

Wiring Matters: Injection Topology and Initialization of Affordance Heads in Vision-Language-Action Policies

Zijian An¹, Linhan Wang², Jiayan Wang³, Shijie Geng⁴, Ran Yang², Yiming Feng², and Lifeng Zhou^{1*}

Abstract—Dense affordance supervision is an appealing auxiliary signal for vision-language-action (VLA) policies, yet naively co-training an affordance head can severely damage instruction following. We present a controlled study of *how* such a head should be wired into a modern VLA on the LIBERO benchmark. Our recipe reads the backbone through a stop-gradient and re-injects an intermediate head feature into the action expert via a learned bridge. The stop-gradient is a precondition: letting affordance gradients reach the backbone drops the policy below the headless base (85.5% vs. 93.1%). With the backbone protected, a same-budget 2×2 ablation over injection topology (concatenation vs. residual) and bridge initialization (zero vs. random) shows initialization is the dominant lever. The best wiring, an actively initialized residual bridge, reaches 96.2%, matching the far more elaborate three-expert AffordanceVLA (95.8%) with under 1% extra parameters. Two probes explain the mechanism: ground-truth affordances fed as an input hurt, and inference-time zeroing shows a lazy bridge acts only as a training-time regularizer while an active bridge becomes load-bearing.

I. INTRODUCTION

Vision-language-action (VLA) models have rapidly become the dominant recipe for language-conditioned robot manipulation. A large vision-language backbone, pretrained on internet-scale image-text data, is fine-tuned by imitation to map camera observations and a natural-language instruction to low-level actions [2], [3], [4], [5], [6]. The recipe inherits the backbone’s open-vocabulary perception and its language grounding, and recent flow-matching action experts [5] have pushed success rates on standard benchmarks such as LIBERO [9] above the 90% mark. Yet the supervision these models receive is curiously indirect. The action loss constrains only the *trajectory*; nothing in the objective ever tells the model *where* in the scene the interaction should happen. Which object the instruction refers to, which part of it to grasp, and where to place it must all be discovered implicitly, by correlating pixels with future gripper positions across the demonstration corpus.

The failure modes of imitation-trained VLAs mirror this gap. In our experiments, a strong 2B-parameter policy is nearly saturated on suites that test object diversity or spatial layout (98+%), but loses most of its remaining errors in

two places. The first is *instruction disambiguation*: scenes containing several plausible targets where only the language picks out the right one. The second is *long-horizon tasks*, where a small placement error early in the episode compounds. Both are precisely the situations in which an explicit, spatially grounded notion of “the place to act next” should help.

Affordances offer exactly that notion. The term, originating in ecological psychology [1], denotes the action possibilities a scene offers an agent; in robot learning it has been operationalized as dense maps of contact or interaction regions [12], [13], [14]. Affordance labels are attractive as auxiliary supervision for three reasons. First, in simulation they are *free*: the physics engine knows every contact between gripper and object, so pixel-accurate interaction maps can be extracted from existing demonstrations without human labeling. Second, they are *dense*: a per-pixel heatmap carries orders of magnitude more spatial signal per frame than a single action vector. Third, they are *task-aligned*: predicting where to interact next requires resolving the instruction, binding it to an object, and localizing a part, which are the exact sub-skills whose failures dominate VLA error budgets.

It is therefore no surprise that several recent systems build affordances into VLAs [15], [16]. What is surprising is how inconsistent the reported outcomes are, and how much architectural machinery is deployed to make the signal help. AffordanceVLA [16], the strongest published system, introduces a purpose-built three-expert mixture-of-transformers with a dedicated affordance-generation expert and a two-stage curriculum, and its own ablation shows that removing the second stage costs almost ten points. Our starting point was the opposite experience: attaching a straightforward affordance head to a VLA and co-training *hurt*, collapsing instruction-following performance. The same auxiliary task, on the same benchmark, can evidently help or harm depending on integration details that papers rarely state, let alone ablate.

This paper treats those details as the object of study. Our method rests on three wiring decisions. The first is a **stop-gradient**: the affordance head reads intermediate backbone layers, but its loss never updates the backbone. This is not a stylistic choice. Ablating it flips the sign of the whole intervention, dropping goal-conditioned success from the 90s to 62% and making the head-equipped policy *worse* than the headless base. With the backbone protected, the head’s only channel of influence on the policy is a single learned **bridge** that re-injects an intermediate head feature into the visual

¹Drexel University, Philadelphia, PA 19104, USA {za382, lz457}@drexel.edu

²Virginia Tech, Virginia Seafood Agricultural Research and Extension Center, Hampton, VA 23669, USA {ryang17, yimingfeng}@vt.edu

³TODO: Jiayan Wang’s affiliation and email

⁴Amazon Store Foundation AI (SFAI), New York, NY 10018, USA jeykigung@gmail.com. This work is independent of and outside of the author’s work at Amazon.

*Corresponding author.

tokens conditioning the action expert. This bridge is defined by two remaining bits of wiring, and we ablate both under a strictly identical data, budget, and seed protocol:

- **Injection topology:** the head feature is either *concatenated* with the visual tokens and linearly re-projected, which re-mixes (and can distort) the pretrained features, or added as a *residual*, which preserves them exactly.
- **Bridge initialization:** the injection weight starts at *zero* (“lazy”: the policy begins identical to the base and must earn a dependence on the head) or at a standard *random* initialization (“active”: the head’s feature participates at full magnitude from the first step).

These bits swing the result by more than the presence of the head itself. Initialization is the dominant lever, worth +2.0 to +2.7 points on either topology; topology contributes a further half point, and only under active initialization. The best cell, residual injection with active initialization, lifts the average success on LIBERO from 93.1% to 96.2%, with the gains concentrated exactly on the two failure modes above (Goal +5.6, Long +6.2).

