\%$	$44.1\% \pm 3.1\%$	$42.5\% \pm 2.7\%$	$30.0\% \pm 3.6\%$	$25.8\% \pm 3.0\%$	$33.2\% \pm 2.8\%$
OpenVLA	$47.2\% \pm 3.0\%$	$49.6\% \pm 3.1\%$	$49.5\% \pm 3.1\%$	$43.7\% \pm 2.9\%$	$34.9\% \pm 3.5\%$	$27.0\% \pm 2.9\%$	$35.8\% \pm 3.2\%$
DP3	$27.2\% \pm 1.9\%$	$23.3\% \pm 1.7\%$	$36.8\% \pm 3.2\%$	$21.1\% \pm 1.8\%$	$63.3\% \pm 3.8\%$	$58.3\% \pm 4.8\%$	$65.2\% \pm 3.0\%$
FVP-DP3	$31.6\% \pm 2.4\%$	$29.5\% \pm 2.0\%$	$40.8\% \pm 3.2\%$	$26.9\% \pm 2.2\%$	$73.9\% \pm 4.7\%$	$70.0\% \pm 5.0\%$	$71.4\% \pm 3.7\%$
RN-VLA (Ours)	$\mathbf{80.8\% \pm 2.8\%}$	$\mathbf{78.7\% \pm 3.1\%}$	$\mathbf{81.7\% \pm 2.9\%}$	$\mathbf{77.3\% \pm 4.4\%}$	$\mathbf{83.1\% \pm 2.8\%}$	$\mathbf{83.9\% \pm 3.3\%}$	$\mathbf{85.8\% \pm 3.0\%}$

4 Experiments

We evaluate RN-VLA on atomic precision, component contributions, and zero-shot long-horizon composability.

4.1 Experimental Setup

Platform & Tasks: Experiments are conducted in Nvidia Isaac Sim (PhysX 5, 500 Hz physics, 10 Hz control) and on a 6-DoF UFACTORY xArm6 with a wrist-mounted Intel L515 and static camera. We evaluate 7 household appliance tasks requiring sub-centimeter atomic manipulation, with success criteria defined in Table 2.

Baselines: We compare RN-VLA with end-to-end VLAs (RT-2, OpenVLA, CoT-VLA) and 3D diffusion policies (DP3, FVP-DP3), representing semantic reasoning and geometric control paradigms, respectively.

4.2 Main Results: Atomic-Task Precision

We first evaluate the core single-step atomic execution accuracy across all 7 tasks. Results are reported with 95% binomial confidence intervals based on rigorous randomized trials (Table 2). And Figure 3 provides qualitative evidence across representative tasks, showing that RN-VLA consistently converges to the correct target instance and achieves precise contact even under dense, visually ambiguous layouts such as keyboards and remotes.

The results demonstrate that baseline models suffer heavily from task-specific bias. End-to-end VLA models (e.g., OpenVLA) perform moderately well on semantically dense tasks (Keyboard, Phone) but fail on high-precision geometric tasks (Safe, Oven) due to coordinate scale mismatch. Conversely, 3D Diffusion models (FVP-DP3) show strong geometry performance but fail catastrophically on dense keyboards where semantic target ambiguity overwhelms their holistic point cloud encoders. RN-VLA breaks this dichotomy, achieving $\sim$80%+ atomic precision across *both* semantic and geometric extremes, proving that SQG and Reference Point Normalization successfully bridge the gap.

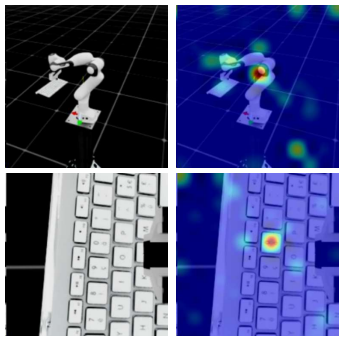

Figure 4: Visualization of task-conditioned attention on the Keyboard task. Left: raw scene observation and a close-up of the target key region. Right: corresponding attention heatmaps, showing that the model's focus concentrates sharply on the correct end-effector–target contact region, with minimal activation on visually similar neighboring keys.

