TAGER: TEMPORAL ACTION GEOMETRY FOR ROBUST ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models inherit rich visual-semantic knowledge from pretrained vision-language models (VLMs), yet remain vulnerable to distribution shifts in visual observations, language instructions, and manipulation conditions. A key limitation is that pretrained VLMs lack explicit geometric awareness and spatial relationships required for manipulation. Recent methods mitigate this limitation by aligning VLA representations with geometric teacher features. However, directly injecting such supervision into shared visual representations may interfere with the visual-semantic priors inherited from pretraining. Moreover, static spatial representations fail to capture the action-conditioned evolution of visual observations and robot states, thereby restricting the policy's capacity to adapt subsequent actions under perturbations. To address these challenges, we introduce **TAGER**, a ***T**emporal **A**ction **G**eometry framework for robust **E**mbodied **R**obot manipulation* that couples **Sparse Spatial Query Distillation** with an **Action Response Residual Flow**. To enhance spatial awareness while while mitigating interference with pretrained visual-semantic priors, sparse spatial query distillation uses dual independent query transformers to distill dense geometric and semantic knowledge from frozen VGGT and SAM3 teachers into compact tokens. These tokens provide complementary world-grounded context to the action expert without directly imposing teacher-feature matching constraints on shared VLA representations. To support adaptive action generation under perturbations, the action response residual flow relates recent executed actions to subsequent changes in robot states and spatial tokens representations, providing evidence about the current visual-motor relationship. The resulting execution context drives a residual flow that adaptively refines subsequent action generation. Experiments on LIBERO and VLABench, together with zero-shot evaluations on LIBERO-Plus and LIBERO-PRO, demonstrate that TAGER achieves outstanding manipulation performance while generalizing robustly across diverse perturbations.

1 INTRODUCTION

Vision-language-action (VLA) models leverage the rich visual-semantic priors of pretrained vision-language models (VLMs) for robotic control (Kim et al., 2024; Black et al., 2024; Physical Intelligence et al., 2025). Despite excellent in-distribution performance, their robustness to shifts in visual observations, language instructions, and manipulation conditions remains limited, posing a major challenge to broader deployment (Fei et al., 2025; Zhou et al., 2025). A key limitation is that VLM pretraining primarily emphasizes semantic understanding rather than the fine-grained geometric and spatial reasoning required for precise manipulation (Li et al., 2026a). Furthermore, policies driven primarily by current observations do not explicitly model how recent actions affect subsequent visual observations and proprioceptive states, potentially restricting their ability to adapt subsequent actions under perturbations.

Existing methods enhance VLA spatial awareness through geometric inputs (Sun et al., 2025), spatial conditioning (Qu et al., 2025; Jiang et al., 2025), and spatial representation learning (Zhang et al., 2025b; Feng et al., 2026). Among these approaches, distilling features from pretrained 3D foundation models, as a strategy for spatial representation learning, is particularly promising. It transfers rich geometric priors into intermediate policy representations without requiring explicit 3D observations or intermediate depth estimates, thereby reducing dependence on their availability

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

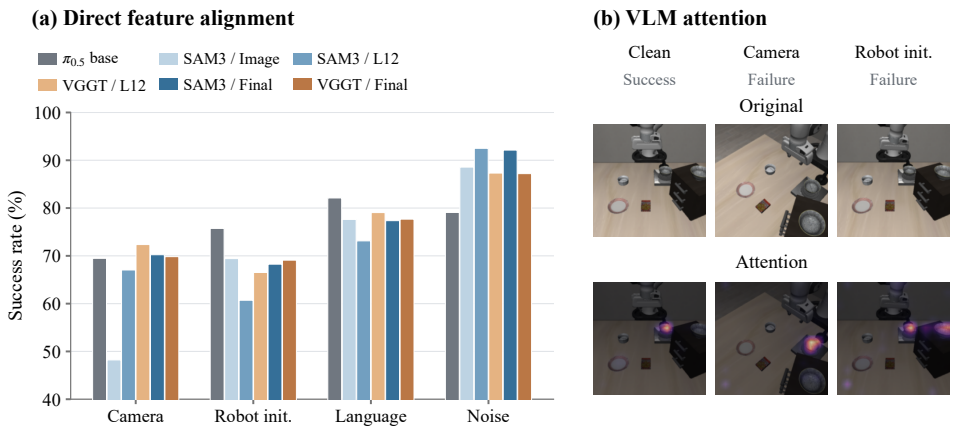


Figure 1: **Perturbation Analysis.** (a) LIBERO-Plus success rates with direct feature alignment. (b) Initial VLM attention for successful clean and failed perturbed runs.

and quality (Li et al., 2026a). However, directly aligning shared VLA visual features with geometric teacher representations may alter the visual-semantic structure inherited from VLM pretraining. Since the same representations also support semantic recognition and instruction grounding, improving geometric alignment alone does not necessarily yield more robust control.

To further investigate how direct feature alignment interferes with pretrained visual-semantic priors, we follow SpatialForcing (Li et al., 2026a) and finetune $\pi_{0.5}$ with an auxiliary objective that aligns its visual representations with frozen VGGT or SAM3 teacher features. We apply the alignment objective at three different stages, including the image encoder, an intermediate VLM layer, and the final VLM layer. Each model is trained on LIBERO and evaluated zero-shot on LIBERO-Plus to assess its robustness under distribution shifts. As seen in Figure 1(a), direct feature alignment consistently improves robustness to visual noise but degrades zero-shot performance under language and robot-state perturbations. For example, aligning SAM3 features at VLM layer 12 increases success under visual noise from 79.08% to 92.50%, while reducing success under robot-state perturbations from 75.74% to 60.71%. These results reveal a clear trade-off: *while direct feature alignment improves robustness to visual perturbations, imposing geometric or visual teacher supervision on shared VLA representations may interfere with the pretrained visual-semantic structure that supports generalization across other dimensions*.

Moreover, static spatial representations are unable to capture the visual and proprioceptive consequences of recent actions, limiting a policy's ability to adjust subsequent actions under perturbations. As seen in Figure 1(b), we evaluate $\pi_{0.5}$ on the same manipulation task under clean conditions as well as camera-viewpoint and robot state perturbations. Although the VLM continues to attend to the target object, the policy fails to complete the task under the perturbed conditions. Limited exposure to such perturbations during finetuning will hinder the policy from translating visual target information into actions appropriate to the current physical state of robot. This motivates incorporating recent action response context to support action adjustment under changing observation and execution conditions.

To address these challenges, we introduce **TAGER**, a *Temporal Action Geometry framework for robust embodied robot manipulation*. Specifically, to improve spatial awareness while mitigating potential interference with pretrained visual-semantic priors, **Sparse Spatial Query Distillation** utilizes two independent query transformers to compress dense geometric and semantic knowledge from frozen VGGT (Wang et al., 2025) and SAM3 (Carion et al., 2026) into compact token states. This design mitigate interference with pretrained visual-semantic priors while provide spatial context to the action learning. Furthermore, to enable action adjustment under perturbations, the **Action Response Residual Flow** infers the current visual-motor relationship from recent actions and their subsequent effects on robot states and query representations, which adaptively refines the generation of the next action chunk through a residual flow ordinary differential equation (ODE).

Our main contributions are summarized as follows:

- We identify two challenges in robust VLA control: direct feature alignment can interfere with pretrained visual-semantic priors, while static spatial representations fail to model the visual-motor relationship under distribution shifts.
- We propose **TAGER**, which couples **Sparse Spatial Query Distillation** with **Action Response Residual Flow**. The former compresses dense teacher knowledge into compact token states, alleviating interference with pretrained priors; the latter leverages historical action response dynamics to adaptively refine action generation.
- We demonstrate competitive manipulation performance on LIBERO and VLABench and enhanced zero-shot robustness on LIBERO-Plus and LIBERO-PRO. Without additional fine-tuning, TAGER outperforms retrained baselines by 5.4 and 1.9 percentage points on LIBERO-Plus and LIBERO-PRO, respectively, achieving outstanding success rates of 87.6% and 62.2%. Extensive ablation studies further confirm the complementary contributions of spatial distillation and action response modeling.

2 PRELIMINARIES

We consider language-conditioned robotic manipulation. At time step t, given multi-view RGB observations I_t, proprioceptive state s_t, and instruction l, the policy predicts an action chunk

$$A_t = (a_t, \dots, a_{t+H-1}) \in \mathbb{R}^{H \times d_a}, \tag{1}$$

where H and d_a denote the prediction horizon and action dimension, respectively. We adopt a flow-matching VLA policy that consists of a pretrained VLM backbone and an action expert. The VLM provides multi-modal context for the flow-matching action expert.

During training, we interpolate the demonstrated action A_t with Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ as $z_\tau = (1-\tau)A_t + \tau\epsilon$, with $\tau \in [0,1]$, and optimize

$$\mathcal{L}_{\text{flow}}(v_\theta) = \mathbb{E}\left[\frac{1}{Hd_a}\left\|v_\theta(z_\tau, \tau \mid I_t, s_t, l) - (\epsilon - A_t)\right\|_F^2\right]. \tag{2}$$

At inference, starting from $z_1 = \epsilon$, the predicted action chunk $\widehat{A}_t = z_0$ is obtained by integrating

$$\frac{\mathrm{d}z_\tau}{\mathrm{d}\tau} = v_\theta(z_\tau, \tau \mid I_t, s_t, l), \qquad \tau : 1 \to 0. \tag{3}$$

3 METHOD

3.1 OVERVIEW

As shown in Figure 2, we propose **TAGER**, a *Temporal Action Geometry framework for robust embodied robot manipulation* under distribution shifts. To improve spatial awareness while avoiding the interference caused by direct feature alignment, **Sparse Spatial Query Distillation** employs dual isolated query transformers to distill dense geometric and semantic knowledge from frozen VGGT and SAM3 teachers into compact token states. These tokens provide complementary spatial context to the action expert without directly imposing teacher-feature matching constraints on shared VLA representations. The reconstruction decoder are used only during training and discarded at inference so that the additional computation cot is limited. To address the lack of action response context in static spatial representations, the **Action Response Residual Flow** relates recent executed actions to subsequent changes in robot states and SAM3-aligned query representations. The resulting execution context provides evidence about the current visual-motor relationship and drives the residual flow that adaptively refines subsequent actions generation under perturbations.

3.2 SPARSE SPATIAL QUERY DISTILLATION

To enhance spatial awareness while mitigating interference with pretrained visual-semantic priors, we introduce two independent sparse query transformers guided by frozen SAM3 and VGGT teachers. As seen in Figure 2(a), dense teacher features are reconstructed from compact token states, thereby routing spatial supervision through the query branches rather than directly constraining shared VLA representations.