Contributions. (1) A simple affordance-head recipe (stop-gradient + feature re-injection) that lifts a strong VLA from 93.1% to 96.2% average success on LIBERO, on par with the dedicated mixture-of-transformers AffordanceVLA (95.8%) [16] at a fraction of its architectural complexity. (2) A stop-gradient ablation showing gradient isolation is a *precondition*: with affordance gradients flowing into the backbone, the head makes the policy *worse* than the headless base (85.5% vs. 93.1%, with goal-conditioned success collapsing to 62%). The sign of the entire intervention flips on this single bit. (3) A same-budget 2×2 factorial ablation showing that, once the backbone is protected, *initialization of the injection bridge, not its topology, is the dominant lever*. Active initialization adds +2.0/+2.7 points over lazy on the concatenation/residual topologies respectively, a two-way internal replication. Topology contributes at most +0.5, and only under active initialization. (4) Two mechanism probes. Ground-truth affordance maps supplied as an *input* hurt (−5.6 points), showing the signal must be internalized by prediction rather than injected as data. Inference-time zeroing of the injected feature is a free diagnostic that separates two regimes. Under a lazy bridge the head is a pure training-time regularizer whose benefit survives ablation. Under an active bridge the affordance pathway becomes load-bearing, and it additionally lifts long-horizon success from 82.6% to 91.4%.

II. RELATED WORK

A. Vision-language-action models

Casting manipulation as next-action prediction on top of a pretrained vision-language model was popularized by RT-2 [2], which showed that web-scale semantics transfer into control when actions are emitted as tokens. Subsequent open efforts scaled the recipe across embodiments and datasets: Octo [4] and OpenVLA [3] established open-weight generalist policies, π_0 [5] replaced autoregressive action decoding with a flow-matching action expert conditioned on backbone

features, and GR00T N1 [6] extended the approach toward humanoid platforms. A parallel line augments the visual side of VLAs with explicit spatial structure, e.g. 3D position encodings in SpatialVLA [7] or intermediate visual sub-goals in CoT-VLA [8]. Our study is orthogonal to the backbone choice and concerns only how auxiliary dense supervision should be attached.

B. The StarVLA platform and the QwenPI model

We conduct all experiments in StarVLA [11], an open-source, “Lego-like” codebase for VLA development in which every functional component (model, data pipeline, trainer, configuration, and benchmark evaluation) is factored into plug-and-play modules with high cohesion and low coupling. The platform lets a VLM backbone be paired with interchangeable action experts and evaluated under standardized pipelines across benchmarks including LIBERO, RoboCasa, and CALVIN. This modularity is not incidental to our study but instrumental. A controlled wiring ablation is only meaningful if inserting the head, swapping the bridge topology, or changing an initialization touches *nothing else* in the stack. StarVLA’s factoring makes each of our variants a small, isolated diff against an otherwise frozen training and evaluation pipeline.

Our base policy is StarVLA’s **QwenPI** instantiation, which realizes the π_0 design [5] on a modern open backbone. A Qwen3-VL-2B VLM [10], a natively multimodal model pairing a ViT vision tower with a language decoder, encodes the camera views and the instruction. Action generation is delegated to a *layerwise flow-matching action expert*: a DiT that carries a noisy action chunk and denoises it under a conditional flow-matching objective, attending at each of its layers to the hidden states of the corresponding VLM layer. Compared with autoregressive action tokenization [2], [3], this keeps actions continuous and chunked while letting the expert read the backbone at every depth, which is precisely the interface our affordance bridge later writes into.

C. Affordances for manipulation

The original notion of affordance, the action possibilities an environment offers [1], entered robot learning as the problem of predicting *where* and *how* an agent can interact. Where2Act [12] learns per-pixel actionability scores for articulated objects from simulated interaction; VRB [13] distills contact regions and post-contact trajectories from human videos; MOKA [14] elicits affordance points from a VLM zero-shot via visual prompting and executes them open-loop. These works treat the affordance predictor as the *product*. Closer to us are systems that use affordances *in service of* an end-to-end policy: DeVA [15] co-trains dense affordance and depth heads on a video-generation backbone for embodied control, and AffordanceVLA [16] interleaves an affordance-generation expert with action generation inside a three-expert mixture-of-transformers, trained with a two-stage curriculum. Both report that affordance awareness improves manipulation, but both couple the signal to substantial bespoke architecture, which leaves open what the *minimal*