4.3 Diagnostic Analysis: Selection vs. Execution

To isolate the sources of atomic precision, we decompose the single-step accuracy (Acc_{single}) into two conditional metrics: Target Selection Accuracy (Acc_{select}) and Execution Accuracy (Acc_{exec}). We define Acc_{select} as the frequency with which the predicted target matches ground truth, and Acc_{exec} as the conditional success rate given correct selection, such that $Acc_{single} Acc_{select} Acc_{exec}$.As shown in Table 3, RN-VLA dramatically improves Acc_{select} (via the SQG mechanism) and Acc_{exec} (via CMGM and Reference Point Normalization), proving that semantic grounding and spatial execution must be optimized sequentially.

Table 3: Diagnostic Decomposition of Atomic Precision (Avg. across all tasks)

Policy Model	Acc_{select}	Acc_{exec}	Acc_{single}
OpenVLA	$64.2\% \pm 3.1\%$	$64.0\% \pm 3.0\%$	41.1%
FVP-DP3	$58.6\% \pm 2.5\%$	$84.0\% \pm 2.2\%$	49.2%
RN-VLA (Ours)	$\mathbf{94.0\% \pm 1.3\%}$	$\mathbf{86.8\% \pm 1.8\%}$	$\mathbf{81.6\%}$

4.4 Ablation Studies

We conduct modular ablations on the Keyboard (Semantic) and Safe (Geometric) tasks to validate the core architectural design (Table 4).

Table 4: Ablation Analysis of Core Modules (Atomic Accuracy)

Model Variant	**Keyboard**	**Safe**
RN-VLA (Full)	$\mathbf{80.8\% \pm 2.8\%}$	$\mathbf{83.9\% \pm 3.3\%}$
w/o RGB Modality	$76.8\% \pm 2.9\%$	$79.8\% \pm 3.7\%$
w/o Point Cloud Modality	$33.2\% \pm 3.4\%$	$38.7\% \pm 4.8\%$
w/o Cross-Modal Gated Modulation	$56.1\% \pm 3.6\%$	$59.9\% \pm 4.4\%$
Reference Point Norm: workspace-fixed	$53.4\% \pm 3.6\%$	$59.1\% \pm 4.2\%$
Reference Point Norm: none (global)	$16.8\% \pm 2.7\%$	$19.5\% \pm 4.1\%$

The absence of the Point Cloud modality causes catastrophic failure across all tasks, proving 3D structures are the bedrock of precision. Removing the Cross-Modal Gated Modulation dilutes semantic-to-geometric routing, dropping performance significantly. Qualitative attention maps (Figure 4) further corroborate this finding, showing that the Cross-Modal Gated Modulation sharply concentrates activation on the instructed key while suppressing responses from visually identical neighbors. Most importantly, replacing target-centered normalization with a workspace-fixed reference (a non-trivial but static local frame) still trails the full model by $27.4/24.8\%$, and removing normalization entirely trails by a further $36.6/39.6\%$—confirming both that RPN's benefit stems specifically from target-relative (not merely local) coordinates, and that the drop is a precision effect rather than an optimization failure, since the workspace-fixed variant remains clearly trainable.

Action Normalization Ablation: On the Keyboard task, our dataset-driven Z-score standardization yields 80.8%. Switching to conventional min-max normalization drops performance to 48.0%, as rare kinematic outliers in demonstrations compress the bounded scale. Disabling normalization entirely collapses accuracy to 22.0%.

4.5 Spatial Generalization under OOD Shifts

To evaluate coordinate space robustness, we test policies under Out-of-Distribution (OOD) object placements (translation $>$ 15cm, yaw $> 45°$). As shown in Table 5, without Reference Point Normalization, RN-VLA's performance collapses to 7.6%. With it, the policy retains a remarkable 52.6% accuracy (65.1% relative retention), proving that centering the coordinate frame strictly on the target part grants the policy inherent translation invariance.

