SEEING WHAT MATTERS FOR ACTION: BRIDGING VISUAL PRIORS AND ROBOTIC CONTROL IN VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained visual foundation models provide rich semantic and spatial priors for Vision-Language-Action (VLA) policies, and recent methods often transfer these priors by directly aligning external visual features with VLA representations. However, direct alignment leaves two gaps between generic perception and robotic control: it ***does not explicitly prioritize visual information relevant to action prediction***, and ***frame-wise supervision leaves cross-frame relational structure only indirectly constrained***. We propose ActSight, a lightweight framework for action-sensitive, temporally structured visual transfer in VLA policies. ActSight consists of two complementary components. Control-Oriented Visual Refinement weights external visual supervision by each tokens influence on the action objective, guiding lightweight residual refinement that preserves the VLA's learned representation while incorporating control-relevant visual knowledge. Temporal-Aware Representation Alignment complements frame-wise supervision by explicitly preserving relational structure across neighboring observations, encouraging the adapted representation to retain how visual states change during interaction. Across multiple manipulation tasks, ActSight improves control performance over direct feature alignment and visual-enhancement baselines. These results demonstrate the effectiveness of organizing visual transfer around action relevance and temporal relational structure.

1 INTRODUCTION

Vision-Language-Action (VLA) models transfer large-scale vision-language knowledge to robotic control (Kim et al., 2024; Li et al., 2024a; Black et al., 2024; Kim et al., 2025). Yet effective control requires more than perceptually informative features. For a fixed grasping task, a change in background appearance may leave the desired action unchanged, whereas a small change in the relative pose of the gripper and target requires a different correction. The challenge is therefore not simply to enrich a policy's visual representation, but to preserve distinctions that matter for action.

Recent studies show that visual adaptation can improve VLA robustness (Li et al., 2026c), while action-focused fine-tuning can weaken capabilities inherited from vision-language pretraining (Hancock et al., 2026). Feature alignment provides a way to introduce external visual priors by encouraging policy representations to match pretrained visual features, including spatial representations from foundation models (Li et al., 2026a; Wang et al., 2026a). Affordance-based transfer improves task relevance by constructing task-conditioned visual targets (Kong et al., 2026). These advances raise a question beyond the choice of visual teacher: *which distinctions in an external representation should be preserved for control, and how should those distinctions be maintained across interaction?*

Our starting point is that *not every visual distinction captured by an external representation matters equally for control*. Feature alignment typically encourages the policy to preserve the teacher's representation as a whole, even though some visual distinctions may have little influence on action prediction. Even when the teacher is task-conditioned, it does not indicate which of these distinctions actually influence the policy's predicted actions. We refer to this as an **action-relevance mismatch**: the structure rewarded by visual alignment is not explicitly prioritized according to its

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.

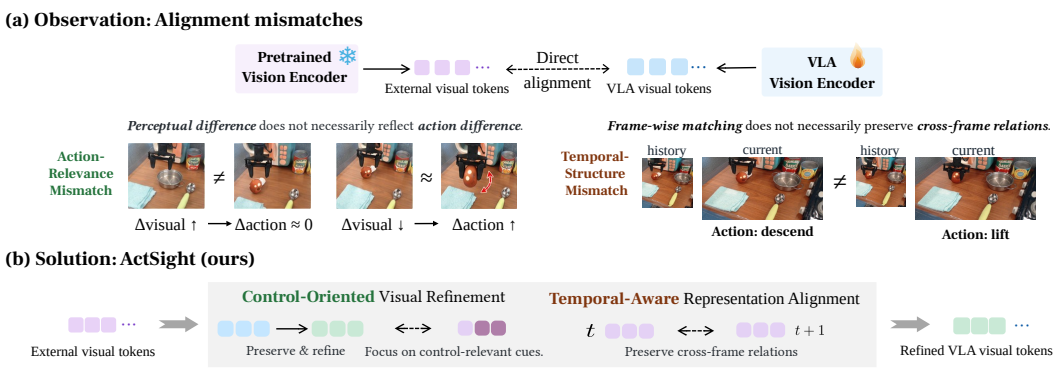


Figure 1: **Motivation of ActSight.** (a) Direct alignment can preserve perceptual distinctions that are *weakly related to control* (action-relevance mismatch) and match individual frames *without preserving their cross-frame relations* (temporal-structure mismatch). (b) ActSight addresses these mismatches by emphasizing control-relevant visual distinctions and preserving cross-frame relations.

influence on action prediction (Figure 1 (a)). This motivates using the action predictor itself to guide the emphasis placed on external visual features.

Visual transfer should preserve not only individual states, but also the relationships between them. For example, after a robot closes its gripper around an object, whether the object moves together with the gripper helps indicate whether the grasp succeeded. Such information lies in the change between observations rather than in either frame alone. Frame-wise alignment, however, matches each observation independently and does not explicitly preserve these cross-frame changes. Consequently, small per-frame alignment errors can distort the relative differences between nearby states. We refer to this as a **temporal-structure mismatch** (Figure 1 (a)). This motivates directly preserving cross-frame relations during visual transfer.

We introduce **ActSight**, a visual adaptation framework that treats visual transfer as *action-sensitive, temporally structured adaptation*. ActSight combines two objectives: using the policy's action sensitivity to prioritize external visual guidance, and preserving relational structure across neighboring observations (Figure 1 (b)). In this way, visual adaptation is guided not only by the external feature space, but also by the policy's control objective and the temporal structure of interaction. ***Control-Oriented Visual Refinement*** estimates the control relevance of visual features from their influence on the action objective, assigning stronger perceptual supervision to features that more strongly affect the objective. A lightweight residual pathway then refines the VLA's native visual representation under this selectively weighted external supervision, allowing external visual knowledge to complement rather than replace the policy's pretrained representation. ***Temporal-Aware Representation Alignment*** complements this refinement by preserving relational structure across neighboring interaction states. Rather than aligning each observation independently, it supervises how representations change across successive observations so that the cross-frame structure of the visual prior is retained. Together, the two components address *what to emphasize during transfer* and *how to organize the representation across interaction*.

Our contributions are threefold: (1) We formulate visual transfer for VLAs from a control-oriented perspective, highlighting action relevance and temporal structure as two complementary factors in visual adaptation. (2) We propose ActSight, which combines Control-Oriented Visual Refinement and Temporal-Aware Representation Alignment with a lightweight residual pathway for incorporating external visual guidance. (3) We demonstrate improvements across multiple manipulation tasks, showing the benefits of the proposed transfer strategy over direct feature alignment and the evaluated visual-enhancement baselines.

2 Related Work

Vision-Language-Action Models Vision-language-action (VLA) models transfer pretrained vision-language knowledge to robotic control through demonstrations, with recent advances substantially improving policy generality and action generation (Brohan et al., 2023; Kim et al., 2024; Li et al.,

2024a; Black et al., 2024; Kim et al., 2025). However, effective manipulation also depends critically on how visual information is represented and utilized by the policy. Recent studies reveal several limitations in current VLAs, including dispersed attention over visual observations, perceptual redundancy, insufficient grounding of task-relevant regions, and weakened utilization of visual information during action generation (Song et al., 2026; Li et al., 2026b; Xiao et al., 2026; Luo et al., 2026). Other work shows that targeted adaptation of visual representations can substantially improve policy robustness (Li et al., 2026c), while adapting pretrained VLMs to action prediction may degrade capabilities inherited from pretraining (Kachaev et al., 2025; Hancock et al., 2026). These findings further motivate improving visual representations in VLA policies.