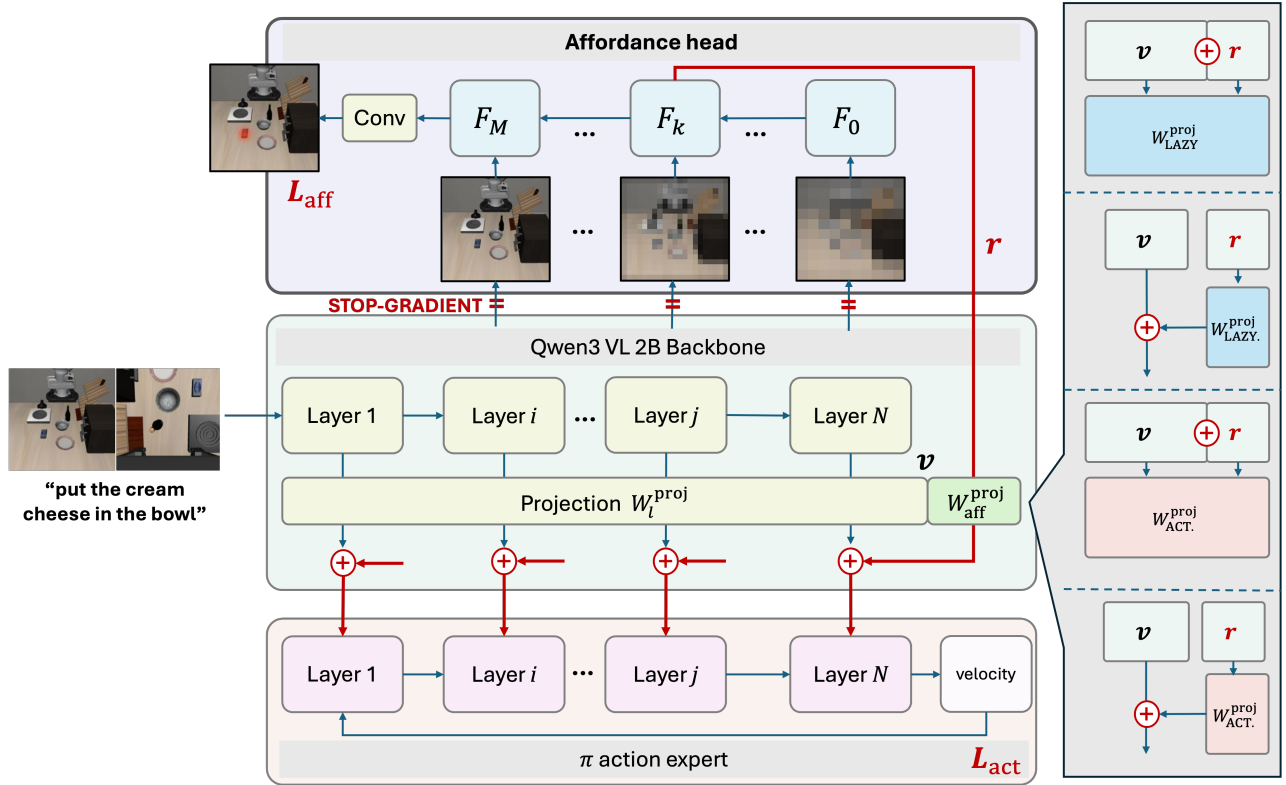


Fig. 1. **Left:** our recipe (RES-ACTIVE). The backbone encodes the camera views and the instruction in a single forward pass; per-layer projections W_l^{proj} hand each layer’s visual tokens v to the corresponding layer of the flow-matching action expert. The affordance head reads uniformly spaced backbone layers through stop-gradient taps, reassembles earlier layers onto finer grids and later layers onto coarser ones, fuses them through a cascade of stages $F_0, \dots, F_M$, and decodes a per-view contact heatmap supervised by $\mathcal{L}_{\text{aff}}$. The feature r of an intermediate fusion stage F_k is carried by the bridge $W_{\text{aff}}^{\text{proj}}$ (written W_{aff} in the text) back into the visual tokens at every conditioning layer. The expert is trained with the action objective $\mathcal{L}_{\text{act}}$ in a single pass; at inference it integrates the learned flow over a few passes. **Right:** the four bridge wirings of our 2×2 ablation, top to bottom CAT-LAZY, RES-LAZY, CAT-ACTIVE, and RES-ACTIVE (ours). Blue denotes a zero-initialized (lazy) bridge and red a randomly initialized (active) one; a $\oplus$ joining the adjacent v and r boxes denotes concatenation before the shared projection, whereas a $\oplus$ on the token stream denotes elementwise addition of the projected r onto the untouched v .

effective coupling is. Our result, a stock VLA plus one head and one linear bridge matching AffordanceVLA on LIBERO, suggests the architecture is not the essential ingredient; the wiring is.

D. Auxiliary objectives, gradient interference, and bridges

Auxiliary losses have long been used to shape representations, from the self-supervised auxiliary tasks UNREAL in RL [17] to depth and segmentation heads in perception stacks. The risk is equally well documented: auxiliary gradients can interfere destructively with the primary task [18]. Our preliminary experiments reproduce exactly this failure, with the affordance loss collapsing instruction following when backpropagated into the backbone. Our stop-gradient design sidesteps interference structurally rather than algorithmically, at the price of raising a new question: if the auxiliary task cannot alter the backbone, how does it influence the policy at all? The answer runs through the injection bridge, which connects our work to a literature on *how to attach new pathways to pretrained models*. Zero-initialized connections are the standard conservative choice: ControlNet’s zero convolutions [19], Flamingo’s tanh-gated cross-attention [20], and the zero-initialized up-projection of

LoRA [21] all guarantee that a new branch starts as an exact identity and grows only as needed. Our central empirical finding is that this orthodoxy can silently fail in the auxiliary-head setting. When the primary task can already be solved from the existing features, the zero-initialized bridge sits at a *lazy attractor*: it never grows, the injected feature stays runtime-inert, and the head’s benefit reduces to a real but smaller implicit regularization effect. Random initialization is exactly what the adapter literature avoids, because it perturbs the pretrained computation. Yet it is what forces the policy’s representation to form around the affordance pathway, and it is worth more than two points on LIBERO. This inverts the usual guidance and, to our knowledge, has not been isolated before in a VLA setting.

III. METHOD

We describe our full method as instantiated in its best configuration (RES-ACTIVE); Sec. III-D defines the three ablated alternatives. Fig. 1 summarizes the data flow.