Table 5: OOD Generalization (Keyboard Task, Oracle BBox)

Model Variant	In-Distribution	Out-of-Distribution	Relative Retention
OpenVLA	$47.2\% \pm 3.0\%$	$12.7\% \pm 2.4\%$	26.9%
RN-VLA (w/o Reference Point Norm)	$16.8\% \pm 2.7\%$	$7.6\% \pm 1.8\%$	45.2%
RN-VLA (Ours, Full)	$\mathbf{80.8\% \pm 2.8\%}$	$\mathbf{52.6\% \pm 3.4\%}$	**65.1%**

4.6 Real-Robot Deployment and Sim-to-Real Gap

We deployed RN-VLA on the physical xArm6 platform to evaluate its execution performance on fine-grained atomic actions. Success or failure in the physical environment is determined strictly via post-hoc visual inspection (aided by fiducial markers for sub-millimeter tasks) serving purely as an evaluation metric rather than an online system component. As detailed in Table 6, taking the simulation performance with predicted bounding boxes ($80.8\% \pm 2.8\%$, aligned with Table 2) as our reference baseline, RN-VLA achieved an end-to-end atomic success rate of $70.0\% \pm 4.8\%$ on the real-world Keyboard task. This yields a modest Sim-to-Real absolute gap of $10.8\% \pm 5.5\%$, demonstrating the robustness of localized 3D control against real-world perceptual noise and domain shift. Furthermore, evaluating under oracle bounding boxes ($88.2\% \pm 3.6\%$) reveals that perfect spatial perception provides an additional $+7.4\%$ performance upper bound. A similar trend is observed on the Remote and Microwave tasks, with Sim-to-Real gaps of $11.1\% \pm 5.4\%$ and $17.8\% \pm 5.9\%$ respectively (Table 6), the latter computed directly between oracle-bbox simulation and real-robot results due to the absence of Sim (Pred BBox) evaluation for this task. Figure 5 shows a representative real-world execution sequence on the xArm6 platform, where RN-VLA executes consecutive atomic actions by first opening the laptop lid and subsequently pressing the target key, confirming that the sub-centimeter precision achieved in simulation transfers effectively to physical hardware.

4.7 Failure Mode Analysis

An analysis of 80 real-world failure trials reveals task-specific limits: 1. **Visual Occlusion (42.5%):** During close-range semantic tasks, the parallel gripper obscures the wrist camera, causing dead-reckoning alignment failures. 2. **Kinematic Singularity (35.0%):** Dominant in precision tasks (e.g., Oven doors, dials) where rigid constraints push the 6-DoF arm to singular limits. 3. **Accumulative Slips (22.5%):** Smooth metallic surfaces cause slight end-effector drifts during continuous contact.

4.8 Zero-Shot Long-Horizon Composability

While our method is explicitly trained to optimize sub-centimeter accuracy for single atomic operations, real-world tasks often manifest as sequential interactions. Because the input definition of our atomic tasks (object, part, action) naturally forms an ordered sequence (e.g., typing 'Hello' translates to an ordered sequence of five precise keypresses), we explore the zero-shot composability of our framework.

By simply concatenating these atomic policies sequentially—without any additional long-horizon fine-tuning, explicit LLM planning, or online state verification—our framework achieves highly

Table 6: Real-Robot Deployment and Sim-to-Real Absolute Gap Analysis across Three Representative Tasks

Task	Sim (Oracle BBox)	Sim (Pred BBox)	Real Robot (Pred BBox)	Sim-to-Real Gap
Keyboard	$88.2\% \pm 3.6\%$	$80.8\% \pm 2.8\%$	$70.0\% \pm 4.8\%$	$10.8\% \pm 5.5\%$
Remote	$85.6\% \pm 4.0\%$	$78.7\% \pm 3.0\%$	$67.6\% \pm 4.3\%$	$11.1\% \pm 5.4\%$
Microwave	$91.0\% \pm 4.2\%$	/	$73.2\% \pm 4.1\%$	$17.8\%^{\dagger} \pm 5.9\%$

† Due to the lack of Sim (Pred BBox) data for the Microwave task, its Sim-to-Real gap is calculated between Sim (Oracle BBox) and Real Robot (Pred BBox).

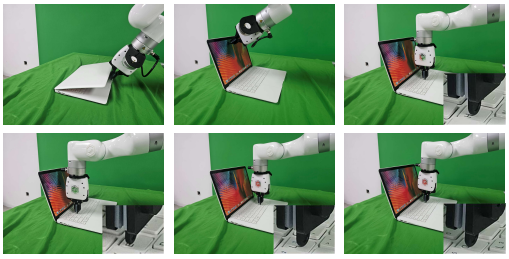

Figure 5: Real-world deployment of RN-VLA on the physical xArm6 platform. The sequence shows the robot opening a laptop lid (top row) and subsequently executing precise keypresses on the laptop keyboard (bottom row), with inset close-ups confirming accurate end-effector contact with the target key.