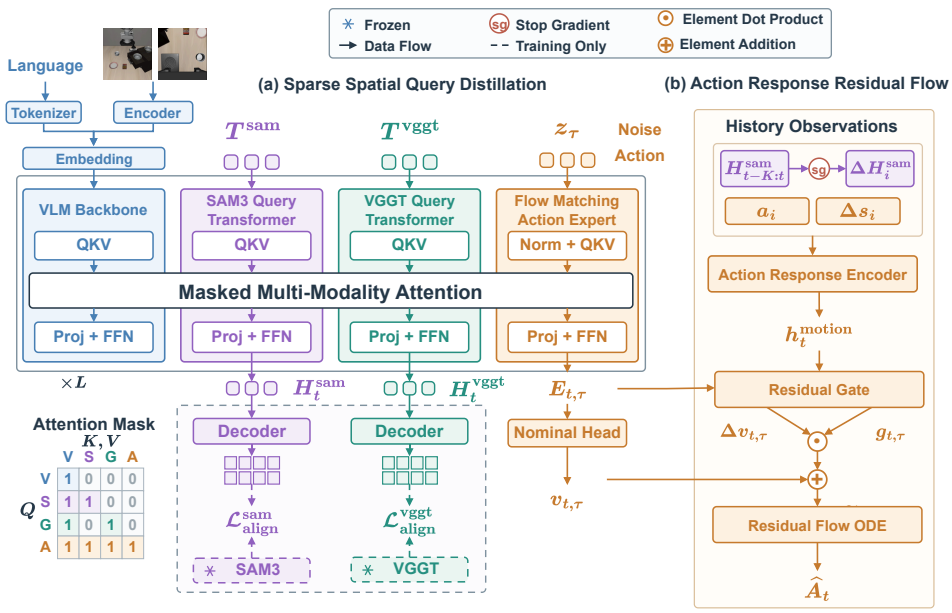


Figure 2: **TAGER Network Architecture.** (a) Sparse Spatial Query Distillation uses two independent transformer networks to encode the SAM3(S) and VGGT(G) learnable tokens(T). Multi-modal attention is masked: the VLM context(V) attends only to itself, each teacher attends to the VLM context and itself, and the two teachers cannot attend to each other. Action(A) Response Residual Flow encodes history observations into a residual flow that refines the flow-matching velocity field.

Sparse Query Transformer. We index the SAM3 and VGGT branches by $k = 0$ and $k = 1$, respectively. Each branch maintains learnable tokens $T^k \in \mathbb{R}^{M_k \times d_k}$, where M_k counts tokens across configured camera groups and d_k is their hidden dimension. At step t, the query transformer T_{ϕ_k}, with trainable parameters ϕ_k, exploits the layerwise VLM context $\mathcal{C}_t$ encoded from images I_t and instruction l:

$$H_t^k = T_{\phi_k}(T^k, \mathcal{C}_t), \qquad H_t^k \in \mathbb{R}^{M_k \times d_k}. \tag{4}$$

Here, T^k contains the learnable initial tokens, while H_t^k contains their final states conditioned on the current observation and instruction. Following the attention mask illustrated in Figure 2(a), each query branch attends jointly to its own stream and the corresponding VLM context, while remaining isolated from the other query branch. Conversely, the VLM attends only within its original stream and the action expert accesses the layerwise keys and values from the VLM and both query branches.

Geometric-Sematic Reconstruction. To provide dense spatial supervision for the sparse query transformers, we employ two frozen teacher models: VGGT supplies geometric representations, while SAM3 provides sematic representations. For teacher k, let $F_t^k \in \mathbb{R}^{N_k \times c_k}$ denote the target feature grid aggregated across camera views, where N_k is the number of dense teacher tokens and c_k is their feature dimension. A teacher-specific decoder D_{ψ_k}, parameterized by ψ_k, reconstructs these dense targets from the final query states:

$$\widehat{F}_t^k = D_{\psi_k}(H_t^k), \qquad M_k \ll N_k. \tag{5}$$

For each training example, the alignment loss $\mathcal{L}_{\text{align}}^k$ minimizes the cosine distance between reconstructed features $\widehat{F}_t^k$ and teacher features F_t^k over valid locations. During inference, the reconstruction decoders are removed, leaving only the query transformers to condition the policy. Decoder architecture, validity masking, and the full alignment objective are detailed in Appendix D.1.

3.3 Action Response Residual Flow

To support action adjustment under perturbations, we introduce **Action Response Residual Flow**, which complements static spatial representations with recent execution context. As shown in Figure 2(b), the module encodes executed actions together with subsequent changes in robot states and

SAM3 token representations, and uses this context to condition a residual flow for action generation. The following analysis motivates the residual formulation and the use of local action–response measurements.

Let s denote the robot state and η the environment state, including object configurations, camera viewpoint, and illumination. With the instruction and network parameters fixed, image formation and query encoding define a mapping from (s, η) to (s, H^{sam}). Its Jacobians under nominal and nearby perturbed conditions are denoted by J_{nom} and J_{pert}, which indicate the local visual–motor relationship.

Proposition 1 (Local Residual Adaptation)**.** For the same policy input, a perturbation in the local visual–motor relationship from J_{nom} to J_{pert} may require different actions to achieve the task. Let $\widetilde{v}^{\star}_{t,\tau}(J)$ denote the corresponding ideal action-generating field and define its nominal value as $v^{\star}_{t,\tau} = \widetilde{v}^{\star}_{t,\tau}(J_{\mathrm{nom}})$. As a local modeling assumption, suppose this field is differentiable with respect to J at J_{nom}. Then, as $J_{\mathrm{pert}} \to J_{\mathrm{nom}}$,

$$\begin{aligned}\widetilde{v}^{\star}_{t,\tau}(J_{\mathrm{pert}}) = v^{\star}_{t,\tau} \\ + D_J \widetilde{v}^{\star}_{t,\tau}(J_{\mathrm{nom}})[J_{\mathrm{pert}} - J_{\mathrm{nom}}] + o(\|J_{\mathrm{pert}} - J_{\mathrm{nom}}\|).\end{aligned} \tag{6}$$

Here, $D_J \widetilde{v}^{\star}_{t,\tau}(J_{\mathrm{nom}})$ is the derivative of the ideal velocity field with respect to J at J_{nom}. It maps the local visual–motor relationship change $J_{\mathrm{pert}} - J_{\mathrm{nom}}$ to the first-order velocity correction.

Appendix B.3 provides the proof and scope. Since the local visual–motor relationship change is not directly available to the policy, we condition the learned correction on recent action–response measurements.

At replanning time t, (s_t, η_t) denotes the current robot and environment states; the subscript t indexes execution time. We collect K historical triplets $(a_i, \Delta s_i, \Delta H_i^{\mathrm{sam}})$, indexed by $i = t-K, \ldots, t-1$, where a_i is the executed action, $\Delta s_i = s_{i+1} - s_i$ and $\Delta H_i^{\mathrm{sam}} = H_{i+1}^{\mathrm{sam}} - H_i^{\mathrm{sam}}$. The following proposition relates these measurements to the local response map under the current operating conditions.

Proposition 2 (Local Responses)**.** Let $f(a; s_t, \eta_t)$ denote the joint robot–environment state increment $(\Delta s, \Delta \eta)$ induced by action a under the fixed current conditions (s_t, η_t). Motivated by local visual-servoing models (Chaumette & Hutchinson, 2006) and attention sensitivity analysis (Castin et al., 2023), we assume that the joint state–query mapping has a locally Lipschitz Jacobian within a common smooth neighborhood. Under the local model and regularity assumptions in Appendix B.4, nearby historical transitions satisfy

$$\begin{bmatrix}\Delta s_i \\ \Delta H_i^{\mathrm{sam}}\end{bmatrix} = J_{\mathrm{pert}} f(a_i; s_t, \eta_t) + \rho_{i,t}. \tag{7}$$

Here, $\rho_{i,t}$ accounts for linearization error and deviations of historical conditions from (s_t, η_t); its bound shrinks as transition sizes and these deviations decrease.

The derivation and error bound are provided in Appendix B.4. These approximate input–output observations of the current response map are encoded by the **Action Response Encoder** into h_t^{motion}, which conditions the **Residual Flow ODE** alongside the current action-expert states.

Action Response Encoder. The encoder processes K historical triplets $(a_i, \Delta s_i, \Delta H_i^{\mathrm{sam}})$ defined above. As shown in Figure 2(b), the encoder uses each action and robot-state change to attend to the corresponding query difference, then aggregates the transition descriptors in temporal order into h_t^{motion}. This context is computed once per replanning step and reused during action-chunk generation.

Residual Flow ODE. Motivated by Proposition 1, we parameterize the corrected velocity as a nominal prediction plus a learned history-conditioned residual. At flow time τ, the final action-expert states $E_{t,\tau}$ and h_t^{motion} jointly condition a velocity residual $\Delta v_{t,\tau}$ and a sigmoid gate $g_{t,\tau}$. The corrected field is

$$\widetilde{v}_{t,\tau} = v_{t,\tau} + g_{t,\tau} \odot \Delta v_{t,\tau}, \tag{8}$$

where $v_{t,\tau}$ is the trainable nominal head's output from $E_{t,\tau}$ and $\odot$ denotes element-wise multiplication. The gated term $g_{t,\tau} \odot \Delta v_{t,\tau}$ supplies the learned correction. The residual and gate are

recomputed at each flow step. The corrected field replaces v_θ in both the flow-matching objective in Eq. (2) and the ODE integration in Eq. (3). The encoder projections, temporal aggregation, and residual heads are specified in Appendix D.2.

3.4 Training Objective

We train TAGER by jointly optimizing the corrected flow-matching objective and the two teacher-feature alignment losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{flow}}(\widetilde{v}) + \lambda_{\text{sam}}\mathcal{L}_{\text{align}}^{\text{sam}} + \lambda_{\text{vggt}}\mathcal{L}_{\text{align}}^{\text{vggt}}, \tag{9}$$

Here, $\mathcal{L}_{\text{flow}}(\widetilde{v})$ evaluates Eq. (2) with the corrected, history-conditioned field in Eq. (8) and the coefficients λ_{sam} and λ_{vggt} weight the two alignment losses.

4 Experiments

In this section, we conduct comprehensive experiments to evaluate the effectiveness of TAGER. Our primary goals are to: 1) Assess the performance of TAGER against existing VLAs on standard benchmarks. 2) Evaluate its generalization across different distribution shifts. 3) Analyze the contribution of each component in TAGER. 4) Examine whether the query transformers improve spatial representations while preserving pretrained visual-semantic priors. 5) Validate the adaptive action correction capability of the Action Response Residual Flow (Appendix F). 6) Examine real-world feasibility through manipulation examples. 7) Quantify inference latency and parameter overhead (Appendix I).

Table 1: **Average success rates (%) comparison on LIBERO**.

Method	Spatial	Object	Goal	10	Avg.
OpenVLA (Kim et al., 2024)	84.7	88.4	79.2	53.7	76.5
CoT-VLA (Zhao et al., 2025)	87.5	91.6	87.6	69.0	81.1
OpenVLA-OFT (Kim et al., 2025)	97.6	98.4	97.9	94.5	97.1
CogVLA (Li et al., 2026c)	98.5	98.8	96.5	95.2	97.4
PixelVLA (Liang et al., 2026)	88.5	90.0	85.8	82.6	86.7
π_0 (Black et al., 2024)	96.8	98.8	95.8	85.2	94.1
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	98.8	98.2	98.0	92.4	96.9
VLA-Adapter (Wang et al., 2026b)	97.8	99.2	97.2	95.0	97.3
MemoryVLA (Shi et al., 2026)	98.4	98.4	96.4	93.4	96.7
X-VLA (Zheng et al., 2026)	98.2	98.6	97.8	97.6	98.1
FASTer (Liu et al., 2026)	98.0	99.4	98.6	95.4	97.9
InstructVLA (Yang et al., 2026)	97.3	99.6	96.5	89.8	95.8
TAGER (Ours)	99.8	99.6	98.3	96.0	98.4

Table 2: **Comparison on VLABench(%).** Each setting reports success rate (SR) and progress score (PS).In-Dis., Category. and Common. are in-distribution, cross category and common sense.

