AffordanceVLA: A Vision-Language-Action Model Empowering Action Generation through Affordance-Aware Understanding

Qize Yu[1,†], **Jiadi You**[2,†], **Yuran Wang**[1], **Jiaqi Liang**[1], **Bowen Ping**[1], **Yang Tian**[1], **Yue Chen**[1], **Minghong Cai**[3], **Zeying Gong**[2], **Ruihai Wu**[1], **Yinchuan Li**[4], **Junwei Liang**[2,*], **Yingcong Chen**[2,*]

[1]Peking University, [2]Hong Kong University of Science and Technology (Guangzhou), [3]The Chinese University of Hong Kong, [4]Knowin AI
[†]Equal contribution. [*]Corresponding authors.

Vision-Language-Action (VLA) models leverage the rich world knowledge of pretrained vision-language models (VLMs) to enable instruction-following robotic manipulation. However, the structural mismatch between VLM semantic spaces and embodied control policies often hinders the learning of precise perception–action mappings. To address this challenge, we propose **AffordanceVLA**, a unified framework that introduces structured affordance forecasting as a task-oriented intermediate representation to establish a more precise and robust perception–action mapping. Specifically, we progressively model manipulation priors through three complementary components: 1) **Which2Act** for object-centric grounding via visual latent prediction to suppress distractions; 2) **Where2Act** for 2D interaction localization via affordance map estimation; and 3) **How2Act** for 3D geometric reasoning to guide manipulation policies. These affordance cues provide spatially grounded, semantically conditioned, and action-coupled intermediate representations, thereby naturally bridging vision, language and action. We integrate these modules into a Mixture-of-Transformer (MoT) architecture with specialized experts and train the model using a three-stage training strategy with a progressive data curriculum. To overcome the scarcity of dense affordance labels in robotic datasets, we also develop a robust automated data augmentation pipeline. Extensive experiments on simulation and real-world demonstrate that AffordanceVLA achieves strong performance across diverse manipulation scenarios.

Date: April 10, 2026
Code: https://github.com/Skywalker-yqz/AffordanceVLA/
Project Page: https://skywalker-yqz.github.io/AffordanceVLA/

1 Introduction

Vision-Language-Action (VLA) models [4, 9, 29, 45, 61, 68] have shown great promise in instruction-following manipulation, largely driven by the rich world knowledge embedded in pretrained Vision-Language Models (VLMs). Many approaches [39, 53, 55, 57, 60, 69, 75] attempt a direct end-to-end mapping from natural language instructions and visual observations to robot actions. However, while the core of VLM pre-training lies in aligning vision and language within a semantic space, robotic actions are inherently representations in the 3D physical space, creating a significant gap that makes learning a direct mapping challenging. Therefore, finding an appropriate intermediate representation to bridge perception and action is crucial.

Previous works have made substantial efforts to bridge perception and action. A conventional paradigm employs video prediction or visual foresight [47, 63, 79, 80, 82, 88] to guide action generation. However, such dense visual signals often contain redundant information, and the inference process of video prediction models is typically time-consuming. In addition to video prediction, many works incorporate visual representations such as depth, visual grounding, and optical flow [2, 7, 38, 84] to enhance visual localization. Overall, the exploration and utilization of intermediate representations remains an open question.

We argue that a robust VLA requires learning a richer, task-oriented representation to enhance generalization across complex manipulation scenarios. Without innovating the model paradigm, blindly scaling up data

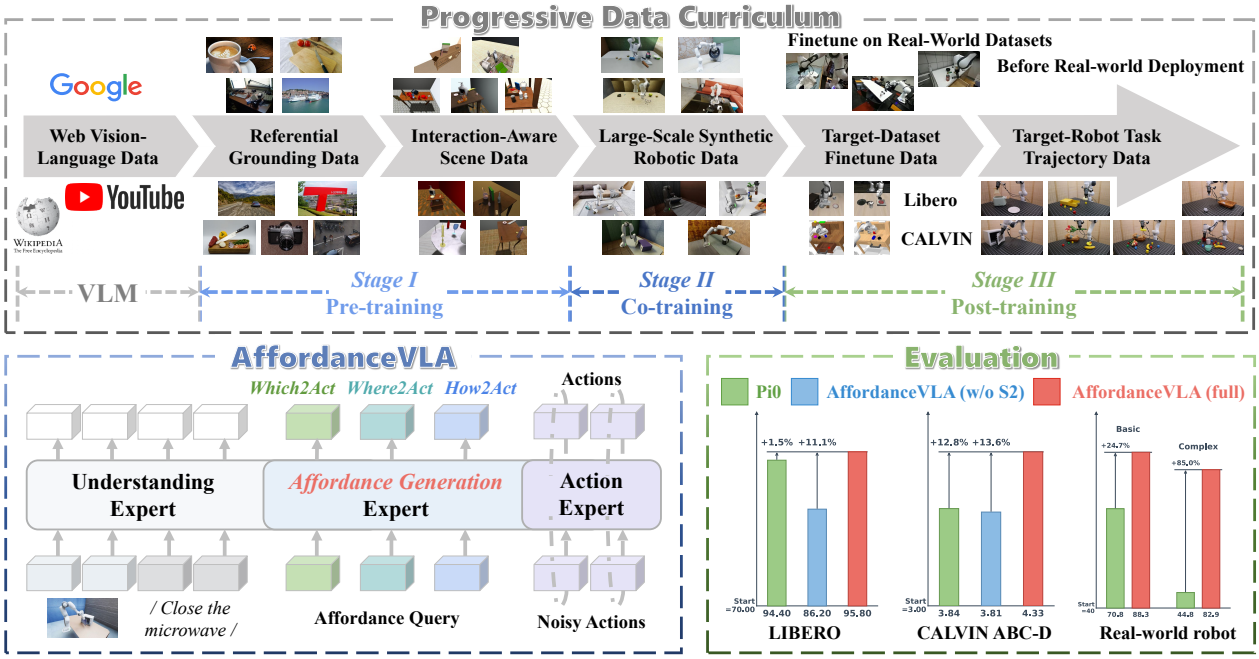


Figure 1 AffordanceVLA Overview. (Bottom-left) AffordanceVLA employs three specialized experts (Understanding, Affordance Generation, and Action), leveraging structured affordance forecasting (Which2Act, Where2Act, and How2Act) as intermediate representations to bridge perception and action. **(Top)** Enabled by a three-stage training strategy with progressive data curriculum, **(Bottom-right)** AffordanceVLA achieves strong performance across both simulation and real-world evaluations.

fails to maximize the intrinsic power within the datasets, and relying solely on scaling is insufficient to resolve the fundamental spatial gap. Indeed, recent strong VLAs increasingly introduce structured, train-only intermediate supervision to better exploit their data and to keep the backbone's vision–language ability from being eroded by low-level action learning—a direction we pursue through affordance.

> Without innovating the model paradigm, blindly scaling up data fails to maximize the intrinsic power within the datasets, and relying solely on scaling is insufficient to resolve the fundamental spatial gap.

Intuitively, much like how humans naturally perceive a mug's handle as an invitation to grasp, an embodied agent should understand its environment. **Affordances** [20]—manipulation priors that explicitly indicate *which* object to manipulate, as well as *where* and *how* to interact—naturally fulfill this role as ideal intermediate prediction targets [10, 11, 36, 73]. They serve as a perfect bridge by seamlessly coupling spatial grounding in vision, semantic conditioning in language, and execution guidance in action. Furthermore, extensive prior work [33, 41, 52, 66, 67, 71, 72, 83] has shown that affordance representations are readily learnable and highly adaptable.

> Affordances serve as a perfect bridge, seamlessly coupling spatial grounding in vision, semantic conditioning in language, and execution guidance in action.

Building upon these insights, we propose **AffordanceVLA** (as illustrated in Figure 1), a novel framework that leverages affordance forecasting to establish a precise and robust perception-action mapping. To comprehensively integrate this task-relevant information, our framework incorporates three streamlined designs: **Which2Act** achieves object-centric grounding by predicting the visual latent of the target to suppress irrelevant distractions; **Where2Act** predicts a 2D affordance map to pinpoint precise interaction areas; and **How2Act** extracts a 3D geometric representation to guide the manipulation policy. Together, these modules progressively weave affordance priors—from coarse to fine, and from 2D to 3D—into the VLA's perception-action loop, equipping the model with a structured and deeply grounded understanding of the physical

world.

To effectively instantiate these structured affordance designs and bridge the modality gap, we build AffordanceVLA upon a Mixture-of-Transformer (MoT) [42] architecture comprising three dedicated experts: an *Understanding Expert*, an *Affordance Generation Expert*, and an *Action Expert*. This decoupled multi-expert design facilitates progressive information fusion and representation propagation from broad perception to closed-loop control. Moreover, the architecture naturally absorbs diverse multimodal knowledge from varied sources, such as VQA and robotic trajectories. Recognizing the inherent lack of dense affordance annotations in large-scale robotic datasets, we develop a robust data augmentation pipeline to synthesize these critical supervisory signals. Supported by this tool, we introduce a three-stage training strategy with a progressive data curriculum: Stage I Pre-training on referential grounding and interaction-aware scene data; Stage II Co-training with large-scale synthetic robotic data; and Stage III Post-training via target-dataset fine-tuning. Through extensive experiments on rigorous simulation benchmarks, including LIBERO [43] and CALVIN [48], alongside real-world evaluations, we demonstrate the effectiveness of our approach. AffordanceVLA achieves success rates competitive with recent strong VLAs and exhibits remarkable generalization, spatial robustness, and cross-modal alignment.

In summary, our key contributions are summarized as follows:

- We propose AffordanceVLA, a perception–prediction–action VLA framework that leverages structured affordance forecasting as intermediate supervision. By modeling Which2Act, Where2Act, and How2Act, our method integrates object grounding, 2D interaction localization, and 3D geometric reasoning to establish a more precise perception–action mapping.
- We design a MoT architecture with specialized experts and a three-stage training strategy with progressive data curriculum. This unified framework bridges perception, prediction and control, enabling a smooth transition from vision–language alignment to embodied manipulation. Furthermore, we develop a novel robotic data augmentation pipeline to overcome the lack of affordance annotations.
- We achieve strong performance in both simulation and real-world experiments. AffordanceVLA demonstrates success rates competitive with recent strong VLAs, strong generalization, and enhanced robustness, supported by comprehensive ablation, qualitative and quantitative analyses.

