TARGET-VLA: MASK-GUIDED VISUAL RETRIEVAL FOR ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

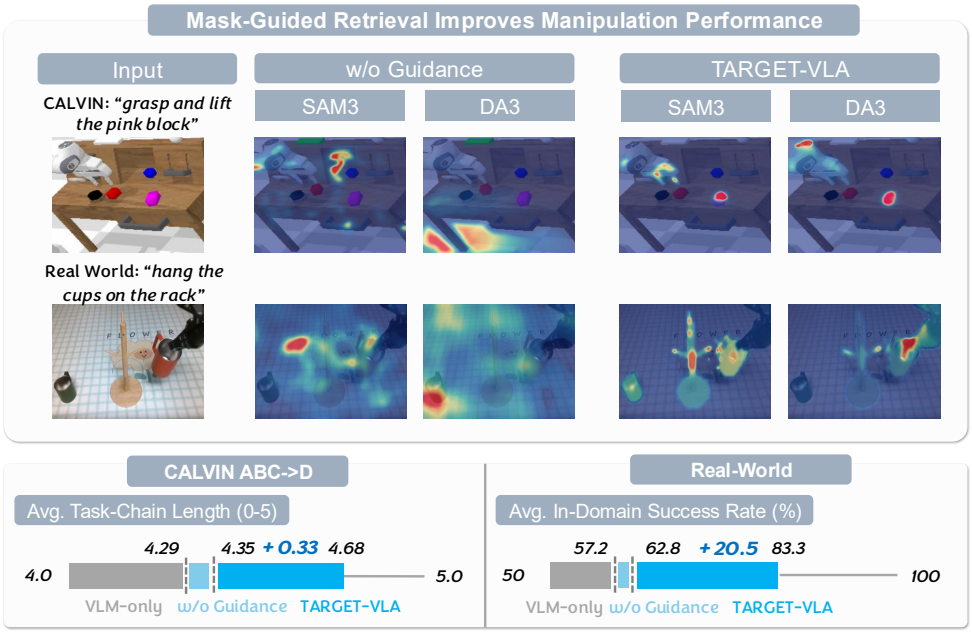


Figure 1: **Mask-guided visual retrieval for robotic manipulation.** Top: SAM3 and DA3 query-to-feature attention with and without mask guidance. Bottom: performance of VLM-only, w/o Guidance, and TARGET-VLA, measured by CALVIN ABC→D average task-chain length and mean success across three real-world in-domain tasks.

ABSTRACT

Vision-language-action (VLA) policies benefit from pretrained vision-language representations, but precise manipulation also requires fine-grained object-level and geometric information. However, our controlled comparison indicates that incorporating complementary representations from specialized vision foundation models without explicit spatial guidance yields only modest gains. Motivated by this observation, we introduce **TARGET-VLA**, a method for mask-guided retrieval of visual representations from frozen segmentation and depth models. It separates the roles of spatial guidance and visual content: prompt-conditioned segmentation masks guide where to retrieve information, while pretrained representations supply the features for action generation. Experiments in simulation and on a physical robot demonstrate improved manipulation performance over retrieval without mask guidance. TARGET-VLA achieves competitive results across simulation benchmarks, including zero-shot transfer to LIBERO-Plus, with real-world gains extending to shifted object positions and object categories absent from the robot demonstrations. These findings demonstrate the value of spatial guidance in integrating semantic and spatial representations to improve robotic manipulation.

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.

1 INTRODUCTION

Human manipulation combines scene understanding with perception of object locations and distances, and robotic policies (Chi et al., 2023) face similar perceptual demands. Recent vision-language-action (VLA) policies (Black et al., 2025; Yuan et al., 2026) adopt pretrained vision-language models (VLMs) as multimodal backbones, leveraging their semantic representations of visual observations and language instructions. However, semantic representations alone may not provide the spatial precision required for precise manipulation, which also depends on accurate localization of task-relevant objects and scene geometry. These spatial requirements have motivated efforts to enhance VLA perception by incorporating explicit 3D information or learning implicit geometric representations (Li et al., 2026a; Sun et al., 2025; Li et al., 2026b).

Pretrained vision foundation models for segmentation and depth estimation (Carion et al., 2026; Lin et al., 2025a; Wang et al., 2025a;b) encode object-level and geometric information that can complement VLM representations. However, incoporating this information into VLAs is non-trivial. We find that simply adding these representations without spatial guidance yields only modest gains over visual conditioning from VLM representations alone (Figures 1 and 3; Table 3). Qualitative attention maps further illustrate that query-to-feature attention can concentrate on background regions and exhibit similar spatial patterns across tasks involving different target objects (Figure 1; Appendix D). These observations motivate a key question: **How can task-relevant information be retrieved more effectively from these complementary representations for action generation?**

We address this question by separating the roles of spatial guidance and visual content during retrieval: **masks guide where queries attend, while pretrained visual features supply the content for action generation.** We introduce *TARGET-VLA*, a framework that integrates frozen representations from Segment Anything Model 3 (SAM3) (Carion et al., 2026) and Depth Anything 3 (DA3) (Lin et al., 2025a) into VLA policies through mask-guided visual retrieval (Figure 2). Given instruction-derived object prompts and a fixed end-effector prompt, SAM3 generates soft masks that guide retrieval from both representations. We inject mask-derived additive biases into query-to-feature cross-attention logits, favoring retrieval from prompted regions while retaining feature-dependent attention. The SAM3 branch retrieves segmentation features using a combined object-and-end-effector mask, while the DA3 branch retrieves geometric information from depth-aware features and predicted depth through separate object and end-effector streams. A query aggregation module consolidates these entity-specific streams into compact DA3 tokens. The retrieved SAM3 and DA3 tokens are concatenated with VLM and robot-state tokens to condition the action expert for action-chunk prediction.

Our experiments demonstrate that TARGET-VLA outperforms the compared VLA policies on multiple evaluation metrics (Tables 1 and 2). Its gains over unguided retrieval extend across simulation and real-world manipulation, including zero-shot transfer and the evaluated visual and object variations (Tables 3 and 1; Figure 3). Controlled comparisons highlight that how masks are used matters: with the same visual sources and condition-token budgets, TARGET-VLA outperforms a variant that concatenates the same masks with visual features instead of using them to guide retrieval (Table 3). Complementary attention analysis shows that guidance shifts attention toward prompted object and end-effector regions (Table 13). Together, these findings support using masks to guide where visual information is retrieved, rather than merely providing additional spatial inputs to the policy.

Our contributions are summarized as follows:

- We introduce TARGET-VLA, a visual retrieval framework that aggregates frozen segmentation and depth representations into compact conditioning tokens to complement VLM features for robotic manipulation.
- We develop a mask-guided cross-attention mechanism that separates the roles of spatial guidance and visual content, using prompt-conditioned soft masks to introduce additive spatial biases into query-to-feature attention.
- We demonstrate that TARGET-VLA outperforms the compared VLA policies on multiple evaluation metrics. Ablation studies further substantiate the effectiveness of mask guidance in improving manipulation performance in simulation and on a physical robot, with benefits extending to zero-shot transfer.

2 Related Work

Vision–language–action policies. VLA policies map visual observations and language instructions to robot actions through tokenized or autoregressive prediction (Liu et al., 2026a; Pertsch et al., 2025; Brohan et al., 2022; Cen et al., 2025), or continuous regression, diffusion, and flow matching (Team et al., 2024; Black et al., 2024; Liu et al., 2025; Li et al., 2024a). Large-scale robot-data pretraining and cross-embodiment learning further improve transfer across tasks and platforms (Black et al., 2025; Bjorck et al., 2025; Zheng et al., 2026; Yuan et al., 2026; Du et al., 2025; Wang et al., 2026c; Wu et al., 2026; Bu et al., 2025a; Zhang et al., 2026a; Team, 2026; Zhang et al., 2026b; Ma et al., 2026; Cheang et al., 2024; 2025). Our work is complementary to these advances and focuses on the visual evidence supplied to the action expert.

Spatial and task-relevant visual representations. Spatially enhanced VLA policies incorporate 3D, depth, point-cloud, multi-view, or geometry-aware representations (Zhen et al., 2024; Qu et al., 2025; Li et al., 2026a; Sun et al., 2025; Yuan et al., 2025; Li et al., 2025b; Xiao et al., 2026; Zhang et al., 2026c), or learn spatial representations through grounding, alignment, reconstruction, and distillation (Lin et al., 2025b; Ye et al., 2026b; Li et al., 2026b; Song et al., 2026; Wang et al., 2026a; Liu et al., 2026b). Other approaches use object-centric representations, visual prompts, affordances, and token selection to emphasize task-relevant information (Zhu et al., 2023; Zheng et al., 2025; Li et al., 2025a; Zhang et al., 2025; Huang et al., 2025; Zhou et al., 2026; Wang et al., 2026b). Closely related methods differ in where they strengthen visual processing: V-Link learns spatial and semantic queries within the VLM and passes them to the action expert (Lu et al., 2026); GuidedVLA supervises specialized attention heads in the action decoder (Jia et al., 2026); and DeepVision-VLA combines visual-feature injection into the VLA backbone with action-guided token pruning (Luo et al., 2026b). TARGET-VLA focuses on retrieval from frozen segmentation and depth representations before they condition the action expert. Prompt-conditioned masks directly bias query-to-feature attention, while feature-dependent attention determines the retrieved content.

Query-based retrieval and masked attention. Learnable-query interfaces, including BLIP-2's Q-Former (Li et al., 2023) and Flamingo's Perceiver Resampler (Alayrac et al., 2022), aggregate frozen visual features into compact token sequences. In segmentation, Mask2Former (Cheng et al., 2022) restricts cross-attention to regions defined by binarized decoder-predicted masks. TARGET-VLA builds on query-based aggregation but uses external, prompt-conditioned soft masks to supply finite additive attention biases over frozen segmentation and depth representations for action generation. The masks specify spatial weighting, while the source features supply the retrieved content.

3 Method

TARGET-VLA augments VLM conditioning with compact tokens retrieved from frozen segmentation and depth representations (Figure 2). Prompt-conditioned masks guide a segmentation branch that jointly aggregates features from the object and end-effector regions, and a depth branch that retrieves entity-specific features separately before aggregation. The resulting tokens are concatenated with VLM and robot-state tokens to condition an action expert trained to predict action chunks.

3.1 Problem Formulation

A VLA policy models a conditional distribution over future robot actions, $\mathbf{A}_t \sim \pi_\theta(\cdot \mid \mathbf{o}_t)$. The action chunk $\mathbf{A}_t = [\mathbf{a}_t, \ldots, \mathbf{a}_{t+H-1}] \in \mathbb{R}^{H \times d_a}$ contains H actions starting at control step t, each of dimension d_a. The observation $\mathbf{o}_t = (\mathcal{I}_t, \ell, \mathbf{s}_t)$ comprises images $\mathcal{I}_t = \{\mathbf{I}_t^v\}_{v=1}^{N_v}$ from N_v cameras, a language instruction ℓ, and robot state $\mathbf{s}_t$. The policy is learned from demonstration pairs $\mathcal{D} = \{(\mathbf{o}_t, \mathbf{A}_t)\}$.