A. Base policy and notation

The base policy is the QwenPI model introduced in Sec. II-B: the Qwen3-VL-2B backbone encodes two RGB views (a

third-person and a wrist camera) and the instruction, and the layerwise flow-matching expert denoises action chunks. The backbone has N layers; we write H_l for the hidden states of backbone layer l and v_l for the visual tokens the expert consumes at conditioning layer l . Writing x_1 for the demonstrated action chunk and x_0 for Gaussian noise, the expert v_θ is trained with the conditional flow-matching objective [5]

$$\mathcal{L}_{\text{act}} = \mathbb{E}_{t, x_0} \|v_\theta(x_t, t, \{v_l\}) - (x_1 - x_0)\|^2, \quad (1)$$

where $x_t = (1-t)x_0 + tx_1$ is the linearly interpolated noisy chunk. The base is trained with $\mathcal{L}_{\text{act}}$ only.

B. Affordance supervision from demonstration replay

Our affordance target answers one question per frame: *where will the gripper next make task-relevant contact?* We extract this from the existing LIBERO demonstrations, with no human labeling, by exploiting privileged simulator state. LIBERO tasks run in a MuJoCo physics engine [28] via the robosuite framework [29]; each demonstration is replayed step by step, and at every timestep the engine’s contact solver exposes the exact set of geometry pairs in contact. Scanning *forward* along the replay from frame t , the first object the gripper contacts defines the frame’s interaction target. Its 3D contact location is projected through each camera’s calibrated intrinsics and extrinsics to a pixel coordinate u^* , and the label is an isotropic Gaussian of fixed scale σ centered there, one map per camera view:

$$A(u) = \exp\left(-\frac{\|u - u^*\|^2}{2\sigma^2}\right). \quad (2)$$

Two properties of this construction matter. First, the label is a *look-ahead* signal: within a task stage it is constant (the whole approach phase points at the same grasp location) and it switches discretely at stage boundaries (the moment the mug is grasped, the label jumps to the target plate). The head is therefore supervised to predict not the present but the immediate motor future, the same latent decision the policy must make. Second, because the Gaussian template is fixed, the per-frame label mass is constant, so the prediction task never degenerates into predicting label *intensity*; only *location* carries information.

C. Affordance head: multi-scale readout of a columnar backbone

Where the head reads from. The head taps the backbone’s hidden states at a small set of uniformly spaced layers. The motivation is the well-established semantic stratification of deep networks: early transformer layers carry local, high-frequency appearance information, while late layers carry increasingly global, language-conditioned semantics. Affordance prediction needs *both ends* of this spectrum simultaneously. Selecting the correct object among distractors is a late-layer, instruction-conditioned decision, but placing the peak on the mug’s *handle* to pixel accuracy is an early-layer, appearance-driven one. A single tap would force a choice between the two; uniformly spaced taps hand the head the full hierarchy and let it learn the mixture.

Why a pyramid. A vision transformer is *columnar*: every layer represents the image at the same, coarse to-token resolution, unlike convolutional hierarchies whose resolution decreases with depth [22]. Dense prediction from such a backbone therefore faces the classic tension between recognition and localization. Semantic strength lives in coarse features, while spatial precision requires fine ones. The canonical resolution of this tension is the top-down pyramid with lateral connections, introduced by FPN for object detection [23] and, in encoder-decoder form, by U-Net for segmentation [25]. In such a pyramid, coarse and semantically strong features are progressively upsampled and *fused at each scale* with finer ones, so every output location is informed by both global context and local evidence. DPT [26] showed how to manufacture the missing multi-scale structure from a columnar ViT by *reassembling* different layers’ tokens at different synthetic resolutions before running such a pyramid. We adopt this principle for the same first-principles reason it was invented: our output must be sharp at pixel level but selected by global, instruction-level semantics. Concretely, each tapped layer is reassembled onto its own grid, earlier layers onto finer grids to preserve their high-frequency content and later layers onto coarser grids matching their semantic granularity. The grids are fused top-down with lateral connections through a cascade of fusion stages $F_0, \dots, F_M$ (coarse to fine) and decoded to the per-view contact heatmap, trained with binary cross-entropy $\mathcal{L}_{\text{aff}}$.

Instruction conditioning. One subtlety of reading a causal multimodal backbone is that the image tokens precede the instruction in the sequence, so under causal attention their hidden states cannot attend to the text that follows. The head therefore receives the instruction through an explicit second route: the instruction token embeddings are pooled and applied as a FiLM modulation [24] to the readouts and to every fusion stage. The head is thus instruction-conditioned by construction, which is what allows it to place its peak on the commanded object even when an identical distractor is present. It also means the injected feature r carries an instruction-resolved spatial decision rather than a purely visual saliency map.

Stop-gradient. Collecting the pieces, the head h_ϕ maps the tapped hidden states and the pooled instruction embedding e_{txt} to a predicted map, and is trained against the label of Eq. (2) with binary cross-entropy:

$$\hat{A} = h_\phi(\{\text{sg}[H_l]\}_{l \in \mathcal{T}}, e_{\text{txt}}), \quad \mathcal{L}_{\text{aff}} = \text{BCE}(\hat{A}, A), \quad (3)$$

where $\mathcal{T}$ is the set of tapped layers and $\text{sg}[\cdot]$ is the stop-gradient operator: $\mathcal{L}_{\text{aff}}$ updates *only* head parameters ϕ , never the backbone. The rationale is asymmetric protection. The backbone’s language grounding is the scarcest resource in the system; it was bought with internet-scale pretraining and is easily corrupted. In preliminary experiments, letting affordance gradients into the backbone collapsed goal-conditioned success from 94% to below 32%. The head, in contrast, is cheap and can adapt to the backbone rather than vice versa. Note that the stop-gradient is not a freeze. Every parameter

of the backbone and the action expert remains trainable end to end under the action loss; we only deny the auxiliary loss a write path into the backbone. The recipe therefore needs no frozen modules, no LoRA-style protection, and no staged curriculum: it is plain full fine-tuning, single-stage, under the combined objective

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \lambda \mathcal{L}_{\text{aff}}, \quad (4)$$

and we found performance insensitive to λ .