Method	In-Dist.		Category.		Common.		Instruction		Texture		Avg.	
	SR	PS	SR	PS	SR	PS	SR	PS	SR	PS	SR	PS
π_0	47.0	62.7	21.2	33.6	29.1	43.0	17.3	38.7	32.2	42.5	29.4	44.1
π_0-FAST	56.2	66.8	31.0	38.2	38.0	48.6	35.0	45.0	39.0	49.0	39.8	49.5
X-VLA	–	67.8	–	25.1	–	48.2	–	63.1	–	–	–	51.1
ACoT-VLA	–	66.1	–	38.9	–	37.8	–	39.6	–	54.6	–	47.4
$\pi_{0.5}$	65.4	77.8	38.2	49.7	43.9	57.3	48.2	64.2	44.9	62.3	48.1	62.3
TAGER (Ours)	69.7	78.8	40.0	46.8	48.3	55.9	59.0	66.7	52.4	65.1	54.1	63.0

4.1 EXPERIMENTS SETUP

Benchmarks and Evaluation Metrics. We evaluate TAGER across multiple robotic manipulation benchmarks, including LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2025), LIBERO-PRO (Zhou et al., 2025) and VLABench (Zhang et al., 2025a). To verify robustness under the perturbation, we transfer the LIBERO-trained model to LIBERO-Plus and LIBERO-PRO without additional fine-tuning. LIBERO-Plus covers seven perturbation dimensions: camera viewpoint, robot initial state, language instruction, lighting, background, visual noise, and scene layout. LIBERO-PRO covers object, position, semantic, and task perturbations. Also, We additionally evaluate TAGER on VLABench across five settings: in-distribution, cross-category, commonsense, semantic instruction, and unseen texture. About the evaluation metric, we report success rate (SR) on all benchmarks and additionally report progress score (PS) on VLABench, with all results expressed as percentages.

Experiments Setting. We use OpenPI as the training framework and $\pi_{0.5}$ (Physical Intelligence et al., 2025) as the baseline model. Training is conducted on 8 H20 GPUs for all datasets, with 128 total batch size and the AdamW optimizer (Loshchilov & Hutter, 2017). For the dual-teacher distillation transformers, learnable tokens are organized per input camera. Both the SAM3 and VGGT teachers employ the main and wrist cameras to supervise the reconstruction of sparse geometric and semantic features, with loss weights of 0.4 and 0.1, respectively. For each camera, we use 16 learnable tokens, each of dimension 512. More experiment settings are in Table 5 and details about baselines are in Appendix C.

4.2 MAIN RESULTS

Tables 1 and 2 demonstrate the outstanding in-distribution performance of TAGER across diverse manipulation tasks. Specifically, on LIBERO, TAGER achieves the highest average success rate of 98.4% and consistently improves upon the reported $\pi_{0.5}$ baseline across all four suites, with the largest gain on LIBERO-10 from 92.4% to 96.0%. This advantage further transfers to VLABench, where TAGER improves the in-distribution success rate from 65.4% to 69.7% and the overall success rate from 48.1% to 54.1%. These results show that TAGER achieve outsanding in-distribution manipulation performance while extending effectively to more diverse task settings.

4.3 TAGER GENERALIZATION ACROSS DISTRIBUTION SHIFTS

To evaluate the generalization of TAGER under different perturbations, we directly transfer the policy, which only trained on LIBERO dataset, to LIBERO-Plus and LIBERO-PRO without additional fine-tuning. As shown in Table 3, TAGER achieves 87.6% overall success rate on LIBERO-Plus, outperforming the official $\pi_{0.5}$ model by 3.0 percentage points. The largest gain occurs under robot state perturbations, where success increases from 75.5% to 86.6%, with further improvements under noise and camera-viewpoint shifts. These results highlight the robustness of TAGER to changes that alter the visual-motor relationship. This generalization advantage extends to LIBERO-PRO. As shown in Table 4, TAGER improves overall success from 60.3% to 62.2% over the retrained $\pi^*_{0.5}$, with improvements in 12 of the 16 suite-perturbation combinations. In particular, TAGER consistently improves performance under semantic shifts across all four suites and also shows gains under object-position and task perturbations. Consistent improvements are also observed on VLABench (Table 2). Compared with $\pi_{0.5}$, TAGER achieves higher success rates across all four out-of-distribution settings, including cross-category, commonsense, instruction, and unseen-texture generalization. The better gain is observed under instruction shifts, where success rate increases from 48.2% to 59.0%. Together, these results demonstrate that TAGER generalizes across diverse visual, semantic, and embodied distribution shifts.

4.4 ABLATION STUDIES

Figure 3 evaluates the cumulative contribution of each component on LIBERO-Plus. Starting from the retrained $\pi_{0.5}$ baseline at 82.2%, adding the VGGT and SAM3 query branches progressively improves overall success to 84.0% and 85.8%, respectively. Incorporating the Action Response Residual Flow further raises performance to 87.6%, yielding a total improvement of 5.4 percent-

Table 3: **Zero-Shot success rates (%) comparison on LIBERO-Plus**. Cam., Rob., Lang., and BG. denote camera viewpoint, robot initial state,language instruction, and background perturbations.

Method	Cam.	Rob.	Lang.	Light	BG.	Noise	Layout	Overall
OpenVLA-OFT (Kim et al., 2025)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
OpenVLA (Kim et al., 2024)	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
SRPO (Fei et al., 2026)	17.1	51.0	81.8	70.4	88.9	35.3	72.4	59.6
MergeVLA (Fu et al., 2026)	64.4	46.4	75.7	92.7	92.7	74.7	73.7	72.4
VLA-JEPA (Sun et al., 2026)	63.3	67.1	85.4	95.6	93.6	66.3	85.1	79.5
Spatial Forcing (Li et al., 2026a)	0.1	0.3	26.8	66.0	45.9	0.1	59.8	25.7
π_0 (Black et al., 2024)	61.0	40.8	63.5	89.3	84.1	80.1	76.4	69.4
π_0-FAST (Pertsch et al., 2025)	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
RIPT-VLA (Tan et al., 2025)	55.2	31.2	77.6	88.4	91.6	73.5	74.2	68.4
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	71.7	75.5	85.9	96.1	95.7	86.4	87.5	84.6
TAGER (Ours)	75.0	86.6	85.9	94.8	94.1	93.0	87.9	87.6

Table 4: **Zero-Shot success rates (%) comparison on LIBERO-PRO.** Obj., Pos., and Sem. denote object, position, and semantic perturbations. An asterisk marks a retrained baseline.

Method	LIBERO-Spatial				LIBERO-Object				LIBERO-Goal				LIBERO-10				Avg.
	Obj.	Pos.	Sem.	Task	Obj.	Pos.	Sem.	Task	Obj.	Pos.	Sem.	Task	Obj.	Pos.	Sem.	Task	
π_0	95.4	8.0	88.4	55.0	87.0	0.0	98.6	9.8	71.0	9.6	82.6	10.6	44.0	0.0	72.8	9.0	46.4
π_0^*	93.4	6.8	62.0	63.6	92.0	0.0	98.2	9.6	75.6	2.4	88.2	10.6	53.2	0.0	82.2	11.0	46.8
$\pi_{0.5}$	97.4	49.6	97.0	54.4	94.0	18.2	99.0	10.4	88.8	34.8	96.4	20.6	67.2	9.6	92.4	17.8	59.2
$\pi_{0.5}^*$	97.6	50.8	95.4	55.8	94.4	36.6	99.0	11.0	85.0	32.2	97.0	17.0	64.0	16.2	92.2	20.8	60.3
TAGER	97.0	49.4	99.4	52.0	96.8	44.8	99.4	11.6	86.0	37.8	97.8	21.2	66.2	15.6	96.2	25.8	62.2

age points. The gains are most pronounced under visual-noise and robot-state perturbations, while lighting and background conditions show slight degradation.

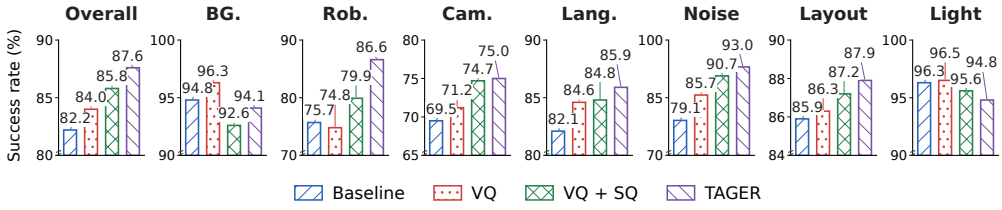


Figure 3: **Component Ablations on LIBERO-Plus.** VQ/SQ denote VGGT/SAM3 query transformer. We compare the performance of different modules under the perturbation.

4.5 Spatial Representation and Visual-Semantic Prior Analysis

We first examine whether the distilled query states acquire meaningful spatial representations. Specifically, we fit linear probes for body-ID segmentation and depth prediction on frozen SAM3 and VGGT supervised token representations. Detains can be seen in Appendix H. As shown in Figure 4, the segmentation probe distinguishes the basket and multiple tabletop objects, while the depth probe captures coarse geometric variations around the moving robot arm. These results indicate that the compact tokens states retain complementary semantic and geometric cues distilled from the two teachers.

We further examine whether such spatial enhancement mitigates the interference with pretrained visual-semantic priors of the VLA context. As shown in Figure 3, introducing the query branches

substantially improves robustness to visual perturbations, while avoiding the degradation under language and robot-state perturbations observed with direct feature alignment. This suggests that routing teacher knowledge through token states effectively mitigates interference with shared VLA representations, enabling stronger spatial awareness without sacrificing the pretrained visual-semantic capabilities required for generalization.

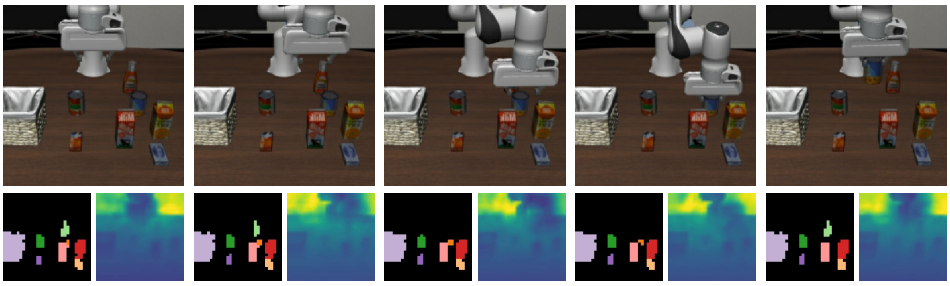


Figure 4: **Spatial Tokens Probe Visualizations.** Top: RGB; Bottom Left: Body Sematic IDs; Bottom Right: Depth.

4.6 Real-World Feasibility Validation

For each real-world task, we collect 100 demonstrations under nominal conditions and use them to fine-tune TAGER. Following the perturbation taxonomy of LIBERO-Plus (Fei et al., 2025), Figure 5 pairs reference and variant sequences for three manipulation goals: apple placement, bowl nesting, and block stacking. Lang illustrate alternative wording for the same apple-placement goal; The latter two illustrate completed goals with different robot initial configurations (Rob.) and object layouts (Layout). These examples provide qualitative evidence of manipulation feasibility across recorded physical configurations. Task scope and recording details are provided in Appendix G.



Figure 5: **Real-world Feasibility Examples.** Each row pairs reference and variant recordings of the same goal, with three chronological frames per sequence. Lang. uses generated descriptions; Rob. and Layout show recorded configuration differences.

5 Conclusion

TAGER couples **Sparse Spatial Query Distillation** with an **Action Response Residual Flow** to enhance spatial representations and adapt action generation from recent execution context. Experiments on LIBERO, VLABench, LIBERO-Plus, and LIBERO-PRO demonstrate outstanding manipulation performance and improved zero-shot robustness, with ablations confirming complementary gains from the spatial and temporal components.