2 Related Work

2.1 Vision-Language-Action Models

The rapid evolution of Multimodal Large Language Models (MLLMs) [44, 54, 74] has catalyzed the development of Vision–Language–Action (VLA) models, which couple a vision–language backbone with an action module to inherit web-scale visual–linguistic priors for flexible instruction following [4, 29, 30, 45, 61, 68]. These models differ chiefly in how actions are represented and decoded: some autoregressively emit discretized action tokens [8, 29], whereas others attach a continuous diffusion or flow-matching action expert for smoother, high-frequency control [4, 24, 39, 45, 57]. To improve transfer and scalability, recent efforts pursue cross-embodiment training with latent action spaces [6, 85], spatially enhanced backbones [55], and efficient lightweight variants [57, 69]. Beyond monolithic policies, dual-system and expert-decoupled designs further disentangle high-level understanding from reactive control [14, 53, 60].

Despite their strong perceptual grounding, directly regressing actions from raw observations leaves VLAs without explicit reasoning about task-relevant scene structure, which often undermines robustness. A large body of work therefore augments VLAs with auxiliary intermediate representations, broadly falling into two families. The first treats future prediction as a *world model*, using generative visual foresight—video or image prediction [3, 5, 16, 22, 70, 82], predictive inverse dynamics [63], and unified video–action or latent-action world models [2, 7, 47, 79–81, 84, 86, 88]—to guide action generation. While visually intuitive, such dense pixel-level targets are highly redundant and their rollout is typically computationally heavy. The second family instead predicts *more compact structured cues*, such as textual rationales or chains of thought [23, 56, 75], keyposes and object pointflow [76], reconstructive or gaze-region targets [58], and implicit 3D/spatial representation alignment [35]. Notably, the latest generalist policies (*e.g.* $\pi_{0.5}$ [24] and $\pi_{0.7}$ [25]) reveal a complementary

trend: introducing train-only structured intermediate supervision (*e.g.* discrete action tokens or bounding boxes that are *not* deployed at inference), so that the low-level control objective does not erode the backbone's vision–language and instruction-following ability.

These intermediates, however, are frequently either overly redundant (dense video) or too coarse (global rationales) to capture the task-critical *what*, *where*, and *how* of manipulation. In contrast, we adopt **affordance** as a structured, task-focused intermediate representation that is simultaneously spatially grounded, semantically conditioned, and action-coupled—distinct from both video-prediction-as-bridge and latent-action world models. By employing a multi-stage training strategy, our approach integrates deep semantic understanding with world knowledge, leading to a more reliable perception–action mapping and enhanced robustness.

2.2 Affordance for Robotic Manipulation

Affordance [20] characterizes the actionable possibilities that objects offer, providing a compact prior for *what* can be manipulated and *how* to interact with it. It has been extensively studied across robotic grasping [13, 17, 32, 78], articulated-object manipulation [10, 11, 18, 41, 49, 52, 66, 71, 77], deformable-object manipulation [37, 65, 67, 72], and broader scene interaction [21, 50, 51], consistently demonstrating strong cross-task generalization. Owing to its expressive power, affordance has been incorporated into diverse learning paradigms: some methods learn it from large-scale human-video datasets or reinforcement learning [1, 19, 49, 83], while others, such as Robo-ABC [27] and RAM [33], enable training-free affordance transfer across novel objects and tasks via semantic correspondence and retrieval.

Nevertheless, most prior work represents affordances as sparse 2D/3D contact points with directions and couples them with external grasp generators [17] and motion planners [12, 59], forming open-loop pipelines that are brittle on long-horizon or contact-rich tasks. While AffordDP [73] and CoA-VLA [36] move toward closed-loop control, affordance is still largely consumed as an external cue and does not fully exploit the rich world knowledge embedded in large-scale pre-trained VLMs. In contrast, we internalize structured affordance forecasting *inside* a web-scale pre-trained VLA, jointly optimizing it with the VLM backbone and the action expert across semantic, operational, and spatial–geometric dimensions. This embedding turns affordance from an externally consumed prior into an action-coupled intermediate representation, improving reasoning, generalization, and performance across diverse manipulation scenarios.

3 Method

AffordanceVLA unifies perception, prediction, and action within a Vision-Language-Action framework by leveraging structured affordance forecasting as intermediate supervision in a Mixture-of-Transformer (MoT) architecture to learn a precise, robust, and generalizable perception action mapping for embodied manipulation. The overall pipeline is shown in Figure 2. In this section, we first present the architecture of AffordanceVLA together with its overall operational pipeline(Section 3.1). We then provide a detailed description of the three affordance knowledge prediction modules(Section 3.2). Finally, we elaborate on the training procedure of the entire framework(Section 3.3).

3.1 AffordanceVLA Architecture

As illustrated in Figure 2, the core of AffordanceVLA comprises three specialized experts: Understanding Expert, Affordance Generation Expert, and Action Expert. This Mixture-of-Transformer (MoT) architecture seamlessly bridges semantic alignment, affordance-oriented state prediction, and action execution.

3.1.1 Understanding Expert.

The Understanding Expert ($\mathcal{M}_{und}$) establishes a fine-grained alignment between visual perception and linguistic intent by leveraging pre-trained VLM priors. Given the visual observation $O_t \in \mathbb{R}^{H\times W\times 3}$ and language instruction l at time step t, $\mathcal{M}_{und}$ dynamically fuses these multimodal inputs into an instruction-aware representation: $h_t^{und} = \mathcal{M}_{und}(O_t, l)$. The proprioceptive state s_t bypasses this expert and is fed directly to $\mathcal{M}_{act}$, following the standard π_0-style decoupling. Moving beyond global feature extraction, this module

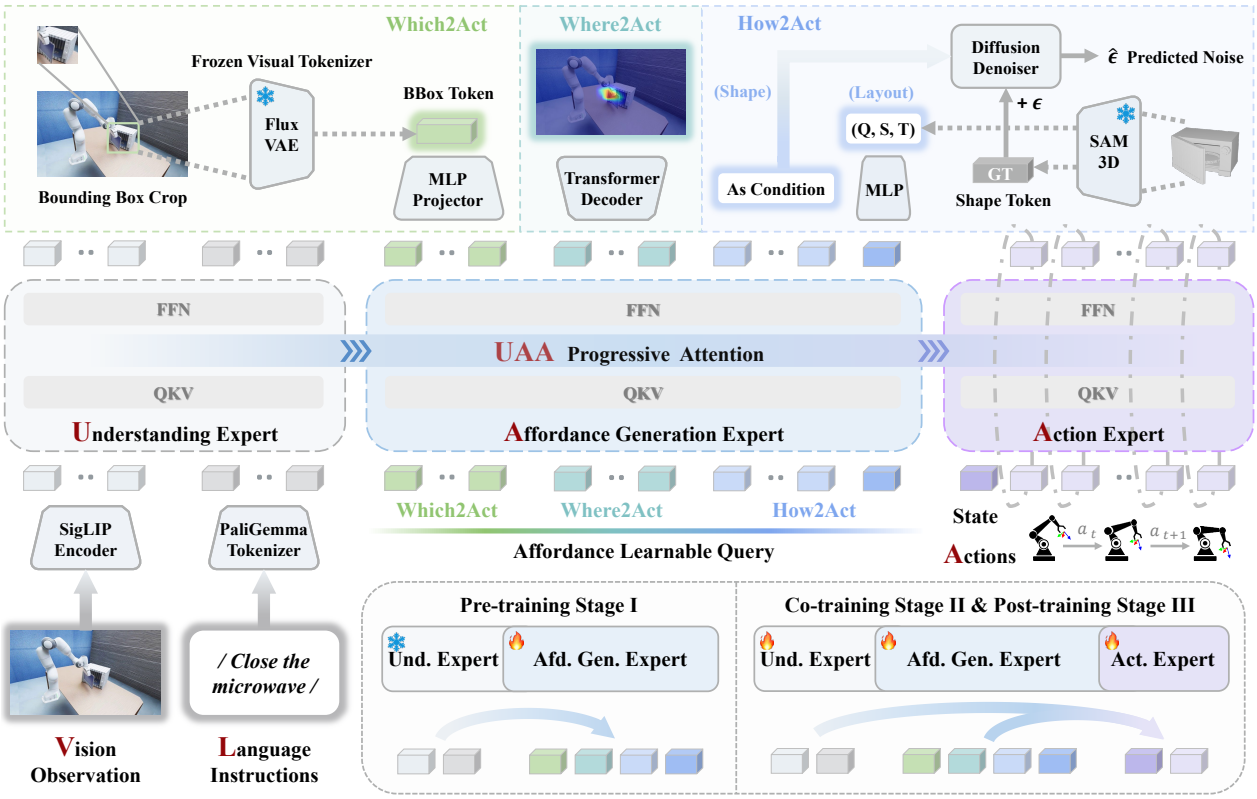


Figure 2 Pipeline. The framework employs a MoT architecture comprising three specialized experts of Understanding ($\mathcal{M}_{und}$), Affordance Generation ($\mathcal{M}_{gen}$), and Action ($\mathcal{M}_{act}$), which are coordinated via a unidirectional Understanding-Affordance-Action (UAA) progressive attention mechanism. Given an RGB observation O_t and instruction l, $\mathcal{M}_{und}$ extracts fused semantics h_t^{und}. $\mathcal{M}_{gen}$ then decodes h_t^{und} into structured affordance tokens $\hat{A}_t$ (*Which2Act*, *Where2Act*, *How2Act*) as intermediate priors. Finally, $\mathcal{M}_{act}$ synthesizes control actions $\hat{a}_{t:t+k}$ conditioned on both h_t^{und} and $\hat{A}_t$.

meticulously grounds linguistic concepts into spatial visual patches, providing a robust semantic foundation for subsequent physically grounded planning.

3.1.2 Affordance Generation Expert.

Acting as a specialized visual planner, this module generates structured affordance priors that bridge the gap between vision, language, and action. Conditioned on the instruction-aware semantics h_t^{und}, it predicts a structured representation $\hat{A}_t = \mathcal{M}_{gen}(h_t^{und})$. Rather than a generic feature, this representation anchors high-level semantic alignment into actionable geometric cues. By encapsulating spatial localization (visual latent of the target), interaction hotspots (2D affordance maps), and 3D geometric structures (shape latents), the module transforms abstract reasoning into grounded physical states. Consequently, it filters out task-irrelevant noise and provides decoupled, highly informative planning targets to assist downstream action generation.