Visual Representation Learning for Robot Control Recent work improves VLA perception through several forms of additional supervision. Some methods introduce visual or spatial guidance via prompts, keypoints, affordances, grounding, or adaptive attention (Liu et al., 2024; Huang et al., 2024; Yuan et al., 2024; Li et al., 2025a; Ye et al., 2026; Song et al., 2026; Li et al., 2026b; Xiao et al., 2026). Others use geometry-aware supervision or distillation (Li et al., 2025b; Wang et al., 2026b; Qian et al., 2026; Chen et al., 2026), or predictive objectives such as future visual-state prediction to capture scene dynamics (Ma et al., 2026; Xu et al., 2026). To transfer rich perceptual priors without increasing deployment burden, recent work has introduced supervision directly at the representation level, using pretrained visual or learned latent representations to constrain internal policy features (Li et al., 2026a; Kachaev et al., 2025; Lin et al., 2026; Wang et al., 2026a). However, such approaches largely treat external representations as uniform, frame-wise adaptation targets, leaving their relevance to control and their structure across interaction implicit. In contrast, we study how representation-level visual transfer itself should be structured for control, rather than treating the external target as a fixed perceptual constraint.

3 METHODOLOGY

Overview. We formulate robotic manipulation as a vision-language-conditioned control problem. Given an RGB observation I_t and a language instruction ℓ, a policy predicts an action chunk $\mathbf{A}_t = \pi_\theta(I_t, \ell)$. We build on a demonstration-trained VLA with Qwen-VL as the vision-language backbone, whose native vision encoder produces visual tokens $\mathbf{V}_t$. ActSight operates on these native visual tokens to obtain a refined representation $\widetilde{\mathbf{V}}_t$ for downstream action prediction, enabling efficient policy adaptation with limited training cost. As shown in Figure 2, the refinement is learned with two complementary forms of training-time supervision from an external visual encoder. ***Control-Oriented Visual Refinement*** (§ 3.1) uses a lightweight residual pathway to refine the VLA's native visual representation under external perceptual supervision, reweighting this supervision according to action sensitivity, while ***Temporal-Aware Representation Alignment*** (§ 3.2) constrains cross-frame relational structure across neighboring observations. The resulting training objective combines the original action loss with these two auxiliary objectives. The external encoder and auxiliary supervision are discarded after training, so deployment adds only the lightweight residual adapter to the original VLA.

3.1 CONTROL-ORIENTED VISUAL REFINEMENT

Because pretrained vision models and VLA policies emphasize perception and action prediction differently, direct visual alignment could over-preserve perceptual distinctions with limited influence on control, leading to an **action-relevance mismatch**. To mitigate this mismatch, ActSight refines the native VLA representation through ***residual adaptation*** and aligns external visual priors with control-relevant information via ***control-relevant alignment***.

3.1.1 RESIDUAL ADAPTATION

Residual Parameterization. We preserve the VLA's native visual representation and refine it through a lightweight residual pathway under external visual supervision. Let $\mathbf{V}_t = E_{\theta_v}(I_t)$ denote the native visual tokens produced by the VLA vision encoder from observation I_t. The residual function R_ϕ predicts a token-wise residual $\Delta\mathbf{V}_t$, which is added to the native representation: $\Delta\mathbf{V}_t = R_\phi(\mathrm{sg}(\mathbf{V}_t))$, $\widetilde{\mathbf{V}}_t = \mathbf{V}_t + \Delta\mathbf{V}_t$, where $\mathrm{sg}(\cdot)$ denotes stop-gradient. We implement R_ϕ as a two-layer MLP with LayerNorm and GELU, and zero-initialize its output layer such that $\Delta\mathbf{V}_t = 0$

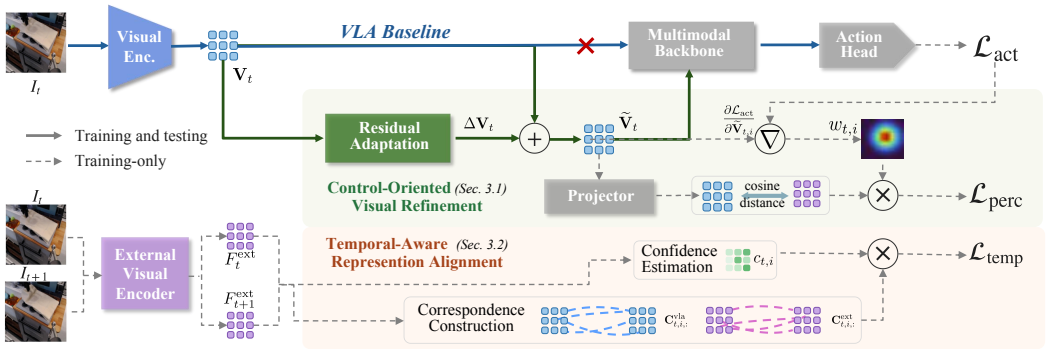


Figure 2: **Overview of ActSight.** Given the native visual tokens $\mathbf{V}_t$ produced by the VLA visual encoder, ActSight learns a lightweight residual adaptation $\Delta\mathbf{V}_t$ and forms refined visual tokens $\widetilde{\mathbf{V}}_t$ for downstream action prediction. Control-Oriented Visual Refinement aligns refined tokens with external perceptual priors, weighted by action sensitivity. Temporal-Aware Representation Alignment preserves reliable cross-frame correspondences through confidence-aware supervision. At inference, the external visual encoder and auxiliary supervision branches are removed, and only the residual refinement pathway is retained alongside the original VLA.

at initialization. The model therefore starts from the unmodified VLA computation and gradually learns residual refinements during training. The refined tokens $\widetilde{\mathbf{V}}_t$ are then passed to the multimodal backbone for action prediction, while the rest of the VLA architecture remains unchanged.

Gradient Isolation. To preserve action-adapted features during residual adaptation, we separate the gradient paths of action and perceptual supervision. Applying perceptual supervision directly to $\widetilde{\mathbf{V}}_t$ would allow gradients to update the native VLA representation through the identity path. We therefore construct a gradient-isolated representation: $\mathbf{S}_t = \mathrm{sg}(\mathbf{V}_t) + \Delta\mathbf{V}_t$. The action objective is computed using $\widetilde{\mathbf{V}}_t$, allowing both the native VLA pathway and residual adapter to be optimized. In contrast, perceptual objectives are computed on $\mathbf{S}_t$, restricting gradients to the residual adapter. Although $\widetilde{\mathbf{V}}_t$ and $\mathbf{S}_t$ are identical in the forward pass, they induce different gradient flows.

3.1.2 Control-Relevant Alignment

External vision models provide perceptual priors, but their relevance to action prediction varies across visual tokens. ActSight introduces control-relevant supervision through three steps: extracting perceptual guidance from pretrained vision models, estimating token-level control relevance using policy action sensitivity, and optimizing perceptual supervision according to this relevance.

Perceptual Alignment. We obtain external perceptual targets $\mathbf{T}_t$ from frozen pretrained visual encoders, resizing their patch-level feature maps to match the spatial resolution of the VLA visual tokens. Since $\mathbf{T}_t$ and $\mathbf{S}_t$ lie in different feature spaces, a linear projector P, shared across spatial tokens, maps $\mathbf{S}_t$ into the external feature space. For token i, we define the perceptual discrepancy as

$$\ell_{t,i} = 1 - \cos\left(P(\mathbf{S}_{t,i}), \mathbf{T}_{t,i}\right). \tag{1}$$

Applying supervision to $\mathbf{S}_t$ encourages the residual to enhance the native VLA representation while preserving its action-oriented structure, rather than forcing the residual to reproduce external visual features independently. In our implementation, we instantiate the perceptual guidance using DINOv3 (Siméoni et al., 2025) and the visual encoder of SigLIP 2 (Tschannen et al., 2025). DINOv3 provides dense visual features with fine-grained local structures, while SigLIP 2 provides representations learned from large-scale image–text pretraining.

