GRAM-VLA: GIVING VISION-LANGUAGE-ACTION MODELS A SENSE OF 3D SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied intelligence is increasingly moving toward more complex manipulation tasks. Compared with single-arm tasks, dual-arm tasks require coordination of the two arms' spatial positions and temporal order. This places greater demands on the model's joint vision-language understanding and spatial perception. Most Vision-Language-Action (VLA) models excel at dual-arm tasks by leveraging large-scale 2D vision-language pre-training, but their reliance on RGB images limits spatial reasoning critical for real-world interaction. Adding 3D information can improve spatial perception. However, without extra large-scale 2D–3D data pre-training, it remains challenging to fully use 3D information for action generation while remaining consistent with pretrained 2D semantics. To address this challenge, we propose GRAM-VLA, a 3D-enhanced dual-arm VLA model without additional large-scale 2D–3D pretraining. GRAM-VLA keeps the pretrained vision-language backbone (VLM) frozen and adds a lightweight depth encoder to form an independent RGB–Depth dual-branch architecture. It establishes token-level spatiotemporal RGB–Depth alignment and alternately injects language, RGB, and depth conditions into the action generation backbone, allowing depth information to fully participate in action generation. Furthermore, we use four-modal GRAM alignment to mitigate multimodal semantic shift caused by the newly introduced depth information. On eight real-robot tasks aligned with the RoboTwin 2.0 simulation protocol, GRAM-VLA achieves an average success rate of 53.6%, outperforming both 2D- and 3D-based models. In controlled experiments, GRAM-VLA outperforms baseline-RDT by 40.0 and 72.5 percentage points under appearance and height shifts, respectively, and by 22.5 percentage points in long-horizon task completion.

1 INTRODUCTION

The goal of embodied intelligence is to free robots from fixed task programs and enable autonomous perception, decision-making, and execution through interaction with the physical environment. Vision-Language-Action (VLA) models use visual observations and language instructions to guide robot action generation and have become an important paradigm for embodied intelligence. Early VLA research mainly focused on single-arm manipulation, achieving substantial progress in language instruction understanding, multi-task manipulation, and scene generalization Lynch & Sermanet (2020); Jang et al. (2022); Shridhar et al. (2022; 2023); Mees et al. (2022); Brohan et al. (2022). As robotic manipulation moves toward more complex and human-like tasks, recent VLA research has increasingly extended to dual-arm manipulation Shi et al. (2025); Tang et al. (2026); Fan et al. (2026); Wang et al. (2026). Compared with single-arm manipulation, dual-arm tasks require finer coordination of the spatial positions, timing, and movements of both arms. This places greater demands on the model's joint understanding of visual scenes and language instructions for accurate task-goal identification and coordinated action planning.

To support fine-grained dual-arm manipulation, current dual-arm VLA models Black et al. (2024); Wen et al. (2025b;a); Bjorck et al. (2025); Liu et al. (2025a); Kim et al. (2025); Im et al. (2026) commonly follow a "vision-language pre-training + robot-data fine-tuning" paradigm. They first learn transferable 2D semantic priors from large-scale RGB image-text pre-training Beyer et al. (2024); Liu et al. (2023); Wang et al. (2024); Karamcheti et al. (2024) and then fine-tune on robot demonstrations for action generation. However, RGB-only observations require the model to infer

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.

object height, distance, and relative position implicitly from 2D appearance, which can introduce spatial ambiguity in height variation, occlusion, and fine-grained manipulation. PointVLA Li et al. (2026a) shows poor generalization of 2D policies to unseen object heights, while SpatialVLA Qu et al. (2025) reports that explicit 3D spatial information improves the average success rate from 45.4% to 70.5%–72.7% across four fine-grained manipulation tasks. These results highlight the limitations of RGB-only observations for spatial reasoning.

To improve spatial perception, some VLA models replace RGB images with 3D information such as point clouds and depth maps, allowing the model to directly access explicit geometric information Ze et al. (2024); Chen et al. (2023a); Ke et al. (2024); Garcia et al. (2025); Yang et al. (2025); Zhu et al. (2025). However, these models cannot directly inherit the 2D semantic priors learned from large-scale RGB image-text pre-training. Large-scale paired 3D observations and language instructions are also scarce, making 3D pre-training costly. At the same time, point-cloud and depth-map representations mainly capture geometric information and may perform poorly on tasks that rely on color or texture recognition.

To preserve 2D appearance semantics while enabling the model to understand 3D geometric information, using both as conditions for action generation is a reasonable approach to improving fine-grained spatial manipulation and generalization in VLA models. However, existing 3D-enhanced VLA models often face a trade-off between fully using 3D geometric information and preserving pretrained 2D semantic priors. On the one hand, direct interaction between newly introduced geometric representations and VLM representations may disrupt pretrained 2D semantics and degrade manipulation performance. BridgeVLA Li et al. (2026b) shows that directly fusing per-pixel 3D position features with 2D image features reduces the average success rate from 88.2% to 56.2%, while its solution requires additional heatmap pretraining, increasing the training cost. On the other hand, some methods restrict the injection of 3D geometric features to reduce interference with pretrained 2D semantics. PointVLA Li et al. (2026a) injects point-cloud features only into selected action layers. This reduces interference, but the geometric information cannot fully participate in action generation.

To address this trade-off, we propose GRAM-VLA, a 3D-enhanced dual-arm VLA model built on RDT-1B Liu et al. (2025b). GRAM-VLA introduces depth maps to provide explicit 3D geometric information. Specifically, we freeze the pretrained vision and language encoders and add an independent lightweight depth encoder, thereby avoiding direct interaction between newly introduced depth features and VLM features. However, RGB and depth provide complementary information about the same scene and jointly guide action generation, while independent encoding alone does not explicitly establish the connection between them. We therefore partition synchronized RGB and depth map using the same spatial grid to establish one-to-one token-level spatial correspondence. Building on this spatial correspondence, we further establish temporal correspondence by constructing Tri-modal Cycle-Phase Positional Encoding (TCPE) and adding it to the language, RGB, and depth condition sequences before they are alternately injected into successive DiT blocks. TCPE encodes a shared temporal relation among the three conditions while preserving their modality-specific positions. At the same time, this alternating injection maintains the separation between depth features and VLM features while allowing 3D geometric information to fully participate in action generation.

However, the independent dual-branch architecture only prevents direct interference with the pretrained semantic space, while conventional diffusion loss mainly supervises action prediction. Neither explicitly constrains the semantic consistency of the newly introduced 3D geometric representations with language, RGB, and expert actions. To address this issue, we introduce four-modal global representation alignment across language, RGB, depth map, and clean expert actions. We map their global representations into a shared space and use the Gramian volume Cicchetti et al. (2025) spanned by the four representations to measure the degree of multimodal alignment. Based on this measure, we introduce a symmetric contrastive learning objective to strengthen cross-modal semantic consistency. This objective is used only during training and adds no inference cost.