3.2 Complementary Visual Representations

The VLM provides language-conditioned visual tokens, while SAM3 and DA3 supply complementary representations for mask-guided retrieval. The VLM processes the available camera views; the frozen SAM3 and DA3 encoders operate on a main-view image $\mathbf{I}_t^m \in \mathcal{I}_t$. For clarity, we omit time indices when describing visual conditioning.

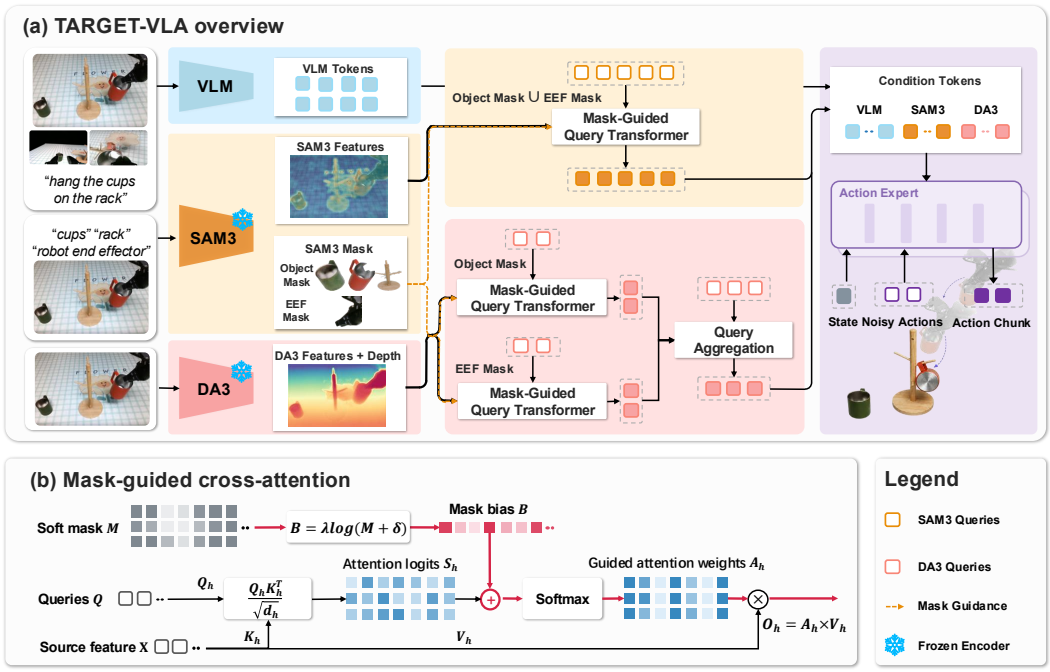


Figure 2: Overview of TARGET-VLA. SAM3 masks guide retrieval from segmentation and depth representations through an additive cross-attention bias (bottom). The resulting visual tokens condition the action expert alongside VLM and state tokens. Attention matrices are schematic.

VLM tokens. The VLM jointly encodes the images and language instruction into contextualized tokens, $\mathbf{Z}_{\text{vlm}} = E_{\text{vlm}}(\mathcal{I}, \ell) \in \mathbb{R}^{L_{\text{vlm}} \times d_{\text{cond}}}$, where L_{vlm} is the sequence length and d_{cond} is the common conditioning-token dimension.

SAM3 features and masks. Given task-object prompts $\mathcal{P}(\ell)$ extracted from instruction ℓ and a fixed end-effector prompt p_{eef}, SAM3 produces dense features and soft masks: $(\mathbf{F}_{\text{sam3}}, \mathbf{M}_{\text{eef}}, \mathbf{M}_{\text{obj}}) = E_{\text{sam3}}(\mathbf{I}^m, \{p_{\text{eef}}\} \cup \mathcal{P}(\ell))$. Here, $\mathbf{F}_{\text{sam3}} \in \mathbb{R}^{H_s \times W_s \times d_s}$ denotes the dense SAM3 representation. The masks $\mathbf{M}_{\text{eef}}, \mathbf{M}_{\text{obj}} \in [0,1]^{H_s \times W_s}$ correspond to the end-effector mask and the combined soft mask for the prompted task objects, respectively. Prompt construction and detection details are provided in Appendix C.1.

DA3 features and depth. DA3 processes the same image to produce depth-aware features $\mathbf{F}_{\text{da}} \in \mathbb{R}^{H_d \times W_d \times d_d}$ and a log-depth estimate $\mathbf{D} \in \mathbb{R}^{H_d \times W_d}$, written as $(\mathbf{F}_{\text{da}}, \mathbf{D}) = E_{\text{da}}(\mathbf{I}^m)$. Both outputs serve as sources of geometric information for retrieval.

3.3 Focused Retrieval with Mask Guidance

Our retrieval mechanism combines feature-dependent attention with mask-based spatial guidance. Rather than relying on query–feature similarity alone, it adds a soft mask bias to cross-attention logits, directly adjusting how spatial features contribute to the retrieved tokens. We implement this operation in a *mask-guided query transformer*, using the same mechanism for both representation sources.

Mask-guided cross-attention. Let $\mathbf{Q} \in \mathbb{R}^{N \times d_{\text{ret}}}$ denote N query tokens and $\mathbf{X} \in \mathbb{R}^{P \times d_{\text{ret}}}$ the source tokens at P spatial locations, where d_{ret} is the internal retrieval dimension. Each transformer block applies query self-attention, mask-guided cross-attention, and a feed-forward network, with pre-normalization and residual connections. For cross-attention head h, $\mathbf{Q}_h$ is projected from the updated queries, while $\mathbf{K}_h$ and $\mathbf{V}_h$ are projected from $\mathbf{X}$. The query–feature logits are $\mathbf{S}_h = \mathbf{Q}_h \mathbf{K}_h^\top / \sqrt{d_h}$, where d_h is the head dimension. We resize the soft mask to the source grid and flatten it into $\mathbf{M} \in [0,1]^{1 \times P}$. As shown in Figure 2, the mask bias $\mathbf{B} = \lambda \log(\mathbf{M} + \delta)$ modifies the attention

weights before feature aggregation:

$$\mathbf{A}_h = \mathrm{softmax}(\mathbf{S}_h + \lambda \log(\mathbf{M} + \delta)), \tag{1}$$

where softmax operates over the P source locations, λ is a learned guidance strength, and $\delta > 0$ ensures numerical stability. The bias is broadcast across queries and attention heads, and each head retrieves features as $\mathbf{O}_h = \mathbf{A}_h \mathbf{V}_h$. Thus, the masks adjust the spatial weighting without replacing the feature-dependent logits. The finite bias reweights source locations without explicitly excluding them. We denote the mask-guided query transformer by $\mathrm{QT}(\mathbf{Q}, \mathbf{X}; \mathbf{M})$ and write $\mathrm{QT}(\mathbf{Q}, \mathbf{X})$ for the same attention operator with $\mathbf{B} = \mathbf{0}$.

Mask-guided retrieval from SAM3 features. The SAM3 branch forms a joint summary of segmentation features for the prompted objects and end effector using a single set of learnable queries. We take the element-wise maximum, $\mathbf{M}_{\mathrm{sam3}} = \max(\mathbf{M}_{\mathrm{eef}}, \mathbf{M}_{\mathrm{obj}})$, and apply it as guidance over SAM3 source tokens. These tokens are formed as $\mathbf{X}_{\mathrm{sam3}} = \mathrm{Flatten}(\phi_{\mathrm{sam3}}(\mathbf{F}_{\mathrm{sam3}})) \in \mathbb{R}^{P_s \times d_{\mathrm{ret}}}$, where ϕ_{sam3} is a lightweight MLP-based encoder mapping each feature to dimension d_{ret} and P_s is the number of spatial tokens. A set of N_{sam3} learnable queries $\mathbf{Q}_{\mathrm{sam3}}$ retrieves a fixed-length representation, which an output MLP ψ_{sam3} projects to the conditioning space:

$$\mathbf{Z}_{\mathrm{sam3}} = \psi_{\mathrm{sam3}}(\mathrm{QT}(\mathbf{Q}_{\mathrm{sam3}}, \mathbf{X}_{\mathrm{sam3}}; \mathbf{M}_{\mathrm{sam3}})) \in \mathbb{R}^{N_{\mathrm{sam3}} \times d_{\mathrm{cond}}}. \tag{2}$$

Entity-wise mask-guided retrieval from DA3 features. To retain entity-type-specific geometric information before aggregation, the DA3 branch uses separate mask-guided retrieval streams for the objects and end effector. The two streams share the query-transformer parameters. Both retrieve from $\mathbf{X}_{\mathrm{da}} = \mathrm{Flatten}(\phi_{\mathrm{da}}(\mathbf{F}_{\mathrm{da}}, \mathbf{D})) \in \mathbb{R}^{P_d \times d_{\mathrm{ret}}}$, where the lightweight MLP-based encoder ϕ_{da} maps the feature and log-depth at each of the P_d locations to dimension d_{ret}. For entity type $r \in \{\mathrm{eef}, \mathrm{obj}\}$, an entity-type embedding $\mathbf{e}_r \in \mathbb{R}^{d_{\mathrm{ret}}}$ is added to shared learnable queries $\mathbf{Q}_{\mathrm{ent}}$, giving $\mathbf{Q}_r = \mathbf{Q}_{\mathrm{ent}} + \mathbf{e}_r$. Entity-specific retrieval yields:

$$\mathbf{U}_r = v_r \ \mathrm{QT}(\mathbf{Q}_r, \mathbf{X}_{\mathrm{da}}; \mathbf{M}_r), \qquad r \in \{\mathrm{eef}, \mathrm{obj}\}. \tag{3}$$

The entity tokens $\mathbf{U}_r$ remain in the d_{ret}-dimensional retrieval space. Here, $v_r \in \{0, 1\}$ indicates whether SAM3 detects the corresponding entity and zeros its retrieved tokens otherwise; the detection threshold is specified in Appendix C.1.

To integrate the two entity-specific streams into a compact conditioning sequence, the *query aggregation* module in Figure 2 uses a second set of learnable queries $\mathbf{Q}_{\mathrm{da}}$ to attend to their concatenated tokens without a spatial mask bias. The aggregated tokens are projected into the conditioning space by ψ_{da}:

$$\mathbf{Z}_{\mathrm{da}} = \psi_{\mathrm{da}}(\mathrm{QT}(\mathbf{Q}_{\mathrm{da}}, [\mathbf{U}_{\mathrm{eef}}; \mathbf{U}_{\mathrm{obj}}])) \in \mathbb{R}^{N_{\mathrm{da}} \times d_{\mathrm{cond}}}. \tag{4}$$

Here, $[\,;\,]$ denotes concatenation along the token dimension. After projection, we add a d_{cond}-dimensional embedding of $[v_{\mathrm{eef}}, v_{\mathrm{obj}}]$ to each token in $\mathbf{Z}_{\mathrm{da}}$ to convey detection status to the action expert.