competitive long-horizon execution. We calculate the theoretical long-horizon success rate (P_{expected}) across a T-step sequence based on the single-step accuracy (Acc_{single}):

$$P_{\text{expected}} = (Acc_{\text{single}})^T \tag{6}$$

We compare this expected rate against the actual measured long-horizon success rate (P_{actual}) achieved by running RN-VLA sequentially on full 7-step trajectories, benchmarking it against the best monolithic baseline (Table 7). We note that Eq. 6 assumes step-wise independence, which likely underestimates Pactual due to inter-step pose correlation in contiguous execution; we therefore treat Pexpected as a conservative lower-bound reference rather than a precise theoretical prediction.

Table 7: Zero-Shot Long-Horizon Composability vs. Best Baselines (Avg. $T = 7$ steps)

Task Category	Acc_{single}	P_{expected}	P_{actual}	Best Baseline (P_{actual})
Semantics-dominant	79.6%	20.2%	22.5%	13.4% (OpenVLA)
Precision-dominant	84.3%	30.3%	33.6%	22.3% (FVP-DP3)

The empirical results show that P_{actual} slightly exceeds P_{expected} (22.5% vs. 20.2% on semantics-dominant tasks; 33.6% vs. 30.3% on precision-dominant tasks), as contiguous multi-step execution benefits from pose pre-alignment compared to randomized single-step resets. Crucially, without any specialized long-horizon training, the pure sequential composition of RN-VLA's highly reliable atomic skills substantially outperforms the strongest state-of-the-art baselines, achieving $1.68\times$ and $1.51\times$ their success rates, respectively. This confirms our core hypothesis: solving the atomic precision bottleneck effectively mitigates error compounding from the ground up, providing a powerful and composable foundation for long-horizon manipulation.

5 Conclusion

We presented RN-VLA, a multimodal diffusion policy that bridges semantic grounding and geometric precision for fine-grained robotic manipulation. By addressing instance ambiguity, geometric dispersion, and coordinate scale mismatch, RN-VLA achieves robust object grounding and precise action generation within a unified framework. Across seven household tasks in simulation and real-world xArm6 deployments, RN-VLA achieves over 80% atomic success while consistently outperforming representative VLA and 3D diffusion baselines. By reducing atomic execution errors, RN-VLA further enables zero-shot long-horizon task composability, approximately doubling the success rate of monolithic baselines without requiring explicit high-level LLM planning.

AI USE STATEMENT

During the preparation and revision of this manuscript, we used generative AI tools to assist with manuscript editing, organization, experimental design discussion, and interpretation/presentation of experimental results. AI tools were not used to generate experimental data or to determine the reported results. All AI-assisted content, methodological suggestions, and analyses were reviewed and independently verified by the authors. The authors remain fully responsible for the final manuscript, including all methodological claims, experimental results, interpretations, and conclusions.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

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.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

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. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Murtaza Dalal, Tarun Chiruvolu, Devendra Chaplot, and Ruslan Salakhutdinov. Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks. *arXiv preprint arXiv:2405.01534*, 2024.

Daniel DeTone, Tianwei Shen, Fan Zhang, Lingni Ma, Julian Straub, Richard Newcombe, and Jakob Engel. Boxer: Robust lifting of open-world 2d bounding boxes to 3d. *arXiv preprint arXiv:2604.05212*, 2026.

Coline Devin, Pieter Abbeel, Trevor Darrell, and Sergey Levine. Deep object-centric representations for generalizable robot learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7111–7118. IEEE, 2018.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Yiguo Fan, Pengxiang Ding, Shuanghao Bai, Xinyang Tong, Yuyang Zhu, Hongchao Lu, Fengqi Dai, Wei Zhao, Yang Liu, Siteng Huang, et al. Long-vla: Unleashing long-horizon capability of vision language action model for robot manipulation. *arXiv preprint arXiv:2508.19958*, 2025.

Peter Florence, Lucas Manuelli, and Russ Tedrake. Self-supervised correspondence in visuomotor policy learning. *IEEE Robotics and Automation Letters*, 5(2):492–499, 2019.

Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023.

Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pp. 694–710. PMLR, 2023.

Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018.