We conduct extensive experiments to validate the effectiveness of GRAM-VLA. First, we transfer eight RoboTwin 2.0 Chen et al. (2025) tasks from simulation to a real dual-arm robot. GRAM-VLA achieves the best results on 6/8 tasks. It preserves color recognition and complex semantic understanding while showing clear advantages on geometry-sensitive tasks involving scale, height, occlusion, and stacking. Next, we conduct controlled experiments against the RDT baseline. In

the standard setting, GRAM-VLA performs comparably to RDT (85.0% vs. 87.5%). Under appearance and height changes, GRAM-VLA outperforms RDT by 40.0 and 72.5 percentage points, respectively. On complex long-horizon spatial manipulation tasks, GRAM-VLA further improves the average task completion rate by 22.5 percentage points over RDT. Our main contributions are summarized as follows:

1. **We propose a four-modal global representation alignment mechanism.** It uses the Gramian volume spanned by language, RGB, depth, and expert-action representations to measure multimodal alignment, together with a contrastive objective to enhance cross-modal semantic consistency.
2. **We build an RGB–Depth dual-branch VLA architecture.** It uses an independent lightweight depth branch, token-level RGB-D alignment, and alternating multimodal conditioning to enable 3D geometric information to fully participate in action generation while avoiding interference with pretrained 2D semantics.
3. **We conduct systematic evaluations on a real dual-arm robotic platform.** The experiments validate the effectiveness of GRAM-VLA in preserving 2D semantics, enhancing 3D geometric perception, improving environmental generalization, supporting real-robot deployment, and enabling complex spatial manipulation.

2 Related Work

To provide the 3D geometric information needed for action generation, recent studies increasingly incorporate 3D geometric information into existing VLA architectures. However, introducing geometric information does not necessarily improve performance. Previous studies show that directly fusing 3D position features with 2D image features, adding depth tokens to a pretrained vision-language backbone, or allowing direct interaction between depth features and the pretrained vision-language branch may reduce manipulation performance Li et al. (2026b); Chen et al. (2026); Yuan et al. (2025). These results indicate a risk of interference between newly introduced 3D geometric information and pretrained priors. Therefore, preserving pretrained 2D semantic priors is a prerequisite for 3D enhancement.

Based on this premise, existing 3D-enhanced VLA methods can be grouped into two main categories. The first category follows a "spatial pre-adaptation + task fine-tuning" paradigm. Before target-task training, these methods use additional 2D–3D data to pretrain the relationships among 3D information, 2D semantics, and robot actions, such as SpatialVLA, VIPA-VLA and UniDex-VLA Qu et al. (2025); Feng et al. (2026); Zhang et al. (2026). Other methods use auxiliary objectives for spatial adaptation of 3D geometric representations and pretrained VLMs. These objectives include geometric reconstruction and target heatmap prediction, such as Lift3D-VLA and BridgeVLA Liu et al. (2026); Li et al. (2026b). A common feature of this category is the need for an additional pretraining stage beyond downstream task fine-tuning, which increases training complexity and computational cost.

The second category aims to incorporate 3D geometric information while minimizing interference with the pretrained VLM. PointVLA Li et al. (2026a) injects point-cloud features only into selected action layers. This limits the interaction between depth information and action representations. GeoVLA Sun et al. (2025) aggregates point-cloud information into a single anchor token. The action network no longer directly accesses geometry tokens at each spatial location. As a result, this category prevents 3D geometric information from fully participating in action generation.

3 Methodology

Without large-scale 2D–3D data pretraining, effectively integrating depth information presents two main challenges. First, depth information should fully participate in multi-layer action modeling. Second, the newly introduced depth representations should not disturb the existing pretrained 2D semantics. We summarize these two goals as the full participation and stable participation of depth information. Based on these goals, we propose GRAM-VLA. The overall architecture is shown in Fig. 1.

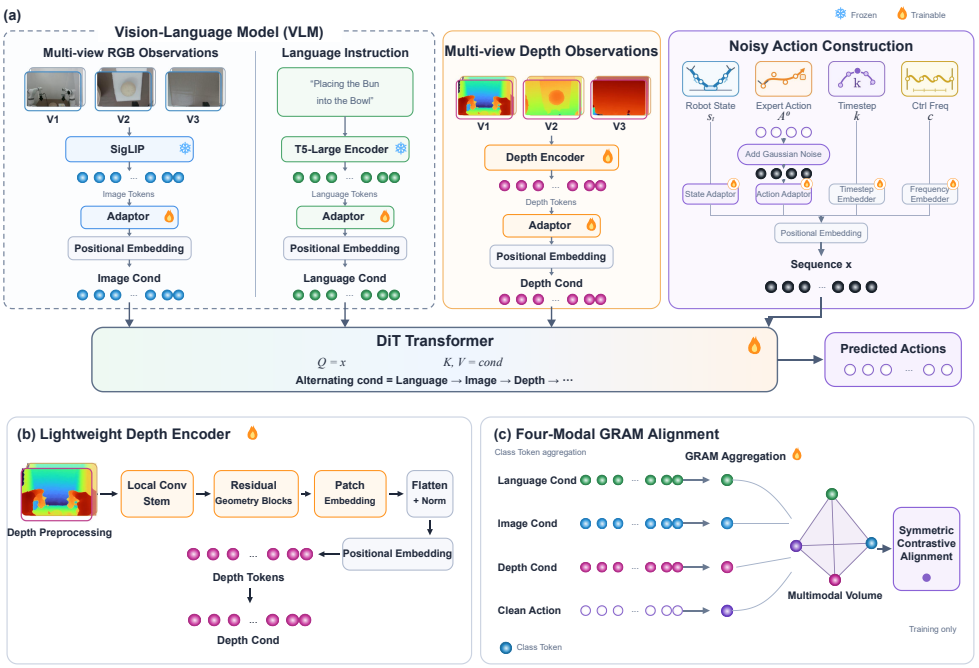

Figure 1: Overall architecture of GRAM-VLA. **(a)** The robot state, noisy action chunk, diffusion timestep, and control frequency form the denoising input. RGB, language, and depth are projected to a shared feature dimension and used as conditions. The "Positional Embedding" blocks in these three conditioning branches implement Tri-modal Cycle-Phase Positional Encoding (TCPE), with temporal and view encodings additionally applied to RGB and depth. The DiT alternately injects language, RGB, and depth conditions. **(b)** Architecture of the lightweight depth encoder. **(c)** Gramian volume measures the global relationships among language, RGB, depth, and clean expert actions to reduce semantic shift across heterogeneous modalities. Snowflakes and flames denote frozen and trainable modules, respectively.

3.1 Task Formulation

As shown in Fig. 1(a), GRAM-VLA formulates robot control as a continuous action chunk generation problem conditioned on a language instruction and multimodal observations. Given a language instruction ℓ, the model receives observation $\mathbf{o}_t = (\mathcal{X}_t, \mathcal{D}_t, \mathbf{s}_t, c)$ at time t. We represent the multi-view RGB and depth observations as

$$\begin{aligned} \mathcal{X}_t &= \{\mathbf{X}_\tau^v \mid \tau = t - T_o + 1, \ldots, t;\ v = 1, \ldots, V\}, \\ \mathcal{D}_t &= \{\mathbf{D}_\tau^v \mid \tau = t - T_o + 1, \ldots, t;\ v = 1, \ldots, V\}. \end{aligned} \tag{1}$$

Here, $\mathbf{X}_\tau^v$ and $\mathbf{D}_\tau^v$ denote the RGB image and depth map from view v at time τ. T_o is the observation history length, V is the number of cameras, $\mathbf{s}_t$ is the current robot state, and c is the control frequency. All observations use the same timestamp. Each language instruction and multimodal observation has a corresponding future expert action sequence. We define the clean expert action chunk as $\mathbf{A}_t^0 = [\mathbf{a}_t, \mathbf{a}_{t+1}, \ldots, \mathbf{a}_{t+T_a-1}] \in \mathbb{R}^{T_a \times D_a}$. Here, T_a is the action chunk length, and D_a is the action dimension. The superscript 0 denotes the clean action without diffusion noise. Based on these definitions, GRAM-VLA learns the continuous action chunk distribution $\mathbf{A}_t^0 \sim p_\theta(\cdot \mid \ell, \mathbf{o}_t)$ under the language instruction and current multimodal observation.