3.4 Action Conditioning and Learning

Action conditioning. The retrieved SAM3 and DA3 tokens complement the VLM and robot-state conditions supplied to the action expert. To distinguish the visual sources, we add learned modality embeddings $\mathbf{m}_r \in \mathbb{R}^{d_{\mathrm{cond}}}$, giving $\widetilde{\mathbf{Z}}_r = \mathbf{Z}_r + \mathbf{m}_r$ for $r \in \{\mathrm{vlm}, \mathrm{sam3}, \mathrm{da}\}$. With the robot state projected into d_{cond}-dimensional tokens $\mathbf{Z}_{\mathrm{state}}$, the conditioning sequence is $\mathbf{C} = [\mathbf{Z}_{\mathrm{state}}; \widetilde{\mathbf{Z}}_{\mathrm{vlm}}; \widetilde{\mathbf{Z}}_{\mathrm{sam3}}; \widetilde{\mathbf{Z}}_{\mathrm{da}}]$. The action expert jointly processes $\mathbf{C}$ and noisy action tokens through self-attention, decoding the hidden states at the action-token positions into the predicted action chunk.

Learning objective. We use clean-action (x_0) prediction (Lipman et al., 2022; Ho et al., 2020) across all benchmarks. Let $\mathbf{C}_t$ denote the conditioning sequence constructed from observation $\mathbf{o}_t$. Given a clean action chunk $\mathbf{A}_t$, we sample a noise level $\tau \in [0, 1]$ and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and construct the interpolated action $\mathbf{A}_t^\tau = (1 - \tau)\mathbf{A}_t + \tau\boldsymbol{\epsilon}$. Conditioned on $\mathbf{C}_t$, the action expert f_θ predicts the clean action chunk from $\mathbf{A}_t^\tau$ and is optimized by

$$\mathcal{L}(\theta) = \mathbb{E}_{(\mathbf{o}_t, \mathbf{A}_t) \sim \mathcal{D},\, \tau \sim p(\tau),\, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\| f_\theta(\mathbf{A}_t^\tau, \tau \mid \mathbf{C}_t) - \mathbf{A}_t \|_2^2 \right]. \tag{5}$$

Table 1: Zero-shot LIBERO-Plus success rates (%) for perception-enhanced VLA methods. Avg. is weighted by evaluation counts; w/o Guidance removes mask guidance from TARGET-VLA.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	**Avg.**
Evo-Depth	47.2	49.2	78.9	88.1	76.4	77.6	69.6	69.6
3DThinkVLA	73.8	64.5	78.0	98.4	94.8	84.7	81.5	81.0
V-Link	42.8	59.1	82.8	95.7	94.9	87.8	74.4	75.0
GuidedVLA	73.7	51.4	62.6	94.6	89.0	85.2	79.9	75.4
Spatial Forcing	20.1	13.4	40.9	29.1	33.4	25.7	39.3	29.1
w/o Guidance	74.3	61.4	84.1	95.5	94.6	90.7	79.8	81.9
TARGET-VLA	**89.7**	**64.8**	**89.7**	**99.5**	**99.2**	**96.4**	**87.9**	**88.8**

4 EXPERIMENTS

We evaluate manipulation and generalization in simulation and the real world, then examine the benefits and cost of mask guidance through component and retrieval ablations. Paired attention analysis is provided in Appendix D.

4.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate TARGET-VLA on SIMPLER WidowX (Li et al., 2024b), CALVIN (Mees et al., 2022) (D→D and ABC→D), LIBERO (Liu et al., 2023), and RoboDojo (Chen et al., 2026), and assess zero-shot transfer from LIBERO to LIBERO-Plus (Fei et al., 2025). Real-world evaluation on an AgileX dual-arm platform covers in-domain manipulation, shifted object positions, and object categories absent from the robot demonstrations. Benchmark descriptions and real-world evaluation protocols are provided in Appendices A and C.3, respectively.

Metrics. We report success rate (%) for SIMPLER, LIBERO, LIBERO-Plus, and real-world tasks. LIBERO-Plus averages are weighted by evaluation counts; real-world in-domain averages weight the three tasks equally. RoboDojo reports task-progress score (Score) and success rate (SR). CALVIN uses average task-chain length (0–5): consecutive subtasks completed before the first failure, averaged over 1,000 five-subtask sequences.

Baseline results and reproduction. External baseline results in Tables 1 and 2 are taken from the original papers or other publicly available studies. The exceptions are X-VLA and $\pi_{0.5}$ on CALVIN D→D in Table 2, which we reproduce using their official recommended training settings and the same training dataset and evaluation scripts as TARGET-VLA.

Implementation details. We initialize the vision-language backbone from Florence-2-large (Xiao et al., 2024a) and train a DiT action expert from scratch on the corresponding task-training data, without initialization from a robot-action-pretrained policy. SAM3 and DA3-Base remain frozen, while all remaining parameters are optimized jointly under the action objective. The default configuration uses 64 SAM3 and 40 DA3 condition tokens. The model contains approximately 1.83B parameters, of which 0.89B are trainable. We use AdamW with a global batch size of 256 and perform inference with 10 denoising steps. Task-object prompts are extracted offline using spaCy-based noun-phrase extraction and deterministic dependency rules. For each instruction, the extracted prompts are fixed across evaluation episodes and shared by all variants that use SAM3 masks. Complete model configurations, optimization settings, and evaluation protocols are provided in Appendix C.

4.2 EVALUATION ON SIMULATION BENCHMARKS

Zero-shot robustness on LIBERO-Plus. We first examine whether TARGET-VLA maintains manipulation performance when visual observations and scene configurations change. Policies trained on LIBERO are evaluated on LIBERO-Plus without additional training. Table 1 compares TARGET-VLA and its unguided variant with the perception-enhanced baselines Evo-Depth,

Table 2: Comparison with representative VLA policies across simulation benchmarks. Metrics are success rates (%) except CALVIN (average task-chain length) and RoboDojo Score (task progress). Robot PT denotes policy pretraining on robot-action data beyond the target training set.

Method	Robot	LIBERO	SIMPLER	LIBERO-Plus	CALVIN		RoboDojo	
	PT	Avg.	WidowX	Zero-Shot	D→D	ABC→D	Score	SR (%)
OpenVLA	✓	76.5	8.3	15.6	1.41	-	-	-
UniVLA (Bu et al.)	✓	95.2	-	42.9	-	3.80	-	-
π_0-FAST	✓	85.5	39.5	61.6	-	-	-	-
Discrete Diffusion VLA	✓	96.3	54.2	-	-	-	-	-
HiMoE-VLA	✓	98.0	-	-	3.98	-	-	-
GR00T-N1.6	✓	97.2	63.2	-	-	4.24	-	-
ABot-M0	✓	98.6	-	80.5	-	-	-	-
OpenVLA-OFT	✓	97.1	31.3	69.6	-	-	0.21	0.02
π_0	✓	94.4	27.8	53.6	3.76	3.92	3.48	1.53
FLOWER	✓	95.7	40.0	-	4.35	4.53	-	-
UniVLA (Wang et al.)	✓	95.5	69.8	-	-	4.41	-	-
Being-H0.5	✓	98.9	-	78.5	-	4.48	-	-
StarVLA-α	✗	98.8	64.6	79.7	-	-	6.40	3.24
Xiaomi-Robotics-0	✓	98.7	79.2	-	-	**4.75**	6.93	4.18
X-VLA	✓	98.1	95.8	-	4.21	4.43	10.13	6.52
$\pi_{0.5}$	✓	97.6	46.9	84.4	4.23	4.13	11.41	6.91
TARGET-VLA	✗	**99.1**	**99.0**	**88.8**	**4.36**	4.68	**12.51**	**8.72**

3DThinkVLA, V-Link, GuidedVLA, and Spatial Forcing (Lin et al., 2026; Shi et al., 2026; Lu et al., 2026; Jia et al., 2026; Li et al., 2026b). TARGET-VLA achieves an overall success rate of 88.8%, 7.8 percentage points above the highest reported external baseline value in the table, obtained by 3DThinkVLA. Under camera-view and object-layout changes, it reaches 89.7% and 87.9%, respectively, supporting robustness to changes in how the scene is observed and how objects are arranged. High success rates under lighting, background, and noise perturbations further show that its performance extends across several forms of visual variation.

Manipulation performance across benchmarks. Compared with representative VLA policies (Kim et al., 2024; Pertsch et al., 2025; Liang et al., 2025; Du et al., 2025; Bjorck et al., 2025; Yang et al., 2026; Kim et al., 2025; Black et al., 2024; Reuss et al., 2025; Luo et al., 2026a; Ye et al., 2026a; Cai et al., 2026; Zheng et al., 2026; Black et al., 2025; Bu et al., 2025b; Wang et al., 2026d), TARGET-VLA achieves the highest reported values on six of the seven benchmark metrics in Table 2, without robot-action pretraining. On CALVIN D→D, it attains an average task-chain length of 4.36, compared with 4.21 for X-VLA and 4.23 for $\pi_{0.5}$, which we reproduce using their official recommended training settings and the same training dataset and evaluation scripts as TARGET-VLA. On RoboDojo, it exceeds the strongest reported comparator, $\pi_{0.5}$, by 1.10 in task-progress score and 1.81 percentage points in success rate. On CALVIN ABC→D, its average task-chain length of 4.68 remains competitive with the highest reported value of 4.75. Detailed results are provided in Appendix A.

4.3 Real-World Evaluation

Platform and tasks. We compare TARGET-VLA with $\pi_{0.5}$ on three AgileX dual-arm tasks: Cup Hanging, Block-to-Drawer, and Fruit-to-Plate (Figure 4). Both methods use the same 613 task-specific demonstrations and matched evaluation instructions and initial layouts; detailed protocols are provided in Appendix C.3.

In-domain manipulation performance. TARGET-VLA achieves higher observed success rates than $\pi_{0.5}$ across all three in-domain tasks, with task-averaged success of 83.3% versus 70.6% and per-task gains of 10.0–15.0 percentage points (Figure 3).