Chengkai Hou, Yanjie Ze, Yankai Fu, Zeyu Gao, Songbo Hu, Yue Yu, Shanghang Zhang, and Huazhe Xu. 4d visual pre-training for robot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8451–8461, 2025.

Haoxu Huang, Fanqi Lin, Yingdong Hu, Shengjie Wang, and Yang Gao. Copa: General robotic manipulation through spatial constraints of parts with foundation models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9488–9495. IEEE, 2024.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: Robot manipulation with multimodal prompts. 2023.

Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *2021 IEEE international conference on robotics and automation (ICRA)*, pp. 4613–4619. IEEE, 2021.

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

Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Hao Li, Zi-han Ding, Shuai Yang, Yilun Chen, Yang Tian, Xiaolin Hu, Tai Wang, Dahua Lin, Feng Zhao, Si Liu, et al. Robointer: A holistic intermediate representation suite towards robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 113314–113381, 2026.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

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

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 *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025.

Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

Yutao Ouyang, Jinhan Li, Yunfei Li, Zhongyu Li, Chao Yu, Koushil Sreenath, and Yi Wu. Long-horizon locomotion and manipulation on a quadrupedal robot with large language models. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11157–11164. IEEE, 2025.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Chuer Pan, Brian Okorn, Harry Zhang, Ben Eisner, and David Held. Tax-pose: Task-specific cross-pose estimation for robot manipulation. In *Conference on Robot Learning*, pp. 1783–1792. PMLR, 2023.

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

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

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.

Yide Shentu, Philipp Wu, Aravind Rajeswaran, and Pieter Abbeel. From llms to actions: Latent codes as bridges in hierarchical robot control. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8539–8546. IEEE, 2024.

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

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*, 2022.

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.

Ajay Kumar Tanwani, Pierre Sermanet, Andy Yan, Raghav Anand, Mariano Phielipp, and Ken Goldberg. Motion2vec: Semi-supervised representation learning from surgical videos. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2174–2181. IEEE, 2020.

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. *arXiv preprint arXiv:2405.12213*, 2024.

Thomas Tian, Chenfeng Xu, Masayoshi Tomizuka, Jitendra Malik, and Andrea Bajcsy. What matters to you? towards visual representation alignment for robot learning. In *International Conference on Learning Representations*, volume 2024, pp. 29609–29628, 2024.

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

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Jianke Zhang, Yanjiang Guo, Xiaoyu Chen, Yen-Jen Wang, Yucheng Hu, Chengming Shi, and Jianyu Chen. Hirt: Enhancing robotic control with hierarchical robot transformers, 2025. URL https://arxiv.org/abs/2410.05273.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1702–1713, 2025.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.

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 *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

A Appendix

This supplementary document provides comprehensive architectural specifications, complete mathematical formulations, dataset characteristics, simulation and hardware environment setups, hyperparameter configurations, extended experimental diagnostic results, per-step failure mode analyses, and real-robot deployment details for the **RN-VLA** (Reference-Point Normalized Vision-Language-Action) framework.

Section A: Extended Model Architecture & Mathematical Formulations

A.1 Perception Frontend & Task-Conditioned Query (TCQ)

The perception frontend bridges unstructured raw sensory observations and 3D spatial priors. We repurpose BoxerNet, an open-vocabulary 3D bounding box regressor, by replacing its original RGB feature encoder with a pre-trained SigLIP (`siglip-base-patch16-224`) vision-language backbone. Given synchronized fixed-camera RGB image $I_{f,t} \in \mathbb{R}^{224\times224\times3}$ and wrist hand-eye camera RGB image $I_{h,t} \in \mathbb{R}^{224\times224\times3}$, SigLIP extracts spatial patch token sequences $T_f, T_h \in \mathbb{R}^{196\times768}$.

The textual sub-goal query $s_i = \{\text{object: } o, \text{part: } p, \text{action: } a\}$ generated by the LLM planner is processed by the SigLIP text encoder to yield sub-goal text embedding $\phi_{\text{text}} \in \mathbb{R}^{768}$. By conditioning box queries directly on joint visual-textual embeddings, the network isolates the specific target part among visually identical candidate parts (e.g., a single key on a cluttered keyboard). The output 9-DOF spatial bounding box is formulated as:

$$\mathcal{B}_t = [c_{\text{obj}}, l, r] \in \mathbb{R}^9, \quad c_{\text{obj}} = (x, y, z) \in \mathbb{R}^3, \quad l = (dx, dy, dz) \in \mathbb{R}^3, \quad r = (\theta_x, \theta_y, \theta_z) \in \mathbb{R}^3 \tag{7}$$

where c_{obj} denotes the target centroid in world coordinates, l represents spatial bounding box dimensions, and r specifies Euler orientation angles.