3.2 2D–3D Dual-Branch Encoding for Full Participation

Based on the task formulation above, GRAM-VLA needs to encode multimodal observations for action generation. To avoid interference between the newly introduced depth representations and

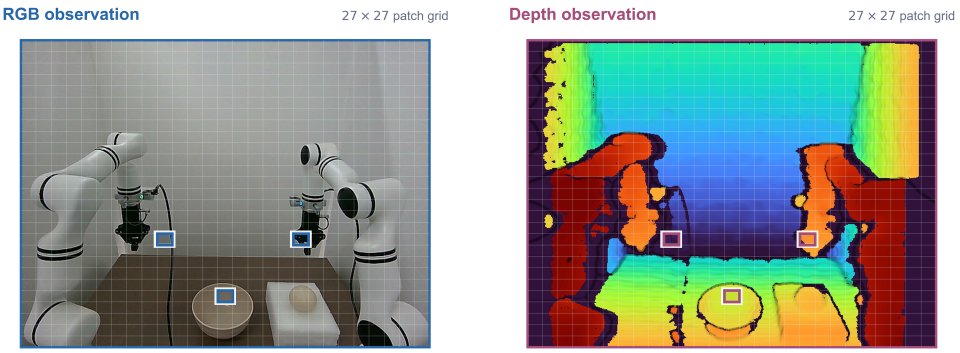

Figure 2: Token-level Correspondence between RGB and Depth.

the pretrained VLM semantics, we build an RGB–Depth dual-branch encoding structure. As shown in Fig. 1(a), for the synchronized RGB–Depth observation $(\mathbf{X}_\tau^v, \mathbf{D}_\tau^v)$ from view v at time τ, the two visual branches generate

$$\mathbf{Z}_{\tau,v}^x = E_x(\mathbf{X}_\tau^v), \qquad \mathbf{Z}_{\tau,v}^d = E_d(\mathbf{D}_\tau^v), \qquad \mathbf{Z}_{\tau,v}^x, \mathbf{Z}_{\tau,v}^d \in \mathbb{R}^{N_p \times d_v}. \tag{2}$$

Here, E_x is the frozen RGB encoder, and E_d is the lightweight depth encoder. N_p is the number of tokens in a single-frame visual observation, and d_v is the token feature dimension.

Depth Encoding Branch. As shown in Fig. 1(b), we first preprocess the depth maps. For each camera, we set a depth range based on the typical object distances in the task, and use it to clip and normalize the depth map. We then resize it to the RGB resolution using nearest-neighbor interpolation to preserve depth values and geometric boundaries. After preprocessing, we use a Conv2d–GroupNorm–GELU convolution module to aggregate nearby depth information. This changes each single depth value into a multi-channel feature with local depth information. In this way, the model can learn local depth structures. Residual convolution blocks further aggregate depth information from a wider local region while keeping the depth feature resolution unchanged. Finally, Patch Embedding encodes the depth feature map into a token sequence. The resulting depth tokens have the same number and feature dimension as the RGB tokens. This depth branch maps depth observations into independent and learnable geometric representations without introducing an additional large-scale 3D backbone.

Token-level 2D–3D alignment. To strengthen the correspondence between depth features and 2D image features, we use the same spatial grid as the RGB encoder for patch partitioning during Patch Embedding. As shown in Fig. 2, RGB tokens and depth tokens with the same time step, view, and patch index describe the 2D appearance semantics and depth geometry of the same local region, respectively:

$$\mathbf{z}_{\tau,v,p}^x \leftrightarrow \mathbf{z}_{\tau,v,p}^d, \qquad p = 1, \ldots, N_p. \tag{3}$$

Furthermore, to achieve temporal associations across modalities, we introduce Tri-modal Cycle-Phase Positional Encoding (TCPE). As shown in Fig. 1(a), for the same sample, the language, RGB, and depth token sequences are mapped to the same feature dimension D through their respective adaptors, resulting in the condition tokens $\mathbf{C}_j^\ell$, $\mathbf{C}_{\tau,v,p}^x$, and $\mathbf{C}_{\tau,v,p}^d$, respectively. These three conditions are then cyclically injected into the DiT in a fixed order. One complete Language $\rightarrow$ RGB $\rightarrow$ Depth injection is defined as a tri-modal cycle. For the k-th DiT block, we define the cycle index g_k and the intra-cycle modality index r_k to indicate its cycle membership and the type of condition it receives, respectively:

$$g_k = \left\lfloor \frac{k}{3} \right\rfloor, \qquad r_k = k \bmod 3. \tag{4}$$

Here, $r_k = 0, 1, 2$ correspond to language, RGB, and depth, respectively. We then encode the two coordinates into a unified D-dimensional positional vector:

$$\mathbf{H}_k = \mathbf{H}(g_k, r_k) = \mathrm{PE}_{D/2}(g_k) \oplus \mathrm{PE}_{D/2}(r_k). \tag{5}$$

Here, $\oplus$ denotes concatenation along the feature dimension. $\mathrm{PE}_q(u) \in \mathbb{R}^q$ denotes a q-dimensional sinusoidal positional encoding:

$$\mathrm{PE}_q(u)[2i] = \sin\left(\frac{u}{10000^{2i/q}}\right), \qquad \mathrm{PE}_q(u)[2i+1] = \cos\left(\frac{u}{10000^{2i/q}}\right). \tag{6}$$

Therefore, the three conditions within the same cycle share the same first $D/2$ dimensions, which encode their common cycle index. The remaining $D/2$ dimensions are determined by r_k and distinguish language, RGB, and depth. The final representations of the three conditions are

$$\begin{aligned} \widetilde{\mathbf{C}}^{\ell}_{g,j} &= \mathbf{C}^{\ell}_{j} + \mathbf{H}(g,0), \\ \widetilde{\mathbf{C}}^{x}_{g,\tau,v,p} &= \mathbf{C}^{x}_{\tau,v,p} + \mathbf{H}(g,1) + \mathbf{T}_{\tau} + \mathbf{V}_{v}, \\ \widetilde{\mathbf{C}}^{d}_{g,\tau,v,p} &= \mathbf{C}^{d}_{\tau,v,p} + \mathbf{H}(g,2) + \mathbf{T}_{\tau} + \mathbf{V}_{v}. \end{aligned} \tag{7}$$

Here, $\mathbf{T}_{\tau}$ and $\mathbf{V}_{v}$ denote the positional encodings for the historical frame and camera view, respectively. This cyclic injection builds temporal connections among multimodal conditions. At the same time, it avoids semantic interference between depth information and the pretrained VLM backbone while allowing depth information to fully participate in action generation.