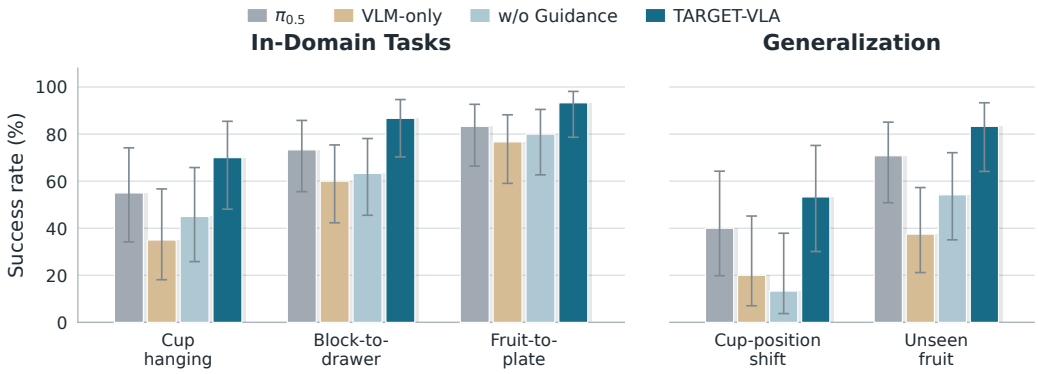


Figure 3: Real-world performance on AgileX. Bars show success rates with Wilson 95% confidence intervals: 20/30/30 trials for the three in-domain tasks and 15/24 for position shifts/unseen fruits. The w/o Guidance variant retains both complementary representations without mask guidance.

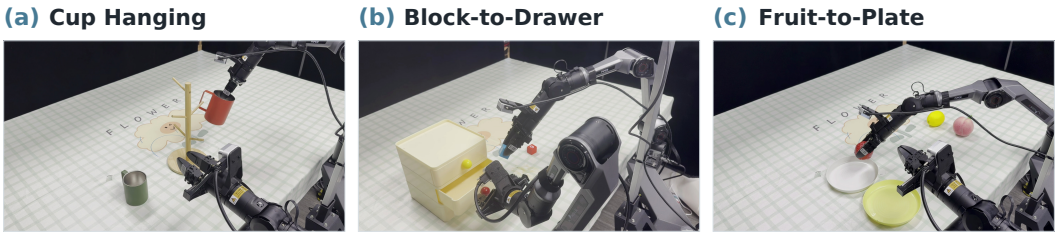


Figure 4: Real-world tasks on AgileX. Execution sequences for in-domain and generalization settings are provided in Appendix C.5.

Generalization to shifted positions and unseen objects. The observed gains extend to both evaluated transfer settings without further training (Figure 3). Under cup-position shifts of up to 10 cm with the rack fixed, TARGET-VLA achieves 53.3% success, compared with 40.0% for $\pi_{0.5}$. On Fruit-to-Plate with avocado and mango, which are absent from the robot demonstrations, TARGET-VLA achieves 83.3% success versus 70.8% for $\pi_{0.5}$.

4.4 Ablation Studies

All CALVIN variants retain VLM conditioning and share the training data and optimization schedule. They are evaluated on the same 1,000 task chains, following Appendix C.4.

Effectiveness of mask guidance. The w/o Guidance variant retains both frozen visual sources and 104 condition tokens, but disables mask-dependent attention biases and DA3 validity gating and embeddings. On CALVIN ABC→D, unguided retrieval achieves an average task-chain length of 4.35, only slightly above VLM-only conditioning (4.29), whereas TARGET-VLA reaches 4.68 (Table 3). The paired bootstrap 95% confidence interval for the gain over w/o Guidance is [0.23, 0.43]. Five-subtask completion also increases from 74.3% to 84.3%, indicating improved completion of full task chains.

Visual feature integration strategies. To distinguish the proposed guidance design from simply providing task-conditioned mask information, we compare against Query retrieval + mask input. This variant concatenates the same predicted masks with spatial features before projection, without adding mask-dependent attention biases. With the same frozen visual sources and 104 final condition tokens, TARGET-VLA achieves an average task-chain length of 4.68, compared with 4.45 for the mask-input variant (Table 3). The advantage also extends to full-chain completion, with Success@5 increasing from 77.0% to 84.3%. These results support the effectiveness of the proposed guidance design beyond supplying task-conditioned masks as additional feature inputs. As another integration

Table 3: CALVIN ABC→D ablations with VLM conditioning. Success@k: first-k subtask completion (%); Avg. Len.: mean task-chain length (0–5). w/o Guidance removes mask-dependent attention biases and validity conditioning.

Variant	Success@k (%) ↑					Avg. Len. ↑
	1	2	3	4	5	
Component ablations						
VLM-only	97.1	93.4	86.9	79.5	72.1	4.29
SAM3, w/o Guidance	98.5	94.4	88.8	82.0	75.0	4.39
DA3, w/o Guidance	98.3	94.2	86.7	80.8	73.7	4.34
w/o Guidance	98.0	93.4	88.0	80.9	74.3	4.35
SAM3 + Guidance	98.4	94.9	89.8	84.6	77.4	4.45
DA3 + Guidance	98.0	94.6	89.0	83.5	76.7	4.42
Retrieval alternatives (SAM3 + DA3)						
Downsample + concatenate	**99.8**	96.2	89.8	82.9	74.7	4.43
Query retrieval + mask input	98.9	95.5	89.2	83.9	77.0	4.45
TARGET-VLA	99.5	**96.4**	**94.7**	**92.7**	**84.3**	**4.68**

baseline, Downsample + concatenate achieves 4.43 despite using 164 condition tokens, compared with 104 for TARGET-VLA.

Choice of visual sources. We compare SAM3-only, DA3-only, and dual-source retrieval with 64, 40, and 104 condition tokens, respectively. DA3 + Guidance uses SAM3 masks but no SAM3 condition tokens; DA3 features and predicted log-depth are evaluated jointly. Guidance improves average task-chain length from 4.39 to 4.45 for SAM3 and from 4.34 to 4.42 for DA3 (Table 3). The dual-source configuration achieves the highest average task-chain length among these variants (4.68), exceeding guided SAM3 and DA3 retrieval by 0.23 and 0.26. The corresponding paired bootstrap 95% confidence intervals are [0.13, 0.32] and [0.16, 0.36].

Transfer and real-world ablations. We compare the LIBERO-trained guided and unguided policies on LIBERO-Plus without additional training, keeping the visual sources and condition-token budget fixed. Guidance increases weighted success from 81.9% to 88.8% (Table 1), with gains across all seven perturbation categories. The largest improvements occur under camera-view (+15.4 percentage points) and object-layout changes (+8.1 points), while robot-initial-state perturbations remain challenging at 64.8% success. Real-world variants use the same demonstrations and matched evaluation layouts (Appendix C.3). TARGET-VLA achieves higher observed success rates than w/o Guidance in all five evaluated settings (Figure 3): in-domain task-averaged success increases from 62.8% to 83.3%, with gains of 40.0 and 29.2 percentage points under position shifts and unseen fruit categories, respectively.

Inference cost. Under the standardized inference settings in Table 11, TARGET-VLA requires 85.3 ms per forward pass, compared with 80.7 ms for w/o Guidance and 56.0 ms for VLM-only conditioning. The complete guidance configuration therefore adds 4.6 ms (5.7%) over the unguided variant. Both retrieval variants report 4.5 GB peak allocated GPU memory, compared with 2.4 GB for VLM-only conditioning.

5 Conclusion

We introduced TARGET-VLA, which integrates frozen segmentation and depth representations into VLA policies through mask-guided retrieval. Prompt-conditioned masks supply spatial guidance, while pretrained features provide visual content for action generation. Experiments demonstrate improved manipulation performance over unguided retrieval, including zero-shot evaluation on LIBERO-Plus and real-world transfer under position and object variations. These findings highlight task-conditioned spatial guidance as an effective way to incorporate pretrained visual representations into robot action generation.

DISCLOSURE OF LLM USE

We used generative AI tools to assist with manuscript drafting and editing, and code revision. The authors reviewed the AI-assisted material incorporated into this work and take full responsibility for the final manuscript, code, and reported findings.

REPRODUCIBILITY STATEMENT

Section 3 describes the model and learning objective. Appendix C documents the model configuration, prompt construction, optimization settings, and evaluation protocols. Upon acceptance, we will publicly release our training and evaluation code and scripts, together with all model checkpoints used for the reported benchmark evaluations.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikołaj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://papers.neurips.cc/paper_files/paper/2022/hash/960a172bc7fbf0177ccccbb411a7d800-Abstract-Conference.html.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *9th Annual Conference on Robot Learning*, 2025.

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.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025a.

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

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, et al. Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution. *arXiv preprint arXiv:2602.12684*, 2026.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

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

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Chilam Cheang, Sijin Chen, Zhongren Cui, Yingdong Hu, Liqun Huang, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Xiao Ma, et al. Gr-3 technical report. *arXiv preprint arXiv:2507.15493*, 2025.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, et al. Robodojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1290–1299, 2022.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Zhiying Du, Bei Liu, Yaobo Liang, Yichao Shen, Haidong Cao, Xiangyu Zheng, Zhiyuan Feng, Zuxuan Wu, Jiaolong Yang, and Yu-Gang Jiang. Himoe-vla: Hierarchical mixture-of-experts for generalist vision-language-action policies. *arXiv preprint arXiv:2512.05693*, 2025.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Haifeng Huang, Xinyi Chen, Yilun Chen, Hao Li, Xiaoshen Han, Zehan Wang, Tai Wang, Jiangmiao Pang, and Zhou Zhao. Roboground: Robotic manipulation with grounded vision-language priors. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22540–22550. IEEE, 2025.

Xiaosong Jia, Bowen Yang, Zuhao Ge, Xian Nie, Yuchen Zhou, Cunxin Fan, Yufeng Li, Yilin Chai, Chao Jing, Zijian Liang, et al. Guidedvla: Specifying task-relevant factors via plug-and-play action attention specialization. *arXiv preprint arXiv:2605.12369*, 1, 2026.

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.

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

Chengmeng Li, Junjie Wen, Yaxin Peng, Yan Peng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. *IEEE Robotics and Automation Letters*, 11(3):2506–2513, 2026a.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 132324–132345, 2026b.

Jinming Li, Yichen Zhu, Zhibin Tang, Junjie Wen, Minjie Zhu, Xiaoyu Liu, Chengmeng Li, Ran Cheng, Yaxin Peng, Yan Peng, and Feifei Feng. Coa-vla: Improving vision-language-action models via visual-text chain-of-affordance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9759–9769, October 2025a.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023. URL https://proceedings.mlr.press/v202/li23q.html.

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*, 2024a.

Xiaoqi Li, Liang Heng, Jiaming Liu, Yan Shen, Chenyang Gu, Zhuoyang Liu, Hao Chen, Nuowei Han, Renrui Zhang, Hao Tang, et al. 3ds-vla: A 3d spatial-aware vision language action model for robust multi-task manipulation. In *9th Annual Conference on Robot Learning*, 2025b.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024b.

Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Liuao Pei, Tian Nian, Shunbo Zhou, Xiaokang Yang, Jiangmiao Pang, et al. Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint arXiv:2508.20072*, 2025.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025a.

Tao Lin, Gen Li, Yilei Zhong, Yanwen Zou, and Bo Zhao. Evo-0: Vision-language-action model with implicit spatial understanding. *arXiv preprint arXiv:2507.00416*, 2025b.