REFERENCES

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

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

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.

Valérie Castin, Pierre Ablin, and Gabriel Peyré. How smooth is attention? *arXiv preprint arXiv:2312.14820*, 2023.

François Chaumette and Seth Hutchinson. Visual servo control. i. basic approaches. *IEEE robotics & automation magazine*, 13(4):82–90, 2006.

Danny Driess, Jost Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *Advances in Neural Information Processing Systems*, 38:102867–102888, 2026.

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.

Senyu Fei, Siyin Wang, Li Ji, Ao Li, Shiduo Zhang, Liming Liu, Jinlong Hou, Jingjing Gong, Xianzhong Zhao, and Xipeng Qiu. Srpo: Self-referential policy optimization for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6718–6728, 2026.

Yicheng Feng, Wanpeng Zhang, Ye Wang, Hao Luo, Haoqi Yuan, Sipeng Zheng, and Zongqing Lu. 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.

Yuxia Fu, Zhizhen Zhang, Yuqi Zhang, Zijian Wang, Zi Huang, and Yadan Luo. Mergevla: Cross-skill model merging toward a generalist vision-language-action agent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22335–22347, 2026.

Hyeongjun Heo, Seungyeon Woo, Sang Min Kim, Junho Kim, Junho Lee, Yonghyeon Lee, and Young Min Kim. Anycamvla: Zero-shot camera adaptation for viewpoint robust vision-language-action models. *arXiv preprint arXiv:2603.05868*, 2026.

Chengyue Huang, Mellon M Zhang, Robert Azarcon, Glen Chou, and Zsolt Kira. Maps: Preserving vision-language representations via module-wise proximity scheduling for better vision-language-action generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32451–32462, 2026.

Tianchong Jiang, Jingtian Ji, Xiangshan Tan, Jiading Fang, Anand Bhattad, Vitor Guizilini, and Matthew R Walter. Do you know where your camera is? view-invariant policy learning with camera conditioning. *arXiv preprint arXiv:2510.02268*, 2025.

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.

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, 2026a.

Hao Li, Shuai Yang, Yilun Chen, Xinyi Chen, Xiaoda Yang, Yang Tian, Hanqing Wang, Tai Wang, Dahua Lin, Feng Zhao, et al. Towards efficient and robust manipulation via multi-frame vision-language-action modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18388–18396, 2026b.

Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. Cogvla: Cognition-aligned vision-language-action models via instruction-driven routing & sparsification. *Advances in neural information processing systems*, 38:137646–137675, 2026c.

Wenqi Liang, Gan Sun, Yao He, Jiahua Dong, Suyan Dai, Ivan Laptev, Salman Khan, and Yang Cong. Pixelvla: Advancing pixel-level understanding in vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 110464–110481, 2026.

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.

Yicheng Liu, Shiduo Zhang, Zibin Dong, Baijun Ye, Tianyuan Yuan, Xiaopeng Yu, Linqi Yin, Chenhao Lu, Junhao Shi, Luca Jiang-Tao Yu, et al. Faster: Toward powerful and efficient autoregressive vision–language–action models with learnable action tokenizer and block-wise decoding. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yuechen Luo, Fang Li, Shaoqing Xu, Yang Ji, Zehan Zhang, Bing Wang, Yuannan Shen, Jianwei Cui, Long Chen, Guang Chen, et al. Last-vla: Thinking in latent spatio-temporal space for vision-language-action in autonomous driving. *arXiv preprint arXiv:2603.01928*, 2026.

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.

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

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.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 18567–18602, 2026.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. In *European Conference on Computer Vision*, pp. 478–497. Springer, 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.

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

Xiaomi Embodied Intelligence Team, Shaoqing Xu, Fang Li, Guozhi Zhan, Zhixiang Duan, Yuhan Wang, Yuechen Luo, Shengyin Jiang, Hanbing Li, Zhiying Du, et al. One policy, many embodiments: Unified camera-centric action geometry pre-training for heterogeneous embodied manipulation. *arXiv preprint arXiv:2608.26058*, 2026.

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, 2025.

Xinkai Wang, Chenyi Wang, Yifu Xu, Mingzhe Ye, Fucheng Zhang, Jialin Tian, Xinyu Zhan, Lifeng Zhu, Cewu Lu, and Lixin Yang. Lamp: Learning vision-language-action policy with 3d scene flow as latent motion prior. In *European Conference on Computer Vision*, pp. 227–244. Springer, 2026a.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 40, pp. 18638–18646, 2026b.

Zhongyu Xia, Yousen Tang, Bingqing Wei, and Yongtao Wang. 3dvla: Enhancing vision-language-action models via 3d spatial and instance understanding. *arXiv preprint arXiv:2605.29416*, 2026.

Shuai Yang, Hao Li, Bin Wang, Yilun Chen, Yang Tian, Tai Wang, Hanqing Wang, Feng Zhao, Yiyi Liao, and Jiangmiao Pang. Vision-language-action instruction tuning: From understanding to manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 152896–152943, 2026.

Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, et al. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11142–11152. IEEE, 2025a.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems*, 38: 24195–24228, 2026.

Zhengshen Zhang, Hao Li, Yalun Dai, Zhengbang Zhu, Lei Zhou, Chenchen Liu, Dong Wang, Francis EH Tay, Sijin Chen, Ziwei Liu, et al. From spatial to actions: Grounding vision-language-action model in spatial foundation priors. In *The Fourteenth International Conference on Learning Representations*, 2025b.

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

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.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Si Liu, and Guanghui Ren. Acot-vla: Action chain-of-thought for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8152–8162, 2026.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

A Related Work

Generalization in VLA models. VLA models connect vision-language pretraining with robot control. RT-2 and OpenVLA predict discrete action tokens from visual and language inputs (Brohan et al., 2023; Kim et al., 2024; Team et al., 2026; Luo et al., 2026), while π_0 uses a flow-matching expert for continuous action generation (Black et al., 2024). Heterogeneous co-training in $\pi_{0.5}$ supports transfer to new environments and objects (Physical Intelligence et al., 2025); X-VLA uses embodiment-specific soft prompts to accommodate diverse robot data (Zheng et al., 2026). Efficient adaptation and decoding are complementary concerns. OpenVLA-OFT combines action chunking, parallel decoding, and continuous action regression (Kim et al., 2025). π_0-FAST compresses action sequences in the frequency domain (Pertsch et al., 2025), whereas FASTer learns a quantized action representation and uses blockwise autoregressive decoding (Liu et al., 2026). CogVLA reduces visual-token computation through instruction-driven routing and sparsification (Li et al., 2026c), and VLA-Adapter uses a lightweight policy module with Bridge Attention to connect vision-language features to actions (Wang et al., 2026b). Temporal context and intermediate predictions provide policies with information about history observations and anticipated task progress. TraceVLA renders tracked trajectories as visual prompts (Zheng et al., 2025); MemoryVLA retrieves perceptual and semantic information from a memory bank (Shi et al., 2026); and CronusVLA aggregates cached per-frame VLM features through a cross-frame decoder (Li et al., 2026b). CoT-VLA predicts future goal images before generating actions (Zhao et al., 2025), while ACoT-VLA conditions its action head on coarse reference trajectories and latent action priors (Zhong et al., 2026). Environment interaction offers another route: RIPT-VLA performs reinforcement-learning post-training with binary success rewards (Tan et al., 2025), and SRPO uses successful rollouts as references to assign progress rewards in a world model's latent space (Fei et al., 2026). Despite progress across these directions, LIBERO-Plus and LIBERO-PRO expose sensitivity to observation, instruction, and manipulation shifts (Fei et al., 2025; Zhou et al., 2025). TAGER learns a history-conditioned residual flow through supervised fine-tuning. It pairs preceding actions with subsequent robot-state and distilled-query changes to condition the action-generating velocity field.

Spatial representation learning. Spatial information affect policies through additional inputs, geometric conditioning, or representation supervision. GeoVLA encodes depth-derived point clouds (Sun et al., 2025), SpatialVLA introduces 3D positional encodings (Qu et al., 2025), and camera-conditioned policies use extrinsic to account for viewpoint changes (Jiang et al., 2025). AnyCamVLA instead transforms test-time observations toward the training viewpoint using novel-view synthesis (Heo et al., 2026). VIPA-VLA learns visual-physical alignment from human videos with 3D supervision (Feng et al., 2026), while 3DVLA combines geometry encoding, instance representations, and masked feature completion to address spatial reasoning and occlusion (Xia et al., 2026). PixelVLA targets finer visual grounding through multi-scale pixel-aware features and visual-prompt instruction tuning (Liang et al., 2026). Teacher supervision and predictive objectives offer ways to enrich the representations used for actions prediction. Spatial Forcing transfers geometric priors by aligning intermediate VLA visual features with spatial teacher features (Li et al., 2026a). FALCON maintains a separate spatial model whose features are fused with VLM features for action prediction (Zhang et al., 2025b). DreamVLA uses modality-specific queries and structured attention to forecast future dynamic, spatial, and semantic information (Zhang et al., 2026). VLA-JEPA predicts future latent representations from current observations during pretraining, then fine-tunes an action head (Sun et al., 2026). LaMP learns a future 3D scene-flow prior and fuses partially denoised motion features into VLM features through gated cross-attention before action generation (Wang et al., 2026a). TAGER focuses on distilling the current observation through dual independently parameterized query transformers. Their compact outputs condition the action expert and reconstruct dense geometric and sematic teacher features during training. This moves teacher matching to query-decoded outputs while retaining the VLM context for control. The teachers and reconstruction heads are omitted at inference. Its temporal branch conditions a velocity-field residual on observed execution responses, rather than generating a future scene-flow prior or applying a gate to the conditioning features.

Preserving pretrained knowledge. Adapting VLMs for control requires balancing action learning with retention of pretrained representations. InstructVLA combines mixture-of-experts adaptation with multi-modal and robotic instruction tuning to retain vision-language capabilities while

learning manipulation (Yang et al., 2026). Knowledge insulation blocks gradients from a continuous action expert into the VLM while adapting the backbone through discrete action prediction and vision-language co-training (Driess et al., 2026). MAPS schedules module-specific proximity constraints to preserve pretrained knowledge while allowing action-oriented components to adapt (Huang et al., 2026). MergeVLA addresses interference between task-specific updates when combining VLA experts, using task-masked LoRA and cross-attention-only action experts to retain mergeability (Fu et al., 2026). Its focus on cross-skill composition is distinct from retaining vision-language priors during adaptation to control. TAGER addresses the related question of where to apply auxiliary spatial supervision. Attention independent query transformers and reconstruction heads replace direct matching of shared visual tokens, providing additional capacity to transform VLM context into teacher-specific representations. The reconstruction losses remain differentiable through that context, so this design does not impose gradient isolation.

B Local Visual–Motor Analysis of Action Response Residual Flow

This appendix provides proofs for Propositions 1 and 2, together with additional interpretations of the Action Response Residual Flow. These analyses serve only as theoretical motivation and do not introduce any additional estimator, controller, or training objective. In implementation, the module directly encodes observed action–response transitions and predicts a residual correction to the generative velocity field, without explicitly estimating the local Jacobians.