D. The affordance bridge

The stop-gradient leaves the head an isolated observer: accurate, but by default mute with respect to the policy. Everything the head contributes to control therefore flows through a single learned **bridge**, and the rest of this section develops the bridge from the inside out: which head feature it carries, and the small design space of ways to write that feature into the policy.

The carried feature is not the final heatmap. The heatmap is a one-channel bottleneck: it reports *where* but discards the head’s intermediate computation about *what* and *why*. We instead take the feature r of an intermediate fusion stage F_k , chosen because its spatial grid coincides with the backbone’s native visual-token grid. Injection is then token-aligned: each visual token receives the head feature for precisely its own spatial cell. At the same time, F_k sits late enough in the cascade that coarse semantics and fine localization have already been married. This feature is written into the visual tokens at *every* conditioning layer of the action expert, so the affordance evidence accompanies the visual stream at whatever depth the expert consults it. Our default writes it as a residual:

$$v'_l = v_l + W_{\text{aff}} r \quad (\text{residual}), \quad (5)$$

on the standard grounds of residual learning [27]: the pre-trained visual features, the base policy’s entire working capital, pass through unmodified, and the head contributes a purely additive refinement. The natural alternative concatenates and re-projects,

$$v'_l = W_{\text{aff}} [v_l \parallel r] \quad (\text{concatenation}), \quad (6)$$

which is more expressive but forces even the original features through a learned re-mixing; Sec. IV-B shows this measurably erodes fine-grained placement.

The bridge has one further degree of freedom, and it turns out to be the consequential one: how W_{aff} is initialized. We use a standard Kaiming random scheme (*active*) rather than zero (*lazy*), contradicting the prevailing adapter-design orthodoxy [19], [20], [21]. The zero-initialized bridge looks safer, since the policy starts exactly at the base model. But it must earn its growth from the gradient of the action loss, and that gradient is starved by construction: the head offers information the action expert can, with effort, extract from v_l anyway. The marginal value of the bridge at $W_{\text{aff}}=0$ is therefore near zero, and the optimizer never leaves the attractor. Empirically, lazily initialized bridges converge to a few percent of the magnitude of the identity path and

TABLE I
LIBERO SUCCESS RATES (%), 50 TRIALS/TASK. ALL RUNS SHARE DATA, BUDGET, AND SEED; ONLY THE HEAD WIRING DIFFERS. **BOLD** = BEST PER COLUMN. [†]MEASURED IN A PRELIMINARY CONFIGURATION OF THE SAME PIPELINE.

Model	Object	Spatial	Goal	Long	Avg
Base (no head)	98.0	98.6	90.6	85.2	93.1
Head w/o stop-grad [†]	100.0	92.0	62.0	88.0	85.5
CAT-LAZY	99.4	98.2	94.8	82.6	93.8
RES-LAZY	98.8	96.8	93.4	85.2	93.6
CAT-ACTIVE	98.0	99.6	96.0	89.4	95.8
RES-ACTIVE	98.8	98.4	96.2	91.4	96.2

are runtime-inert; zeroing them at inference changes nothing (Sec. IV-F). Active initialization removes the chicken-and-egg problem by fiat. The feature r is present at full magnitude from step one, the expert’s representation forms around it, and the pathway becomes genuinely load-bearing, at the cost of a violent start: the action loss opens an order of magnitude higher before recovering within a few hundred steps.

Crossing the two topologies with the two initializations yields the four wirings we evaluate (Fig. 1, right): CAT-LAZY, RES-LAZY, CAT-ACTIVE, and RES-ACTIVE, of which RES-ACTIVE is our method and the other three are ablations.

E. What can the head even contribute?

The affordance target is a deterministic function of the observation and instruction, and the head reads the same backbone features the action expert already sees; information-theoretically, r can offer the policy *no new information*, only a more explicit format. Any benefit must therefore come from (a) the optimization dynamics induced by the head’s presence, or (b) the explicit format easing the action expert’s job at run time. The two initializations of Sec. III-D turn out to select between exactly these mechanisms, and the experiments below separate them.

IV. EXPERIMENTS

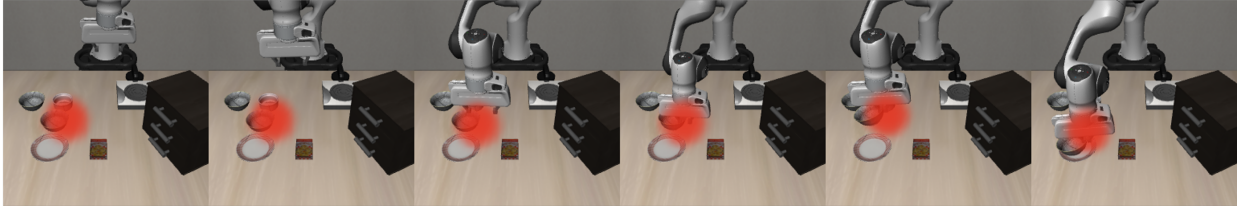
A. Setup

All models are fine-tuned on the LIBERO demonstration corpus (all four suites jointly) from the same Qwen3-VL-2B initialization, with an identical budget: global batch 128, 20k steps (≈ 9.4 epochs), identical hyperparameters and seed. The *only* difference between the five runs is the head wiring. Evaluation is 50 trials per task (500 per suite) on LIBERO-Spatial, -Object, -Goal, and -Long (LIBERO-10).