Tao Lin, Yuxin Du, Jiting Liu, Nuobei Zhu, Yunhe Li, Yuqian Fu, Yinxinyu Chen, Hongyi Cai, Zewei Ye, Bing Cheng, et al. Evo-depth: A lightweight depth-enhanced vision-language-action model. *arXiv preprint arXiv:2605.14950*, 2026.

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

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.

Yicheng Liu, Zibin Dong, Baijun Ye, Tianyuan Yuan, Tao Jiang, Anqi Yang, Shicheng Cao, Haonan Liu, Yue Sun, Zihan Guo, Xiao Liu, Ke Dong, Changxun Pan, Chenru Wu, Tailai Cheng, Xiaoshu Ren, Xinlei Zhang, Jianning Cui, Zijie Zhao, Haoyu Zhang, Kaiming Xu, Haodong Yang, Bowen Zhang, Jiahui Niu, Shaoting Zhu, Shiduo Zhang, and Hang Zhao. G0.5: One autoregressive stream for robot reasoning and action, 2026a. URL https://arxiv.org/abs/2608.11739.

Zonghe Liu, Shanyuan Jie, Xiaoquan Sun, Chen Cao, Zetian Xu, Zongsheng Liu, and Jiayu Chen. Sam3d-guided object-centric representation alignment for vision-language-action models. *arXiv preprint arXiv:2607.25912*, 2026b.

Yehao Lu, Jiarui Yang, Yuning Su, Yufeng Xie, Yu Zhong, Yazhou Zhang, Haiyu Lan, Kaixiang Lu, Peiwen Lin, Chuang Wang, et al. V-link: Recovering lost visual representations in action dit for vision-language-action models. *arXiv preprint arXiv:2608.25308*, 2026.

Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, et al. Being-h0. 5: Scaling human-centric robot learning for cross-embodiment generalization. *arXiv preprint arXiv:2601.12993*, 2026a.

Yulin Luo, Hao Chen, Zhuangzhe Wu, Bowen Sui, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Qiuxuan Feng, Jiale Yu, Shuo Gu, et al. Look before acting: Enhancing vision foundation representations for vision-language-action models. *arXiv preprint arXiv:2603.15618*, 2026b.

Haoxiang Ma, Junhao Cai, Xiaoxu Xu, Hao Li, Yuyin Yang, Yang Tian, Jiafei Cao, Hongrui Zhu, Zherui Qiu, Yuqiang Yang, et al. Internvla-a1. 5: Unifying understanding, latent foresight, and action for compositional generalization. *arXiv preprint arXiv:2607.04988*, 2026.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

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

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.

Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies. *arXiv preprint arXiv:2509.04996*, 2025.

Jiaxin Shi, Xidong Zhang, Fucai Zhu, Zhe Li, Siyu Zhu, and Weihao Yuan. 3dthinkvla: Endowing vision-language-action models with latent 3d priors via 3d-thinking-guided co-training. *arXiv preprint arXiv:2606.04436*, 2026.

Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. Reconvla: Reconstructive vision-language-action model as effective robot perceiver. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18549–18557, 2026.

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.

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.

Spirit AI Team. Spirit-v1.5: Clean data is the enemy of great robot foundation models. *Spirit AI Blog*, 2026. https://www.spirit-ai.com/en/blog/spirit-v1-5.

Chenyi Wang, Xinkai Wang, Bokai Lin, Jialin Tian, Fucheng Zhang, Cewu Lu, and Lixin Yang. Track4action: Distilling world-centric 3d tracker into vision-language-action policies. *arXiv preprint arXiv:2608.03727*, 2026a.

Fangyuan Wang, Peng Zhou, Jiaming Qi, Shipeng Lyu, David Navarro-Alarcon, and Guodong Guo. Think proprioceptively: Embodied visual reasoning for vla manipulation. *arXiv preprint arXiv:2602.06575*, 2026b.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025a.

Qiuyue Wang, Mingsheng Li, Jian Guan, Jinhui Ye, Sicheng Xie, Yitao Liu, Junhao Chen, Zhixuan Liang, Jie Zhang, Xintong Hu, et al. Qwen-vla: Unifying vision-language-action modeling across tasks, environments, and robot embodiments. *arXiv preprint arXiv:2605.30280*, 2026c.

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5261–5271. IEEE, 2025b.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 80929–80944, 2026d.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4818–4829. IEEE, 2024a.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024b.

Junjin Xiao, Dongyang Li, Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Feng Xiong, Mu Xu, Xing Wei, Zhiheng Ma, et al. Learning action manifold with multi-view latent priors for robotic manipulation. *arXiv preprint arXiv:2605.11832*, 2026.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026.

Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. Starvla-α: Reducing complexity in vision-language-action systems. *arXiv preprint arXiv:2604.11757*, 2026a.

Jinhui Ye, Fangjing Wang, Ning Gao, Junqiu Yu, Yangkun Zhu, Bin Wang, Jinyu Zhang, Weiyang Jin, Yanwei Fu, Feng Zheng, et al. St4vla: Spatially guided training for vision-language-action models. *arXiv preprint arXiv:2602.10109*, 2026b.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, et al. Qwen-robotmanip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026.

Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. Depthvla: Enhancing vision-language-action models with depth-aware spatial reasoning. *arXiv preprint arXiv:2510.13375*, 2025.

He Zhang, Lingzhu Xiang, Haitao Lin, Zeyu Huang, Minghui Wang, Dingyan Zhong, Yubo Dong, Yihao Wu, Yongming Rao, Dongsheng Zhang, et al. Hy-embodied-0.5-vla: From vision-language-action models to a real-world robot learning stack. *arXiv preprint arXiv:2606.14409*, 2026a.

Jesse Zhang, Marius Memmel, Kevin Kim, Dieter Fox, Jesse Thomason, Fabio Ramos, Erdem Bıyık, Abhishek Gupta, and Anqi Li. Peek: Guiding and minimal image representations for zero-shot generalization of robot manipulation policies. *arXiv preprint arXiv:2509.18282*, 2025.

Kaidong Zhang, Jian Zhang, Rongtao Xu, Yu Sun, Shuoshuo Xue, Youpeng Wen, Xiaoyu Guo, Minghao Guo, Weijia Liufu, Liu Zihou, et al. A1: A fully transparent open-source, adaptive and efficient truncated vision-language-action model. *arXiv preprint arXiv:2604.05672*, 2026b.

Yuan Zhang, Shiqi Zhang, Yedong Shen, Shuai Dong, Jiajun Deng, Xin Zhang, Yuxuan Gao, Jiajia Wu, Xin Nie, Zhiyuan Cheng, et al. Gear-vla: Learning geometry-aware action representations for generalizable robotic manipulation. *arXiv preprint arXiv:2606.08530*, 2026c.

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.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *International Conference on Learning Representations*, volume 2025, pp. 54277–54296, 2025.

Jiaying Zhou, Zhihao Zhan, Ruifeng Zhai, Qinhan Lyu, Hao Liu, Keze Wang, Liang Lin, and Guangrun Wang. Tag: Target-agnostic guidance for stable object-centric inference in vision-language-action models. *arXiv preprint arXiv:2603.24584*, 2026.

Yifeng Zhu, Zhenyu Jiang, Peter Stone, and Yuke Zhu. Learning generalizable manipulation policies with object-centric 3d representations. *arXiv preprint arXiv:2310.14386*, 2023.

A Detailed Simulation Results

We use the official evaluation settings for each simulation benchmark. The following tables report per-task, per-suite, or per-capability results for TARGET-VLA.

SIMPLER WidowX. We evaluate four tasks: putting a spoon on a towel, putting a carrot on a plate, stacking a green block on a yellow block, and putting an eggplant in a yellow basket.

Table 4: Success rates (%) on the four SIMPLER WidowX tasks.

Method	Put Spoon	Put Carrot	Stack Block	Put Eggplant	Avg.
TARGET-VLA	100.0	95.8	100.0	100.0	**99.0**

CALVIN. D→D uses environment D for both training and evaluation, whereas ABC→D trains on environments A, B, and C and evaluates on the unseen environment D.

Table 5: Task-chain performance on CALVIN D→D. Success@k denotes the percentage of sequences completing the first k subtasks; Avg. Len. is the average task-chain length (0–5).

Method	Success@k (%) ↑					Avg. Len. ↑
	1	2	3	4	5	
TARGET-VLA	96.5	92.3	87.4	82.5	77.0	**4.36**

Table 6: Task-chain performance on CALVIN ABC→D, using the metrics in Table 5.

Method	Success@k (%) ↑					Avg. Len. ↑
	1	2	3	4	5	
TARGET-VLA	99.5	96.4	94.7	92.7	84.3	**4.68**

LIBERO and LIBERO-Plus. The four LIBERO suites evaluate spatial relations, object variation, different task goals, and long-horizon manipulation. We evaluate the LIBERO-trained policy zero-shot on LIBERO-Plus using its official settings. LIBERO-Plus evaluates perturbations to camera viewpoints, robot initial states, language instructions, lighting, backgrounds, image noise, and object layouts. Category-level results are reported in Table 1, with the overall success rate weighted by evaluation counts.

Table 7: Success rates (%) across the four LIBERO suites.

Method	Spatial	Object	Goal	Long	Avg.
TARGET-VLA	99.0	100.0	99.6	97.8	**99.1**

RoboDojo. The simulation suite tests generalization, precision, long-horizon execution, memory, and open-vocabulary instruction following. Score measures task progress, while SR measures task success; each overall result is the equally weighted mean across the five capabilities.

Table 8: RoboDojo results by capability. Open denotes open-vocabulary instruction following.

Method	Metric	Generalization	Precision	Long-Horizon	Memory	Open	Avg.
TARGET-VLA	Score	13.82	20.60	21.80	5.83	0.50	**12.51**
	SR (%)	10.17	15.25	13.00	4.67	0.50	**8.72**

B Exploratory Feature-Integration Variants

We report two exploratory extensions: auxiliary prediction added to downsample-and-concatenate, and sink tokens added to w/o Guidance. Table 9 includes each underlying variant, along with VLM-only and TARGET-VLA as references. All variants share the VLM backbone, training data, and optimization schedule; those using complementary representations share the frozen SAM3 and DA3 encoders. Results use the final 100k checkpoint on CALVIN ABC→D, following Appendix C.4.

Auxiliary feature and depth prediction. For the Downsample + concatenate variant, two convolutional heads read the hidden states at the SAM3 and DA3 token positions in the final action-expert block. The SAM3 head reshapes its 100 tokens into a 10×10 grid and reconstructs the corresponding frozen SAM3 FPN features on a 20×20 grid using a 1×1 projection, one stride-2 transposed convolution, and a 3×3 output convolution, with 256 hidden channels. The DA3 head reshapes its 64 tokens into an 8×8 grid and predicts a 32×32 log-depth map using a 1×1 projection, two stride-2 transposed convolutions with 128 and 64 output channels, and a 3×3 output convolution. Both heads use GroupNorm and GELU before the output convolution.