3.3 Global Representation Alignment for Stable Participation

To achieve stable participation of depth information, we use the idea of multimodal representation alignment. However, existing multimodal alignment methods often choose one modality as an anchor and align the other modalities with it separately (Ruan et al., 2023; Girdhar et al., 2023; Zhu et al., 2024; Zhao et al., 2022; Chen et al., 2023b; Liu et al., 2024; Xu et al., 2023; **?**). These alignment schemes do not directly constrain the relations among non-anchor modalities, making it difficult to achieve joint alignment across all modalities. We therefore adopt the multimodal alignment strategy of GRAM (Cicchetti et al., 2025) and use the Gramian volume to jointly model the overall relation among language, RGB, depth, and expert actions under a unified objective.

Four-Modal Global Representations and Gramian Volume Measure. As shown in Fig. 3, we use learnable-query attention aggregation to extract global class tokens from the language, RGB, depth, and clean expert-action token sequences:

$$\mathbf{g}^{m} = \begin{cases} \mathcal{P}_m\left(\mathbf{C}^{m}; \mathbf{q}^{m}\right), & m \in \{\ell, a\}, \\ \dfrac{1}{T_o V} \displaystyle\sum_{r=1}^{T_o V} \mathcal{P}_m\left(\mathbf{C}^{m}_{r}; \mathbf{q}^{m}_{r}\right), & m \in \{x, d\}. \end{cases} \tag{8}$$

Here, $\mathbf{C}^{\ell}$, $\mathbf{C}^{x}$, $\mathbf{C}^{d}$, and $\mathbf{C}^{a}$ denote the token sequences of language, RGB, depth, and clean expert actions, respectively, with the same feature dimension. $\mathcal{P}_m$ denotes the attention aggregator, and $\mathbf{q}^{m}$ is a learnable query vector. We directly aggregate the language and action sequences into global representations. For RGB and depth, we independently aggregate the tokens within $T_o V$ time-view groups and then use mean pooling to combine the group representations into global representations. Finally, we obtain the four-modal class tokens $\mathbf{g}^{\ell}$, $\mathbf{g}^{x}$, $\mathbf{g}^{d}$, and $\mathbf{g}^{a}$.

Next, as shown on the right side of Fig. 3, we apply L_2 normalization to the four-modal class tokens to prevent the model from reducing the Gramian volume by simply shrinking the feature norms. The normalized language, RGB, depth, and expert-action representations are then arranged as columns to form the representation matrix $\mathbf{A}$, from which the Gram matrix $\mathbf{G} = \mathbf{A}^{\top}\mathbf{A}$ is constructed. Since all representations are normalized, the diagonal elements of $\mathbf{G}$ are 1, while the off-diagonal elements encode the inner-product relations between modalities, thereby jointly characterizing the overall relation among the four modalities. According to the Gram determinant theorem, the Gramian volume is defined as $\mathrm{Vol} = \sqrt{\det(\mathbf{G})}$, where a smaller volume indicates stronger joint alignment among the four modalities.

Bidirectional Contrastive Loss. The Gramian volume measures the relations among the four modalities but does not directly enforce semantic matching. Therefore, we further introduce the Gramian volume into contrastive learning. For a batch of size B, we pair the i-th language representation with the j-th RGB–Depth–Action group and define

$$V_{ij} = \mathrm{Vol}\left(\bar{\mathbf{g}}^{\ell}_{i}, \bar{\mathbf{g}}^{x}_{j}, \bar{\mathbf{g}}^{d}_{j}, \bar{\mathbf{g}}^{a}_{j}\right), \qquad s_{ij} = -\frac{V_{ij}}{\tau}. \tag{9}$$

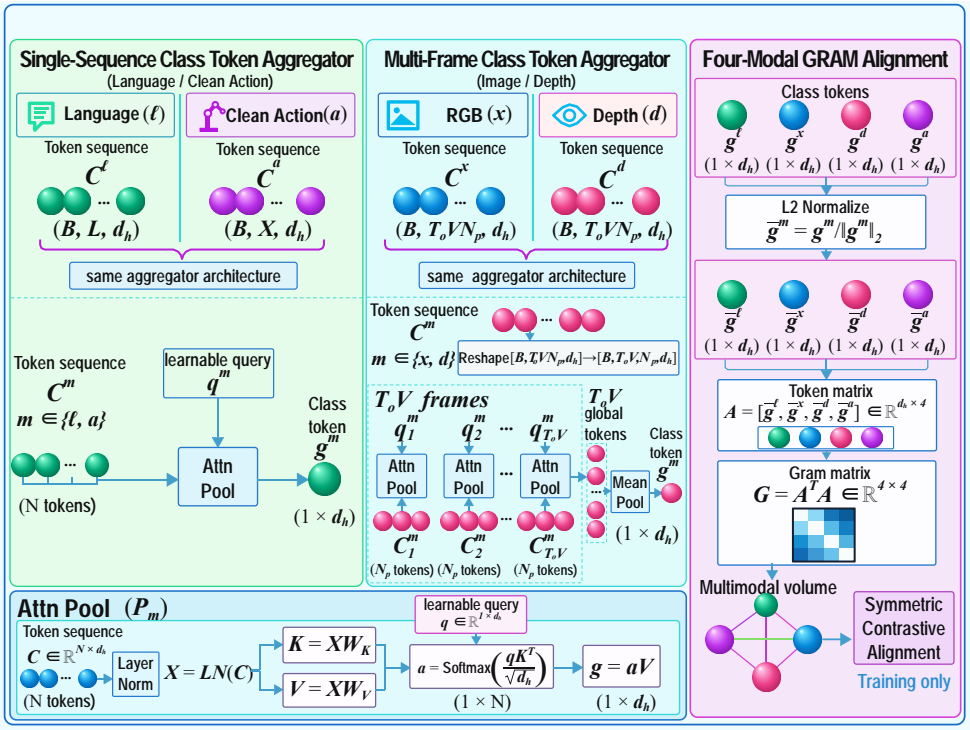

Figure 3: Four-modal Global Representation Aggregation and GRAM Alignment.

Here, $i = j$ denotes a matched combination, while $i \neq j$ denotes an unmatched combination. We optimize both retrieval directions using the symmetric contrastive loss

$$\mathcal{L}_{\text{GRAM}} = -\frac{1}{2B}\sum_{i=1}^{B}\left[\log\frac{\exp(s_{ii})}{\sum_{j=1}^{B}\exp(s_{ij})} + \log\frac{\exp(s_{ii})}{\sum_{j=1}^{B}\exp(s_{ji})}\right]. \tag{10}$$

This objective encourages matched combinations to have smaller Gramian volumes than unmatched combinations. Distributed batch construction is detailed in Appendix A.