Action-Sensitive Weighting. A large perceptual discrepancy does not necessarily imply high relevance to action prediction. We therefore use the policy's action sensitivity to estimate token-level control relevance and weight the perceptual supervision accordingly. Tokens with greater influence on the action objective receive stronger supervision, prioritizing visual distinctions that matter more for control. For token i at timestep t, we compute the gradient of the action objective with respect to

the refined visual representation: $\mathbf{g}_{t,i} = \frac{\partial \mathcal{L}_{\text{act}}}{\partial \tilde{\mathbf{V}}_{t,i}}$. We aggregate the gradient magnitude across feature dimensions to obtain an action-sensitivity score: $r_{t,i} = \|\text{sg}(\mathbf{g}_{t,i})\|_2$. A larger $r_{t,i}$ indicates greater local sensitivity of the action loss to token i, suggesting that the corresponding visual information is more relevant to the current control decision. Since the sensitivity is derived from the current policy and its action objective, it provides a dynamic estimate of token-level control relevance. To reduce the influence of different gradient scales across observations, we normalize the sensitivity score relative to the mean sensitivity within each observation: $q_{t,i} = \frac{r_{t,i}}{r_{t,i} + \frac{1}{N}\sum_j r_{t,j} + \epsilon}$, where ϵ is a small constant for numerical stability. We then obtain the token-level supervision weights by centering the normalized scores: $w_{t,i} = \text{sg}\left[1 + \alpha\left(q_{t,i} - \frac{1}{N}\sum_j q_{t,j}\right)\right]$. The centering operation keeps the average weight equal to one within each observation, allowing perceptual supervision to be redistributed toward more action-sensitive tokens while maintaining the overall supervision scale. Using the action-sensitive weights, we formulate the objective as:

$$\mathcal{L}_{\text{perc}} = \mathbb{E}_{t,i}\left[w_{t,i}\ell_{t,i}\right]. \tag{2}$$

This objective selectively allocates perceptual supervision according to control relevance, allowing useful visual priors to be transferred while prioritizing action-relevant information.

3.2 Temporal-Aware Representation Alignment

Frame-wise visual alignment does not explicitly preserve relational structure across neighboring interaction states, resulting in the **temporal-structure mismatch**. We address this with ***Temporal-Aware Representation Alignment***, which uses confidence-weighted cross-frame correspondences to preserve this structure in the refined VLA representation.

Temporal Correspondence Construction. To capture how visual states evolve during interaction, we construct cross-frame correspondence structures between consecutive observations. For each observation I_t, we pair it with the observation I_{t+1}. Let $\mathbf{S}_t$ and $\mathbf{S}_{t+1}$ denote the refined VLA representations. We detach the future representation: $\bar{\mathbf{S}}_{t+1} = \text{sg}(\mathbf{S}_{t+1})$, so that the future observation serves only as a temporal reference. Given two token sets $\mathbf{X}$ and $\mathbf{Y}$, we define a soft cross-frame correspondence: $\mathcal{C}_{ij}(\mathbf{X}, \mathbf{Y}) = \text{softmax}_j\left(\frac{\cos(\mathbf{X}_i, \mathbf{Y}_j)}{\tau}\right)$, where τ is a temperature parameter. Using this correspondence function, we construct forward and backward correspondence distributions for the refined VLA representation and the frozen external visual features: $\mathbf{C}_t^{\text{vla},\rightarrow} = \mathcal{C}(\mathbf{S}_t, \bar{\mathbf{S}}_{t+1})$, $\mathbf{C}_t^{\text{vla},\leftarrow} = \mathcal{C}(\bar{\mathbf{S}}_{t+1}, \mathbf{S}_t)$, $\mathbf{C}_t^{\text{ext},\rightarrow} = \mathcal{C}(\mathbf{F}_t^{\text{ext}}, \mathbf{F}_{t+1}^{\text{ext}})$, $\mathbf{C}_t^{\text{ext},\leftarrow} = \mathcal{C}(\mathbf{F}_{t+1}^{\text{ext}}, \mathbf{F}_t^{\text{ext}})$, where $\mathbf{F}_t^{\text{ext}}$ denotes frozen external visual features used for temporal correspondence supervision. The subscript t indexes the frame pair $(t, t+1)$ in both directions. Each distribution describes how tokens in one observation are associated with tokens in the neighboring observation.

Confidence Estimation. Temporal correspondences obtained from external visual features may contain unreliable matches due to object motion, occlusion, or visual ambiguity. We therefore estimate correspondence confidence using forward-backward cycle consistency. For token i in the current observation, the forward confidence is defined as: $c_{t,i}^{\rightarrow} = \sum_j C_{t,ij}^{\text{ext},\rightarrow} C_{t,ji}^{\text{ext},\leftarrow}$. A larger $c_{t,i}^{\rightarrow}$ indicates that the correspondence from the current observation to the neighboring observation can be consistently recovered through the reverse mapping, suggesting a more reliable temporal relation. The backward confidence $c_{t,i}^{\leftarrow}$ is computed analogously.

Selective Alignment. We align the temporal correspondence structure of the refined VLA representation with that of the external vision model using confidence-weighted KL divergence:

$$\mathcal{L}_{\text{temp}} = \mathbb{E}_t\left[\frac{1}{2}\sum_{r\in\{\rightarrow,\leftarrow\}} \frac{\sum_i c_{t,i}^r D_{\text{KL}}\left(\mathbf{C}_{t,i,:}^{\text{ext},r} \| \mathbf{C}_{t,i,:}^{\text{vla},r}\right)}{\sum_i c_{t,i}^r + \epsilon}\right], \tag{3}$$

where the expectation is over sampled adjacent frame pairs, r denotes the correspondence direction, and $c_{t,i}^r$ is the confidence of source token i in direction r for pair $(t, t+1)$. The KL divergence aligns the refined VLA representation with the cross-frame correspondence distributions of an external vision model, while confidence weighting downweights ambiguous matches. We use frozen DINOv3 features to construct these temporal correspondences, leveraging their dense patch-level representations for fine-grained spatial matching.

Table 1: Success rates (%) on **SimplerEnv-WidowX**. **Bold**/underline indicate the best/second-best results in each column. Green values in parentheses denote percentage-point gains over the same-backbone Baseline.

METHOD	BASE VLM	CARROT ON PLATE	STACK CUBE	SPOON ON TOWEL	EGGPLANT IN BASKET	AVG.
π_0 (Black et al., 2024)	PaliGemma (3B)	0.0	16.7	29.1	62.5	27.1
OpenVLA-OFT (Kim et al., 2025)	Prismatic (7B)	4.2	8.3	12.5	37.5	15.6
Baseline	Qwen-VL (0.8B)	16.7	0.0	62.5	83.3	40.6
Baseline	Qwen-VL (2B)	16.7	0.0	75.0	87.5	44.8
QDepth-VLA (Li et al., 2025b)	PaliGemma (3B)	57.5	39.6	82.0	95.0	68.5
VITA (Ma et al., 2026)	PaliGemma (3B)	68.8	37.5	84.2	95.6	71.5
ST4VLA (Ye et al., 2026)	Qwen-VL (3B)	**79.2**	35.4	80.2	98.0	73.2
LA-Align (Lin et al., 2026)	Qwen-VL (0.8B)	25.0	37.5	79.2	**100.0**	60.4
FeatAlign (Li et al., 2026a; Kachaev et al., 2025)	Qwen-VL (0.8B)	45.8	33.3	**91.7**	83.3	63.5
RelAlign (Zhang et al., 2026; Bhowmik et al., 2026)	Qwen-VL (0.8B)	29.2	54.2	75.0	**100.0**	64.6
ActSight	Qwen-VL (0.8B)	45.8 (↑ 29.2)	**66.7** (↑ 66.7)	79.2 (↑ 16.7)	**100.0** (↑ 16.7)	72.9 (↑ 32.3)
LA-Align (Lin et al., 2026)	Qwen-VL (2B)	25.0	41.7	83.3	95.8	61.5
FeatAlign (Li et al., 2026a; Kachaev et al., 2025)	Qwen-VL (2B)	37.5	37.5	87.5	95.8	64.6
RelAlign (Zhang et al., 2026; Bhowmik et al., 2026)	Qwen-VL (2B)	37.5	50.0	87.5	95.8	67.7
ActSight	Qwen-VL (2B)	41.7 (↑ 25.0)	62.5 (↑ 62.5)	**91.7** (↑ 16.7)	**100.0** (↑ 12.5)	**74.0** (↑ 29.2)

4 EXPERIMENTS

We organize our experiments into three parts: (1) simulation experiments comparing ActSight with baseline methods (§ 4.1); (2) ablation studies and analysis of the proposed design (§ 4.2); and (3) real-world experiments evaluating robotic manipulation performance (§ 4.3).