B.1 Physical motion and visual changes

For a fixed camera, a three-dimensional point $P_c = (X_c, Y_c, D_c)$ with positive depth projects to

$$\pi(P_c) = \begin{bmatrix} f_x X_c / D_c + c_x \\ f_y Y_c / D_c + c_y \end{bmatrix}, \qquad D\pi(P_c) = \begin{bmatrix} f_x/D_c & 0 & -f_x X_c / D_c^2 \\ 0 & f_y/D_c & -f_y Y_c / D_c^2 \end{bmatrix}. \tag{10}$$

Here f_x, f_y, c_x, c_y are intrinsic camera parameters. If the point is attached to a robot link, let q^{rob} be local robot configuration coordinates, J_{kin} its position Jacobian, and R_{cw} the world-to-camera rotation. The chain rule gives

$$\Delta\pi = D\pi(P_c) R_{cw} J_{\mathrm{kin}} \Delta q^{\mathrm{rob}} + \mathcal{O}(\|\Delta q^{\mathrm{rob}}\|^2). \tag{11}$$

This recovers the local interaction-matrix perspective commonly used in visual servoing (Chaumette & Hutchinson, 2006): the mapping from physical motion to image-space displacement depends explicitly on depth, viewpoint, and robot configuration. Since the projection of a single point has rank at most two, its image coordinates alone do not determine a unique three-dimensional motion. This derivation is intended only as an illustrative local example. Our latent query representations are not tracked image points, and practical manipulation additionally involves moving wrist cameras, object motion, and contact dynamics. We therefore adopt the more general local state representation introduced below.

Fix the language instruction and the network parameters. Let $\xi = (s, \eta)$ collect the robot state s and the environment state η needed to describe the local configuration and its transitions, including object configurations, camera viewpoint, and illumination. Commas separate components of this ordered pair. In the noiseless local model, the multi-view RGB images I are observations formed under these conditions; the fixed vision-language and query encoders then produce H^{sam} from I and the instruction. Thus, the joint map below includes both image formation and query encoding. The images are intermediate observations of the underlying robot and environment states, not additional coordinates of ξ. We write

$$\Phi(\xi) = \begin{bmatrix} s \\ H^{\mathrm{sam}} \end{bmatrix}.$$

Brackets collect the robot-state vector and query-token matrix as two blocks. The norm on this product space combines the Euclidean norm of the state block and the Frobenius norm of the query block in quadrature. The Jacobian acts as a linear map into this product space, so no explicit flattening is needed. Assume that Φ is continuously differentiable on a convex neighborhood containing the

nominal and current operating points and all historical transition segments considered below, and that its Jacobian

$$J(\xi) = \frac{\partial \Phi(\xi)}{\partial \xi}$$

is L-Lipschitz in operator norm. For the perturbed rollout, let t denote the current replanning time and $\xi_t = (s_t, \eta_t)$ the robot and environment states at that time. The corresponding Jacobian is $J_{\mathrm{pert}} = J(\xi_t)$: t indexes execution time, while pert labels the operating conditions.

Local regularity. Visual servoing relates visual-feature changes to physical motion through interaction matrices (Chaumette & Hutchinson, 2006), while Castin et al. (2023) bound self-attention sensitivity on compact token domains. These results motivate a local analysis; they do not establish the Lipschitz-Jacobian condition for the full state–query map. A sufficient condition is that observation formation and the fixed query encoder are twice continuously differentiable on an open neighborhood with unchanged visibility and contact modes. With a fixed attention mask, their composition then has bounded second derivatives on a compact convex subset, yielding a Lipschitz Jacobian there. The state block is linear. We retain this as a local modeling assumption, with noise and changes of visibility or contact regime discussed below.

B.2 How perturbations change the local map

Under a fixed instruction and fixed network parameters, let $\xi_t^{\mathrm{nom}} = (s_t^{\mathrm{nom}}, \eta_t^{\mathrm{nom}})$ denote a nominal operating point and $\xi_t = \xi_t^{\mathrm{nom}} + e_t$ a nearby perturbed operating point, with both lying within the same smooth neighborhood. The perturbation e_t compares nominal and perturbed operating conditions at the current replanning step; it does not denote the state increment of a historical transition. The labels nom and pert distinguish operating conditions at the same replanning time t, rather than two time steps. In particular, ξ_t^{nom} is a nearby reference at time t, not the initial state of the rollout. Writing $J_{\mathrm{nom}} = J(\xi_t^{\mathrm{nom}})$ and $\delta J = J_{\mathrm{pert}} - J_{\mathrm{nom}}$, the L-Lipschitz continuity of J implies

$$\|\delta J\| = \|J(\xi_t^{\mathrm{nom}} + e_t) - J(\xi_t^{\mathrm{nom}})\| \le L\|e_t\|. \tag{12}$$

Thus, δJ characterizes the change in the local representation map induced by the perturbation e_t, rather than the physical perturbation itself. Equation (11) provides a concrete geometric interpretation: viewpoint and depth modify the projection geometry and camera transformation, whereas robot configuration alters the kinematic Jacobian and the corresponding point location. This local analysis applies to continuous changes in operating conditions; it does not treat instruction variations or arbitrary image noise as smooth perturbations of the same local map.

B.3 Residual adaptation of the action-generating field

Even for the same policy input, a change in the local visual–motor relationship may require different actions to achieve the task. We describe this dependence through the ideal velocity field of the action-generating ODE. Here, J denotes the joint robot-state and query Jacobian defined above, with $J_{\mathrm{nom}} = J(\xi_t^{\mathrm{nom}})$ serving as the nominal local map. We write $\widetilde{v}_{t,\tau}^{\star}(J)$ for the ideal field at fixed current policy inputs, including the noisy action chunk z_τ, flow time τ, and conditioning context, and define its nominal value as $v_{t,\tau}^{\star} = \widetilde{v}_{t,\tau}^{\star}(J_{\mathrm{nom}})$. These symbols mirror the nominal and corrected learned velocities $v_{t,\tau}$ and $\widetilde{v}_{t,\tau}$, while $\star$ distinguishes the ideal fields from their learned counterparts.

Proof of Proposition 1. Equation (12) gives $\|\delta J\| \le L\|e_t\|$. As a local modeling assumption, $\widetilde{v}_{t,\tau}^{\star}(J)$ is Fréchet differentiable with respect to J at J_{nom}, with the current policy inputs held fixed. By the definition of differentiability,

$$\widetilde{v}_{t,\tau}^{\star}(J_{\mathrm{nom}} + \Delta J) = v_{t,\tau}^{\star} + D_J \widetilde{v}_{t,\tau}^{\star}(J_{\mathrm{nom}})[\Delta J] + R(\Delta J), \tag{13}$$

Here, $D_J \widetilde{v}_{t,\tau}^{\star}(J_{\mathrm{nom}})$ is the derivative of the ideal velocity field with respect to J, evaluated at J_{nom} while holding the current policy inputs fixed. It maps a Jacobian change ΔJ to the corresponding first-order change in the ideal velocity field. The remainder R satisfies

$$R(0) = 0, \qquad \lim_{\substack{\Delta J \to 0 \\ \Delta J \neq 0}} \frac{\|R(\Delta J)\|}{\|\Delta J\|} = 0. \tag{14}$$

Setting $\Delta J = \delta J$ gives the $o(\|J_{\text{pert}} - J_{\text{nom}}\|)$ remainder in Eq. (6), proving Proposition 1. Equation (12) ensures that $\delta J \to 0$ as the robot and environment state perturbation e_t tends to zero; no quadratic rate for the velocity remainder is assumed. □

The field $\widetilde{v}^{\star}_{t,\tau}(J)$ is a generative velocity over action chunks. Its differentiability with respect to J is an independent local modeling assumption; it is not implied by encoder smoothness, the flow-matching objective, or the local response relation in Proposition 2. In practice, a perturbation may simultaneously affect the current policy inputs, the local representation map J, and the physical transition dynamics. Our expansion isolates only the variation induced through J while holding the current inputs fixed, and therefore does not capture all possible changes in the action-to-state dynamics.

B.4 Why differences and a local window

To provide observed execution context for the learned residual, we use $i = t - K, \ldots, t - 1$ to index the completed historical transitions before the current replanning time t. In the local state coordinates defined above, let the executed command a_i induce

$$\xi_{i+1} = \xi_i + f(a_i; s_i, \eta_i). \tag{15}$$

Here, f is a fixed, possibly nonlinear transition-increment function. The semicolon separates the action from the initial conditions under which its response is evaluated. Its output contains both robot and environment increments: $\Delta\xi_i = (\Delta s_i, \Delta\eta_i) = f(a_i; s_i, \eta_i)$, where $\Delta\eta_i = \eta_{i+1} - \eta_i$. The environment increment belongs to the theoretical state model; the encoder instead observes the visual response through ΔH_i^{sam}. For the short-window approximation, we use the current conditions (s_t, η_t) as a common reference and account for deviations of historical conditions through an error term. This does not require the historical states to be identical or the state to remain unchanged during execution. Importantly, the commanded action need not coincide with the realized state change; a_i may represent a target position or another control input whose physical effect depends on the current local state. This formulation assumes only that ξ_i contains the variables required to characterize the local transition. The implementation does not estimate or evaluate f explicitly, but instead directly uses the executed actions and their observed responses.

Transition regularity. We additionally assume that the transition function is defined at both the historical and current initial conditions for each historical action, within this common regime, and satisfies

$$\|f(a_i; s_i, \eta_i) - f(a_i; s_t, \eta_t)\| \le L_f \|\xi_i - \xi_t\|. \tag{16}$$

Here, L_f is a local Lipschitz constant uniform over the historical actions. This bounds the error from using the current conditions as a reference; it imposes no linearity assumption on the dependence on action. This is an independent local dynamics assumption. A sufficient condition is that f is continuously differentiable in (s, η), with derivatives uniformly bounded over the historical actions on the common convex neighborhood.

Proof of Proposition 2. Using the common representation map and regularity assumptions above, denote $J_i = J(\xi_i)$ and $\Delta\xi_i = \xi_{i+1} - \xi_i$. The fundamental theorem of calculus yields

$$\begin{aligned}\begin{bmatrix}\Delta s_i \\ \Delta H_i^{\text{sam}}\end{bmatrix} - J_{\text{pert}}\Delta\xi_i &= (J_i - J_{\text{pert}})\Delta\xi_i \\ &\quad + \int_0^1 \big[J(\xi_i + u\Delta\xi_i) - J_i\big]\Delta\xi_i \,\mathrm{d}u.\end{aligned} \tag{17}$$

Bounding the first term by $L\|\xi_i - \xi_t\|\|\Delta\xi_i\|$ and the integral by $\int_0^1 Lu\|\Delta\xi_i\|^2 \,\mathrm{d}u$ yields

$$\|r_{i,t}^{\Phi}\| \le L\|\xi_i - \xi_t\|\,\|\Delta\xi_i\| + \frac{L}{2}\|\Delta\xi_i\|^2, \tag{18}$$

where $r_{i,t}^{\Phi} = \Phi(\xi_{i+1}) - \Phi(\xi_i) - J_{\text{pert}}\Delta\xi_i$ is the representation linearization remainder. Substituting $\Delta\xi_i = f(a_i; s_i, \eta_i)$ and adding and subtracting $J_{\text{pert}} f(a_i; s_t, \eta_t)$ gives

$$\begin{aligned}\begin{bmatrix}\Delta s_i \\ \Delta H_i^{\text{sam}}\end{bmatrix} &= J_{\text{pert}} f(a_i; s_t, \eta_t) + \rho_{i,t}, \\ \rho_{i,t} &= r_{i,t}^{\Phi} + J_{\text{pert}}\big[f(a_i; s_i, \eta_i) - f(a_i; s_t, \eta_t)\big].\end{aligned} \tag{19}$$

Combining Eqs. (16) and (18) yields

$$\|\rho_{i,t}\| \le \big(\|J_{\mathrm{pert}}\|L_f + L\|\Delta\xi_i\|\big)\|\xi_i - \xi_t\| + \frac{L}{2}\|\Delta\xi_i\|^2. \tag{20}$$

This proves Proposition 2. The robot-state block of $r^{\Phi}_{i,t}$ is zero because that block of Φ is the identity on s. The total error $\rho_{i,t}$ need not have a zero robot-state block, since it also includes the change in physical response between historical and current initial conditions. □

Conditioned on (s_t, η_t), the function $a \mapsto J_{\mathrm{pert}} f(a; s_t, \eta_t)$ is a common local response map that may be nonlinear in action. Historical triplets provide approximate input–output observations of this map at the executed commands. The error bound separates finite-transition linearization from drift in the initial conditions, including both the representation Jacobian and the physical response. A short history favors this approximation when recent transitions remain near the current operating point; temporal proximity alone does not ensure this condition. Small transitions alone also do not eliminate the condition-drift term $\|J_{\mathrm{pert}}\|L_f\|\xi_i - \xi_t\|$. Shorter histories are not always preferable, since very short windows may contain too few transitions to characterize recent execution responses.