A.2 Task-Guided Multimodal Fusion & FiLM Bridge

To fuse task semantics with multi-view visual tokens, the sub-goal text embedding ϕ_{text} and spatial prior $\mathcal{B}_t$ are projected into a unified embedding space of dimension $D = 512$:

$$z_{\text{task}} = \text{MLP}_{\text{task}}([W_{\text{text}}\phi_{\text{text}}; W_{\text{bbox}}\mathcal{B}_t]) \in \mathbb{R}^{512} \tag{8}$$

Visual patch streams T_f and T_h are projected to dimension $D = 512$ and added with learnable view embeddings $e_{\text{fixed}}, e_{\text{hand-eye}} \in \mathbb{R}^{1\times512}$. The task guiding token $t_{\text{task}} = \text{MLP}_{\text{fuse}}(z_{\text{task}})$ is prepended to form input token sequence $X = [t_{\text{task}}, t_f, t_h] \in \mathbb{R}^{393\times512}$. This sequence is processed by a 2-layer Transformer Encoder (8 heads, hidden dimension 512, feedforward dimension 2048, dropout 0.1). The output token at index 0 is extracted and normalized to yield the task-aware 2D visual feature $f_{2\text{D}} = \text{LayerNorm}(Y_0) \in \mathbb{R}^{512}$.

For 3D point cloud processing, the Anchor-normalized point cloud $P'_t = P_t - c_{\text{obj}} \in \mathbb{R}^{1024\times3}$ passes through PointNet++ set abstraction layers to extract per-point features $F_{\text{geo}} \in \mathbb{R}^{128\times128}$. The 2D feature $f_{2\text{D}}$ generates channel-wise affine parameters via a linear layer initialized with bias $b_{\text{film}} = [1, 0]$:

$$[\gamma, \beta] = W_{\text{film}} f_{2\text{D}} + b_{\text{film}}, \quad \gamma, \beta \in \mathbb{R}^{1\times128}, \quad F'_{\text{geo}} = \gamma \odot F_{\text{geo}} + \beta \tag{9}$$

Max-pooling over points yields task-aware 3D feature $f_{\text{geo}} \in \mathbb{R}^{512}$. The multimodal conditioning token sequence is assembled as $F_{\text{unified}} = [f_{\text{geo}}; f_{\text{2D}}] \in \mathbb{R}^{2\times 512}$.

A.3 Action Expert: Conditional 1D-UNet & Diffusion Generation

The low-level policy models conditional distribution $p(A \mid O)$ over trajectory horizon $H = 16$. The continuous action vector $\Delta a_t = [\Delta p_t, \Delta r_t, g_t] \in \mathbb{R}^7$ consists of relative translation $\Delta p_t \in \mathbb{R}^3$, relative axis-angle orientation $\Delta r_t \in \mathbb{R}^3$ (derived from relative quaternion $q_{\text{diff}} = q_t^{-1} \otimes q_{t+1}$), and binary gripper state $g_t \in [0, 1]$.

The architecture is a 1D Conditional UNet with 4 residual downsampling blocks and 4 residual upsampling blocks (channel widths: 128, 256, 512, 1024). The side-branch condition $C_{\text{side}} = [s_t; z_{\text{text}}; z_{\text{bbox}}] \in \mathbb{R}^{d_{\text{side}}}$ concatenates robot proprioceptive state $s_t \in \mathbb{R}^8$ with sub-goal embeddings. During training, Modality Dropout ($p = 0.15$) zeroes out F_{unified} with probability p to enforce side-branch regularization. Timestep $k \in [1, K]$ embedding $\psi(k) \in \mathbb{R}^{128}$ is combined with external conditions $g_{\text{ext}} = [F_{\text{unified}} \cdot m_{\text{drop}}; C_{\text{side}}]$ to modulate every UNet residual block via FiLM layers.