3.1.3 Action Expert.

Conditioned on the high-level semantic context h_t^{und} and physically grounded affordance $\hat{A}_t$, the Action Expert ($\mathcal{M}_{act}$) employs a robust generative process to decode these unified representations into low-level action. It synthesizes a sequence of smooth, temporally coherent, and physically plausible action chunks: $\hat{a}_{t:t+k} = \mathcal{M}_{act}(h_t^{und}, \hat{A}_t, s_t)$. By conditioning on the intermediate affordance bridge, $\mathcal{M}_{act}$ is relieved from the burden of complex visual reasoning, allowing it to focus entirely on precise and robust physical execution.

3.1.4 Attention Mechanism.

To coordinate the heterogeneous experts efficiently, AffordanceVLA employs a structured *Understanding–Affordance–Action (UAA) progressive attention* mechanism. Within each expert, bidirectional intra-expert attention is applied to ensure thorough contextual fusion. Across modules, a strict causal inter-expert attention is enforced. Specifically, the Affordance Generation Expert queries features exclusively from the Understanding Expert (i.e., Attention$(Q_{gen}, K_{und}, V_{und})$), while the Action Expert attends to the outputs of both preceding experts. This unidirectional information flow prevents action information leakage into the prediction stage, thereby maintaining the purity of affordance features and enhancing generalization in complex environments.

3.2 Affordance Knowledge Prediction

To bridge high-level semantics and low-level control, the Affordance Generation Expert predicts structured affordance knowledge rather than monolithic global features. We disentangle learnable affordance queries into three parallel sub-modules to concurrently decode manipulation priors: *object-centric visual grounding (Which2Act)*, *fine-grained interaction localization (Where2Act)*, and *3D spatial-geometric reasoning (How2Act)*. By leveraging bidirectional attention to jointly refine their representations, this design yields task-relevant priors that naturally unify vision, language, and action.

Affordance is a natural vision–language–action bridge—spatially grounded, semantically conditioned, and action-coupled—that anchors the VLM's semantics while directly serving action generation.

3.2.1 Which2Act Forecasting.

To align semantic intents with specific visual entities, the **Which2Act** module performs object-centric grounding by localizing target bounding boxes and extracting their visual representations. Specifically, we crop the observation based on the target bounding box and extract a continuous visual latent $z_q \in \mathbb{R}^{C\times H\times W}$ using a frozen pre-trained encoder (e.g., Flux VAE [34]). The module utilizes Which2Act queries to reconstruct the predicted latent $\hat{z}$, optimizing alignment via a Mean Squared Error (MSE) loss:

$$\mathcal{L}_{\text{which}} = \frac{1}{C \cdot H \cdot W} \sum_{c,h,w} \|\hat{z}_{c,h,w} - z_{q,c,h,w}\|^2 \tag{1}$$

This auto-encoding formulation translates spatial localization into latent reconstruction, compelling the model to isolate the interacting entity while filtering out background distractions to provide a pure visual anchor.

3.2.2 Where2Act Forecasting.

The **Where2Act** module achieves fine-grained interaction localization by predicting a 2D affordance map to pinpoint interactive regions. To transform 1D query tokens into a 2D spatial distribution, we employ a lightweight Transformer decoder that treats spatial position embeddings as queries to extract interaction cues via cross-attention. The resulting features are mapped into spatial logits $\hat{y} \in \mathbb{R}^{H_t\times W_t}$ and aligned with the ground-truth mask $M \in [0,1]^{H_t\times W_t}$ using a pixel-wise Binary Cross-Entropy (BCE) loss:

$$\mathcal{L}_{\text{where}} = -\frac{1}{H_t W_t} \sum_{i=1}^{H_t W_t} \Big[M_i \log \sigma(\hat{y}_i) + (1 - M_i) \log \big(1 - \sigma(\hat{y}_i)\big) \Big] \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function. By "unfolding" compressed 1D queries into intuitive 2D hotspots, this module translates semantic intent into actionable visual affordance, providing explicit contact point guidance for low-level planning.

3.2.3 How2Act Forecasting.

The **How2Act** module performs 3D spatial–geometric reasoning by bifurcating its tokens into two synergistic branches: 3D shape generation and spatial layout regression. The shape generation branch formulates 3D

voxel prediction as a conditional diffusion process, where an iterative Transformer denoiser $\hat{\epsilon}_\theta$ generates the target's 3D shape latent, optimized via a standard noise-prediction objective:

$$\mathcal{L}_{\text{shape}} = \mathbb{E}_{t\sim\mathcal{U}(0,T),\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[\left\|\epsilon - \hat{\epsilon}_\theta(x_t, t, \bar{h}_{shape})\right\|^2\right] \tag{3}$$

Concurrently, the layout regression branch employs an MLP to regress a 10-DoF spatial layout vector $\hat{y}_{layout}$ (rotation, scale, and translation), optimized via a component-wise Smooth-L1 loss:

$$\mathcal{L}_{\text{layout}} = \frac{1}{10}\sum_{j=1}^{10} \text{SmoothL}_1(\hat{y}^{(j)}_{layout}, y^{(j)}_{layout}) \tag{4}$$

Together, these branches thoroughly characterize the target's 3D geometry and spatial posture, equipping the Action Expert with comprehensive spatial priors and kinematic constraints.

3.3 Training Strategy

To fully unlock the potential of the MoT architecture and effectively bridge semantic reasoning with physical execution, we employ a three-stage training strategy with a progressive data curriculum.

3.3.1 Stage I: General Affordance Grounding Pre-training.

The initial stage aims to endow the Affordance Generation Expert with robust spatial and geometric reasoning capabilities. We utilize broad VQA datasets—specifically Referential Grounding Data (AGD20K [46] and RefSpatial [87])—and perform an initial alignment on embodied VQA data, namely Interaction-Aware Scene Data (PRISM [15]). To preserve pre-trained semantic priors, the Vision Encoder, Understanding Expert, and Action Expert remain frozen; only the Affordance Generation Expert, learnable queries, and decoders are optimized. The model is supervised by a multi-task affordance objective:

$$\mathcal{L}_{\text{Stage1}} = \lambda_{\text{which}}\mathcal{L}_{\text{which}} + \lambda_{\text{where}}\mathcal{L}_{\text{where}} + \lambda_{\text{shape}}\mathcal{L}_{\text{shape}} + \lambda_{\text{layout}}\mathcal{L}_{\text{layout}} \tag{5}$$

where weights ($\lambda_{\text{which, where, shape}} = 0.1$, $\lambda_{\text{layout}} = 0.04$) balance optimization across physical dimensions, using native annotations augmented with SAM-3D [62] labels.

3.3.2 Stage II: Affordance-Augmented Robotic Data Co-Training.

Building upon the initial alignment, Stage II performs a rigorous re-alignment of affordance generation and robotic execution by co-training the entire AffordanceVLA on Large-Scale Synthetic Robotic Data (e.g., InternData-A1 [64]). Crucially, the perfect annotation of high-quality data maximizes the inherent power within the dataset. During this stage, the Understanding and Action Experts are unfrozen for end-to-end alignment, while the Vision Encoder is fine-tuned with a reduced learning rate. To address the scarcity of affordance labels, we engineered an automated data processing pipeline (Please refer to the **supplementary materials** for detailed content and analysis.). Specifically, we extract keyframes rule-based on action sequences [38]. A text LLM (Claude Opus 4.5) then decomposes the global instruction into per-keyframe sub-instructions, and a vision–language model (Qwen3-VL [74]) converts each keyframe into a detection category and a spatial affordance query. These guide the fine-tuned RexOmni [26] (via PRISM [15]), integrated with SAM [31] and SAM-3D [62], yielding over 100,000 affordance annotations. The joint training objective is:

$$\mathcal{L}_{\text{Stage2}} = \lambda_{\text{act}}\mathcal{L}_{\text{act}} + \lambda_{\text{afd}}\mathcal{L}_{\text{afd}} \tag{6}$$

where $\mathcal{L}_{\text{afd}}$ aggregates the losses from Stage I. We set $\lambda_{\text{act}} = 1.0$ and $\lambda_{\text{afd}} = 0.5$, slightly down-weighting the affordance objective relative to policy execution while retaining strong intermediate foresight supervision.

3.3.3 Stage III: Target Task Post-Training.

The final stage adapts the generalized VLA policy to specific downstream environments, such as the LIBERO [43] and CALVIN [48] benchmarks. Maintaining the same trainable parameters and loss formulation as Stage II (Equation (6)), but with the affordance weight further annealed to $\lambda_{\text{afd}} = 0.15$ to

prioritize precise control adaptation, this targeted post-training enables the model to deeply adapt its semantic alignment and affordance reasoning to the unique visual distributions, camera viewpoints, and kinematic constraints of the target platforms. This ensures high-fidelity, environment-aware action generation across diverse robotic settings. For real-world deployment, this stage further includes a brief post-training phase on a subset of the DROID dataset [28] to narrow the sim-to-real gap before the final in-house fine-tuning.

3.3.4 Design Philosophy.

Underlying this curriculum is a data-centric view of what supervision should carry. By enriching robotic trajectories with structured affordance labels, each demonstration encodes not merely *what* to do (the action) but also *how* to do it—which object to engage, where to interact, and the target's geometry. Crucially, because the affordance objective is anchored to the vision–language semantics rather than to raw control, it supervises the backbone with a signal that is far more informative than the action loss alone, helping the VLM retain its instruction-following ability instead of having it eroded by control-only optimization.

Rich annotations should encode not just *what* to do but *how* to do it; structured affordance supervision preserves the backbone's vision–language ability instead of eroding it under the action loss (*cf.* $\pi_{0.5}/\pi_{0.7}$).