The SAM3 loss is the mean per-location cosine distance across feature channels. For DA3, the frozen log-depth target is bilinearly resized to the prediction grid. Prediction and target are each centered by their spatial mean and divided by their spatial standard deviation plus 10^{-4}, independently per sample, before computing mean squared error. The SAM3 and DA3 losses have weights 0.05 and 0.1 and are added to the action objective. Both targets are detached from the computation graph.

Additional sink tokens. Softmax attention distributes unit mass across all feature positions, potentially assigning attention to irrelevant regions when no feature strongly matches a query. Inspired by attention sinks (Xiao et al., 2024b), we append four learnable tokens to each projected SAM3 and DA3 feature sequence. The condition queries attend jointly to the spatial features and these non-spatial key–value tokens, providing an alternative destination for attention mass.

Table 9: Exploratory variants on CALVIN ABC→D. Avg. Len. is the average task-chain length (0–5); indented rows add an extension to the preceding variant.

Variant	Avg. Len. ↑
VLM-only	4.29
Downsample + concatenate	4.43
+ auxiliary prediction	4.43
w/o Guidance	4.35
+ sink tokens	4.45
TARGET-VLA	**4.68**

Auxiliary prediction achieves the same reported average task-chain length as its baseline (4.43), while sink tokens improve the unguided variant from 4.35 to 4.45 but remain below TARGET-VLA (4.68).

C Implementation and Evaluation Details

This appendix documents the default model configuration, optimization settings, and evaluation procedures. Unless otherwise specified, the settings below are shared across benchmarks.

C.1 Model Configuration

Pretrained models and action expert. We initialize the vision-language backbone from `florence-community/Florence-2-large`, remove its language decoder, and use the encoder outputs as VLM tokens that condition the action expert. SAM3 and DA3 are initialized

from `facebook/sam3` and `depth-anything/da3-base`, respectively, and remain frozen throughout training. The action expert uses a DiT architecture with 24 Transformer blocks, a hidden dimension of 1,024, and 16 attention heads. Noise-level embeddings are added to the projected conditioning tokens. All reported TARGET-VLA variants use clean-action (x_0) prediction across benchmarks, with 10 denoising steps at inference.

Visual feature extraction and spatial sources. SAM3 and DA3 process the main-view image at resolutions of 280×280 and 224×224, respectively. For SAM3, we use the penultimate FPN hidden state, `fpn_hidden_states[-2]`, retaining its spatial organization for query-based retrieval. This feature map is resized to 80×80 and passed through an MLP-based encoder to form the SAM3 source tokens. For DA3, we use `fused_main`, the multi-scale representation produced by the frozen DualDPT feature-fusion stage immediately before the depth-prediction head, together with the log-depth decoded from this feature. The fused feature and log-depth are resized to 40×40 and passed through an MLP-based encoder to form the DA3 source tokens.

Query-transformer configuration. The SAM3 Mask-Guided Query Transformer uses one block with 64 learnable queries. The DA3 entity streams share a single mask-guided block and a set of 32 base queries, with separate entity-type embeddings for the end effector and objects. Each entity stream produces 32 tokens; a query aggregation block uses 40 learnable queries to aggregate their 64 concatenated tokens without a spatial attention bias. All three blocks use an internal dimension of $d_{\mathrm{ret}} = 256$, eight attention heads, and a two-layer feed-forward network with 1,024 hidden units and GELU activation. Each block applies query self-attention, cross-attention, and the feed-forward network with pre-layer normalization and residual connections. The two-layer output MLPs ψ_{sam3} and ψ_{da} project the branch outputs to the common conditioning-token dimension $d_{\mathrm{cond}} = 1024$, yielding 64 SAM3 and 40 DA3 tokens.

Prompt construction and detection validity. Task-object prompts are extracted offline using spaCy-based noun-phrase extraction followed by deterministic dependency rules, then cached before training and evaluation. Each instruction uses up to two task-object prompts plus the fixed end-effector prompt. For each prompt, SAM3's semantic segmentation logits are passed through a sigmoid to form a soft mask. A prompted entity is valid if its maximum detection logit is at least 0; otherwise, its mask is set to zero. The object mask is the pixelwise maximum of the valid task-object masks, and object validity indicates that at least one such prompt is detected. The SAM3 guidance mask is the pixelwise maximum of the end-effector and object masks. If neither the end effector nor any prompted object is detected, the SAM3 attention bias is spatially constant and does not change the softmax distribution; SAM3 tokens are not zeroed. In the DA3 branch, intermediate tokens for an undetected entity are set to zero before Query Aggregation. A learned linear projection of the two validity indicators $[v_{\mathrm{eef}}, v_{\mathrm{obj}}]$ is added to the final DA3 condition tokens.

Mask-guidance parameters. SAM3 retrieval and DA3 entity retrieval each use a learned scalar guidance strength λ, initialized to 1. The DA3 scalar is shared between the end-effector and object streams. We use $\delta = 10^{-6}$ in the logarithmic mask bias. For the final CALVIN ABC→D checkpoint used in evaluation, the learned guidance strengths are $\lambda_{\mathrm{sam3}} = 0.9595$ and $\lambda_{\mathrm{da}} = 0.9932$.

C.2 Training and Inference Settings

Hardware and training budget. Training uses 8 NVIDIA A100 GPUs with 80 GB memory each, a global batch size of 256, and a per-GPU batch size of 32. We use BF16 mixed precision and gradient checkpointing. By default, simulation policies are trained for 100k optimization steps and real-world policies for 50k steps. For each training observation–action pair, we sample four independent noise-level and Gaussian-noise pairs, reusing the visual conditioning across the four samples. Visual conditions are computed once and shared across copies. Unless stated otherwise, we sample $z \sim \mathcal{N}(0, 1)$ and set $\tau = 0.001 + 0.999\,\sigma(z)$, where σ is the sigmoid function.

Optimization and action representation. We use AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The Florence backbone uses a learning rate of 10^{-5} and no weight decay; the remaining trainable modules use a learning rate of 10^{-4} and weight decay of 0.01 for decay-eligible parameters. During the first 1,000 steps, the backbone learning rate is held at zero, while the learning rate of the remaining

modules is linearly warmed up. Both groups subsequently follow cosine decay to 10% of their respective peak learning rates. Gradients are clipped to a maximum norm of 1.0. The action loss is evaluated only over the valid dimensions of each robot's action representation. Simulation policies use end-effector actions with a 6D rotation representation, while the AgileX policy uses joint-angle actions.

Table 10: Benchmark-specific training configurations. H denotes both the predicted action-chunk length and the execution horizon used during benchmark evaluation. LIBERO-Plus uses the LIBERO-trained policy without additional training.

Benchmark	Training Steps	H	Data Augmentation	Action Space
SIMPLER WidowX	100k	10	ColorJitter	EEF (6D rot.)
CALVIN D→D	100k	10	Crop + ColorJitter	EEF (6D rot.)
CALVIN ABC→D	100k	10	Crop + ColorJitter	EEF (6D rot.)
LIBERO	100k	16	ColorJitter	EEF (6D rot.)
RoboDojo	100k	30	ColorJitter	EEF (6D rot.)
AgileX real world	50k	30	ColorJitter	Joint Angles

Inference benchmarking. Table 11 reports inference cost on an RTX 4090. VLM-only, w/o Guidance, and TARGET-VLA use BF16, batch size 1, 10 denoising steps, and `torch.compile`. Latency covers the complete model forward pass, including visual encoding, mask generation where applicable, and action generation. We report the mean and standard deviation over 100 measurements after compilation and warm-up; memory is peak allocated GPU memory. The three-view, 50-step action-chunk configuration is shared across the timing comparison, including the optimized JAX $\pi_{0.5}$ reference. This is a standardized timing setup, not the AgileX deployment configuration, whose horizon is 30 (Table 10).

Table 11: Inference cost on an RTX 4090 with three observation views and 50-step action chunks. Latency is reported as mean $\pm$ standard deviation. The official optimized JAX implementation of $\pi_{0.5}$ is included as an implementation-specific reference rather than a controlled comparison of architectural efficiency.

Variant	Latency (ms)	GPU Memory (GB)
VLM-only	56.0 ± 0.3	2.4
w/o Guidance	80.7 ± 0.5	4.5
TARGET-VLA	85.3 ± 0.6	4.5
$\pi_{0.5}$ (reference)	63.7 ± 0.3	8.5

C.3 Real-World Data and Evaluation Protocols

Platform and demonstrations. The AgileX dual-arm platform uses one fixed camera and two wrist-mounted cameras. The demonstration dataset contains 121 Cup Hanging trajectories, 306 Block-to-Drawer trajectories, and 186 Fruit-to-Plate trajectories. The drawer tasks cover six combinations of block and ball colors, while the fruit tasks cover strawberry, lemon, and peach with white and yellow plates.

In-domain evaluation and success criteria. Each method is evaluated on 20 Cup Hanging trials, 30 Block-to-Drawer trials, and 30 Fruit-to-Plate trials using demonstrated object categories and in-domain configurations. The latter two tasks each include five trials per instruction combination. Instructions and initial layouts are matched across methods.

Success requires releasing both cups onto rack hooks by their handles, placing the specified block inside the drawer containing the specified ball, or placing the specified fruit on the specified plate, respectively. Success is assessed after release, without human intervention.

Position changes. Both cups are jointly translated to positions within a 10 cm neighborhood of their reference locations, while preserving their relative arrangement and keeping the rack fixed. We evaluate 15 layouts that remain within the fixed camera's field of view and the robot's reachable workspace. The same layouts are used for all methods.

Unseen fruits. Avocado and mango are absent from the robot demonstration dataset. Combining each fruit with the white and yellow plates yields four settings: avocado-to-white, avocado-to-yellow, mango-to-white, and mango-to-yellow. Each setting is evaluated six times, totaling 24 trials per method, with initial layouts matched across methods. Task-object prompts are extracted from the evaluation instructions using the same offline procedure as for training.

Method comparison and reporting. We compare TARGET-VLA with $\pi_{0.5}$ and evaluate VLM-only and w/o Guidance as ablations. All methods use the same task-specific demonstration data and matched evaluation layouts, with 50,000 training or fine-tuning steps. We initialize $\pi_{0.5}$ from its official released checkpoint. TARGET-VLA and its ablation variants initialize the VLM from its pretrained checkpoint, without robot-action-pretrained policy weights; variants using SAM3 or DA3 also load the corresponding pretrained encoder weights.

Table 12: Real-world success rates (%), with successful/total trials in parentheses.