The observed robot state does not, in general, fully determine the underlying scene configuration. To account for additional query noise, partition the local Jacobian as $J_{\mathrm{pert}} = [J^s_{\mathrm{pert}}, J^{\eta}_{\mathrm{pert}}]$ and replace the ideal observation model by

$$\begin{bmatrix} s_i \\ H^{\mathrm{sam}}_i \end{bmatrix} = \Phi(\xi_i) + \begin{bmatrix} 0 \\ n_i \end{bmatrix},$$

where n_i denotes query representation noise. This gives the observed relation

$$\begin{bmatrix} \Delta s_i \\ \Delta H^{\mathrm{sam}}_i \end{bmatrix} = J^s_{\mathrm{pert}}\Delta s_i + J^{\eta}_{\mathrm{pert}}\Delta\eta_i + \begin{bmatrix} 0 \\ \Delta n_i \end{bmatrix} + r^{\Phi}_{i,t}. \tag{21}$$

Here, $r^{\Phi}_{i,t}$ is the Taylor remainder of the noise-free map Φ, rather than the total reference-condition error $\rho_{i,t}$ in Proposition 2. Object motion, changes in imaging conditions, and measurement noise cannot in general be neglected without additional assumptions. If an unmodeled disturbance w_i enters the physical transition as $\Delta\xi_i = f(a_i; s_i, \eta_i) + w_i$, the response relative to the current reference conditions also contains $J_{\mathrm{pert}} w_i$, with the representation remainder evaluated along the realized transition. The action–response triplets record commanded inputs and realized outcomes. They do not attribute every observed change solely to the executed action, nor do they identify the full Jacobian or transition function. Their role in TAGER is to supply observed execution context. Likewise, temporal differences provide local change measurements; they are neither inherently denoised nor guaranteed to correspond to fixed physical points. When abrupt contact events or visibility changes violate the single smooth-neighborhood assumption, the local expansion should instead be interpreted piecewise over regimes in which the smoothness conditions remain valid.

The two propositions serve distinct roles: the first motivates an additive parameterization for a change of the ideal generative field, and the second relates nearby historical responses to a common current-condition response map with bounded approximation error. Neither establishes that a short history identifies J or that the learned encoder can recover every required correction. In Eq. (8), the full gated residual $g_{t,\tau} \odot \Delta v_{t,\tau}$ plays the role of the learned correction. Neither it nor the nominal prediction $v_{t,\tau}$ is assumed to equal its ideal counterpart in Eq. (6); their jointly trained decomposition need not be unique. Whether this history conditioning improves action adjustment is evaluated empirically. The local analysis does not guarantee closed-loop stability or robustness to arbitrarily large perturbations.

C Details of Compared Models

We summarize the problems and core designs of the baseline methods included in our experiments.

OpenVLA. OpenVLA (Kim et al., 2024) provides an open-source generalist policy that can be adapted to new manipulation tasks with limited computing resources. It combines a Llama 2 backbone with fused DINOv2 and SigLIP visual features, and learns to predict discrete action tokens from diverse robot demonstrations.

CoT-VLA. CoT-VLA (Zhao et al., 2025) introduces intermediate visual planning into policies that map observations and instructions to actions. It predicts a future subgoal image and then generates a short action sequence conditioned on that image, connecting the anticipated scene with low-level robot control.

OpenVLA-OFT. OpenVLA-OFT (Kim et al., 2025) studies how to adapt pretrained VLAs to new robot setups while improving control performance and inference speed. Its optimized fine-tuning recipe combines parallel decoding, action chunking, continuous action outputs, and an L_1 regression objective.

CogVLA. CogVLA (Li et al., 2026c) targets the computational cost of processing dense visual tokens in VLA models. It uses instruction-conditioned visual aggregation and token pruning to retain task-relevant information, together with an attention scheme that couples causal vision-language processing and parallel action decoding.

PixelVLA. PixelVLA (Liang et al., 2026) addresses coarse scene perception and reliance on textual instructions in VLA policies. It combines multiscale pixel-aware perception with a visual prompt encoder, using automatically generated pixel-level annotations from robot data for visuomotor instruction tuning.

π_0. π_0 (Black et al., 2024) connects pretrained vision-language knowledge with continuous control for dexterous manipulation. It uses a flow-matching action expert to generate action chunks conditioned on a VLM backbone, with broad training across robot platforms followed by task-specific fine-tuning.

$\pi_{0.5}$. $\pi_{0.5}$ (Physical Intelligence et al., 2025) focuses on generalization to environments and objects outside the robot training distribution. It combines heterogeneous robot data, semantic prediction tasks, and web data in a co-training recipe that connects high-level task understanding with continuous action generation.

VLA-Adapter. VLA-Adapter (Wang et al., 2026b) studies efficient adaptation from vision-language representations to robot actions without large-scale robotic pretraining. Its lightweight policy module uses Bridge Attention to inject relevant vision-language features into action generation, reducing the cost of connecting a pretrained VLM to control.

MemoryVLA. MemoryVLA (Shi et al., 2026) addresses the limited temporal context available to policies that primarily use the current observation. It maintains perceptual details and cognitive semantics in a memory bank, retrieves relevant historical entries, and fuses them with current features to condition a diffusion action expert.

X-VLA. X-VLA (Zheng et al., 2026) addresses differences in observation and action spaces across robot embodiments. It conditions a shared flow-matching transformer on learned embodiment-specific soft prompts, allowing the policy to use heterogeneous robot data and adapt to new embodiments.

FASTer. FASTer (Liu et al., 2026) targets the trade-off between action reconstruction fidelity and autoregressive inference efficiency. It learns a compact action representation through residual vector quantization and uses blockwise autoregressive decoding, predicting tokens within each block in parallel.

InstructVLA. InstructVLA (Yang et al., 2026) addresses the loss of pretrained vision-language capabilities during manipulation-focused adaptation. It jointly tunes embodied reasoning and action generation through mixture-of-experts adaptation, combining vision-language corpora with an instruction dataset for robotic interaction.

π_0-FAST. π_0-FAST (Pertsch et al., 2025) addresses the difficulty of learning high-frequency actions with independently discretized action values. It uses discrete cosine transforms and byte-pair

encoding to compress action chunks into tokens, enabling autoregressive action prediction with shorter target sequences.

Spatial Forcing. Spatial Forcing (Li et al., 2026a) addresses limited spatial awareness in VLAs built on vision-language backbones pretrained primarily on 2D images. It aligns intermediate visual embeddings with geometric features from pretrained 3D foundation models, providing spatial supervision without requiring explicit 3D observations as policy inputs.

SRPO. SRPO (Fei et al., 2026) addresses sparse success rewards that overlook partial progress in failed rollouts. It uses successful trajectories from the current training batch as references and compares trajectories in a world model's latent space to assign progress-based rewards for policy optimization.

MergeVLA. MergeVLA (Fu et al., 2026) addresses interference between task-specific updates when combining independently trained VLA experts. It uses task-masked LoRA and action experts composed of cross-attention-only blocks to facilitate model merging, with a task router selecting the appropriate mask and action head.

VLA-JEPA. VLA-JEPA (Sun et al., 2026) addresses appearance bias and reliance on future observations when inferring latent actions from video. It predicts future representations from the current observation while using future frames only as supervision, then fine-tunes an action head to connect the learned representations with control.

RIPT-VLA. RIPT-VLA (Tan et al., 2025) studies how demonstration-trained VLA policies can improve through environment interaction when expert data are scarce. It fine-tunes policies with binary success rewards, combining dynamic rollout sampling and leave-one-out advantage estimation for policy optimization.

ACoT-VLA. ACoT-VLA (Zhong et al., 2026) addresses the limited action detail conveyed by intermediate language descriptions or synthesized goal images. It combines explicit coarse reference trajectories with implicit action priors extracted from multimodal representations, using both forms of action-space reasoning to condition the action head.

D Implementation Details

D.1 Spatial Query Attention and Reconstruction

Layerwise query attention. Let L be the number of transformer layers and $\ell \in \{1, \dots, L\}$ the layer index. As in Section 3.2, $k \in \{0, 1\}$ indexes SAM3 and VGGT, respectively. Descriptive superscripts identify the same branches: $H_t^{\mathrm{sam}} \equiv H_t^0$ and $H_t^{\mathrm{vggt}} \equiv H_t^1$, with the same convention for T^k and $\mathcal{L}_{\mathrm{align}}^k$. The VLM context at observation step t in Section 3.2 is $\mathcal{C}_t = \{\mathcal{C}_{t,\ell}\}_{\ell=1}^L$, where $\mathcal{C}_{t,\ell}$ supplies the VLM keys $K_{t,\ell}^{\mathrm{vlm}}$ and values $V_{t,\ell}^{\mathrm{vlm}}$. Write $H_{t,0}^k = T^k$ for the initial tokens and $H_{t,L}^k = H_t^k$ for the final output. At layer ℓ, branch k projects $H_{t,\ell-1}^k$ into attention queries $q_{t,\ell}^k$, keys $K_{t,\ell}^k$, and values $V_{t,\ell}^k$. Lowercase q denotes an attention projection; uppercase T^k denotes the learnable initial tokens. The attention output is

$$O_{t,\ell}^k = \mathrm{Attn}\left(q_{t,\ell}^k, [K_{t,\ell}^k; K_{t,\ell}^{\mathrm{vlm}}], [V_{t,\ell}^k; V_{t,\ell}^{\mathrm{vlm}}]\right), \tag{22}$$

where Attn is scaled dot-product attention and $[\,;\,]$ concatenates tokens. The transformer block uses $O_{t,\ell}^k$ to update the query states. The attention mask permits each query branch to read itself and the VLM, but blocks attention between the two query branches and from the VLM to the queries. The action expert reads cached keys and values from all three streams at corresponding layers. These forward attention restrictions do not stop reconstruction gradients from reaching the VLM through its context.

Dense reconstruction and alignment loss. In Eqs. (4)–(5), M_k and N_k count all query tokens and teacher target locations, respectively, across configured camera groups. In the LIBERO implementation, these groups include a padded camera slot in addition to the two valid camera views; the

Table 5: Training and model configurations of TAGER.