Joint Optimization Objective. GRAM-VLA uses the denoising prediction error between the predicted clean action chunk and the expert action chunk as the primary objective for action generation:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{A}_t^0,k,\epsilon}\left[\left\|\mathbf{A}_t^0 - f_\theta\left(\mathbf{A}_t^k, k; \ell, \mathbf{o}_t\right)\right\|_2^2\right]. \tag{11}$$

The bidirectional GRAM contrastive loss is introduced as an auxiliary relational constraint. The final training objective is defined as:

$$\mathcal{L}_{\text{total}}(t) = \mathcal{L}_{\text{diff}} + \omega(t)\mathcal{L}_{\text{GRAM}}. \tag{12}$$

$$\omega(t) = \frac{\omega_0}{2}\left[1 + \cos\left(\frac{\pi t}{T}\right)\right]. \tag{13}$$

Here, $\omega(t)$ is a dynamic weight with cosine decay, where t and T denote the current and total training steps, respectively, and ω_0 is the initial weight. This schedule imposes a stronger four-modal relational constraint early in training and gradually reduces its influence, allowing the model to focus more on conditional action generation at later stages.

4 Experiments

We conduct real-world dual-arm robot experiments to answer the following three questions: **Q1:** Can GRAM-VLA preserve its pretrained 2D semantic understanding and bimanual coordination

Table 1: Success rate comparison on real-robot aligned tasks.

Task	DP	ACT	RDT	DP3	GRAM-VLA
Blocks Ranking RGB	0%	1%	3%	3%	**13%**
Blocks Ranking Size	1%	0%	0%	2%	**22%**
Pick Dual Bottles	24%	31%	42%	**60%**	45%
Place Fan	3%	1%	12%	36%	**52%**
Place Object Stand	22%	1%	15%	60%	**79%**
Press Stapler	6%	31%	41%	**69%**	62%
Stack Bowls Two	61%	82%	76%	83%	**89%**
Stack Bowls Three	63%	48%	51%	57%	**67%**
Average	22.5%	24.4%	30.0%	46.3%	**53.6%**

Table 2: Real-robot success rates under different environment changes.

Model	Standard	Appearance Shift	Height Shift	Average
RDT-Scratch	25.0%	0.0%	0.0%	8.3%
GRAM-VLA-Scratch	**35.0%**	**12.5%**	**17.5%**	**21.7%**
RDT-Pretrained	**87.5%**	25.0%	0.0%	37.5%
GRAM-VLA-Pretrained	85.0%	**65.0%**	**72.5%**	**74.2%**

capabilities after the introduction of 3D geometric information? **Q2:** Can the introduction of 3D geometric information improve the model's performance on spatially sensitive tasks? **Q3:** Can the 2D–3D collaborative modeling method proposed in this paper improve the model's generalization ability? To answer these questions, this paper conducts two types of real-robot experiments.

4.1 Real-Robot Reproduction of RoboTwin 2.0 Tasks

Task description. Following RoboTwin 2.0 (Chen et al., 2025), we reproduce eight representative tasks on a real robot, covering semantic understanding, color discrimination, spatial reasoning, and bimanual coordination. Each task uses 50 expert demonstrations for training and 100 autonomous rollouts for evaluation, with full-task success rate as the metric. Task details are provided in Appendix B.2.

Baselines. We compare GRAM-VLA with Diffusion Policy (DP) (Chi et al., 2025), ACT (Zhao et al., 2023), RDT (Liu et al., 2025b), and 3D Diffusion Policy (DP3) (Ze et al., 2024). These methods cover 2D visual policies, pretrained robot policies, and explicit 3D policies.

Experimental results. As shown in Table 1, GRAM-VLA achieves the highest average success rate of 53.6% across eight tasks and obtains the best performance on six of them, demonstrating strong semantic understanding, spatial reasoning, and bimanual manipulation capabilities and thereby answering Q1 and Q2. Real-robot experiments also include sensor noise, control errors, and contact dynamics. GRAM-VLA can still complete different bimanual tasks in closed-loop control. This further shows its real-robot deployment capability.

4.2 Ablation and Controlled Experiments

To further reduce the effect of the simulation-to-real gap, we directly compare RDT and GRAM-VLA using the same real-robot platform, training data, and evaluation settings.

4.2.1 Generalization Evaluation

Experimental setup. We build three real-robot test settings for the bun pick-and-place task. The models are trained or fine-tuned only on 200 demonstrations collected in the Standard setting. We compare RDT and GRAM-VLA under both Scratch and Pretrained settings. Each model is evaluated with 40 real-robot rollouts in each setting. Detailed scene settings and examples are provided in Appendix B.3.

Experimental results. Under pretraining, GRAM-VLA remains comparable to RDT in Standard (85.0% versus 87.5%) while substantially improving Appearance Shift (65.0% versus 25.0%) and

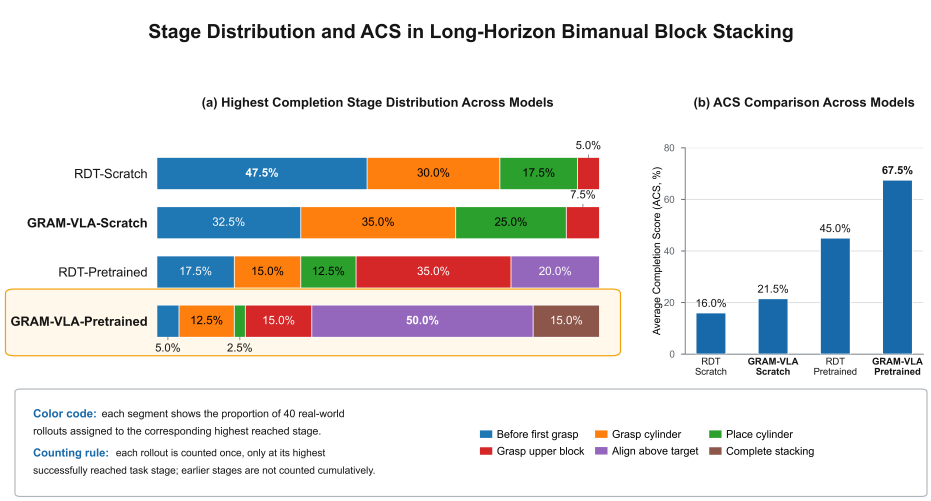

Figure 4: Quantitative results of the bimanual block-stacking task. **(a)** Distribution of the highest completion stage reached over 40 rollouts, where each rollout is counted once at its highest successfully completed stage. **(b)** Average Completion Score (ACS), which summarizes the overall task progress across rollouts.

Height Shift (72.5% versus 0.0%). Its average success rate reaches 74.2%, outperforming RDT by 36.7 percentage points; the same trend holds under Scratch, validating Q3.

4.2.2 Complex Long-Horizon Spatial Manipulation Evaluation

Experimental setup. We evaluate the long-horizon spatial reasoning and bimanual coordination capabilities of GRAM-VLA using a multi-stage block-stacking task. RDT and GRAM-VLA are compared under the same Scratch and Pretrained settings, using 200 demonstrations for training or fine-tuning. Each model is evaluated with 40 real-robot rollouts. Each rollout is assigned to its highest successfully completed stage, and we report both the distribution across completion stages and the Average Completion Score (ACS) to characterize overall task progress. Detailed task stages, scoring rules, and the ACS definition are provided in Appendix B.4.

Experimental results. As shown in Fig. 4(a), GRAM-VLA shifts the completion distribution toward later task stages under both Scratch and Pretrained settings. Fig. 4(b) further shows that GRAM-VLA achieves a higher ACS than RDT in both settings. Under Pretrained, its ACS reaches 67.5%, outperforming RDT by 22.5 percentage points. These results demonstrate stronger long-horizon spatial reasoning and bimanual coordination, further validating Q2.