4 Experiment

We conduct extensive experiments on both simulation benchmarks and real-world tasks to comprehensively evaluate our proposed AffordanceVLA. These experiments not only validate the model's performance but also aim to answer the following questions:

- **Q1 (Representation Strategy):** Can structured affordance forecasting (indicating which, where, and how to act) serve as a more effective intermediate representation?
- **Q2 (Architecture Design):** Does the decoupled design of the Mixture-of-Transformer (MoT) architecture, with specialized Understanding, Affordance Generation, and Action experts, effectively prevent representation collapse and outperform unified network structures?
- **Q3 (Training Paradigm):** How does the proposed three-stage progressive training strategy contribute to bridging the gap between broad visual-language pre-training and task-specific embodied control?

4.1 Baseline Comparison

4.1.1 Simulation Setup.

We compare AffordanceVLA with a wide range of baselines on LIBERO [43] and CALVIN [48], as shown in Table 1 and Table 2. To better isolate the contributions of our architecture and training paradigm, we report the performance of two model variants in our main results:

1) **AffordanceVLA (w/o stage II):** A variant that skips Stage II (affordance-augmented robotic data co-training), proceeding directly from Stage I (general affordance grounding pre-training) to Stage III (target task fine-tuning).

2) **AffordanceVLA (full):** The complete model trained with our full three-stage progressive strategy (Stage I → Stage II → Stage III).

4.1.2 LIBERO Benchmark.

As shown in Table 1, our full model attains a strong average success rate of 95.8%—the highest among the methods compared here and competitive with the best recent VLAs. This consistent performance across all four suites validates the effectiveness and robustness of the AffordanceVLA framework. Notably, even without the extensive affordance-augmented robotic data co-training phase, **AffordanceVLA (w/o stage II)** still achieves an impressive 86.2% average success rate. This directly answers **Q2**, demonstrating the inherent advantage of our MoT architecture. By decoupling the affordance generation and action processes into specialized experts,

Table 1 Quantitative Results on the LIBERO Benchmark. We report the success rates (%) across four distinct task suites: Spatial, Object, Goal, Long, along with the average success rate. All methods are evaluated over 50 rollouts. AffordanceVLA outperforms baselines with a higher average success rate. The best results are **bolded**.

Method	Spatial	Object	Goal	Long	**Average**
OpenVLA [29]	84.7	88.4	79.2	53.7	76.5
SpatialVLA [55]	88.2	89.9	78.6	55.5	78.1
CoT-VLA [82]	87.5	91.6	87.6	69.0	83.9
ThinkAct [23]	88.3	91.4	87.1	70.9	84.4
Pi0 [4]	98.0	96.8	94.4	88.4	94.4
gr00t-N1 [53]	94.4	97.6	93.0	90.6	93.9
F1-VLA [47]	98.2	97.8	95.4	**91.3**	95.7
AffordanceVLA (w/o stage II)	88.5	91.7	91.3	73.3	86.2
AffordanceVLA (full)	**98.6**	**98.4**	**96.2**	89.8	**95.8**

Table 2 Quantitative Results on CALVIN ABC→D. Success rates (%) are reported for completing 1 to 5 consecutive language-conditioned tasks over 1000 rollouts. Best results are **bolded**.

Method	Completed Tasks (%)					**Avg. Len**
	1/5	2/5	3/5	4/5	5/5	
RoboFlamingo [40]	82.4	61.9	46.6	33.1	23.5	2.48
SuSIE [3]	87.0	69.0	49.0	38.0	26.0	2.69
GR-1 [70]	85.4	71.2	59.6	49.7	40.1	3.06
OpenVLA [29]	91.3	77.8	62.0	52.1	43.5	3.27
CLOVER [5]	96.0	83.5	70.8	57.5	45.4	3.53
UniVLA [6]	95.5	85.8	75.4	66.9	56.5	3.80
Pi0 [4]	93.8	85.0	76.7	68.6	60.1	3.84
Seer [63]	94.4	87.2	79.9	72.2	64.3	3.98
VPP [22]	95.3	88.2	80.3	72.9	64.5	4.01
Seer-Large [63]	96.3	91.6	86.1	80.3	74.0	4.28
AffordanceVLA (w/o stage II)	93.4	84.7	75.4	68.1	58.9	3.81
AffordanceVLA (full)	**96.8**	**92.0**	**87.5**	**80.8**	**75.9**	**4.33**

the model effectively isolates task-relevant semantics from raw control signals, preventing the representation collapse that typically plagues VLA models under limited supervision. While AffordanceVLA excels on the Spatial, Object, and Goal suites, the margin narrows on LIBERO-Long (89.8%), suggesting that extremely long-horizon sequential tasks would further benefit from explicit long-term memory—a limitation shared across current VLAs that we revisit qualitatively in our real-world long-horizon study (Table 5).

4.1.3 CALVIN ABC→D Benchmark.

As shown in Table 2, AffordanceVLA (full) achieves strong performance with an Average Length of 4.33, completing 5 consecutive tasks in 75.9% of the rollouts—competitive among recent VLAs on this challenging zero-shot OOD protocol. This margin directly answers **Q1**: compared to redundant dense visual forecasting, our structured affordance prediction explicitly filters out superficial scene correlations. By forcing the model to focus purely on task-critical entities, interaction regions, and spatial layouts, the perception-action mapping becomes highly resilient to novel visual disturbances. Crucially, the substantial performance jump from **AffordanceVLA (w/o stage II)** (3.81) to our full model (4.33) highlights the absolute necessity of the Stage II progressive training strategy. It systematically accumulates generic world knowledge, preventing overfitting to training environments (**Q3**). Furthermore, this OOD robustness is safeguarded by the MoT architecture (**Q2**). Its attention mechanism ensures that the generalized semantic representations learned during Stage I and II are strictly protected from high-frequency control policy updates in Stage III, thereby smoothly unlocking strong generalization.

“‘latex “‘

4.2 Ablation Studies

To systematically isolate the contributions of our proposed components and answer the questions posed in Section 4, we conduct comprehensive ablation studies on the LIBERO [43] and CALVIN ABC→D [48] benchmarks. As shown in table 3, we evaluate our framework from the following three perspectives:

4.2.1 Architecture Design & Training Strategy.

First, we investigate the necessity of the Affordance Generation expert and our three-stage training strategy, **specifically addressing Q2 and Q3**. A natural concern is *where* the improvement truly originates—the high-quality data, the added supervision density, or the structured representation itself. To disentangle these competing explanations, we design a set of controls that each hold one factor fixed; we name them explicitly for clarity.

(i) Data-Only Control (No-Afd, Pi0 Arch). We train a plain Pi0 architecture on the same Stage II InternData-A1 data *without* the affordance objective or the Affordance Generation expert, isolating the effect of high-quality robotic data alone. It improves only marginally over the vanilla Pi0 baseline (LIBERO 92.4%, CALVIN 3.93 vs. Pi0's 3.84) and remains far below the full model. **Hence the gain cannot be attributed to data volume per se**—data alone is insufficient to bridge the spatial gap.

(ii) Frozen-Representation Control (Frozen-Afd). Freezing the Affordance Generation expert after Stage I—*i.e.* treating affordance as a fixed external prior fed to the Action expert—triggers a severe collapse (LIBERO 67.1%, CALVIN 2.83). This confirms that affordance must be *co-optimized* with the control policy: a statically pre-trained representation cannot adapt to the embodied control space, reproducing exactly the structural mismatch we set out to resolve, and sharply distinguishing our internalized affordance from prior external-cue pipelines.

(iii) Stage-II Ablation (w/o Stage II). Skipping the affordance-augmented co-training markedly hurts Out-of-Distribution (OOD) generalization (CALVIN 3.81), evidencing the role of large-scale, affordance-centric robotic data as an inductive bridge between broad vision–language understanding and task-specific embodied control **(Q3)**.

Taken together, controls (i) and (ii) **directly answer Q2**: it is the *decoupled, jointly-optimized* MoT design—not merely more data nor an off-the-shelf affordance module—that prevents representation collapse and unlocks the gains.

Table 3 Ablation Study on LIBERO and CALVIN ABC→D Benchmarks. We report success rates (%) and their averages on four LIBERO task suites. On CALVIN, we report the success rates for multi-step task sequences (1/5, 3/5, 5/5) and the average completed chain length (Avg. Len). The best results are in **bold**.

	LIBERO (Success Rate %)					**CALVIN ABC→D**			
Method	Spatial	Object	Goal	Long	Avg.	1/5	3/5	5/5	Avg. Len
Architecture Design & Training Strategy									
No-Afd (Pi0 Arch)	96.0	95.4	92.4	85.8	92.4	94.5	78.0	62.8	3.93
Frozen-Afd	68.0	71.1	66.4	62.9	67.1	85.3	55.9	26.3	2.83
AffordanceVLA w/o stage II	88.5	91.7	91.3	73.3	86.2	93.4	75.4	58.9	3.81
Affordance Representation									
w/o Which2Act	97.5	97.6	95.0	88.1	94.6	96.7	83.3	72.1	4.20
w/o Where2Act	95.5	96.0	93.4	88.0	93.2	96.2	81.9	69.8	4.13
w/o How2Act	96.1	96.5	93.9	88.2	93.7	95.0	79.4	65.9	4.01
Attention Mechanism for Affordance									
Block-wise Tokens	92.4	92.9	89.8	86.0	90.3	94.1	77.1	61.7	3.89
AffordanceVLA (full)	**98.6**	**98.4**	**96.2**	**89.8**	**95.8**	**96.8**	**87.5**	**75.9**	**4.33**

4.2.2 Affordance Representation.

Next, we ablate the internal affordance representation to validate our structured forecasting strategy, **aiming to answer Q1**. 1) Removing the **Which2Act** module deprives the model of explicit object-centric grounding, making it susceptible to background distractions and reducing the CALVIN average length from 4.33 to 4.20. 2) Eliminating the **Where2Act** prediction removes precise 2D interaction localization, heavily impacting tasks requiring fine-grained manipulation and dropping the LIBERO average to 93.2%. 3) Discarding the **How2Act** module strips the agent of 3D geometric and spatial-layout reasoning, causing a clear performance decay (4.01 on CALVIN) during complex 6-DoF execution. Importantly, removing any *single* head yields only a *graceful* degradation rather than a catastrophic collapse. This is direct evidence that the three sub-modules are *not* chained as a brittle Which→Where→How pipeline in which an upstream error deterministically propagates; instead, they are jointly refined under a shared instruction-aware representation and consumed *together* by the Action expert (Section 3.1). We further observe that How2Act's benefit is comparatively modest under the simple tabletop, two-finger settings of LIBERO/CALVIN, yet becomes pronounced on complex real-world 6-DoF interactions (Table 5), precisely where 3D shape and layout priors are most valuable. The progressive synergy of these three components—integrating information from coarse to fine granularity and from 2D to 3D representations—is key to our strongest results, **firmly establishing that structured affordance forecasting serves as a more effective intermediate representation (Q1).** Detailed analysis about the effectiveness of affordance subgoals can be found in the **supplementary material**.