Parameter	**Setting**
Training configuration	
Training framework	JAX
Number of GPUs	8
FSDP devices	2
Global batch size	128
Training steps	30,000
Data loader workers	8
Optimizer	AdamW
Peak learning rate	5×10^{-5}
Learning-rate schedule	Warmup cosine decay
Warmup steps	10,000
Decay steps	1,000,000
Final learning rate	5×10^{-5}
AdamW coefficients	$\beta_1 = 0.9,\ \beta_2 = 0.95$
Weight decay	1×10^{-10}
Gradient clipping norm	1.0
EMA decay	0.999
Model configuration	
Action prediction horizon H	10
History transitions K	3
Teacher streams	SAM3 and VGGT
Camera/query slots per teacher	3
Tokens per camera slot per teacher	16
Total tokens per teacher M_k	48
Query hidden dimension d_k	512
SAM3 target feature dimension c_0	1024
VGGT target feature dimension c_1	2048
SAM3 alignment loss coefficient λ_{sam}	0.4
VGGT alignment loss coefficient λ_{vggt}	0.1

masks below select valid reconstruction targets. The initial tokens T^k and final states H_t^k both have shape $M_k \times d_k$. The final states retain M_k tokens, while the teacher-specific decoder D_{ψ_k} expands them to N_k predictions. For each camera, learned output slots with two-dimensional positional embeddings attend to projected query states from that view. Each slot predicts the feature at one target grid location. The reconstructed and teacher grids $\widehat{F}_t^k$ and F_t^k both have shape $N_k \times c_k$. Thus, $\widehat{F}_{t,j}^k$ is a decoder prediction and $F_{t,j}^k$ is the frozen teacher feature at spatial location j, both with teacher feature dimension c_k. We use j for spatial locations and reserve i for historical transitions.

Let $m_{t,j}^k \in \{0, 1\}$ mark valid teacher locations, excluding padding and invalid camera views. The alignment objective is

$$\mathcal{L}_{\text{align}}^k = \mathbb{E}\left[\frac{\sum_{j=1}^{N_k} m_{t,j}^k \left[1 - \cos\left(\widehat{F}_{t,j}^k, F_{t,j}^k\right)\right]}{\max\left(1, \sum_{j=1}^{N_k} m_{t,j}^k\right)}\right], \tag{23}$$

where cos denotes cosine similarity. We first average over each example's valid locations, with the denominator clamped to at least one, then take the expectation over training examples. The objective compares reconstructed features with teacher targets; it does not directly match the shared VLM states to teacher features.

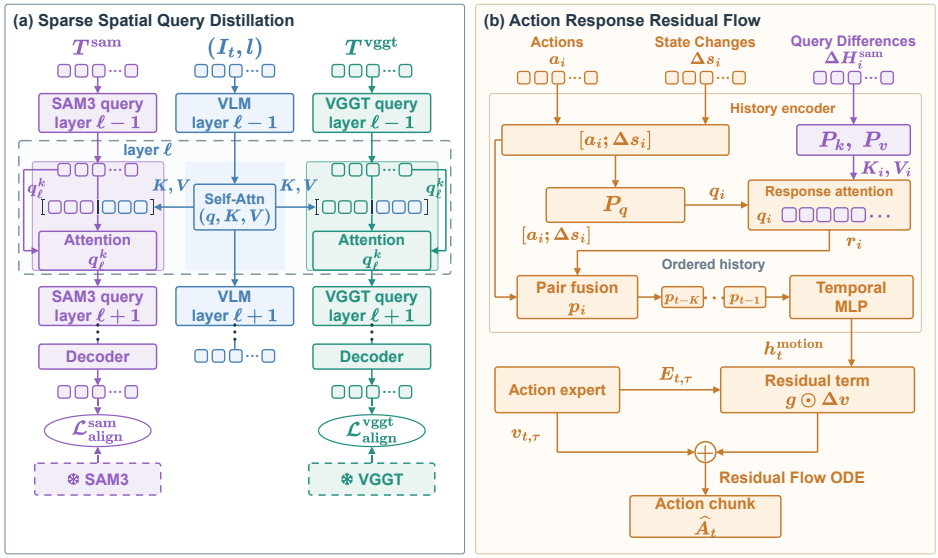


Figure 6: **Spatial Supervision and Action Adjustment.** (a) Query attention and dense teacher-feature reconstruction. (b) History encoding and gated residual heads for action generation. Uppercase T denotes learnable initial tokens and lowercase q denotes projected attention queries; time indices on g and Δv are omitted for clarity.

D.2 Action Response Encoding and Residual Heads

History encoder. At replanning step t, the encoder uses the K past transitions $i = t-K, \ldots, t-1$, requiring $K+1$ observations including the current one. Each pairs executed action a_i with the robot-state and SAM3 query-state changes Δs_i and ΔH_i^{sam} defined in Section 3.3. Learned projections P_q, P_k, and P_v form $q_i = P_q([a_i; \Delta s_i])$, $K_i = P_k(\Delta H_i^{\text{sam}})$, and $V_i = P_v(\Delta H_i^{\text{sam}})$, respectively. Their attention output is

$$r_i = \text{Attn}(q_i, K_i, V_i). \tag{24}$$

Here, r_i summarizes the query-state response conditioned on the action and robot-state change; K_i denotes attention keys, distinct from the history length K. We form transition descriptors $p_i = f_{\text{pair}}([r_i; a_i; \Delta s_i])$ and the execution context $h_t^{\text{motion}} = f_{\text{temp}}([p_{t-K}; \ldots; p_{t-1}])$. Both f_{pair} and f_{temp} are MLPs with input layer normalization; $[\,;\,]$ here concatenates features and, for f_{temp}, preserves temporal order. The context is computed once per replanning step and reused throughout the generated action chunk.

Residual and gate. Let $E_{t,\tau} \in \mathbb{R}^{H \times d_e}$ contain the final action-expert features at flow time τ, where H is the action chunk length and d_e the expert feature dimension. A learned projection P_{m} maps the execution context to d_e dimensions and broadcasts it across the H action tokens:

$$\begin{aligned} U_{t,\tau} &= E_{t,\tau} + \mathbf{1}_H [P_{\text{m}}(h_t^{\text{motion}})]^\top, \\ \Delta v_{t,\tau} &= f_{\text{res}}(U_{t,\tau}), \qquad g_{t,\tau} = \sigma(f_{\text{gate}}(U_{t,\tau})). \end{aligned} \tag{25}$$

Here, $\mathbf{1}_H$ is a length-H column of ones, $U_{t,\tau}$ contains the conditioned features, f_{res} and f_{gate} are lightweight MLP heads, and σ is the elementwise sigmoid. The nominal velocity $v_{t,\tau}$, residual $\Delta v_{t,\tau}$, gate $g_{t,\tau}$, and corrected velocity $\widetilde{v}_{t,\tau}$ all have shape $H \times d_a$, matching the action chunk A_t. The residual and gate modify the nominal field as specified in Eq. (8). This velocity belongs to the generative flow over action chunks, rather than denoting physical robot velocity.

Training. The VLM backbone and action expert are initialized from pretrained $\pi_{0.5}$ weights. VGGT and SAM3 remain frozen. Each query branch has independent parameters and a teacher-specific reconstruction decoder. Query states receive supervision from both teacher-feature reconstruction and action prediction, whereas the reconstruction decoders are optimized only by their respective reconstruction objectives. The residual scaling coefficient is fixed to 1 in both training and inference.

Inference. In our LIBERO configuration, proprioceptive state enters the policy through the action–response history encoder. The noisy action chunk z_τ is embedded into the action stream, while flow time τ conditions adaptive normalization in the action expert. Teacher encoders and reconstruction decoders are discarded. The VLM context, query representations, and action–response execution context are computed once per replanning step and reused throughout action-chunk generation. The residual and gate are recomputed at each flow step from the current action-expert states, as described in Section 3.3.

E Additional Temporal Ablations

Temporal Context and Camera Views. Table 6 varies the temporal window and the camera views used for action–response conditioning. With both main and wrist views, increasing the window from one to three observation frames raises success from 86.1% to 87.6%; five frames provide no further gain at the reported precision. At a fixed three-frame window, wrist-only history achieves success rate of 85.6%, compared with 86.4% for main-view-only history. These results favor a short recent history in this setting and show no additional gain from using both views over wrist history alone at the reported precision.

Table 6: Action Response Residual Flow ablations on LIBERO-Plus. We vary the temporal window and the camera views used for motion conditioning. Frames denotes the number of observations, including the current frame; a window of F frames contains $K = F - 1$ historical transitions. Main and Wrist denote the main and wrist-mounted cameras, respectively. Success rates (SR) are reported as percentages (%).

Frames	Camera	SR (%)
1	Main + Wrist	86.1
3	Main + Wrist	**87.6**
5	Main + Wrist	**87.6**
3	Main	86.4
3	Wrist	86.5

F Action Adjustment under Perturbations

We examine Action Response Residual Flow through matched LIBERO-Plus rollouts. Base denotes the retrained $\pi_{0.5}$ policy, Query adds query distillation, and Query + Residual Flow further adds the action response encoder and residual flow. For this component comparison, the three policies share the task, instruction, perturbation, and simulator initial state. We visualize intermediate action changes and each rollout's actual endpoint. Numbers denote control steps, blue outlines highlight adjustments, and the last column shows each actual endpoint. Checks and crosses follow the logged full-task success criteria.

Correction after contact. After initial contact without lifting the bowl, Query + Residual Flow retracts, reorients, and completes the transfer at step 171 (Figure 7). Base and Query also adjust their poses but do not finish by step 220.

Adjustment between task stages. Figure 8 shows the task of placing a bowl in the bottom drawer and then closing it under a robot-initial-state perturbation. After placing the bowl, the residual-flow variant withdraws the gripper, reorients toward the drawer front, and completes the task at step 224. The query-only variant remains close to the bowl during the corresponding trajectory segment, while neither Base nor Query satisfies the full-task success criterion within 520 steps. This example illustrates adjustment between successive task stages.

Execution with degraded visual observations. In Figure 9, Query + Residual Flow completes the transfer at step 190, whereas Base and Query do not satisfy the task-success criterion within 280

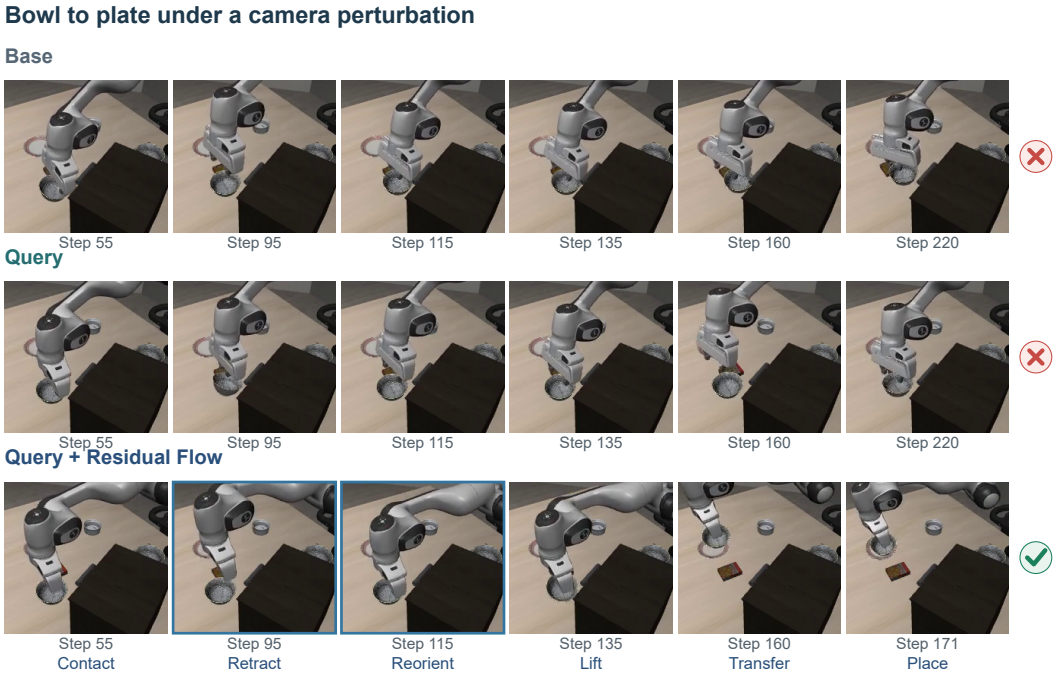


Figure 7: **Camera perturbation: post-contact adjustment.** Blue outlines highlight retraction and reorientation after initial contact.