5 Conclusion

In this work, we presented GRAM-VLA, a 2D–3D collaborative VLA model that incorporates depth information into dual-arm action generation without additional large-scale 2D–3D pretraining. GRAM-VLA combines independent RGB–Depth encoding with four-modal global representation alignment to enhance 3D spatial perception while preserving pretrained 2D semantics. Extensive real-robot experiments demonstrate improved generalization and complex bimanual manipulation, especially under spatial and environmental variations.Our current evaluation is limited to a dual-arm platform and a finite set of tasks. Future work will explore broader robot embodiments and richer 3D observations.

References

Lucas Beyer, Andreas Steiner, André Susano Pinto, et al. PaliGemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Johan Bjorck, Fernando Castañeda, Nikolai Cherniadev, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, 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, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Shizhe Chen, Ricardo Garcia, Cordelia Schmid, et al. PolarNet: 3d point clouds for language-guided robotic manipulation. *arXiv preprint arXiv:2309.15596*, 2023a.

Shizhe Chen, Paul Pacaud, and Cordelia Schmid. PointACT: Vision-language-action models with multi-scale point-action interaction. *arXiv preprint arXiv:2605.21414*, 2026.

Sihan Chen, Handong Li, Qunbo Wang, et al. Vast: A vision-audio-subtitle-text omni-modality foundation model and dataset. In *Advances in Neural Information Processing Systems*, volume 36, pp. 72842–72866, 2023b.

Tianyi Chen, Zhe Chen, Bowen Chen, 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.

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. doi: 10.1177/02783649241273668.

Giovanni Cicchetti, Eleonora Grassucci, Lorenzo Sigillo, et al. Gramian multimodal representation learning and alignment. In *International Conference on Learning Representations*, pp. 42128–42149, 2025.

Q. Fan, Z. Li, J. Hu, et al. Peafowl: Perception-enhanced multi-view vision-language-action for bimanual manipulation. *IEEE Robotics and Automation Letters*, 2026.

Yicheng Feng, Wanpeng Zhang, Ye Wang, et al. Spatial-aware vla pretraining through visual-physical alignment from human videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 712–723, 2026.

Ricardo Garcia, Shizhe Chen, and Cordelia Schmid. Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy. In *2025 IEEE International Conference on Robotics and Automation*, pp. 8996–9002. IEEE, 2025.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, et al. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15180–15190, 2023.

Hyunsoo Im, Eunsol Jeong, Andrey Kolobov, et al. TwinVLA: Data-efficient bimanual manipulation with twin single-arm vision-language-action models. In *International Conference on Learning Representations*, pp. 61969–61995, 2026.

Eric Jang, Alex Irpan, Mohi Khansari, et al. BC-Z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pp. 991–1002. PMLR, 2022.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, et al. Prismatic vlms: Investigating the design space of visually-conditioned language models. *arXiv preprint arXiv:2402.07865*, 2024.

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

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

Cheng Li, Junjie Wen, Yichen Peng, et al. PointVLA: Injecting the 3d world into vision-language-action models. *IEEE Robotics and Automation Letters*, 11(3):2506–2513, 2026a.

Peng Li, Yiming Chen, Hao Wu, et al. BridgeVLA: Input-output alignment for efficient 3d manipulation learning with vision-language models. *Advances in Neural Information Processing Systems*, 38:63635–63673, 2026b.

Haotian Liu, Chunyuan Li, Qingyang Wu, et al. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916, 2023.

J. Liu, S. Chen, X. He, et al. Valor: Vision-audio-language omni-perception pretraining model and dataset. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(2):708–724, 2024.

Jiaming Liu, Hao Chen, Peng An, et al. HybridVLA: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025a.

Jiaming Liu, Q. Wuwu, Nuowei Han, et al. Lift3D-VLA: Lifting vla models to 3d geometry and dynamics-aware manipulation. *arXiv preprint arXiv:2607.06564*, 2026.

Shizhe Liu, Lingxiao Wu, Bing Li, et al. RDT-1B: A diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, pp. 29982–30009, 2025b.

Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*, 2020.

Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters*, 7(4):11205–11212, 2022.

Da Qu, Hao Song, Qiyu Chen, et al. SpatialVLA: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

Ludan Ruan, Anwen Hu, Yuqing Song, et al. Accommodating audio modality in clip for multimodal processing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9641–9649, 2023.

Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, et al. Hi Robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*, 2025.

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

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.

Lin Sun, Bin Xie, Yingfei Liu, et al. GeoVLA: Empowering 3d representations in vision-language-action models. *arXiv preprint arXiv:2508.09071*, 2025.

T. Tang, X. Ji, W. Xing, et al. Towards human-like manipulation through rl-augmented teleoperation and mixture-of-dexterous-experts vla. *arXiv preprint arXiv:2603.08122*, 2026.

Peng Wang, Shuai Bai, Sinan Tan, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Y. Wang, J. Yu, X. Peng, et al. Co-VLA: Coordination-aware structured action modeling for dual-arm vision-language-action systems. *arXiv preprint arXiv:2606.20285*, 2026.

Junjie Wen, Yichen Zhu, Jinming Li, et al. DexVLA: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025a.

Junjie Wen, Yichen Zhu, Jinming Li, et al. TinyVLA: Toward fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 10(4):3988–3995, 2025b.

Haiyang Xu, Qinghao Ye, Mingshi Yan, et al. mplug-2: A modularized multi-modal foundation model across text, image and video. In *International Conference on Machine Learning*, pp. 38728–38748. PMLR, 2023.

Ruihan Yang, Guangyao Chen, Chao Wen, et al. FP3: A 3d foundation policy for robotic manipulation. *arXiv preprint arXiv:2503.08950*, 2025.

Tianyu Yuan, Yiming Liu, Cewu Lu, et al. DepthVLA: Enhancing vision-language-action models with depth-aware spatial reasoning. *arXiv preprint arXiv:2510.13375*, 2025.

Yanjie Ze, Gu Zhang, Kangning Zhang, et al. 3D Diffusion Policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

Gu Zhang, Qicheng Xu, Haozhe Zhang, et al. UniDex: A robot foundation suite for universal dexterous hand control from egocentric human videos. *arXiv preprint arXiv:2603.22264*, 2026.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, 2023. doi: 10.15607/RSS.2023.XIX.016.

Yanpeng Zhao, Jack Hessel, Youngjae Yu, et al. Connecting the dots between audio and text without parallel data through visual knowledge transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4492–4507, 2022.

Bin Zhu, Bin Lin, Munan Ning, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *International Conference on Learning Representations*, 2024.

Xiaoyu Zhu, Yiming Qi, Yuke Zhu, et al. EquAct: An SE(3)-equivariant multi-task transformer for open-loop robotic manipulation. *arXiv preprint arXiv:2505.21351*, 2025.

APPENDIX A.