Setting	Task	$\pi_{0.5}$	VLM-only	w/o Guidance	TARGET-VLA
In-domain	Cup Hanging	55.0 (11/20)	35.0 (7/20)	45.0 (9/20)	**70.0 (14/20)**
	Block-to-Drawer	73.3 (22/30)	60.0 (18/30)	63.3 (19/30)	**86.7 (26/30)**
	Fruit-to-Plate	83.3 (25/30)	76.7 (23/30)	80.0 (24/30)	**93.3 (28/30)**
Generalization	Cup-position shift	40.0 (6/15)	20.0 (3/15)	13.3 (2/15)	**53.3 (8/15)**
	Unseen fruit	70.8 (17/24)	37.5 (9/24)	54.2 (13/24)	**83.3 (20/24)**

C.4 Ablation Protocols

Controlled-ablation protocol. All component and retrieval variants in Table 3 use the same training seed, CALVIN ABC→D data, and optimization schedule (Appendix C.2). We report the final 100k checkpoint and evaluate all variants on the same 1,000 task chains with matched environment seeds and sequence order. The paired comparisons in Section 4.4 jointly resample aligned sequence outcomes 10,000 times to obtain percentile 95% confidence intervals for the mean difference. These intervals measure uncertainty over evaluation sequences. Within each guided–unguided component comparison, the visual sources and condition-token budget are matched. Unguided variants remove mask guidance; in the DA3 branch, this includes entity-validity gating and validity embeddings in addition to the attention bias.

Query retrieval + mask input. This variant concatenates soft mask values with spatial features before projection into source tokens, without adding mask-dependent biases to the attention logits. It uses the same predicted masks and final condition-token counts as TARGET-VLA, retaining task-conditioned spatial information through the instruction-derived object masks.

C.5 Real-World Execution Sequences

Figures 5 and 6 show execution sequences for the three in-domain tasks and the position and object generalization settings described in Appendix C.3. Each task is illustrated by six chronological keyframes, with its instruction shown above the sequence.

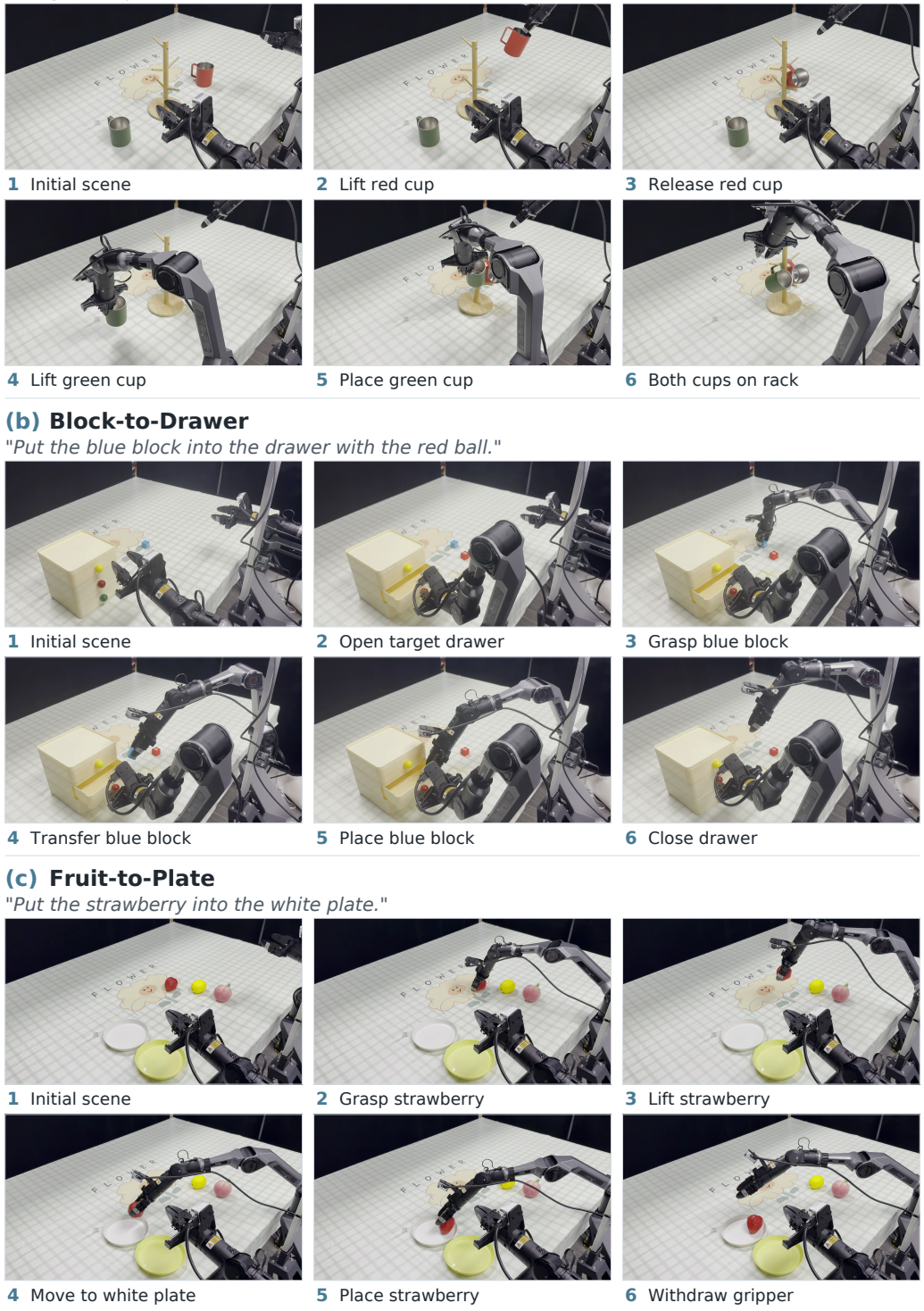


Figure 5: In-domain execution sequences on AgileX: Cup Hanging, Block-to-Drawer, and Fruit-to-Plate. Within each task, frames proceed from left to right across the top row, followed by the bottom row.

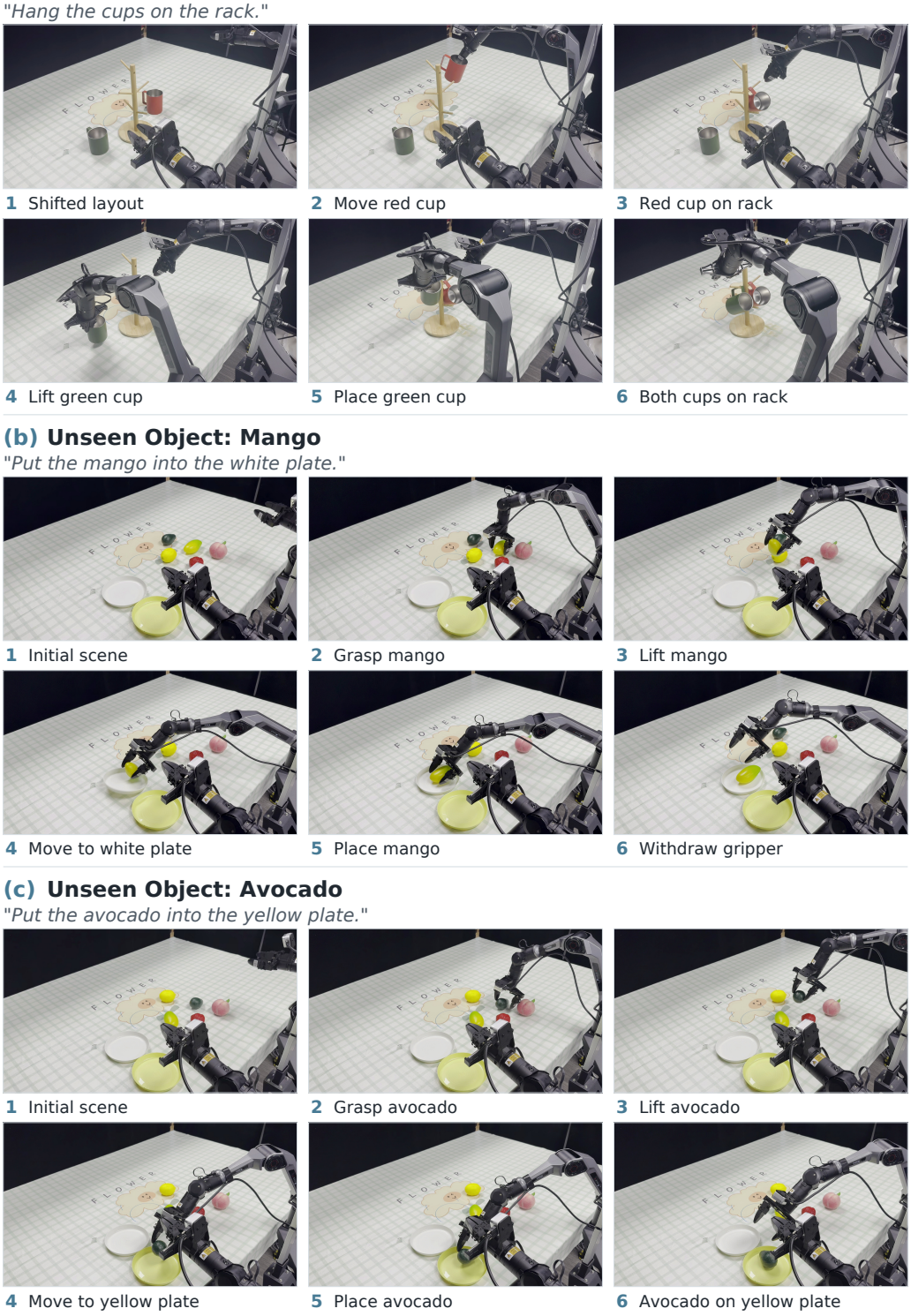


Figure 6: Execution sequences with shifted cup positions and unseen fruit categories (mango and avocado). Frames follow the same temporal order as in Figure 5.

D ATTENTION ANALYSIS AND VISUALIZATION

Attention extraction and visualization. The SAM3 heatmaps visualize cross-attention from the SAM3 queries to the projected SAM3 spatial features. The DA3 heatmaps visualize the first-stage cross-attention from the end-effector and object queries to the shared DA3 source tokens. In the additional CALVIN visualizations, the two DA3 entity maps are smoothed and normalized separately, then combined with equal weight for display. Each displayed map is independently normalized, upsampled, and overlaid on the observation, with blue-to-red colors indicating increasing relative attention. Colors therefore describe the spatial pattern within each map rather than absolute attention magnitudes across panels. The same rendering procedure is applied to both variants; quantitative statistics use the unprocessed attention probabilities and evaluate the DA3 entity maps separately.

Quantitative attention analysis. We measure how mask guidance changes the spatial allocation of query-to-feature attention in the SAM3 and DA3 retrieval branches. To hold the policy input fixed, we randomly sample 500 observations from TARGET-VLA's CALVIN ABC→D evaluation and save the robot state, language instruction, and camera observations. We then run the w/o Guidance variant offline on the same inputs, producing paired attention maps for all 500 observations. The target masks are the same soft masks used to guide retrieval in TARGET-VLA.