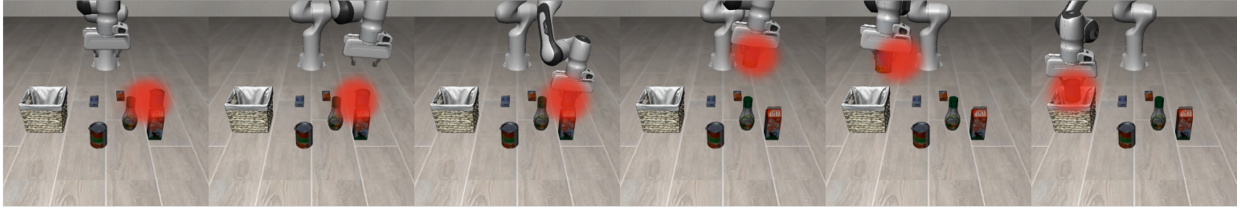
B. Main result: the head helps, and wiring decides how much

Table I and Fig. 3 show the full 2×2 plus the headless base. Every wired variant matches or beats the base on average, and the best cell, RES-ACTIVE, improves the average from 93.1% to 96.2%, with the gains exactly where a policy needs help: Goal +5.6 and Long +6.2, at parity on the near-saturated Object/Spatial suites. Reading the table factorially:

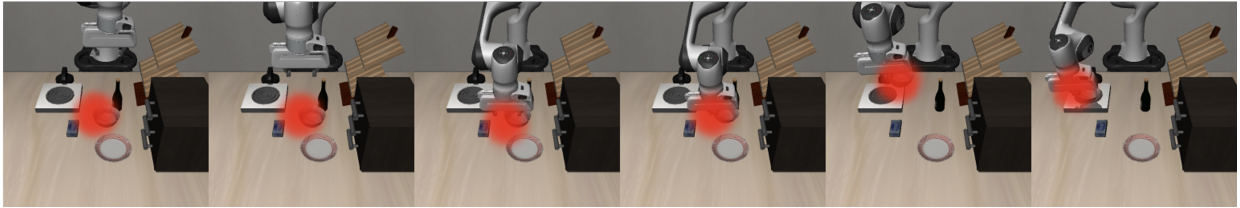
Spatial: “pick up the black bowl between the plate and the ramekin and place it on the plate”



Object: “pick up the alphabet soup and place it in the basket”



Goal: “put the bowl on the stove”



Long: “turn on the stove and put the moka pot on it”

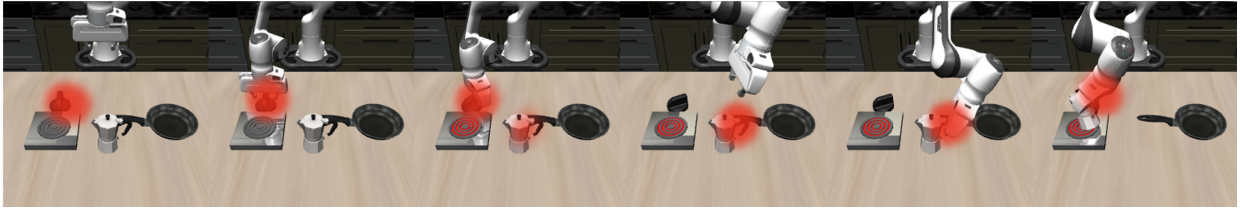


Fig. 2. RES-ACTIVE rollouts with the affordance head’s own predictions overlaid in red. One task per suite, six evenly spaced frames per episode; all four episodes succeed. The predicted contact region locks onto the instruction-designated object among distractors and switches discretely at stage boundaries, e.g. from the stove knob to the moka pot to the burner in the long-horizon task.

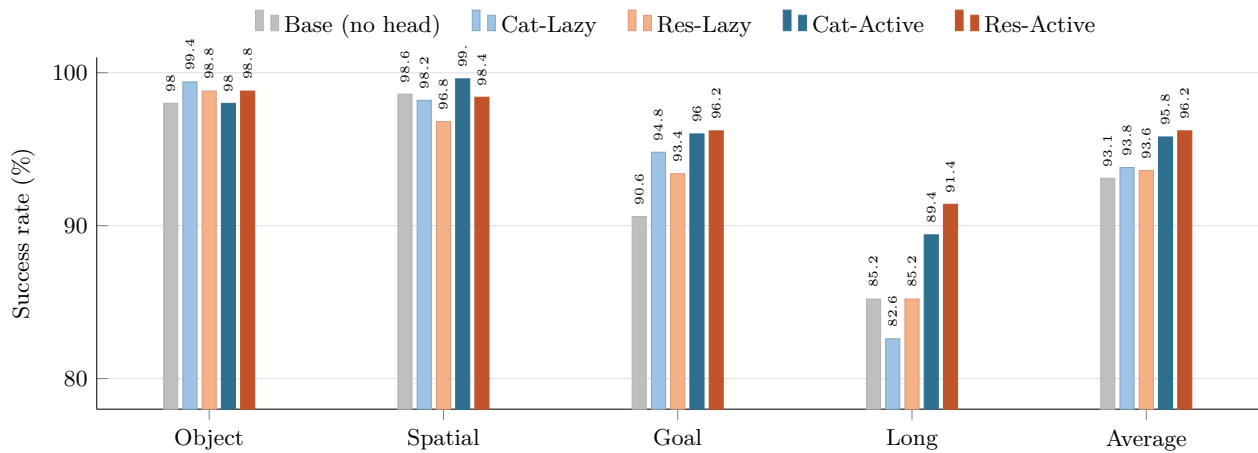


Fig. 3. Per-suite success rates of the headless base and the four bridge wirings (50 trials/task). Color encodes topology (blue: concatenation, orange: residual) and saturation encodes initialization (light: lazy, dark: active). The two active wirings dominate on Goal and Long, the suites that stress instruction disambiguation and long-horizon precision, while all five models are near saturation on Object and Spatial.