Section B: Task Specifications & Simulation/Real Setups

B.1 Dataset Scale & Quantified Task Definitions

The dataset contains 5,800 long-horizon trajectories and 9,280 atomic single-step trajectories across 116 physical asset models. Table 8 lists the quantified verification thresholds and operational constraints for all primary tasks.

Table 8: Detailed task specifications, dataset trajectory counts, maximum step budgets, and physical quantitative success criteria.

Task Category	Assets	Long / Atomic Traj.	Max Steps / Sub-goal	Quantitative Success Thresholds
Keyboard	25	1,250 / 2,000	150 steps	Key depression depth ≥ 5 mm in exact sequence
Remote	20	1,000 / 1,600	150 steps	Power button depth > 4 mm; correct channel sequence
Laptop	15	750 / 1,200	150 steps	Lid opening angle $\geq 90^\circ$; PIN depth ≥ 4 mm
Phone	13	650 / 1,040	150 steps	Numeric button depth ≥ 3 mm; app navigation executed
Oven	15	750 / 1,200	Full Horizon	Door open $\geq 80^\circ$; dial error $\leq 2^\circ$; latched
Safe	13	650 / 1,040	Full Horizon	Rotary dial accuracy $\leq \pm 2^\circ$; deadbolt force ≥ 3 N
Dishwasher	15	750 / 1,200	Full Horizon	Door open $\geq 70^\circ$; rack linear travel $\geq 90\%$ stroke

B.2 Simulation & Hardware Configurations

Simulation environments are built in Nvidia Isaac Sim 2023.1.1 (PhysX 5 engine) operating at 500 Hz physics simulation frequency and 10 Hz action control frequency. Static friction coefficient $\mu = 0.5$, dynamic friction $\mu = 0.5$, contact restitution = 0.1, solver position iterations = 8. Camera specifications: (1) Overhead fixed camera (1920×1080 RGB-D downsampled to 224×224, FOV = 60°); (2) Wrist hand-eye camera (1280×720 RGB-D downsampled to 224×224, FOV = 70°).

Section C: Hyperparameter Configurations & Training Protocols

Table 9 lists the hyperparameters across all modules of RN-VLA.

Section D: Prompts & Closed-Loop Verification Protocols

D.1 High-Level LLM Planner Prompting (Chain-of-Thought)

Below is the system prompt and structured JSON schema used by the LLM planner:

Listing 1: System Prompt and Chain-of-Thought template for LLM semantic planner.

```
SYSTEM PROMPT:
Decompose high-level instructions into atomic sub-goals for a
```

Table 9: Hyperparameter configurations for perception frontend, multimodal fusion, and action diffusion policy.

Parameter / Module	Value	Parameter / Module	Value
SigLIP Backbone	siglip-base-patch16-224	Optimizer & Weight Decay	AdamW, 1×10^{-4}
Transformer Fusion Layers	2 layers, 8 heads	Learning Rate (Policy)	1×10^{-4} (Cosine schedule)
Fusion Hidden Dim (D)	512	Learning Rate (BoxerNet)	2×10^{-5}
PointNet++ Sample Points	$1024 \rightarrow 512 \rightarrow 128$	Batch Size	64
PointNet++ Radii	[0.02, 0.05, 0.10] m	Training Epochs	50 epochs
Action Horizon (H)	16 steps	Modality Dropout Prob (p)	0.15
Execution Horizon (n_a)	8 steps (MPC)	Centroid Noise (σ_c)	0.01 m (10 mm)
DDPM Sampling Steps (K)	100 steps	Point Cloud Jitter (σ_{jitter})	0.003 m (3 mm)
Flow Matching Steps (K)	20 steps	Point Dropout Rate	10%

```
6-DoF robot.

CONSTRAINTS:
1. Laptop: Open "lid" before accessing keys ("key_X").
2. Remote: Power "on" before pressing buttons ("btn_X").
3. Oven: Open "door" -> turn "dial" -> close "door" -> press "start".

OUTPUT SCHEMA (JSON):
[{"obj": str, "part": str,
"action": "press"|"flip"|"rotate"|"pull"|"push"}]

EXAMPLE:
Input: "Type 'H' on laptop."
CoT: Laptop is closed; must open lid before keypress.

JSON Output:
[
  {"obj": "laptop", "part": "lid", "action": "flip"},
  {"obj": "laptop", "part": "key_H", "action": "press"}
]
```