For a spatial attention map, let a_p denote the attention weight at position p, normalized such that $\sum_{p=1}^{P} a_p = 1$, and let $m_p \in [0, 1]$ denote the corresponding soft target mask. We define target attention mass (TAM) as

$$\mathrm{TAM} = \sum_{p=1}^{P} a_p m_p, \tag{6}$$

where P is the number of spatial positions. TAM lies in $[0, 1]$ and measures attention weighted by the soft target mask; for a binary mask, it is the total attention mass inside the masked region. For SAM3, the target is the combined mask $\mathbf{M}_{\mathrm{sam3}} = \max(\mathbf{M}_{\mathrm{eef}}, \mathbf{M}_{\mathrm{obj}})$ used for retrieval guidance. For DA3, the end-effector and object retrieval maps are evaluated against their respective masks. We first average TAM over the layer, head, and query dimensions within each observation and then average over the 500 observations. Table 13 reports $100 \times \mathrm{TAM}$ as a percentage.

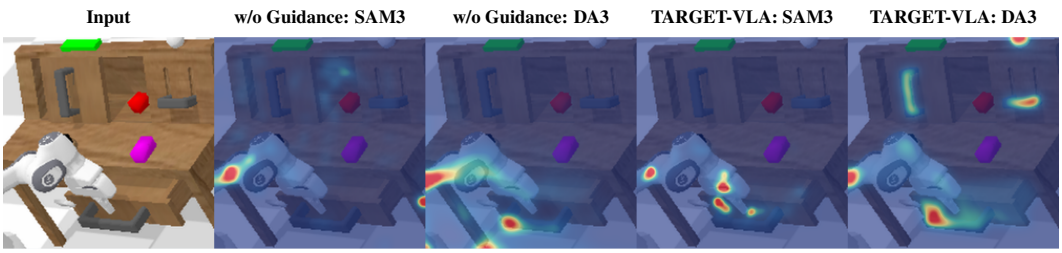
We compare TAM between policies on each matched observation, holding scene state and mask support fixed. Because absolute TAM depends on mask size and softness, we interpret the within-pair change as a measure of attention redistribution. We estimate uncertainty in the mean paired difference using 10,000 paired bootstrap resamples and report percentile 95% confidence intervals.

Table 13: Target attention mass (TAM, %) on 500 matched CALVIN ABC→D observations. Δ is TARGET-VLA minus w/o Guidance, with paired-bootstrap 95% confidence intervals. Differences are computed before rounding.

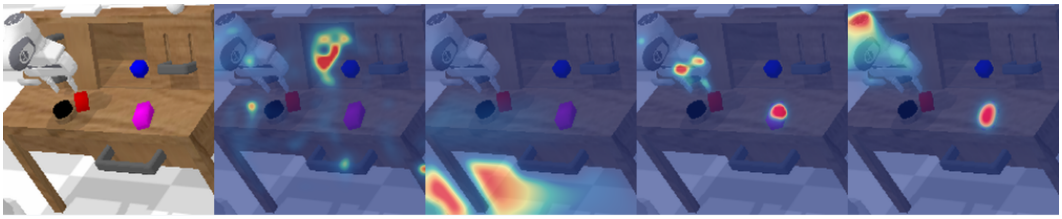
Target region	w/o Guidance	TARGET-VLA	Δ (pp)	95% CI (pp)
SAM3 combined mask	30.2	78.8	+48.6	[46.9, 50.3]
DA3 end-effector mask	8.4	74.8	+66.4	[65.0, 67.7]
DA3 object mask	7.4	76.3	+69.0	[66.5, 71.5]

TARGET-VLA has higher TAM in all three comparisons, with paired increases of 48.6–69.0 percentage points. Because the guidance masks also define the measurement targets, these results quantify attention redistribution toward the prompted regions rather than independently establishing the action relevance of the retrieved features. Table 3 and Figure 3 provide the corresponding task-performance evidence.

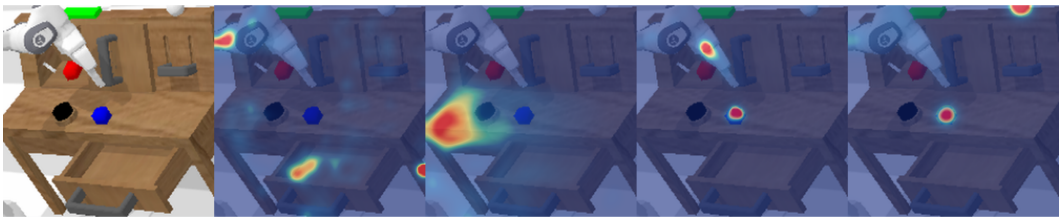
Additional qualitative results. Figures 7 and 8 provide ten additional paired visualizations on CALVIN ABC→D. Each row uses the same observation for w/o Guidance and TARGET-VLA, with separate columns for the SAM3 and DA3 retrieval maps; the corresponding language instruction is shown below the row.



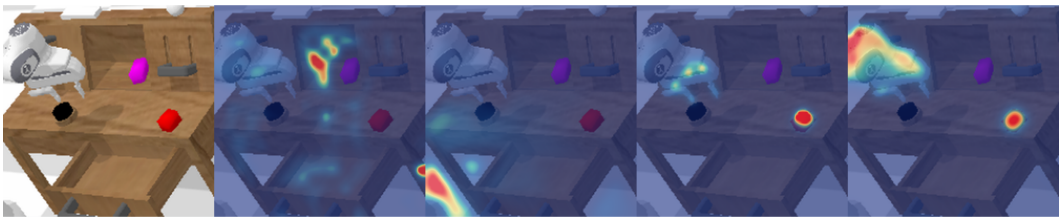
Instruction: "push the handle to close the drawer"



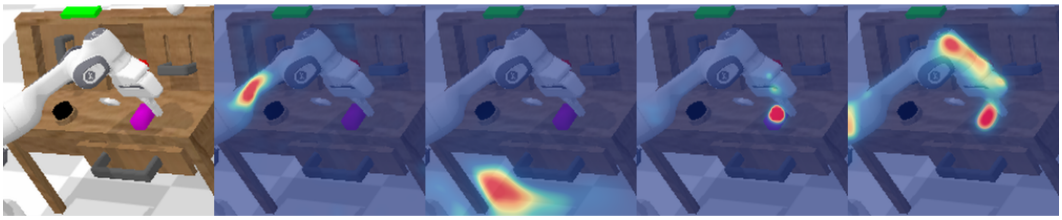
Instruction: "grasp and lift the pink block"



Instruction: "grasp and lift the blue block"



Instruction: "grasp and lift the red block"



Instruction: "go push the pink block left"

Figure 7: Paired query-to-feature attention on CALVIN ABC→D (examples 1–5). Colors show independently normalized attention within each map, from blue (low) to red (high).

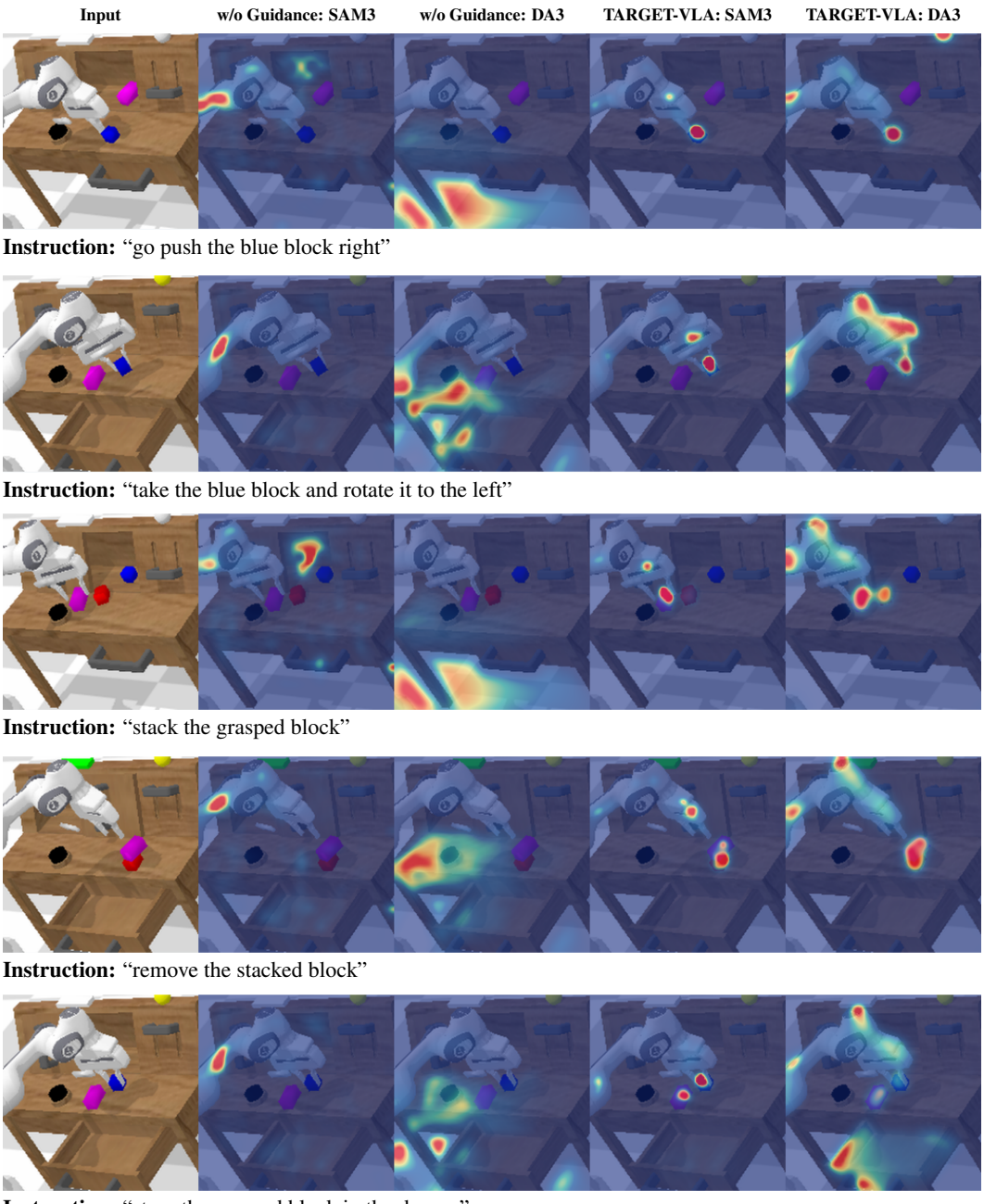


Figure 8: Paired query-to-feature attention on CALVIN ABC→D (examples 6–10), using the same normalization and color scale as Figure 7.

E LIMITATIONS AND FUTURE WORK

TARGET-VLA currently uses an offline instruction-to-prompt mapping based on noun-phrase extraction and dependency rules.This mapping operates on the instruction alone, while the resulting task-object masks are merged into a single guidance mask. Consequently, the retrieval interface does not explicitly preserve object roles, such as the manipulated object and its destination, or encode source–target relations. These limitations may affect tasks that require visual disambiguation or changes in relevant entities across subgoals. A scene-aware high-level policy, coupled with a role-aware retrieval interface, could jointly interpret instructions and observations, assign object roles, and update role-specific prompts and masks during execution.