4.2.3 Attention Mechanism for Affordance (Same-Density Control).

As mentioned in Section 3.1.4, bidirectional intra-expert attention ensures thorough contextual fusion within the Affordance Generation expert. This mechanism also enables our most pointed control for the *representation vs. supervision-density* question: the **Block-wise Tokens** variant keeps the affordance losses and the data—and therefore the total *supervision density*—exactly identical, and only replaces the bidirectional attention with a causal scheme so that the three modules (Which2Act, Where2Act, and How2Act) are strictly prohibited from cross-attending to each other. This degenerates the organic affordance representation into three independent, equally-dense auxiliary tasks. As shown in Table 3, this single change causes a sharp decline to 90.3% on LIBERO and 3.89 on CALVIN. **Because the supervision density is held constant, this directly rules out the "multi-task density" explanation**: what drives the gain is the structured, jointly-refined representation, not the mere addition of more loss terms. The drop further reveals a strong implicit dependency among the affordance dimensions—robust spatial localization relies on accurate object grounding, and 3D structural reasoning builds upon 2D interaction points. The shared attention lets the three heads "promote each other", yielding a unified and coherent perception–prediction representation.

4.2.4 Data Efficiency.

We evaluate data efficiency by scaling the downstream fine-tuning data from 10% to 100%; the four training configurations under comparison are summarized in Table 4, and the corresponding curves are reported in Figure 3.

Initial distribution shift. At the extreme low-data regime (10%), the vanilla **Pi0** starts strong thanks to its original

Table 4 Training Configurations of Methods on Data Efficiency Analysis.

Method	Stage I	Stage II	Stage III
Pi0 (Finetuned)	×	×	Phased (5k∼Full)
No-Afd (Pi0 Arch)	×	Action loss Only	Phased (5k∼Full)
AffordanceVLA (w/o stage II)	VQA Afford.	×	Phased (5k∼Full)
AffordanceVLA (full)	VQA Afford.	Act loss + Afd loss	Phased (5k∼Full)

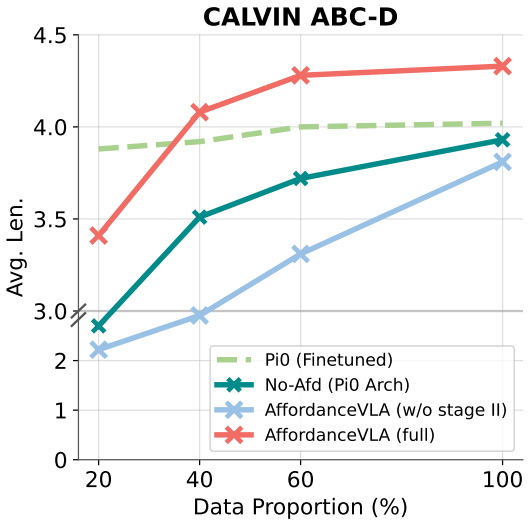


Figure 3 Data Efficiency Curve.

pre-trained weights, whereas all models that undergo specialized pre-training—**No-Afd**, **AffordanceVLA (w/o stage II)**, and the full **AffordanceVLA**—begin lower. This dip is an expected consequence of the temporary weight misalignment introduced by their respective pre-training stages.

Why AffordanceVLA surges. Despite this initial shift, the full model exhibits a rapid surge and superior sample efficiency: with only 40% of the fine-tuning data, it already attains ∼92% on LIBERO and an average chain length above 4.0 on CALVIN, shattering the ceiling of the fully fine-tuned **Pi0**. We attribute this to the structured affordance representation (Which2Act + Where2Act + How2Act), which decomposes the perception–action mapping into interpretable sub-problems, so that each additional sample supervises not only the final action but also object grounding, spatial localization, and 3D geometric reasoning—effectively multiplying the learning signal per sample.

Ablation trajectories. The recovery trajectories of the ablations are equally telling. **No-Afd** recovers slowly and struggles to clearly surpass the original **Pi0**, indicating that robotic pre-training data alone offers limited downstream scalability without affordance structure; **AffordanceVLA (w/o stage II)** suffers the most severe shift and the slowest recovery, reaffirming Stage II as an indispensable bridge between broad vision–language alignment and sample-efficient embodied adaptation.

The takeaway. Rather than reading this as a deficiency of any baseline, we read it as a statement about *representation*: the datasets themselves are not the bottleneck—the representation is. Architectural innovation that structurally grounds manipulation through affordances is what converts additional data into proportional gains, so that architecture, representation, and data quality become mutually amplifying rather than mutually substitutable.

Architecture unlocks data potential: Pi0 saturates while we break its ceiling at 40% data—architecture × representation × data are mutually amplifying.

4.3 Real-world Experiments

4.3.1 Basic Task Performance.

To bridge the sim-to-real gap, our policies are initially pre-trained on the part of DROID dataset [28] and detailed experimental setups are provided in the **supplementary material**. As shown in Table 5, AffordanceVLA exhibits strong generalizability in *Basic Tasks*, achieving an average success rate of 88.3% and consistently outperforming the Pi0 [4] baseline across diverse objects, spatial relations, and semantic categories (e.g., color, shape).

4.3.2 Instruction Sensitivity via Affordance Grounding.

To verify the sensitivity of our V–L bridge under severe visual aliasing, we evaluate *Complex Tasks* like *Drawer* and *Toaster* (Fig. 4). Given identical initial visual observations but distinct instructions (*e.g.* "pick" vs. "close"), AffordanceVLA decisively outperforms Pi0. For instance, our model achieves 86.7% and 100.0% success rates in the *Drawer (pick)* and *Drawer (close)* tasks, compared to Pi0's 46.7% and 40.0%, and reaches an aggregate complex-task average of 82.9% versus Pi0's 44.8%. Crucially, because the two policies share identical visual inputs and differ *only* in language, this gap isolates instruction-following fidelity from visual perception. It thus provides concrete empirical support for our hypothesis (discussed below) that affordance supervision preserves the VLM's vision–language ability rather than letting it be eroded by the action loss: by unambiguously grounding concise linguistic intents into highly localized affordance heatmaps, our model reliably disambiguates the required action.

4.3.3 Emergent Long-Horizon Execution via Intent Enrichment.

The continuous *Pick all the rubbish* task demonstrates how Affordance Subgoals compensate for the lack of explicit long-term planning architectures. By dynamically re-evaluating the workspace, our single-frame model generates sequential visual targets to translate broad instructions into continuous execution. While Pi0's performance sharply degrades by the 3rd execution (6.7%), AffordanceVLA maintains a robust 46.7% success rate. Furthermore, our affordance-driven approach drastically improves execution efficiency, reducing

Table 5 Results on Real-world Tasks. We report the average success ratio (%) over 15 trials per task. Tasks are categorized into Basic and Complex scenarios. The best results are denoted in **bold**.

Method	Task-specific Success Rates								Average
Basic Real-world Tasks									
	Close		Pick up (Color)		Pick up (Shape)		Pick up		
	microwave	safe	red	green	duck	banana	flower	bear	
Pi0 [4]	86.7	86.7	80.0	80.0	26.7	73.3	53.3	80.0	70.8
AffordanceVLA	**93.3**	**100.0**	**86.7**	**80.0**	**86.7**	**86.7**	**80.0**	**93.3**	**88.3**
Complex Real-world Tasks									
	Drawer		Toaster		Pick all the rubbish				
	pick	close	pick	toast	1st (↑)	2nd (↑)	3rd (↑)	Empty (↓)	
Pi0 [4]	46.7	40.0	46.7	26.7	93.3	53.3	6.7	33	44.8
AffordanceVLA	**86.7**	**100.0**	**80.0**	**86.7**	**100.0**	**80.0**	**46.7**	**11**	**82.9**

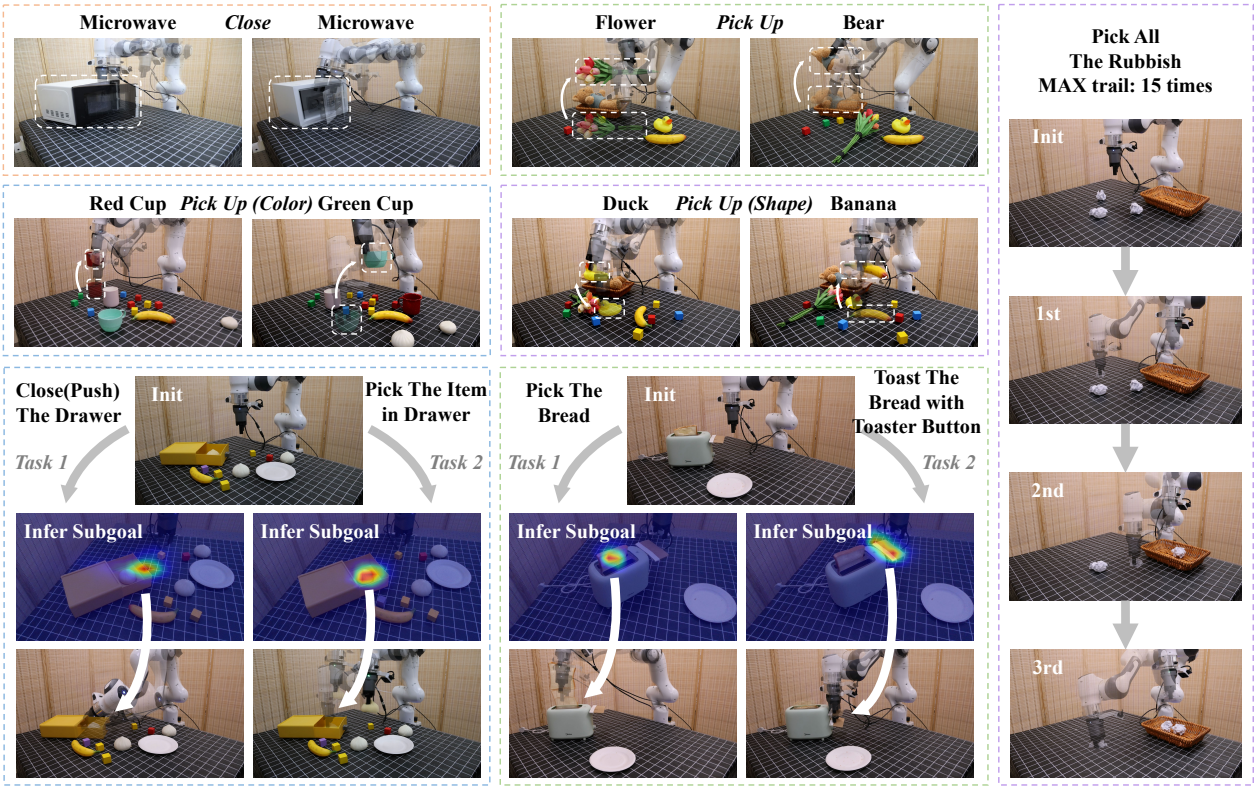


Figure 4 Real-World Experiment Visualizations. Top: Qualitative results for Basic tasks. **Bottom Left:** Visualizations of Where2Act token in the *Drawer* and *Toaster* tasks. **Right:** Sequential execution of the continuous *Pick all the rubbish* task.

total redundant actions (Empty Picks) to 11 versus Pi0's 33. This intent enrichment empowers the model to naturally exhibit long-horizon capabilities.