A.1 LANGUAGE-TO-MULTIMODAL RETRIEVAL

We use distributed training with two GPUs as an example. The local batch size is $B1 = 4$. The RGB–Depth–Action representations from the two GPUs are gathered into a global batch with $B2 = 8$. Taking language as the anchor, the four local language representations on Rank 0 retrieve the eight global multimodal groups.

$$\mathcal{M}_j = \left(\bar{\mathbf{g}}_j^x, \bar{\mathbf{g}}_j^d, \bar{\mathbf{g}}_j^a\right), \qquad j = 0, \ldots, 7.$$

Language-to-multimodal volume matrix

Rank 0	M0	M1	M2	M3	M4	M5	M6	M7
L0	**V(0,0)**	V(0,1)	V(0,2)	V(0,3)	V(0,4)	V(0,5)	V(0,6)	V(0,7)
L1	V(1,0)	**V(1,1)**	V(1,2)	V(1,3)	V(1,4)	V(1,5)	V(1,6)	V(1,7)
L2	V(2,0)	V(2,1)	**V(2,2)**	V(2,3)	V(2,4)	V(2,5)	V(2,6)	V(2,7)
L3	V(3,0)	V(3,1)	V(3,2)	**V(3,3)**	V(3,4)	V(3,5)	V(3,6)	V(3,7)

The diagonal entries are matched samples and are treated as positives; the remaining entries are negatives.

For example, $V(0,0)$ is constructed from the language, RGB, depth, and clean expert-action representations of sample 0. The corresponding four-modal representation matrix is

$$\mathbf{A}_{00} = \left[\bar{\mathbf{g}}_0^{\ell}, \bar{\mathbf{g}}_0^x, \bar{\mathbf{g}}_0^d, \bar{\mathbf{g}}_0^a\right] \in \mathbb{R}^{d_h \times 4}.$$

The corresponding Gram matrix is:

$$\mathbf{G}_{00} = \mathbf{A}_{00}^{\top}\mathbf{A}_{00} = \begin{bmatrix} 1 & \langle \bar{\mathbf{g}}_0^{\ell}, \bar{\mathbf{g}}_0^{x} \rangle & \langle \bar{\mathbf{g}}_0^{\ell}, \bar{\mathbf{g}}_0^{d} \rangle & \langle \bar{\mathbf{g}}_0^{\ell}, \bar{\mathbf{g}}_0^{a} \rangle \\ \langle \bar{\mathbf{g}}_0^{x}, \bar{\mathbf{g}}_0^{\ell} \rangle & 1 & \langle \bar{\mathbf{g}}_0^{x}, \bar{\mathbf{g}}_0^{d} \rangle & \langle \bar{\mathbf{g}}_0^{x}, \bar{\mathbf{g}}_0^{a} \rangle \\ \langle \bar{\mathbf{g}}_0^{d}, \bar{\mathbf{g}}_0^{\ell} \rangle & \langle \bar{\mathbf{g}}_0^{d}, \bar{\mathbf{g}}_0^{x} \rangle & 1 & \langle \bar{\mathbf{g}}_0^{d}, \bar{\mathbf{g}}_0^{a} \rangle \\ \langle \bar{\mathbf{g}}_0^{a}, \bar{\mathbf{g}}_0^{\ell} \rangle & \langle \bar{\mathbf{g}}_0^{a}, \bar{\mathbf{g}}_0^{x} \rangle & \langle \bar{\mathbf{g}}_0^{a}, \bar{\mathbf{g}}_0^{d} \rangle & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4}.$$

$$V(0,0) = \mathrm{Vol}(0,0) = \sqrt{\det(\mathbf{G}_{00})}.$$

A.2 Multimodal-to-Language Retrieval

For the reverse direction, the batch roles are exchanged. The four local RGB–Depth–Action groups on Rank 0 use $B1 = 4$, while the language representations from both GPUs are gathered into a global batch with $B2 = 8$. The multimodal group is used as the anchor to retrieve the eight global language representations.

Multimodal-to-language volume matrix

Rank 0	L0	L1	L2	L3	L4	L5	L6	L7
M0	**V(0,0)**	V(0,1)	V(0,2)	V(0,3)	V(0,4)	V(0,5)	V(0,6)	V(0,7)
M1	V(1,0)	**V(1,1)**	V(1,2)	V(1,3)	V(1,4)	V(1,5)	V(1,6)	V(1,7)
M2	V(2,0)	V(2,1)	**V(2,2)**	V(2,3)	V(2,4)	V(2,5)	V(2,6)	V(2,7)
M3	V(3,0)	V(3,1)	V(3,2)	**V(3,3)**	V(3,4)	V(3,5)	V(3,6)	V(3,7)

A.3 Masked and Gripper-Weighted Diffusion Objective

The diffusion objective in the main text is presented in a compact form. During training, Gaussian noise is added to the clean expert action chunk $\mathbf{A}_t^0$ at a randomly sampled diffusion step k:

$$\mathbf{A}_t^k = \sqrt{\bar{\alpha}_k}\mathbf{A}_t^0 + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

The denoising network predicts the corresponding clean action chunk conditioned on the noisy action, diffusion step, language instruction, and robot observation:

$$\widehat{\mathbf{A}}_t^0 = f_\theta\left(\mathbf{A}_t^k, k; \ell, \mathbf{o}_t\right).$$

The clean expert action chunk is represented as

$$\mathbf{A}_t^0 \in \mathbb{R}^{T_a \times D_a},$$

where $T_a = 64$ denotes the action chunk length and $D_a = 128$ denotes the unified action dimension used by the model. Among the 128 dimensions, only 16 correspond to valid physical controls of the dual-arm robot, including 14 arm-joint dimensions and 2 gripper dimensions. We therefore introduce a binary validity mask

$$M_{r,d} \in \{0, 1\},$$

where $r = 1, \ldots, T_a$ indexes the action step within the chunk and $d = 1, \ldots, D_a$ indexes the action dimension.

Since the two gripper dimensions are more difficult to learn, we assign them a larger loss weight:

$$w_d = \begin{cases} 5, & d \in \mathcal{D}_{\text{gripper}}, \\ 1, & d \in \mathcal{D}_{\text{joint}}. \end{cases}$$

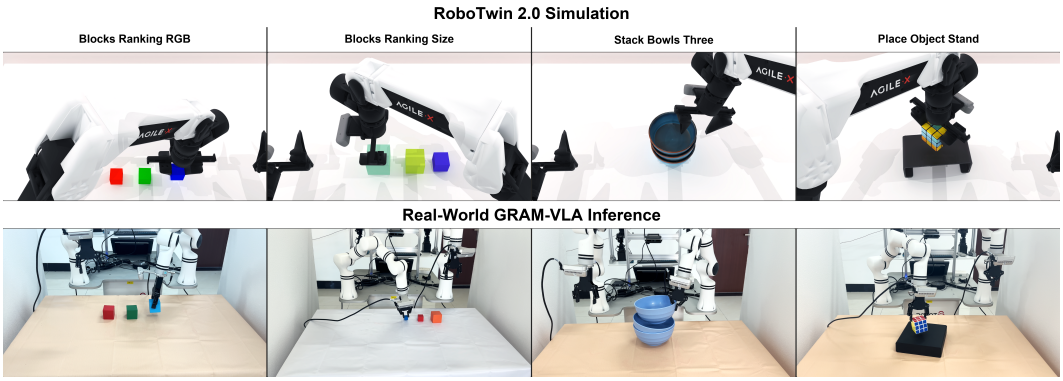

Figure B2: RoboTwin 2.0 simulation tasks and their corresponding real-robot tasks.