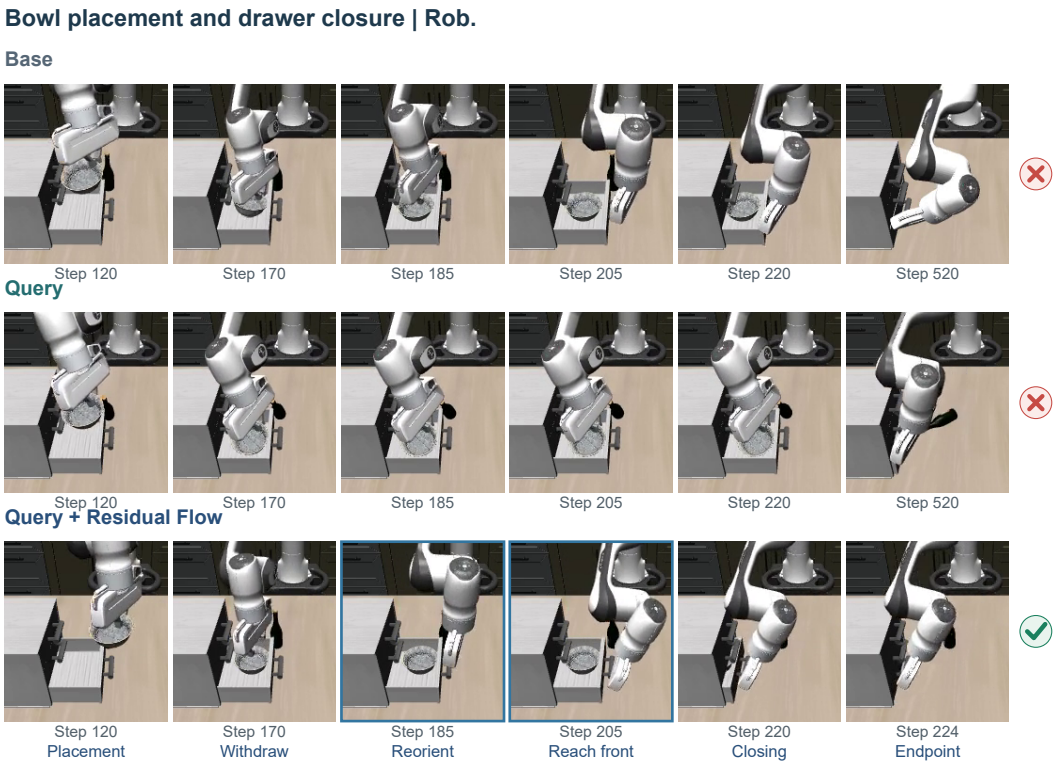


Figure 8: **Robot-initial-state perturbation: placement-to-closing transition.** Blue outlines mark gripper reorientation and the approach to the drawer front.

steps. Severe blur obscures fine contact events, so this case supports successful execution with degraded visual input without identifying a specific recovery mechanism. Together, the three selected examples complement the aggregate component ablations in Figure 3; they illustrate execution behavior rather than estimate overall success rates or establish causal attribution.

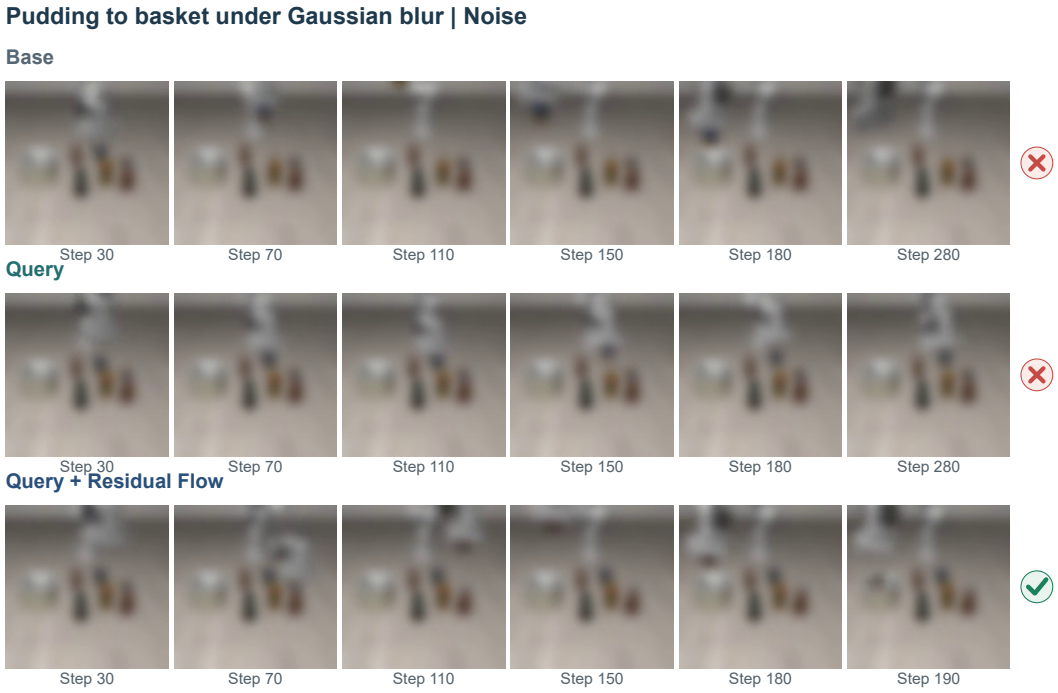


Figure 9: **Noise perturbation: execution under Gaussian blur.** Matched rollouts for placing the chocolate pudding in the basket under LIBERO-Plus `noise_19` ($\sigma = 9$ pixels). The original blurred observations are shown without additional processing.

G Real-World Feasibility Details

Clean-condition fine-tuning. We collect 100 demonstration episodes per task under clean conditions and use these demonstrations to fine-tune TAGER. The zero-shot setting of interest concerns transfer to perturbed conditions after this task-specific data collection, without additional adaptation to those conditions.

Experiment setting. For each task, the source collection contains 100 episodes, with three synchronized 640×480 RGB video streams at 30 Hz. The recorded platform is identified as ALOHA. State and action vectors each contain six joint coordinates and one gripper coordinate per arm. The task names in Figure 5 describe visible behaviors.

H Spatial Query Probe Visualization

Data and checkpoints. Figure 4 uses main-camera frames 0, 20, 40, 60, and 90 from LIBERO demonstration episode 8, whose instruction is "put both the alphabet soup and the tomato sauce in the basket." The sequence contains 259 observations. SAM3-query and VGGT-query readouts use separately fine-tuned policy checkpoints. Both policies remain frozen during linear-probe training.

Readouts and interpretation. For the SAM3 branch, we apply an affine probe to the final query token of each camera stream, mapping its 512-dimensional representation to 32×32 body-ID logits.

The body IDs distinguish object instances within the simulator scene and should not be interpreted as task-invariant semantic categories. For the VGGT branch, we train a linear depth probe on frozen query representations. Since the exported predictions do not provide a calibrated metric depth scale, we retain the original depth visualization rather than assigning absolute depth values. Importantly, Figure 4 visualizes predictions from the learned probes, rather than teacher outputs or ground-truth segmentation and depth annotations.

I Inference Efficiency and Model Size

Evaluation setting. We compare the $\pi_{0.5}$ baseline, the VGGT query transfomer(VQ) + SAM3 query transformer(SQ) configuration, and the full TAGER model in a single-task evaluation on an NVIDIA H20 GPU. VQ + SQ includes both spatial query branches without the Action Response Residual Flow. Each latency trace contains ten measured control steps. TAGER initializes the temporal-difference state at the first step (193.76 ms in this run) and reuses cached history thereafter. Figure 10 shows all ten steps in a single latency plot. We compare post-initialization means over steps 2–10 for all three configurations and retain their reported parameter counts.

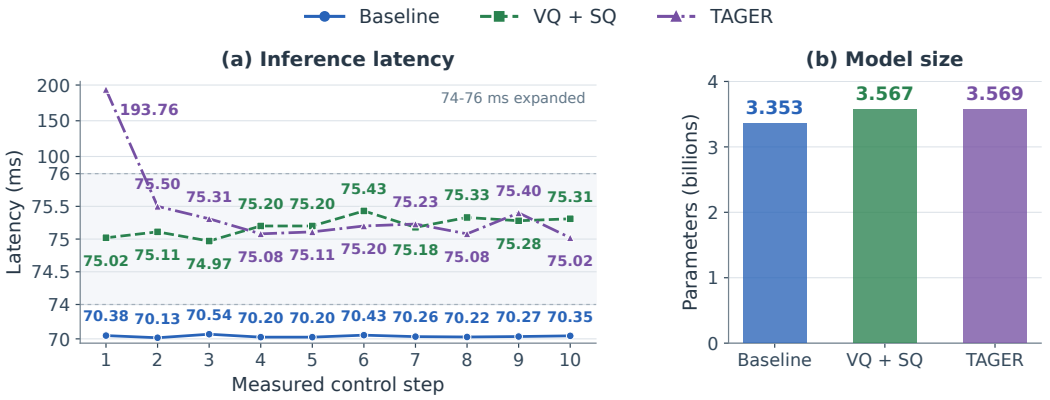


Figure 10: **Inference cost on an NVIDIA H20 GPU.** (a) Latency over all ten measured steps, including temporal-state initialization at step 1; (b) Reported parameter counts. The latency axis uses a piecewise-linear scale that expands 74–76 ms; the parameter axis is linear.

Inference latency. Over steps 2–10, their mean latencies are 70.29, 75.22, and 75.21 ms for the baseline, VQ + SQ, and TAGER, respectively. Thus, TAGER adds approximately 4.93 ms (7.0%) relative to the baseline over these subsequent steps.

Parameter overhead. The reported model sizes are 3.353B, 3.567B, and 3.569B parameters for the baseline, VQ + SQ, and TAGER, respectively. TAGER therefore increases the reported parameter count by approximately 6.4% over the baseline, while adding the Action Response Residual Flow to VQ + SQ accounts for approximately 2M parameters at the reported precision. The query branches account for most of the parameter increase, and the full model has a similar measured latency to VQ + SQ after the first step in this evaluation. These are descriptive measurements from one task, rather than estimates across repeated runs or benchmark suites.

J Limitations

Despite improved robustness, TAGER remains less reliable under large spatial shifts and tasks requiring category transfer or commonsense reasoning. Current-scene distillation and short execution histories may provide insufficient context for unfamiliar goals and extended recovery. The declines under background and lighting perturbations further indicate that action–response conditioning does not ensure appearance robustness. Future work will explore background and lighting augmentation

and more appearance-invariant temporal features, alongside broader training data, longer-term memory, and stronger task-level reasoning. Our real-world evaluation remains qualitative, motivating controlled studies across additional tasks, embodiments, and perturbations.