4.3.4 Why Does Affordance Help? A Discussion.

A closer look at the failure modes is revealing. In the *Toaster* task, Pi0 performs poorly (toast 26.7% vs. our 86.7%), and its bad cases concentrate almost entirely at the button-pressing step: instead of extending to press the button, it frequently *closes the gripper* as if still executing a pick-and-place, largely disregarding the "press the button" instruction. This exposes that Pi0, even after real-trajectory fine-tuning, still suffers from weak instruction following—its behavior is driven by the dominant action prior rather than by the language command.

We offer an intuition—admittedly a hypothesis rather than a proven mechanism—for why affordance grounding alleviates this. In an action-only VLA such as Pi0, the low-level action loss is back-propagated directly into the VLM backbone; this signal is arguably neither sufficiently informative nor well-aligned with the VLM's vision–language semantics, and may gradually erode the instruction-following ability that large-scale pre-training endowed. Recent strong generalist policies appear to mitigate this implicitly: $\pi_{0.5}$ [24] and $\pi_{0.7}$ [25] introduce structured intermediate supervision (*e.g.* discrete action tokens or bounding boxes) that is used only during training and *not* decoded at deployment, plausibly acting as a semantic anchor that keeps the backbone from drifting under the control objective. We conjecture that affordance plays a similar—and arguably more natural—role: as an intrinsic vision–language–action bridge, its training signal stays close to the VLM's semantic space, anchoring the backbone while still directly serving action. This is consistent with our *Drawer* and *Toaster* results, where AffordanceVLA responds correctly to the language command under identical visual observations while Pi0 does not. We stress that this anchoring account is an interpretive hypothesis meant to guide intuition, not a claim backed by direct mechanistic evidence.

Hypothesis. We conjecture that affordance acts as a structured semantic anchor: rather than letting the low-level action loss reshape the VLM directly, the affordance objective—being close to vision–language semantics—helps preserve the backbone's instruction-following ability, in spirit with the train-only intermediate cues adopted by recent strong VLAs.

5 Conclusion

In this work, we present AffordanceVLA to bridge the structural gap between the semantic space of Vision-Language Models and the 3D physical requirements of embodied control. Instead of relying on direct end-to-end mappings or redundant visual foresight, our framework adopts affordances as a task-oriented intermediate representation and decomposes affordance forecasting into Which2Act, Where2Act, and How2Act. With a Mixture-of-Transformer architecture and a progressive data curriculum, AffordanceVLA achieves strong, competitive performance on LIBERO, CALVIN, and real-world experiments, demonstrating strong generalization and robust reasoning. Future work will explore explicit temporal modeling as well as extensions to bimanual and deformable object manipulation.

References

[1] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics, 2023. https://arxiv.org/abs/2304.08488.

[2] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model, 2025. https://arxiv.org/abs/2512.13030.

[3] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models, 2023. https://arxiv.org/abs/2310.10639.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. https://arxiv.org/abs/2410.24164.

[5] Qingwen Bu, Jia Zeng, Li Chen, Yanchao Yang, Guyue Zhou, Junchi Yan, Ping Luo, Heming Cui, Yi Ma, and Hongyang Li. Closed-loop visuomotor control with generative expectation for robotic manipulation, 2024. https://arxiv.org/abs/2409.09016.

[6] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions, 2025. https://arxiv.org/abs/2505.06111.

[7] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. Worldvla: Towards autoregressive action world model, 2025. https://arxiv.org/abs/2506.21539.

[8] Chilam Cheang, Sijin Chen, Zhongren Cui, Yingdong Hu, Liqun Huang, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Xiao Ma, Hao Niu, Wenxuan Ou, Wanli Peng, Zeyu Ren, Haixin Shi, Jiawen Tian, Hongtao Wu, Xin Xiao, Yuyang Xiao, Jiafeng Xu, and Yichu Yang. Gr-3 technical report, 2025. https://arxiv.org/abs/2507.15493.

[9] Tianxing Chen, Yuran Wang, Mingleyang Li, Yan Qin, Hao Shi, Zixuan Li, Yifan Hu, Yingsheng Zhang, Kaixuan Wang, Yue Chen, Hongcheng Wang, Renjing Xu, Ruihai Wu, Yao Mu, Yaodong Yang, Hao Dong, and Ping Luo. Rmbench: Memory-dependent robotic manipulation benchmark with insights into policy design, 2026. https://arxiv.org/abs/2603.01229.

[10] Yue Chen, Chenrui Tie, Ruihai Wu, and Hao Dong. Eqvafford: Se(3) equivariance for point-level affordance learning, 2024. https://arxiv.org/abs/2408.01953.

[11] Yue Chen, Muqing Jiang, Kaifeng Zheng, Jiaqi Liang, Chenrui Tie, Haoran Lu, Ruihai Wu, and Hao Dong. Learning part-aware dense 3d feature field for generalizable articulated object manipulation, 2026. https://arxiv.org/abs/2602.14193.

[12] David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study, 2014. https://arxiv.org/abs/1404.3785.

[13] Enric Corona, Albert Pumarola, Guillem Alenya, Francesc Moreno-Noguer, and Grégory Rogez. Ganhand: Predicting human grasp affordances in multi-object scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5031–5041, 2020.

[14] Can Cui, Pengxiang Ding, Wenxuan Song, Shuanghao Bai, Xinyang Tong, Zirui Ge, Runze Suo, Wanqi Zhou, Yang Liu, Bofang Jia, Han Zhao, Siteng Huang, and Donglin Wang. Openhelix: A short survey, empirical analysis, and open-source dual-system vla model for robotic manipulation, 2025. https://arxiv.org/abs/2505.03912.

[15] Abhay Deshpande, Yuquan Deng, Arijit Ray, Jordi Salvador, Winson Han, Jiafei Duan, Kuo-Hao Zeng, Yuke Zhu, Ranjay Krishna, and Rose Hendrix. Graspmolmo: Generalizable task-oriented grasping via large-scale synthetic data generation, 2025. https://arxiv.org/abs/2505.13441.

[16] Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation, 2023. https://arxiv.org/abs/2302.00111.

[17] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains, 2023. https://arxiv.org/abs/2212.08333.

[18] Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Act the part: Learning interaction strategies for articulated object part discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15752–15761, October 2021.

[19] Yiran Geng, Boshi An, Haoran Geng, Yuanpei Chen, Yaodong Yang, and Hao Dong. End-to-end affordance learning for robotic manipulation, 2022. https://arxiv.org/abs/2209.12941.

[20] James Jerry Gibson. The theory of affordances. 1977. https://api.semanticscholar.org/CorpusID:60688620.

[21] Mohammed Hassanin, Salman Khan, and Murat Tahtali. Visual affordance and function understanding: A survey. *ACM Computing Surveys (CSUR)*, 54(3):1–35, 2021.

[22] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations, 2025. https://arxiv.org/abs/2412.14803.

[23] Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning, 2025. https://arxiv.org/abs/2507.16815.

[24] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. https://arxiv.org/abs/2504.16054.

[25] Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities, 2026. https://arxiv.org/abs/2604.15483.

[26] Qing Jiang, Junan Huo, Xingyu Chen, Yuda Xiong, Zhaoyang Zeng, Yihao Chen, Tianhe Ren, Junzhi Yu, and Lei Zhang. Detect anything via next point prediction, 2025. https://arxiv.org/abs/2510.12798.

[27] Yuanchen Ju, Kaizhe Hu, Guowei Zhang, Gu Zhang, Mingrun Jiang, and Huazhe Xu. Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation, 2024. https://arxiv.org/abs/2401.07487.

[28] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu,

Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset, 2025. https://arxiv.org/abs/2403.12945.

[29] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. https://arxiv.org/abs/2406.09246.

[30] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. https://arxiv.org/abs/2502.19645.

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. https://arxiv.org/abs/2304.02643.

[32] Mia Kokic, Danica Kragic, and Jeannette Bohg. Learning task-oriented grasping from human activity datasets, 2020. https://arxiv.org/abs/1910.11669.

[33] Yuxuan Kuang, Junjie Ye, Haoran Geng, Jiageng Mao, Congyue Deng, Leonidas Guibas, He Wang, and Yue Wang. Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation, 2024. https://arxiv.org/abs/2407.04689.

[34] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. https://arxiv.org/abs/2506.15742.

[35] 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, 2025. https://arxiv.org/abs/2510.12276.

[36] Jinming Li, Yichen Zhu, Zhibin Tang, Junjie Wen, Minjie Zhu, Xiaoyu Liu, Chengmeng Li, Ran Cheng, Yaxin Peng, Yan Peng, and Feifei Feng. Coa-vla: Improving vision-language-action models via visual-textual chain-of-affordance, 2025. https://arxiv.org/abs/2412.20451.