Table B2: Task-specific evaluation focus.

Task	**Primary capability evaluated**
Blocks Ranking RGB	Semantic understanding and color discrimination
Blocks Ranking Size	Object-size and spatial-order reasoning
Pick Dual Bottles	Coordinated bimanual grasping
Place Fan	Semantic recognition and target placement
Place Object Stand	Height and spatial-relation reasoning
Press Stapler	Fine manipulation
Stack Bowls Two	Occlusion, spatial reasoning, and continuous manipulation
Stack Bowls Three	Occlusion and longer-horizon continuous manipulation

The diffusion loss used in training is therefore

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{A}_t^0,k,\boldsymbol{\epsilon}} \left[\frac{\sum_{r=1}^{T_a} \sum_{d=1}^{D_a} M_{r,d} w_d \left(\widehat{A}_{t,r,d}^0 - A_{t,r,d}^0 \right)^2}{\sum_{r=1}^{T_a} \sum_{d=1}^{D_a} M_{r,d} w_d} \right].$$

Thus, only the 16 valid physical action dimensions contribute to the diffusion objective. The 14 arm-joint dimensions use a weight of 1, while the two gripper dimensions use a weight of 5.

APPENDIX B.

B.1 EXPERIMENTAL PLATFORM AND SETUP

All experiments are performed on the ALOHA dual-arm robot platform with 16-dimensional joints. Intel RealSense D435i cameras are mounted on the wrists of both arms to capture local observations near the end effectors, while Intel RealSense D455 cameras are placed above and below the workspace to provide global scene observations. Both RGB images and depth maps are collected at an original resolution of 640×480. Robot control and data collection are performed at 25 Hz.

B.2 ROBOTWIN 2.0 TASK DEFINITIONS

The task-specific evaluation focus is summarized below, and Fig. B2 illustrates the correspondence between representative RoboTwin 2.0 simulation tasks and their real-robot counterparts.

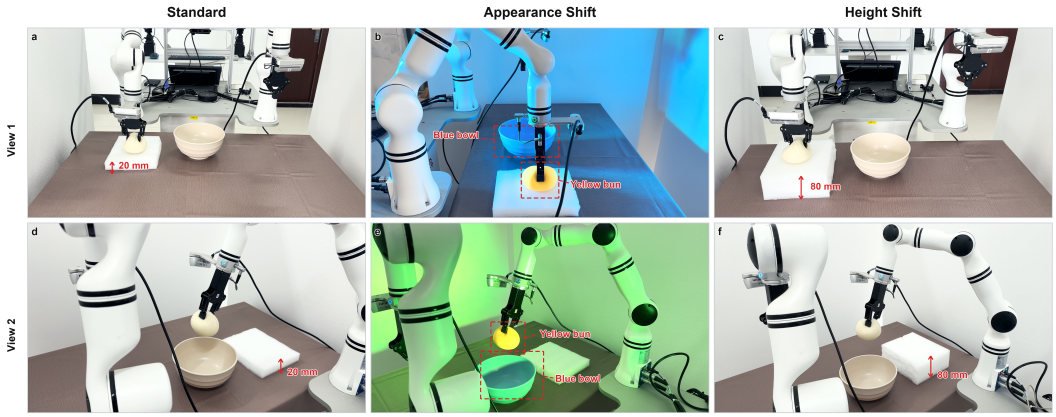

Figure B3: Real-robot scenes for generalization evaluation. The left, middle, and right columns correspond to **Standard**, **Appearance Shift**, and **Height Shift**, respectively. The top and bottom rows show two different camera views.

B.3 Generalization Scene Configuration

The generalization evaluation is based on the bun pick-and-place task and includes three test conditions: **Standard**, **Appearance Shift**, and **Height Shift**. The Standard setting uses the same visual and spatial conditions as the training setting, with a 20 mm spacer. Appearance Shift changes the environmental lighting and the color and appearance of the target objects. Height Shift increases the spacer height to 80 mm, thereby changing the initial height of the target and introducing a spatial geometric variation that is not observed during training. All three settings use the same robot platform, action space, and evaluation protocol, and no additional adaptation data are introduced during evaluation.

B.4 Long-Horizon Block-Stacking Protocol

The task requires the robot to stack a chess-piece block on a cylindrical block on the round tray. The right arm first grasps and places the cylinder. The left arm then grasps the upper block, aligns it above the cylinder, and releases it to complete the stack. Because errors in early stages may prevent subsequent operations, each rollout is evaluated according to the highest successfully completed stage.

As shown in Fig. B4, the task contains five key stages corresponding to 20%, 40%, 60%, 80%, and 100% completion: grasping the cylinder, placing the cylinder, grasping the upper block, aligning it above the cylinder, and completing the stack. A rollout is assigned 0% if the first grasp is not completed. Each rollout is counted only once at its highest successfully completed stage.

Evaluation metric. We use the Average Completion Score (ACS) to summarize the overall task progress across rollouts:

$$\mathrm{ACS} = \frac{1}{N}\sum_{i=1}^{N} C_i, \tag{14}$$

where $C_i \in \{0, 20, 40, 60, 80, 100\}$ denotes the highest completion score reached in the i-th rollout, and $N = 40$. A higher ACS indicates that the model reaches later stages more reliably.

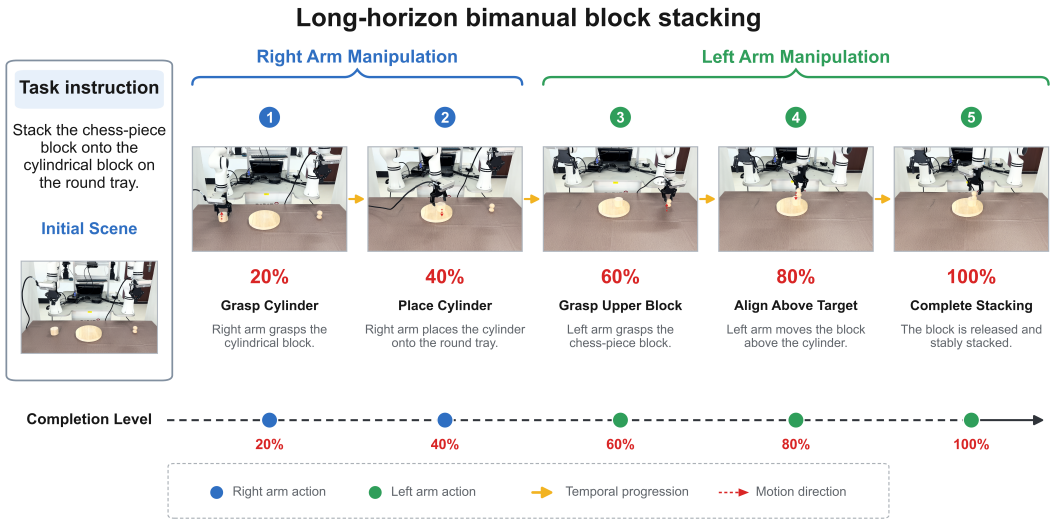

Figure B4: Stage breakdown of the block-stacking task. The left side shows the task instruction and initial scene. The right side shows the five key stages from 20% to 100% task completion. Yellow arrows show the task sequence, and red dashed arrows show the robot arm motion direction.