4.1 SIMULATION EXPERIMENTS

Tasks and Benchmarks. We evaluate ActSight on: (1) LIBERO (Liu et al., 2023), which includes diverse manipulation tasks spanning spatial, object, goal, and long-horizon variations; and (2) the SimplerEnv-WidowX setup (Li et al., 2024b), where policies trained on the real-world BridgeData V2 (Walke et al., 2023) dataset are evaluated on corresponding manipulation tasks in simulation. Additional details on the task settings are provided in Appendix B, while implementation details and hyperparameters for ActSight on both benchmarks are provided in Appendix C.

Baselines. We compare ActSight with three baseline categories. Implementation details of all baselines are provided in Appendix A.

(1) ***General VLA Policies.*** We compare with π_0 (Black et al., 2024) and *OpenVLA-OFT* (Kim et al., 2025). We also include our Qwen-VL-based *Baseline*, which serves as the base policy and initialization for ActSight. It uses a continuous action head trained with L1/L2 losses and excludes diffusion- or flow-based action generation, providing a controlled reference for isolating the effect of visual representation refinement.

(2) ***Structured Perceptual Supervision.*** *QDepth-VLA* (Li et al., 2025b) introduces depth supervision to enhance geometric reasoning, *VITA* (Ma et al., 2026) incorporates visual-state dynamics into policy learning, and *ST4VLA* (Ye et al., 2026) uses spatial grounding to strengthen spatial understanding for manipulation.

(3) ***External Representation Alignment.*** *LA-Align* (Lin et al., 2026) improves policy learning by aligning latent-action representations. We additionally construct two alignment baselines under matched training conditions: *FeatAlign*, following feature-level alignment methods (Yu et al., 2025; Li et al., 2026a; Kachaev et al., 2025), and *RelAlign*, instantiating relation matching (Zhang et al., 2026; Bhowmik et al., 2026) to align intra-frame spatial and cross-frame temporal token relations.

Results. *SimplerEnv-WidowX.* As shown in Table 1, ActSight consistently improves over the general VLA and performs particularly well on the more challenging Stack Cube task. Feature-alignment methods such as FeatAlign, LA-Align, and RelAlign also benefit from external visual supervision, showing that pretrained visual knowledge is useful for control. Their gains, however, are less pronounced on stacking, where the policy must reason about fine-grained object relations

Table 2: Success rates (%) on the **LIBERO** benchmark. **Bold**/underline indicate the best/second-best results in each column. Green values in parentheses denote percentage-point gains over the same-backbone Baseline.

METHOD	BASE VLM	SPATIAL	OBJECT	GOAL	LONG	AVG.
π_0 (Black et al., 2024)	PaliGemma (3B)	96.8	98.8	95.8	85.2	94.2
OpenVLA-OFT (Kim et al., 2025)	Prismatic (7B)	97.6	98.4	97.9	94.5	97.1
Baseline	Qwen-VL (0.8B)	93.8	93.8	94.2	78.6	90.1
Baseline	Qwen-VL (2B)	96.6	97.2	92.8	85.8	93.1
QDepth-VLA (Li et al., 2025b)	PaliGemma (3B)	97.6	96.6	95.2	90.0	94.9
VITA (Ma et al., 2026)	PaliGemma (3B)	95.9	98.9	95.1	**96.8**	96.7
ST4VLA (Ye et al., 2026)	Qwen-VL (3B)	98.0	99.0	93.8	92.6	95.9
LA-Align (Lin et al., 2026)	Qwen-VL (0.8B)	93.2	**99.8**	96.2	92.2	95.4
FeatAlign (Li et al., 2026a; Kachaev et al., 2025)	Qwen-VL (0.8B)	93.6	99.0	96.0	88.2	94.2
RelAlign (Zhang et al., 2026; Bhowmik et al., 2026)	Qwen-VL (0.8B)	92.6	98.8	97.8	89.4	94.7
ActSight	Qwen-VL (0.8B)	94.4 (↑ 0.6)	99.0 (↑ 5.2)	97.4 (↑ 3.2)	95.0 (↑ 16.4)	96.5 (↑ 6.4)
LA-Align (Lin et al., 2026)	Qwen-VL (2B)	97.4	98.6	97.2	94.8	97.0
FeatAlign (Li et al., 2026a; Kachaev et al., 2025)	Qwen-VL (2B)	97.4	97.8	**98.2**	90.6	96.0
RelAlign (Zhang et al., 2026; Bhowmik et al., 2026)	Qwen-VL (2B)	97.4	99.0	97.8	94.0	97.1
ActSight	Qwen-VL (2B)	**98.8** (↑ 2.2)	99.2 (↑ 2.0)	**98.2** (↑ 5.4)	95.0 (↑ 9.2)	**97.8** (↑ 4.7)

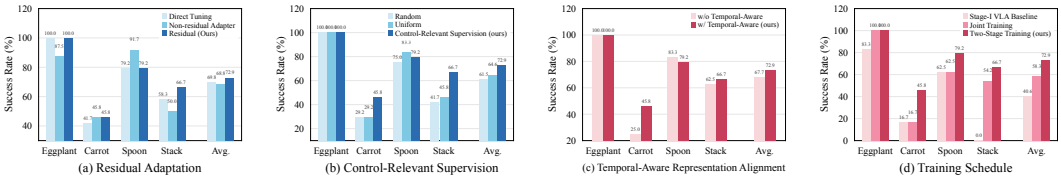
and continuously adjust to changes during interaction. This highlights the main difference of ActSight: rather than directly aligning the VLA representation with an external feature space, it adapts external perceptual knowledge to the control-specific representation already learned by the policy. ST4VLA, which explicitly strengthens spatial reasoning, remains highly competitive on tasks dominated by geometric understanding. In contrast, ActSight shows a clearer advantage when successful execution additionally depends on action-relevant and evolving visual information.

LIBERO. The results in Table 2 show a similar trend. ActSight remains strong across Spatial, Object, Goal, and Long, while its advantage is most evident on LIBERO-Long. Alignment-based methods improve the visual representation, but mainly supervise how external perceptual structure should be transferred into the policy. ActSight further considers whether the transferred information is relevant to control and whether it preserves changes across time, which becomes increasingly important in multi-stage manipulation. Methods such as OpenVLA-OFT and VITA improve long-horizon control from complementary directions, focusing more on action generation and trajectory-level modeling, respectively. ActSight instead targets the interface between perception and control. Its consistent performance across task suites suggests that adapting visual priors to control is complementary to strengthening either perception or action modeling alone.

4.2 ABLATIONS AND ANALYSIS

In this section, we analyze the effectiveness and underlying behavior of ActSight's core components and examine the impact of its training schedule. We organize these analyses around the following four research questions. Efficiency and cost are discussed in Appendix B.3.1.

- Does Residual Adaptation better preserve the VLA's action-adapted representation? (Figure 3 (a), Figure 4, and Figure 5)
- Does Control-Relevant Supervision mitigate action-relevance mismatch by prioritizing control-relevant visual guidance? (Figure 3 (b), Figure 6)
- Does Temporal-Aware Representation Alignment mitigate temporal-structure mismatch? (Figure 3 (c), Figure 7)



(a) Residual Adaptation (b) Control-Relevant Supervision (c) Temporal-Aware Representation Alignment (d) Training Schedule

Figure 3: (a) Residual Adaptation outperforms direct tuning. (b) Control-relevant weighting outperforms uniform and random weighting. (c) Temporal alignment further improves control performance. (d) ActSight benefits more from prior action-supervised training.