[37] Mingleyang Li, Yuran Wang, Yue Chen, Tianxing Chen, Jiaqi Liang, Zishun Shen, Haoran Lu, Ruihai Wu, and Hao Dong. Garmentpile++: Affordance-driven cluttered garments retrieval with vision-language reasoning, 2026. https://arxiv.org/abs/2603.04158.

[38] Peiyan Li, Yixiang Chen, Hongtao Wu, Xiao Ma, Xiangnan Wu, Yan Huang, Liang Wang, Tao Kong, and Tieniu Tan. Bridgevla: Input-output alignment for efficient 3d manipulation learning with vision-language models, 2025. https://arxiv.org/abs/2506.07961.

[39] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation, 2024. https://arxiv.org/abs/2411.19650.

[40] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators, 2024. https://arxiv.org/abs/2311.01378.

[41] Jiaqi Liang, Yue Chen, Qize Yu, Yan Shen, Haipeng Zhang, Hao Dong, and Ruihai Wu. A3d: Adaptive affordance assembly with dual-arm manipulation, 2026. https://arxiv.org/abs/2601.11076.

[42] Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models, 2025. https://arxiv.org/abs/2411.04996.

[43] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

[44] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. https://arxiv.org/abs/2304.08485.

[45] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation, 2025. https://arxiv.org/abs/2410.07864.

[46] Hongchen Luo, Wei Zhai, Jing Zhang, Yang Cao, and Dacheng Tao. Grounded affordance from exocentric view, 2023. https://arxiv.org/abs/2208.13196.

[47] Qi Lv, Weijie Kong, Hao Li, Jia Zeng, Zherui Qiu, Delin Qu, Haoming Song, Qizhi Chen, Xiang Deng, and Jiangmiao Pang. F1: A vision-language-action model bridging understanding and generation to actions, 2025. https://arxiv.org/abs/2509.06951.

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

[49] Kaichun Mo, Leonidas Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects, 2021. https://arxiv.org/abs/2101.02692.

[50] Tushar Nagarajan and Kristen Grauman. Learning affordance landscapes for interaction exploration in 3d environments. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[51] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *CVPR*, 2020.

[52] Chuanruo Ning, Ruihai Wu, Haoran Lu, Kaichun Mo, and Hao Dong. Where2explore: Few-shot affordance learning for unseen novel categories of articulated objects, 2023. https://arxiv.org/abs/2309.07473.

[53] NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. https://arxiv.org/abs/2503.14734.

[54] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick

Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. https://arxiv.org/abs/2303.08774.

[55] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model, 2025. https://arxiv.org/abs/2501.15830.

[56] Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models, 2025. https://arxiv.org/abs/2502.19417.

[57] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. https://arxiv.org/abs/2506.01844.

[58] 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, 2025. https://arxiv.org/abs/2508.10333.

[59] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. curobo: Parallelized collision-free minimum-jerk robot motion generation, 2023. https://arxiv.org/abs/2310.17274.

[60] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, Steven Bohez, Konstantinos Bousmalis, Anthony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Oscar Chang, Jose Enrique Chen, Xi Chen, Hao-Tien Lewis Chiang, Krzysztof Choromanski, David D'Ambrosio, Sudeep Dasari, Todor Davchev, Coline Devin, Norman Di Palo, Tianli Ding, Adil Dostmohamed, Danny Driess, Yilun Du, Debidatta Dwibedi, Michael Elabd, Claudio Fantacci, Cody Fong, Erik Frey, Chuyuan Fu, Marissa Giustina, Keerthana Gopalakrishnan, Laura Graesser, Leonard Hasenclever, Nicolas Heess, Brandon Hernaez, Alexander Herzog, R. Alex Hofer, Jan Humplik, Atil Iscen, Mithun George Jacob, Deepali Jain, Ryan Julian, Dmitry Kalashnikov, M. Emre Karagozler, Stefani Karp, Chase Kew, Jerad Kirkland, Sean Kirmani, Yuheng Kuang, Thomas Lampe, Antoine Laurens, Isabel Leal, Alex X. Lee, Tsang-Wei Edward Lee, Jacky Liang, Yixin Lin, Sharath Maddineni, Anirudha Majumdar, Assaf Hurwitz Michaely, Robert Moreno, Michael Neunert, Francesco Nori, Carolina Parada, Emilio Parisotto, Peter Pastor, Acorn Pooley, Kanishka Rao, Krista Reymann, Dorsa Sadigh, Stefano Saliceti, Pannag Sanketi, Pierre Sermanet, Dhruv Shah, Mohit Sharma, Kathryn Shea, Charles Shu, Vikas Sindhwani, Sumeet Singh, Radu Soricut, Jost Tobias Springenberg, Rachel Sterneck, Razvan Surdulescu, Jie Tan, Jonathan Tompson, Vincent Vanhoucke, Jake Varley, Grace Vesom, Giulia Vezzani, Oriol Vinyals, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Fei Xia, Ted Xiao, Annie Xie, Jinyu Xie, Peng Xu, Sichun Xu, Ying Xu, Zhuo Xu, Yuxiang Yang, Rui Yao, Sergey Yaroshenko, Wenhao Yu, Wentao Yuan, Jingwei Zhang, Tingnan Zhang, Allan Zhou, and Yuxiang Zhou. Gemini robotics: Bringing ai into the physical world, 2025. https://arxiv.org/abs/2503.20020.

[61] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy, 2024. https://arxiv.org/abs/2405.12213.

[62] SAM 3D Team, Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jiawei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. Sam 3d: 3dfy anything in images, 2025. https://arxiv.org/abs/2511.16624.

[63] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation, 2024. https://arxiv.org/abs/2412.15109.

[64] Yang Tian, Yuyin Yang, Yiman Xie, Zetao Cai, Xu Shi, Ning Gao, Hangxu Liu, Xuekun Jiang, Zherui Qiu, Feng Yuan, Yaping Li, Ping Wang, Junhao Cai, Jia Zeng, Hao Dong, and Jiangmiao Pang. Interndata-a1: Pioneering high-fidelity synthetic data for pre-training generalist policy, 2025. https://arxiv.org/abs/2511.16651.

[65] Chenrui Tie, Yue Chen, Ruihai Wu, Boxuan Dong, Zeyi Li, Chongkai Gao, and Hao Dong. Et-seed: Efficient trajectory-level se(3) equivariant diffusion policy, 2025. https://arxiv.org/abs/2411.03990.

[66] Yuanfei Wang, Xiaojie Zhang, Ruihai Wu, Yu Li, Yan Shen, Mingdong Wu, Zhaofeng He, Yizhou Wang, and Hao Dong. Adamanip: Adaptive articulated object manipulation environments and policy learning, 2025. https://arxiv.org/abs/2502.11124.

[67] Yuran Wang, Ruihai Wu, Yue Chen, Jiarui Wang, Jiaqi Liang, Ziyu Zhu, Haoran Geng, Jitendra Malik, Pieter Abbeel, and Hao Dong. Dexgarmentlab: Dexterous garment manipulation environment with generalizable policy, 2025. https://arxiv.org/abs/2505.11032.

[68] Junjie Wen, Minjie Zhu, Yichen Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Chengmeng Li, Xiaoyu Liu, Yaxin Peng, Chaomin Shen, and Feifei Feng. Diffusion-vla: Generalizable and interpretable robot foundation model via self-generated reasoning, 2025. https://arxiv.org/abs/2412.03293.

[69] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation, 2025. https://arxiv.org/abs/2409.12514.

[70] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation, 2023. https://arxiv.org/abs/2312.13139.

[71] Ruihai Wu, Yan Zhao, Kaichun Mo, Zizheng Guo, Yian Wang, Tianhao Wu, Qingnan Fan, Xuelin Chen, Leonidas Guibas, and Hao Dong. Vat-mart: Learning visual action trajectory proposals for manipulating 3d articulated objects, 2022. https://arxiv.org/abs/2106.14440.

[72] Ruihai Wu, Ziyu Zhu, Yuran Wang, Yue Chen, Jiarui Wang, and Hao Dong. Garmentpile: Point-level visual affordance guided retrieval and adaptation for cluttered garments manipulation, 2025. https://arxiv.org/abs/2503.09243.

[73] Shijie Wu, Yihang Zhu, Yunao Huang, Kaizhen Zhu, Jiayuan Gu, Jingyi Yu, Ye Shi, and Jingya Wang. Afforddp: Generalizable diffusion policy with transferable affordance, 2025. https://arxiv.org/abs/2412.03142.

[74] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. https://arxiv.org/abs/2505.09388.

[75] Shuai Yang, Hao Li, Yilun Chen, Bin Wang, Yang Tian, Tai Wang, Hanqing Wang, Feng Zhao, Yiyi Liao, and Jiangmiao Pang. Instructvla: Vision-language-action instruction tuning from understanding to manipulation, 2025. https://arxiv.org/abs/2507.17520.

[76] Yuyin Yang, Zetao Cai, Yang Tian, Jia Zeng, and Jiangmiao Pang. Gripper keypose and object pointflow as interfaces for bimanual robotic manipulation, 2025. https://arxiv.org/abs/2504.17784.

[77] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024.

[78] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo Grau, Nima Fazeli, Ferran Alet, Nikhil Dafle, Rachel Holladay, Isabella Morena, Prem Nair, Druck Green, Ian Taylor, Weber Liu, and Alberto Rodriguez. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. pages 1–8, 05 2018. doi: 10.1109/ICRA.2018.8461044.

[79] Hongyin Zhang, Zifeng Zhuang, Han Zhao, Pengxiang Ding, Hongchao Lu, and Donglin Wang. Reinbot: Amplifying robot visual-language manipulation with reinforcement learning, 2025. https://arxiv.org/abs/2505.07395.

[80] Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent, 2025. https://arxiv.org/abs/2501.18867.

[81] Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge, 2025. https://arxiv.org/abs/2507.04447.

[82] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models, 2025. https://arxiv.org/abs/2503.22020.

[83] Yan Zhao, Ruihai Wu, Zhehuan Chen, Yourong Zhang, Qingnan Fan, Kaichun Mo, and Hao Dong. Dualafford: Learning collaborative visual affordance for dual-gripper manipulation, 2023. https://arxiv.org/abs/2207.01971.

[84] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model, 2024. https://arxiv.org/abs/2403.09631.

[85] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model, 2025. https://arxiv.org/abs/2510.10274.