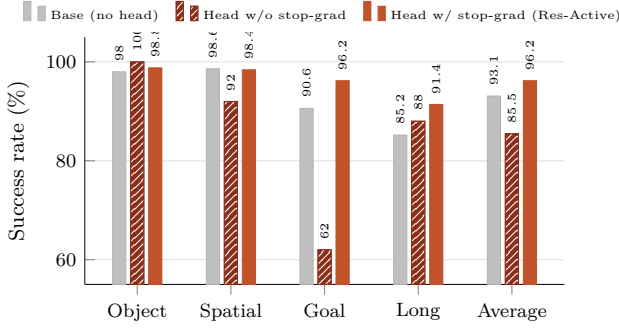


Fig. 4. The stop-gradient is necessary. The same head helps or hurts depending on this single bit: with the gradient blocked (RES-ACTIVE) the head lifts the average to 96.2, while letting the affordance gradient reach the backbone drops it to 85.5, a 10.7-point swing. The gradient-exposed variant even falls 7.6 points below the headless base, and the damage is selective: Goal, the suite that leans hardest on instruction grounding, collapses from the 90s to 62.0.

Initialization is the dominant lever. Active vs. lazy is worth +2.0 points on the concatenation topology (93.8→95.8) and +2.7 on the residual topology (93.6→96.2): the same direction and magnitude on both rows, a two-way replication. Inspecting checkpoints explains why. With lazy initialization the bridge weight converges to only a few percent of the magnitude of the identity path, i.e. the optimizer never grows a real dependence on r ; with active initialization the bridge retains full magnitude and the action expert’s representation forms around it.

Topology is a secondary lever, visible only under active init. Under lazy initialization the two topologies tie (93.8 vs. 93.6). Under active initialization the residual form adds +0.5 over concatenation, with the gain concentrated on LIBERO-Long (89.4→91.4). The Long-suite column tells the cleanest story: 82.6 (CAT-LAZY) → 85.2 (RES-LAZY) → 89.4 (CAT-ACTIVE) → 91.4 (RES-ACTIVE). Concatenation’s re-mixing of visual tokens measurably harms fine-grained, long-horizon placement: under lazy initialization the bridge is inert, so re-mixing is the only active difference, and it costs 2.6 points against the base. The residual’s identity path recovers exactly that loss. Active initialization then adds its larger, orthogonal gain on top.

C. Stop-gradient is the precondition

Fig. 4 makes the case with a three-way comparison: the headless base, the head with its gradient allowed into the backbone, and the same head behind a stop-gradient (RES-ACTIVE). The first comparison is against our best model. Blocking the gradient is worth 10.7 points on average (85.5 vs. 96.2), so the stop-gradient is not an optimization detail but the difference between the head helping and the head failing. The second comparison is against the base, and it is the more damning one. The gradient-exposed variant lands 7.6 points *below* a model with no head at all: the auxiliary gradient does not merely waste the head, it actively degrades the backbone’s own reasoning. The damage is selective and diagnostic. Object-level success is untouched

TABLE II
COMPARISON WITH AFFORDANCEVLA ON LIBERO (%). THEIR NUMBERS AS REPORTED IN [16].

Method	Object	Spatial	Goal	Long	Avg
AffordanceVLA (w/o stage II)	91.7	88.5	91.3	73.3	86.2
AffordanceVLA (full)	98.4	98.6	96.2	89.8	95.8
RES-ACTIVE (ours)	98.8	98.4	96.2	91.4	96.2

(a perfect 100), since recognizing and reaching a unique object needs little language. Goal-conditioned success instead collapses to 62%. The auxiliary gradient specifically erodes the backbone’s *instruction grounding*, the capability bought with vision-language pretraining and consumed most heavily by disambiguation. We suspect this failure mode accounts for much of the mixed evidence on auxiliary heads in the literature. All results in the rest of the paper therefore keep the stop-gradient in place; it is the precondition that makes the remaining 2×2 worth studying.

D. Comparison with AffordanceVLA

Table II compares our best variant with AffordanceVLA [16], the closest prior work. The two reach essentially the same performance, but by very different means. AffordanceVLA introduces a purpose-built three-expert Mixture-of-Transformers with a dedicated affordance-generation expert and a two-stage training curriculum, and its own ablation shows the design is fragile, dropping to 86.2% without the second stage. Our result comes from a stock VLA plus one DPT-style head and one linear bridge, which together add under 1% to the policy’s parameter count. Training is a single stage with no frozen modules, and the inference-time overhead of running the head is negligible. The entire “architecture” of our method is two wiring decisions. Explicit affordance grounding, in other words, does not require an affordance-specialized architecture. It requires getting the wiring right.

E. Probe 1: ground-truth affordances as input are net harmful

Is the affordance signal itself valuable enough that the policy should simply be *given* it? To test this, we trained a variant that receives the ground-truth affordance heatmap as a third input image, with an oracle rendering the map online at evaluation (in a preliminary configuration of the same pipeline, against its own matched base at 94.0% average). Despite privileged information, this variant reaches only 88.3%, a net -5.6 . It is roughly neutral on Object and Spatial but loses 5.5 on Goal and 19 on Long: it specifically damages instruction disambiguation and multi-stage tasks. Externally supplied answers create an input shortcut that competes with the instruction pathway; the signal must be *internalized* by prediction, not injected as data.