- When should ActSight be introduced during policy learning? (Figure 3 (d))

Residual Adaptation Preserves Action-Adapted Representations. As shown in Figure 4, the original VLA and external visual models exhibit noticeably different spatial response patterns under the same observations. The VLA focuses more strongly on regions related to action execution, whereas external visual models respond to a broader range of information. ***This suggests that action supervision has shaped the VLA toward a representation distinct from the external perceptual prior.*** We then examine whether different adaptation strategies preserve this native representation. As shown in Figure 5, Residual Adaptation maintains higher similarity to the original VLA in deeper visual blocks, while both methods remain close to the original representation in earlier blocks. The token-level analysis in Figure 4 shows the same trend: Residual Adaptation preserves higher token-wise similarity than Direct Tuning. ***These results show that Residual Adaptation better retains the VLA's action-adapted representation.*** This representation preservation is also reflected in control performance. Figure 3(a) compares Residual Adaptation with Direct Tuning and a parameter-matched non-residual adapter. Residual Adaptation achieves the highest average success rate, whereas both alternatives perform worse. The non-residual comparison further shows that the improvement cannot be explained simply by additional trainable capacity.

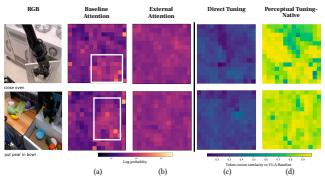


Figure 4: Spatial responses and token-level preservation of the action-adapted VLA representation.

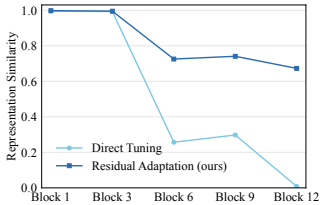


Figure 5: Layer-wise preservation of the original VLA representation after adaptation.

Control-Relevant Supervision Mitigates Action-Relevance Mismatch. To examine the action-relevance mismatch, we compare perceptual discrepancy with the policy's action sensitivity. As shown in Figure 6(a,b), perceptual discrepancy is distributed broadly, whereas action sensitivity concentrates more strongly on tokens that have a larger influence on the action objective. ***This illustrates the central mismatch: visual distinctions emphasized by external perceptual supervision are not necessarily equally relevant to control.*** Figure 6(c) shows how action-sensitive weighting redistributes perceptual supervision toward tokens with greater control relevance. Finally, Figure 3(b) compares Uniform, Random, and Control-Relevant Supervision. Control-Relevant Supervision achieves the best performance. Compared with Random, which preserves the same weight distribution but permutes its spatial assignment, Control-Relevant Supervision performs better.

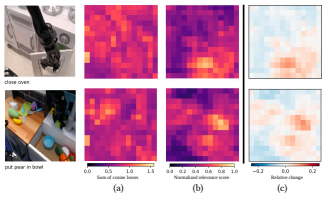


Figure 6: Control-relevant supervision redirects perceptual learning toward action-relevant tokens.

Temporal-Aware Representation Alignment Mitigates the Temporal-Structure Mismatch. Figure 3 (c) compares ActSight with and without Temporal-Aware Representation Alignment. Adding temporal alignment improves overall control performance, suggesting that preserving cross-frame structure provides benefits beyond frame-wise visual refinement. To further examine whether frame-wise refinement disrupts such structure, we evaluate cross-frame correspondence using the gripper as a trackable region. The gripper can undergo substantial spatial displacement across neighboring frames while retaining the same physical identity, making it a natural probe of temporal correspondence. If cross-frame structure is preserved, tokens associated with the gripper in the current frame should remain strongly associated with the gripper at its future location. For frame pairs with substantial gripper motion, we manually annotate the gripper regions in the current and future frames and report three metrics in Figure 7. Region Lift measures how strongly correspondence is concentrated within the future gripper region relative to its spatial size; Top-k Region IoU measures the overlap between the strongest correspondence locations and the future gripper region; and Top-1 Hit Rate measures whether the strongest correspondence falls inside that region. Without Temporal-Aware Representation Alignment, all three metrics decrease relative to the native VLA, indicating that

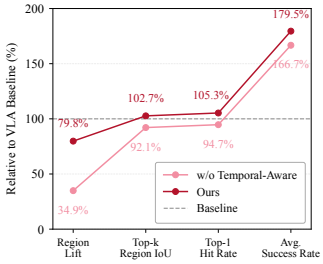


Figure 7: Temporal-aware alignment better preserves action-relevant correspondence across frames.

frame-wise visual refinement can weaken cross-frame correspondence. Adding temporal alignment recovers these metrics toward the native VLA while retaining the control gains from visual refinement. These results support the temporal-structure mismatch identified above: ***frame-wise adaptation can disrupt relations across neighboring states, whereas Temporal-Aware Representation Alignment better preserves such structure while improving control performance***.

ActSight Benefits More from Prior Action-Supervised Training. We compare applying ActSight from the start of policy learning with adding it after an initial stage of action-supervised training. As shown in Figure 3(d), the latter yields stronger control performance, while training with ActSight from the beginning provides only modest gains. ***This suggests that ActSight is most effective after the VLA has learned meaningful control-oriented representations.*** This behavior arises because ActSight relies on policy-derived signals. In particular, Control-Relevant Supervision uses action sensitivity to weight external perceptual guidance. Early in training, these signals are not yet well established and provide limited guidance. After action-supervised learning forms a stronger control representation, ActSight can more effectively refine it with external visual priors.

4.3 Real-World Evaluation

We evaluate ActSight on a Franka Research 3 robot across six tasks spanning pick-and-place, deformable-object manipulation, and cup nesting, with 50 demonstrations per task. The tasks include placing a banana, starfruit, or snack package onto a plate, folding a towel, and two cup-nesting tasks: placing one cup into another and sequentially placing two cups into a target cup. The towel-folding and cup-nesting tasks involve substantial interaction-induced state changes and require repeated perception and action adjustment as the scene evolves. Representative rollouts for three tasks are shown in Figure 8 (bottom). We further evaluate scene-level OOD robustness on the three pick-and-place tasks by changing the tabletop background from black to white, introducing unseen task-irrelevant distractors, and modifying the spatial arrangement of objects. As shown in Figure 8 (top), ActSight improves average performance across the six in-domain tasks, with pronounced gains on tasks involving evolving object configurations and repeated adjustment, such as towel folding and sequential cup nesting. Under scene-level OOD variations, ActSight also achieves higher average success on the pick-and-place tasks, demonstrating improved robustness to changes in visual appearance and scene composition.

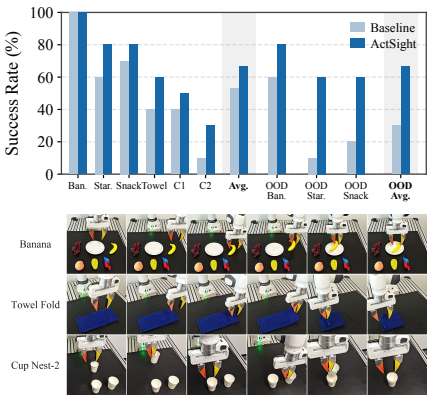


Figure 8: Top: success rates (%). Bottom: qualitative results for three real-world tasks. Additional results and details are provided in Appendix D.

5 Conclusion

We present ActSight, a framework for adapting external visual priors to VLA policies. ActSight prioritizes visual guidance using local, policy-dependent action sensitivity and supervises relations between neighboring observations, with a residual pathway integrating the learned adaptation into pretrained policy features. Across the evaluated simulation and real-robot settings, ActSight improves over native VLA policies and the evaluated visual-enhancement baselines, including direct feature alignment. These findings suggest that effective visual transfer for control benefits not merely from richer representations, but from guidance tailored to the policy's action predictions and explicit supervision of local temporal relations.

Limitations. ActSight currently models temporal structure only between neighboring observations and does not explicitly capture longer-range dependencies across an interaction sequence. In addition, external visual encoders and action-sensitivity estimation introduce additional computation during training, although they are not required at deployment. Our evaluation is also limited to a set of VLA backbones and manipulation settings. Extending relational supervision to longer temporal contexts, reducing adaptation-time overhead, and evaluating across broader architectures, embodiments, and tasks are promising directions for future work.

AI USE STATEMENT

We used generative AI tools, including OpenAI Codex, to assist with drafting and polishing parts of the manuscript, particularly in improving the presentation and clarity of the method. The authors are fully responsible for the final content of the paper, including all AI-assisted text, claims, references, and artifacts.