D.2 Closed-Loop State Verification Protocol

Semantic State Verification: A GPT-4o vision query evaluates pre-execution image I_{pre} and post-execution image I_{post} with prompt: *"Compare image A (before) and image B (after). Has the [object] [part] completed the [action] transition? Answer ONLY 'TRUE' or 'FALSE'."*

Geometric Precision Verification: Physical micro-fiducial markers are placed on precision target surfaces (e.g., dial rim, key cap). Multi-view camera feeds track 3D rigid pose change $\Delta T \in \text{SE}(3)$. Verification outputs $\mathcal{V} =$ True if translation displacement $||\Delta p|| \geq \delta_{\text{thresh}}$ or angular change $|\Delta\theta| \geq \theta_{\text{thresh}}$ matching parameters in Table 8.

Section E: Extended Quantitative Results & Analysis

E.1 Per-Step Long-Horizon Success Degradation

Table 10 details empirical step-by-step success rates across execution horizons $t = 1 \ldots 7$ for all 7 evaluation tasks.

E.2 Robustness to Upstream Perception Noise

Table 11 evaluates single-step accuracy on the Keyboard task under varying Gaussian noise standard deviations σ_c injected into target bounding box centroids c_{obj}.



Figure 6: Representative atomic manipulation tasks in real-world and simulation environments. For each task, an overview image and a close-up observation are shown. The benchmark includes diverse household objects and fine-grained interaction scenarios.

SECTION F: HARDWARE SETUP & DETAILED FAILURE DIAGNOSTICS

F.1 HARDWARE SETUP

Hardware experiments utilize a 6-DoF UFACTORY xArm6 equipped with a parallel-jaw gripper. Sensing includes a wrist-mounted Intel RealSense L515 LiDAR depth camera (hand-eye view) and an overhead static RGB-D camera. End-effector delta trajectories predicted at 10 Hz are converted to joint motor commands using a 100 Hz Operational Space Impedance Controller (OSIC).

Table 10: Empirical step-by-step success rates (%) across execution horizons $t = 1 \dots 7$ in simulation environments.

Task Name	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Full Task Success
Keyboard	88.2%	81.4%	74.6%	68.0%	59.2%	41.6%	27.4%	**27.4% ± 4.2%**
Remote	86.5%	78.2%	71.0%	62.4%	51.8%	35.2%	22.7%	**22.7% ± 4.4%**
Laptop	89.0%	82.1%	73.5%	64.2%	52.0%	36.8%	23.7%	**23.7% ± 5.3%**
Phone	84.1%	75.3%	66.8%	57.2%	45.0%	30.1%	19.2%	**19.2% ± 5.7%**
Oven	91.2%	85.6%	78.0%	70.4%	61.2%	51.5%	42.3%	**42.3% ± 5.6%**
Safe	88.5%	80.2%	72.4%	63.1%	52.6%	42.0%	33.4%	**33.4% ± 6.1%**
Dishwasher	87.0%	79.5%	71.2%	61.8%	50.4%	40.2%	31.2%	**31.2% ± 5.1%**

Table 11: Single-step accuracy on Keyboard task under varying centroid perturbation noise σ_c.

Centroid Noise σ_c	0 mm (Oracle)	5 mm	10 mm (Default)	15 mm	20 mm	25 mm
Single-Step Acc (%)	80.1% ± 2.8%	78.4% ± 3.0%	75.2% ± 3.2%	68.9% ± 3.5%	54.1% ± 4.1%	38.2% ± 4.5%

F.2 Failure Case Analysis

Analysis of 80 failed trials collected during real-robot deployment reveals three primary failure modes:

- **Close-range Gripper Occlusion (42.5% of failures):** When the gripper descends within 5–8 mm of micro-keys or dials, the wrist camera's field-of-view is occluded by gripper fingers, causing target visual feature collapse and local displacement drift.
- **Kinematic Singularities & Joint Limits (35.0% of failures):** Large-range motions (e.g., pulling microwave doors through 80° or executing full 360° dial rotations) force the 6-DoF manipulator near joint limits, triggering joint-space singularities and controller safe-halts.
- **Accumulative Surface Slips (22.5% of failures):** Smooth metallic dials and glass surfaces cause micro-slips under sustained contact, accumulating positional drift over multi-step execution chains.