F. Probe 2: inference-time zeroing separates two regimes

We replace the injected feature with zeros at inference; the head still runs, only the bridge is silenced. For CAT-LAZY, nothing changes (Goal 94.8→95.6, Long 82.6→83.2,

within noise). This is consistent with the small bridge magnitude: the run-time contribution of r is negligible, yet the variant still beats the base on Goal by +4.2. The head’s entire benefit in this regime is a *training-time regularization* effect. Its presence alters the optimization trajectory of the action expert, not the test-time computation. In contrast, in preliminary experiments with an actively initialized bridge, the same ablation collapsed Goal success by 28 points: an active pathway becomes load-bearing. The zeroing probe is free to run and we recommend it as a standard diagnostic for auxiliary-task pathways in VLAs.

G. Head quality is not the bottleneck

Across variants the head itself trains to high fidelity: its predicted contact peaks land within a few pixels of ground truth on held-out frames and correctly disambiguate even duplicate objects. Rollout inspection of the remaining Long-suite failures shows correct target selection with second-stage precision errors (e.g., placing onto an occupied burner). The open problem is thus not *predicting* affordances but *transducing* them into finer control.

V. LIMITATIONS AND CONCLUSION

Our study is limited to one simulated benchmark, one backbone scale, and one seed per cell, partially mitigated by the two-way replication of the initialization effect and a point-identical retest of the Long suite. Affordance labels also come from privileged simulator state. Within this scope the conclusions are sharp. First, a stop-gradient affordance head with feature re-injection reliably helps. Both actively initialized wirings beat the headless base, CAT-ACTIVE by +2.7 points on average and RES-ACTIVE by +3.1, with the gains concentrated on instruction disambiguation and long-horizon precision. Second, the two wiring choices help for separable reasons. Active initialization is the main lever on both topologies: it forces the action expert to build its representation around the affordance pathway instead of leaving the bridge at its lazy attractor. The residual topology adds its own margin on top, chiefly on the Long suite, because its identity path hands the pretrained visual features to the expert untouched, while concatenation re-mixes and slightly degrades them. RES-ACTIVE combines both advantages and is the best model in every comparison we ran, matching the purpose-built AffordanceVLA architecture with one head and one linear bridge. More broadly, the wiring bits that most papers leave unstated, bridge initialization and injection topology, swing the result by more than the presence of the head itself. We hope the 2×2 protocol and the zeroing probe become standard reporting practice for auxiliary heads in VLAs.

REFERENCES

- [1] J. J. Gibson, *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.
- [2] A. Brohan *et al.*, “RT-2: Vision-language-action models transfer web knowledge to robotic control,” in *CoRL*, 2023.
- [3] M. J. Kim *et al.*, “OpenVLA: An open-source vision-language-action model,” in *CoRL*, 2024.
- [4] Octo Model Team, “Octo: An open-source generalist robot policy,” in *RSS*, 2024.
- [5] K. Black *et al.*, “ π_0 : A vision-language-action flow model for general robot control,” *arXiv:2410.24164*, 2024.
- [6] NVIDIA, “GR00T N1: An open foundation model for generalist humanoid robots,” *arXiv:2503.14734*, 2025.
- [7] D. Qu *et al.*, “SpatialVLA: Exploring spatial representations for visual-language-action models,” in *RSS*, 2025.
- [8] Q. Zhao *et al.*, “CoT-VLA: Visual chain-of-thought reasoning for vision-language-action models,” in *CVPR*, 2025.
- [9] B. Liu *et al.*, “LIBERO: Benchmarking knowledge transfer for lifelong robot learning,” in *NeurIPS Datasets and Benchmarks*, 2023.
- [10] Qwen Team, “Qwen3-VL technical report,” 2025.
- [11] StarVLA Community, “StarVLA: A Lego-like codebase for vision-language-action model developing,” *arXiv:2604.05014*, 2026.
- [12] K. Mo, L. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, “Where2Act: From pixels to actions for articulated 3D objects,” in *ICCV*, 2021.
- [13] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, “Affordances from human videos as a versatile representation for robotics,” in *CVPR*, 2023.
- [14] F. Liu, K. Fang, P. Abbeel, and S. Levine, “MOKA: Open-world robotic manipulation through mark-based visual prompting,” in *RSS*, 2024.
- [15] DeVA Authors, “DeVA: Depth and affordance heads for embodied video models,” *arXiv:2607.24159*, 2026.
- [16] Q. Yu *et al.*, “AffordanceVLA: A vision-language-action model empowering action generation through affordance-aware understanding,” *arXiv:2606.06155*, 2026.
- [17] M. Jaderberg *et al.*, “Reinforcement learning with unsupervised auxiliary tasks,” in *ICLR*, 2017.
- [18] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, “Gradient surgery for multi-task learning,” in *NeurIPS*, 2020.
- [19] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *ICCV*, 2023.
- [20] J.-B. Alayrac *et al.*, “Flamingo: A visual language model for few-shot learning,” in *NeurIPS*, 2022.
- [21] E. J. Hu *et al.*, “LoRA: Low-rank adaptation of large language models,” in *ICLR*, 2022.
- [22] A. Dosovitskiy *et al.*, “An image is worth 16×16 words: Transformers for image recognition at scale,” in *ICLR*, 2021.
- [23] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *CVPR*, 2017.
- [24] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, “FiLM: Visual reasoning with a general conditioning layer,” in *AAAI*, 2018.
- [25] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *MICCAI*, 2015.
- [26] R. Ranftl, A. Bochkovskiy, and V. Koltun, “Vision transformers for dense prediction,” in *ICCV*, 2021.
- [27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.
- [28] E. Todorov, T. Erez, and Y. Tassa, “MuJoCo: A physics engine for model-based control,” in *IROS*, 2012.
- [29] Y. Zhu *et al.*, “robosuite: A modular simulation framework and benchmark for robot learning,” *arXiv:2009.12293*, 2020.