REFERENCES

Aritra Bhowmik, Denis Korzhenkov, Cees G Snoek, Amirhossein Habibian, and Mohsen Ghafoorian. Moalign: Motion-centric representation alignment for video diffusion models. In *International Conference on Learning Representations*, volume 2026, pp. 122379–122397, 2026.

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.

Yizhi Chen, Zhanxiang Cao, Xinyi Peng, Yixiao Zheng, Xiaxi Si, Yiheng Li, Liyun Yan, Keqi Zhu, Xueyun Chen, Shengcheng Fu, et al. Geoalign: Beyond semantics with state-guided spatial alignment in vla models. *arXiv preprint arXiv:2606.03240*, 2026.

Asher Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as language: Fine-tuning vlms into vlas without catastrophic forgetting. In *International Conference on Learning Representations*, volume 2026, pp. 75333–75360, 2026.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024.

Nikita Kachaev, Mikhail Kolosov, Daniil Zelezetsky, Alexey K Kovalev, and Aleksandr I Panov. Don't blind your vla: Aligning visual representations for ood generalization. *arXiv preprint arXiv:2510.25616*, 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.

Weijie Kong, Zhian Su, Wei Yu, and Huixu Dong. Affordvla: Injecting affordance representations into vision-language-action models via implicit feature alignment, 2026. URL https://arxiv.org/abs/2605.17517.

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.

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

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.

Wei Li, Renshan Zhang, Rui Shao, Zhijian Fang, Kaiwen Zhou, Zhuotao Tian, and Liqiang Nie. Semanticvla: Semantic-aligned sparsification and enhancement for efficient robotic manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18397–18405, 2026b.

Weiqi Li, Quande Zhang, Ruifeng Zhai, Liang Lin, and Guangrun Wang. Vla models are more generalizable than you think: Revisiting physical and spatial modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35025–35035, 2026c.

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.

Yixuan Li, Yuhui Chen, Mingcai Zhou, Haoran Li, Zhengtao Zhang, and Dongbin Zhao. Qdepth-vla: Quantized depth prediction as auxiliary supervision for vision-language-action models. *arXiv preprint arXiv:2510.14836*, 2025b.

Yihan Lin, Haoyang Li, Yang Li, Haitao Shen, Yihan Zhao, Chao Shao, and Jing Zhang. From pixels to tokens: A systematic study of latent action supervision for vision-language-action models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=VMsumctGvg.

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.

Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-world robotic manipulation through mark-based visual prompting. *arXiv preprint arXiv:2403.03174*, 2024.

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*, 2026.

Xiangkai Ma, Lekai Xing, Han Zhang, Wenzhong Li, and Sanglu Lu. Unifying perception and action: A hybrid-modality pipeline with implicit visual chain-of-thought for robotic action generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22380–22390, 2026.

Jingjing Qian, Boyao Han, Chen Shi, Lei Xiao, Long Yang, Shaoshuai Shi, and Li Jiang. Geopredict: Leveraging predictive kinematics and 3d gaussian geometry for precise vla manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13529–13539, 2026.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

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.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on robot learning*, pp. 1723–1736. PMLR, 2023.

Hao Wang, Xiaobao Wei, Jingyang He, Chengyu Bai, Chun-Kai Fan, Jiajun Cao, Jintao Chen, Ying Li, Shanyu Rong, Ming Lu, et al. Vega: Visual encoder grounding alignment for spatially-aware vision-language-action models. *arXiv preprint arXiv:2605.10485*, 2026a.

Ziru Wang, Mengmeng Wang, Guang Dai, Yongliu Long, and Jingdong Wang. Monolift: Learning 3d manipulation policies from monocular rgb via distillation. *Advances in Neural Information Processing Systems*, 38:5100–5120, 2026b.

Lei Xiao, Jifeng Li, Juntao Gao, Feiyang Ye, Yan Jin, Jingjing Qian, Jing Zhang, Yong Wu, and Xiaoyuan Yu. Ava-vla: improving vision-language-action models with active visual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13453–13463, 2026.

Xiaoxu Xu, Hao Li, Jinhui Ye, Yilun Chen, Jia Zeng, Xinyi Chen, Linning Xu, Dahua Lin, Weixin Li, and Jiangmiao Pang. Futurevla: Joint visuomotor prediction for vision-language-action model. *arXiv preprint arXiv:2603.10712*, 2026.

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*, 2026.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=DJSZGGZYVi.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024.

Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. Videorepa: Learning physics for video generation through relational alignment with foundation models. *Advances in Neural Information Processing Systems*, 38:122647–122676, 2026.

A Detailed Baselines

A.1 Baseline Categories and Comparison Settings

We categorize the baselines evaluated on LIBERO and SimplerEnv into three groups according to how perceptual information is incorporated into policy learning.

General VLA Policies. We include π_0 (Black et al., 2024) and OpenVLA-OFT (Kim et al., 2025) as representative VLA baselines on LIBERO and SimplerEnv. These methods primarily optimize action prediction without introducing dedicated perceptual supervision from external visual representations. We additionally construct a Qwen-VL-based *Baseline* equipped with a continuous action head trained using L1/L2 regression losses. This baseline excludes diffusion- or flow-based action generation, providing a controlled reference for isolating the effect of visual representation refinement. It also serves as the common vanilla-VLA starting point for our controlled comparisons below.

Representation Alignment. This category includes LA-Align from P2T (Lin et al., 2026), together with our controlled FeatAlign and RelAlign baselines. LA-Align aligns internal VLM representations with learned latent-action representations, providing a reference for representation transfer through latent-action supervision. FeatAlign and RelAlign are controlled adaptations built on the same VLA backbones as ActSight. FeatAlign follows the feature-level alignment paradigm explored in REPA (Yu et al., 2025), Spatial Forcing (Li et al., 2026a), and Don't Blind Your VLA (Kachaev et al., 2025), directly encouraging VLA representations to match features extracted by frozen visual teachers. RelAlign instead follows the relation-matching principle of VideoREPA (Zhang et al., 2026) and MoAlign (Bhowmik et al., 2026), matching both intra-frame spatial relations and cross-frame temporal relations among visual tokens. For each model scale, FeatAlign, RelAlign, and ActSight share the same VLA initialization, policy backbone, robot training data, action supervision, and optimization setup. Baseline uses action supervision only, whereas FeatAlign, RelAlign, and ActSight introduce different perceptual learning strategies from the same vanilla-VLA starting point. This design enables controlled comparisons of alternative representation-learning strategies under otherwise matched training conditions.

Auxiliary Perceptual Supervision. We further compare against QDepth-VLA (Li et al., 2025b), VITA (Ma et al., 2026), and ST4VLA (Ye et al., 2026). These methods augment action learning with explicit perceptual objectives based on depth prediction, visual-state dynamics, and spatial grounding, respectively. Together, they provide complementary references for distinguishing standard action learning, representation alignment, and auxiliary perceptual supervision from ActSight's control-oriented perceptual adaptation.

A.2 Construction of Alignment-Based Baselines

We construct FeatAlign and RelAlign as controlled alignment baselines to study how external perceptual priors are transferred into a VLA policy. For each model scale, both baselines use exactly the same VLA backbone and initialization as ActSight, together with the same robot demonstrations, action supervision, optimization setup, and external perceptual guidance. The only change is how the external perceptual information supervises the policy representation. These baselines should therefore be viewed as controlled instantiations of feature- and relation-alignment strategies. Both baselines preserve the original VLA deployment pathway. The external perceptual feature pipeline and all auxiliary supervision branches are used only during training and are removed at inference.

Let I_t denote the current RGB observation, ℓ the language instruction, and

$$\mathbf{V}_t = E_{\theta_v}(I_t) \in \mathbb{R}^{N \times d_v} \tag{4}$$

the native visual tokens produced by the VLA vision encoder. The action objective $\mathcal{L}_{\text{act}}$ and action chunk $\mathbf{A}_t$ follow the definitions in the main text. We denote by

$$\mathbf{T}_t \in \mathbb{R}^{N \times d_e} \tag{5}$$

the frozen external perceptual target obtained after feature extraction and spatial resizing to the VLA token grid. The target representation is detached from optimization. Spatial resampling is

performed before normalization or similarity computation, and similarity matrices themselves are never interpolated.

FeatAlign. FeatAlign instantiates a feature-level representation-alignment strategy inspired by REPA (Yu et al., 2025), Spatial Forcing (Li et al., 2026a), and Don't Blind Your VLA (Kachaev et al., 2025). It tests whether directly aligning the native VLA representation with external perceptual features is sufficient to improve policy learning.

Because the native VLA tokens and external perceptual targets may have different feature dimensions, we introduce a lightweight linear projector

$$P_\psi : \mathbb{R}^{d_v} \to \mathbb{R}^{d_e}, \tag{6}$$

shared across spatial tokens. FeatAlign minimizes the token-wise cosine discrepancy

$$\mathcal{L}_{\text{Feat}} = \frac{1}{N} \sum_{i=1}^{N} \left[1 - \cos\left(P_\psi(\mathbf{V}_{t,i}), \mathbf{T}_{t,i}\right)\right]. \tag{7}$$

The full training objective is

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \lambda_{\text{Feat}} \mathcal{L}_{\text{Feat}}, \tag{8}$$

where $\lambda_{\text{Feat}} > 0$ controls the strength of feature alignment.

Unlike ActSight's Control-Oriented Visual Refinement, FeatAlign applies perceptual supervision directly to the native VLA representation. Its gradients therefore update the native visual encoder as well as the training-only projector. The action decoder continues to consume the native visual tokens through the unchanged VLA pathway and never receives the projected external-space features. This baseline isolates direct point-wise feature alignment without residual adaptation or action-sensitive weighting.

RelAlign. RelAlign tests a complementary hypothesis: whether transferring the *relational structure* of external perceptual representations is sufficient without directly matching their feature vectors. Inspired by VideoREPA (Zhang et al., 2026) and MoAlign (Bhowmik et al., 2026), RelAlign matches token-to-token relations both within individual observations and across adjacent observations. For each valid adjacent pair (I_t, I_{t+1}) from the same demonstration and camera, we extract native VLA representations $\mathbf{V}_t$ and $\mathbf{V}_{t+1}$ together with the corresponding external perceptual targets $\mathbf{T}_t$ and $\mathbf{T}_{t+1}$. The future observation I_{t+1} is processed only to construct the auxiliary relation objective; its representation never enters the action decoder for the current observation.

RelAlign does not reproduce the motion-specific architectures or auxiliary pipelines of the cited systems. Instead, it provides a minimal controlled realization of relation matching using the same policy and external perceptual guidance as ActSight. For $u \in \{t, t+1\}$, we normalize native and external tokens along the feature dimension:

$$\bar{\mathbf{V}}_{u,i} = \frac{\mathbf{V}_{u,i}}{\max(\|\mathbf{V}_{u,i}\|_2, \epsilon)}, \qquad \bar{\mathbf{T}}_{u,i} = \frac{\mathbf{T}_{u,i}}{\max(\|\mathbf{T}_{u,i}\|_2, \epsilon)}, \tag{9}$$

where $\epsilon = 10^{-8}$. Because relation matching depends only on pairwise similarities, the feature dimensions of the VLA and external representations need not match, and no cross-space projector is required.

Intra-frame spatial relations. For each frame u, we construct the corresponding similarity matrices

$$\mathbf{G}_u^V = \bar{\mathbf{V}}_u \bar{\mathbf{V}}_u^\top, \qquad \mathbf{G}_u^T = \bar{\mathbf{T}}_u \bar{\mathbf{T}}_u^\top \in \mathbb{R}^{N \times N}. \tag{10}$$

For two similarity matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{N \times N}$, we define

$$\text{MAE}(\mathbf{X}, \mathbf{Y}) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} |X_{ij} - Y_{ij}|. \tag{11}$$

The spatial relation objective is then

$$\mathcal{L}_{\text{spatial}} = \frac{1}{2} \left[\text{MAE}(\mathbf{G}_t^V, \mathbf{G}_t^T) + \text{MAE}(\mathbf{G}_{t+1}^V, \mathbf{G}_{t+1}^T)\right]. \tag{12}$$

All entries, including the diagonal, are retained. Averaging over the N^2 entries prevents the loss scale from depending on the number of visual tokens.

Cross-frame temporal relations. We additionally match the cross-frame relation structure between adjacent observations:

$$\mathbf{H}^V_{t,t+1} = \bar{\mathbf{V}}_t \bar{\mathbf{V}}^\top_{t+1}, \qquad \mathbf{H}^T_{t,t+1} = \bar{\mathbf{T}}_t \bar{\mathbf{T}}^\top_{t+1}, \tag{13}$$

$$\mathcal{L}_{\text{temporal}} = \text{MAE}\left(\mathbf{H}^V_{t,t+1}, \mathbf{H}^T_{t,t+1}\right). \tag{14}$$

Both matrices have shape $N \times N$. Gradients from the relation objective update the native VLA visual encoder, whereas the external perceptual targets remain detached. Although RelAlign also uses adjacent observations, its temporal objective is fundamentally different from ActSight's Temporal-Aware Representation Alignment. RelAlign directly matches raw pairwise cosine relations with uniform weighting. It does not construct soft cross-frame correspondence distributions, estimate correspondence confidence through forward–backward consistency, use confidence-weighted KL alignment, or incorporate action-sensitive weighting.

Overall relational objective. We combine spatial and temporal relation matching as

$$\mathcal{L}_{\text{Rel}} = \mathcal{L}_{\text{spatial}} + \beta \mathcal{L}_{\text{temporal}}, \qquad \beta = 1, \tag{15}$$

and optimize

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \lambda_{\text{Rel}} \mathcal{L}_{\text{Rel}}, \tag{16}$$

where $\lambda_{\text{Rel}} > 0$ controls the strength of relational supervision.

B Experimental Environments

B.1 LIBERO

We evaluate ActSight on LIBERO (Liu et al., 2023), a benchmark for language-conditioned robotic manipulation. We consider four commonly evaluated 10-task suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-10, the latter also commonly referred to as LIBERO-Long. These suites cover complementary aspects of manipulation, including spatial relations, object variation, goal-conditioned behavior, and longer-horizon multi-stage interaction. We report the success rate for each suite and the average success rate across all four suites.

B.2 SimplerEnv-WidowX

We further evaluate ActSight in the WidowX-Bridge setting of SimplerEnv (Li et al., 2024b), which provides a complementary real-to-sim evaluation scenario. Policies are trained on real Bridge robot demonstrations and evaluated in the corresponding simulated WidowX environments. This setting introduces a visual distribution shift between real-world training observations and simulated evaluation observations, allowing us to examine whether improvements in visual representation learning generalize beyond the training distribution. Each observation consists of a primary-camera RGB image and a language instruction. We evaluate four WidowX manipulation tasks: putting a carrot on a plate, stacking a green cube on a yellow cube, placing a spoon on a table cloth, and putting an eggplant in a basket.

B.3 Additional Results

B.3.1 Deployment Efficiency and Computational Cost

We further examine the computational and deployment cost of ActSight on SimplerEnv-WidowX using the Qwen3.5-0.8B VLM backbone, with computational profiling conducted on NVIDIA A100 GPUs.

Limited Deployment Overhead. As shown in Table 3, ActSight substantially improves average task success while introducing only a modest increase in training-time parameters. The additional trainable parameters arise from the lightweight refinement pathway and auxiliary components used to incorporate external perceptual supervision during adaptation. At deployment, the external visual

Table 3: **Computational and deployment efficiency of ActSight on SimplerEnv-WidowX.** ActSight substantially improves control performance while introducing only limited training-time and deployment overhead.

	Baseline	ActSight
Avg. Success (%) ↑	40.6	72.9
Trainable Params. ↓	856.80 M	860.08 M
Deployment Params. ↓	856.80 M	857.85 M
Inference Latency ↓	141.1 ms	142.2 ms

encoders and training-only auxiliary branches are removed, leaving only the lightweight residual refinement pathway alongside the original VLA. As a result, ActSight preserves nearly the same inference efficiency as the baseline while achieving substantially better task performance. These results show that the gains of ActSight do not depend on deploying the external perceptual models or substantially increasing online computation.

C Hyperparameters and Implementation Details

Table 4 summarizes the training and implementation configurations used for the LIBERO and Bridge experiments. Bridge-trained policies are evaluated on SimplerEnv-WidowX.

Table 4: **Training and implementation configurations.**

Configuration	LIBERO	Bridge
Policy and optimization		
Optimizer	AdamW	AdamW
Learning rate	5×10^{-6}	5×10^{-6}
Adam coefficients (β_1, β_2)	$(0.9, 0.999)$	$(0.9, 0.999)$
Optimizer constant ϵ_{opt}	10^{-8}	10^{-8}
Global batch size	128	256
Gradient accumulation	1	4
Gradient clipping	1.0	Disabled
Action head	Attention-based regression head (width 1024, 11 blocks)	FCDecoder
Visual representation and residual adaptation		
Policy visual inputs	Primary + wrist cameras	Primary camera only
External encoder input resolution	256×256	256×256
Native visual-token grid	8×8 per camera	14×14 at residual interface
Residual MLP hidden dimension	512	512
Residual output initialization	Zero weight and bias	Zero weight and bias
Control-oriented perceptual supervision		
Perceptual coefficient λ_{perc}	0.25	0.25
Action-sensitive strength α	0.10	0.25
Sensitivity constant ϵ_{sens}	10^{-8}	10^{-8}
Temporal-aware representation alignment		
Temporal coefficient λ_{temp}	$0.03 \min(1, u/500)$	$0.03 \min(1, u/500)$
Correspondence temperature τ	0.1	0.1
Confidence constant ϵ_{temp}	10^{-6}	10^{-6}
Runtime and policy execution		
Training devices	4	4
Mixed precision	BF16	BF16
Predicted action-chunk length	8	4
Action ensembling	Disabled	Disabled

D Real-World Tasks and Experimental Setup

D.1 Platform and Training

We evaluate ActSight on a Franka Research 3 (FR3) robot across six real-world manipulation tasks. Baseline and ActSight use the same policy backbone and the same robot demonstrations. For each task, we collect 50 demonstration episodes, resulting in 300 demonstrations in total. RGB observations are captured at 15 FPS from a fixed frontal viewpoint using an Orbbec camera. The recorded image stream is temporally downsampled before being used for policy learning, resulting in approximately 50–100 policy steps per demonstration episode. The fixed frontal viewpoint provides a consistent observation interface across tasks, but does not provide the additional viewpoints available from a wrist-mounted camera. As the robot and manipulated objects move through the workspace, the policy must handle partial occlusion, viewpoint ambiguity, and interaction-induced changes using a single external view. This setting therefore provides a challenging test of visual representation learning for robotic control. External perceptual models used by ActSight provide supervision only during training and are removed before deployment. Baseline and ActSight consequently use the same observation interface during real-world evaluation.

D.2 Tasks and Evaluation Protocol

Tasks and language instructions. We consider six manipulation tasks spanning rigid-object placement, deformable-object manipulation, and sequential object interaction. The language instructions supplied to the policy are:

- **Banana:** "pick up the banana and place it on the plate."
- **Starfruit:** "pick up the starfruit and place it on the plate."
- **Snack:** "pick up the snack and place it on the plate."
- **Towel Fold:** "fold the towel in half."
- **Cup Nest-1:** "pick up the cup and place it into the target cup."
- **Cup Nest-2:** "pick up the two cups one by one and place them into the target cup."

The first three tasks require the robot to identify the instructed object, grasp it, transport it to the plate, and release it at the target location. Towel Fold introduces deformable-object manipulation, in which the appearance and geometry of the towel evolve continuously during execution. Cup Nest-1 requires placing one cup inside a target cup. Cup Nest-2 extends the task to sequential manipulation: two cups are picked up and placed into the target cup one after another. The second manipulation is therefore executed after the first placement has already changed the scene configuration.

Evaluation protocol. For each of the six in-distribution (ID) tasks, we evaluate each method over 10 real-world rollouts. Baseline and ActSight use the same task definitions, camera configuration, initialization procedure, and evaluation protocol. Each rollout is assigned a binary success/failure outcome according to the task objective. For the three pick-and-place tasks, success requires the instructed object to be placed on the target plate. Towel Fold requires completion of the specified folding operation. Cup Nest-1 and Cup Nest-2 require the instructed cup or cups to be successfully placed inside the target cup.

D.3 Scene-Level OOD Evaluation

We additionally evaluate scene-level visual distribution shifts on the three pick-and-place tasks: Banana, Starfruit, and Snack. The robot embodiment, camera viewpoint, language instructions, manipulation objectives, and success criteria remain unchanged from the corresponding ID setting. The OOD scenes introduce three types of visual changes. First, the tabletop background is changed from black to white. Second, task-irrelevant objects are added to the scene as visual distractors. Third, the spatial arrangement of objects in the scene is changed relative to the ID configuration. The rearranged objects are not restricted to the task-relevant target object. These modifications alter the visual appearance and scene composition while preserving the underlying manipulation objective. For each of the three OOD pick-and-place tasks, we conduct 10 real-world evaluation

rollouts, resulting in 30 rollouts in total. The OOD evaluation is intended to test robustness to changes in visual context, object arrangement, and irrelevant scene content rather than changes in robot embodiment or task semantics.

D.4 Qualitative Results

Figure 9 provides an overview of representative successful manipulation sequences for the six real-world tasks. Frames are ordered chronologically from left to right and illustrate the evolution of the robot–scene interaction throughout each task.

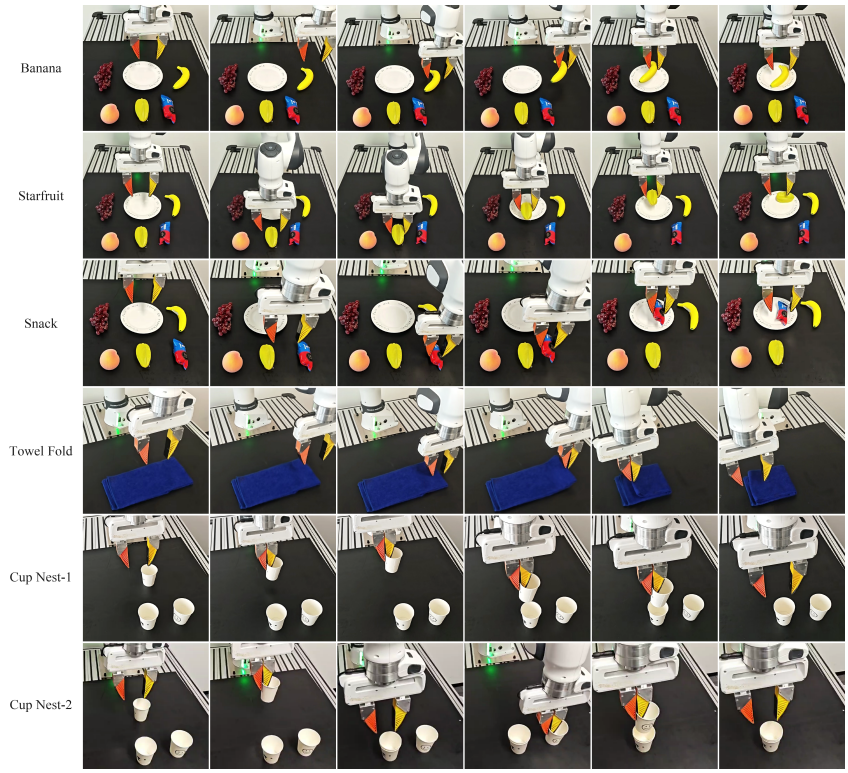


Figure 9: **Qualitative overview of the six real-world FR3 tasks.** Rows show Banana, Starfruit, Snack, Towel Fold, Cup Nest-1, and Cup Nest-2. Each row contains six chronological frames illustrating the corresponding manipulation sequence.