[86] Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Tianran Zhang, Wenxuan Song, Jiayi Chen, Xinhu Zheng, Hesheng Wang, and Haoang Li. Flowvla: Visual chain of thought-based motion reasoning for vision-language-action models, 2025. https://arxiv.org/abs/2508.18269.

[87] Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, and Shanghang Zhang. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics, 2026. https://arxiv.org/abs/2506.04308.

[88] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets, 2025. https://arxiv.org/abs/2504.02792.

Appendix

Outline In this supplementary material, we provide additional details and extra experimental results to support the main manuscript:

1. **Section A: The Effectiveness of Affordance Subgoals**
    - **Section A.1**: Quantitative Validation of Subgoal Representations
    - **Section A.2**: Representation Decoupling: Backbone vs. Decoder
    - **Section A.3**: Qualitative Analysis of Affordance Grounding
2. **Section B: Model Variants and Design Choice Details**
    - **Section B.1**: Design Insight
    - **Section B.2**: Which2Act Redesign
    - **Section B.3**: Model Variants
3. **Section C: Data-Centric Methodology**
    - **Section C.1**: Data Quality as the Performance Ceiling
    - **Section C.2: Affordance Annotation Pipeline**
4. **Section D: Dataset Details**
5. **Section E: Training Details**
6. **Section F: Inference Latency**
7. **Section G: Real-World Experiments Details and Extra Experiments**

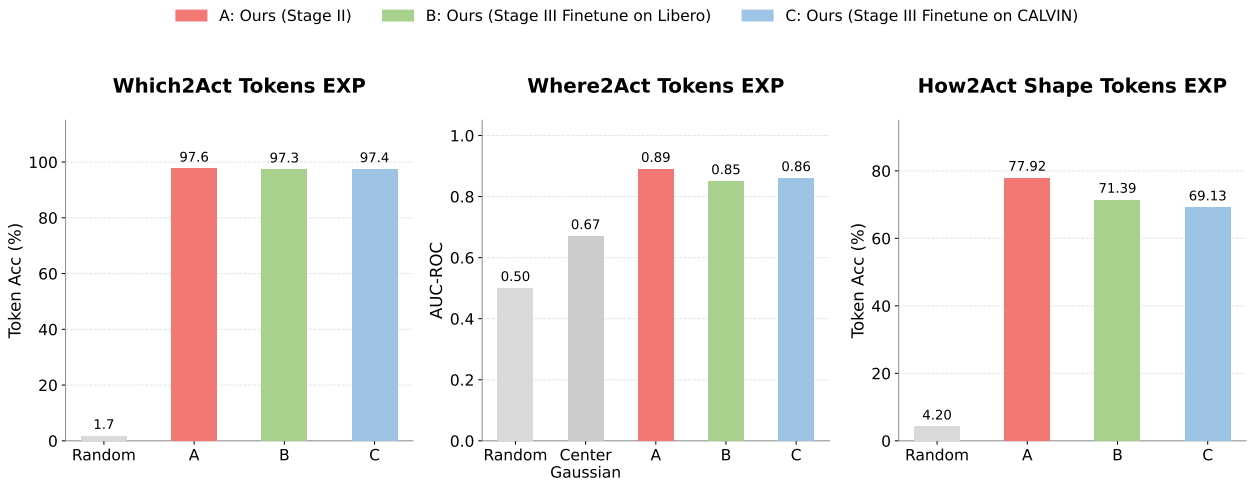
A The Effectiveness of Affordance Subgoals

In this section, we comprehensively evaluate the proposed affordance subgoals, Which2Act, Where2Act, and How2Act, to verify two core hypotheses: first, the three Affordance Query tokens successfully extract crucial task-relevant information (i.e., each predictive head functions effectively); second, these learned representations genuinely provide substantial guidance for downstream action generation. All quantitative evaluations are conducted on the Unseen PRISM subset, comprising 1,000 validation samples excluded from the Stage I training phase.

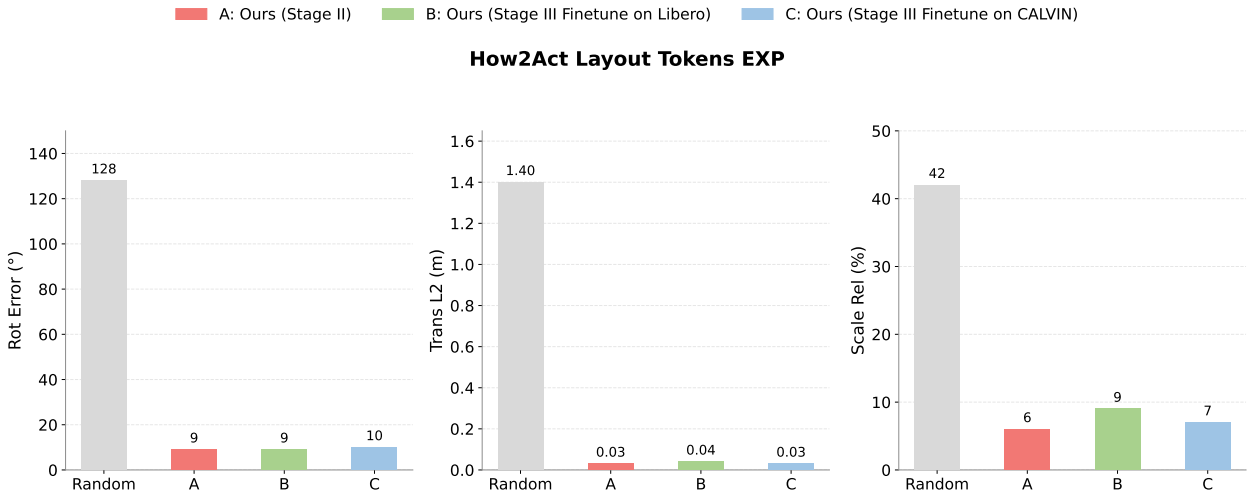
A.1 Quantitative Validation of Subgoal Representations

To ascertain that the three subgoals successfully capture the requisite semantic and geometric information, we adopt a set of targeted metrics, evaluated on the Prev (VQ-VAE) variant (Table 6). For Which2Act and How2Act Shape, we utilize *Token Accuracy* (Token Acc), which calculates the mean element-wise match between the predicted logits and the ground truth codebook indices. For Where2Act, we report the *AUC-ROC* score. Unlike MSE or KLD, AUC-ROC is threshold-free and robust against the vast zero-value regions typical of sparse backgrounds, making it well-suited for affordance prediction. For the How2Act Layout, we measure the *Rotation Angular Error* (Rot Error) using the geodesic angular distance, the *Translation L2 Distance* (Trans L2) in meters, and the *Scale Relative Error* (Scale Rel) as a percentage.

As illustrated in Fig. 5a and Fig. 5b, all three affordance query tokens successfully extract crucial environmental context, substantially outperforming random baselines. Notably, while Which2Act and Where2Act achieve near-perfect performance, the absolute accuracy for the How2Act Shape token is comparatively lower. This aligns with the intrinsic difficulty of reconstructing high-fidelity 3D voxels from highly compressed tokens within the VLA backbone.



(a) Quantitative Results for Subgoal Tokens.



(b) Quantitative Results for How2Act Layout Tokens.

Figure 5 Comprehensive Quantitative Evaluation of Subgoal Tokens. Our models demonstrate effective acquisition of task-relevant semantics and 3D spatial reasoning capabilities across all metrics.

However, this limitation does not impede downstream control. The generated 3D representations sufficiently capture essential task progression characteristics—such as the coarse physical structure of the manipulated object and the anticipated interaction modality—serving as effective explicit planning targets for the Action Expert. Furthermore, our **extra ablation studies** reveal that without Which2Act and Where2Act, the model experiences a severe performance degradation, with the LIBERO average success rate dropping to 11.8%. This underscores that Which2Act and Where2Act provide critical implicit object-centric priors (functioning similarly to bounding boxes or segmentation masks in traditional explicit 3D models) that facilitate the learning of How2Act. Conversely, the successful convergence of How2Act further corroborates that Which2Act and Where2Act have indeed learned robust grounding capabilities. This reciprocal relationship strongly validates our architectural design, demonstrating that the three tokens operate synergistically to comprehend the manipulation scene.

A.2 Representation Decoupling: Backbone vs. Decoder

To confirm that the subgoals genuinely facilitate action generation rather than merely overfitting a high-capacity decoder, it is imperative to verify that the subgoal losses primarily optimize the backbone representations. To

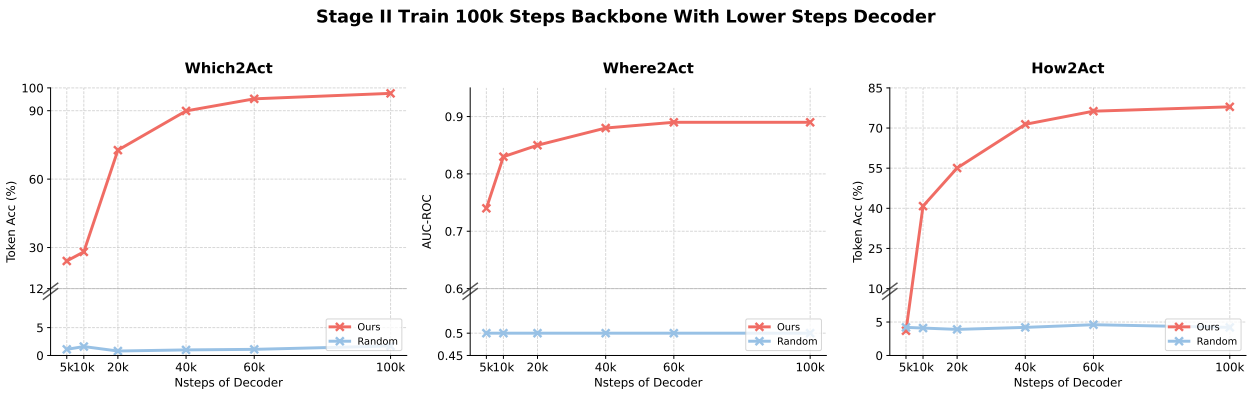


Figure 6 Backbone Representation Evaluation. Performance of a fully trained VLA backbone (100k steps) when evaluated with decoders trained for fewer steps (5k to 100k). The backbone maintains robust feature extraction even when paired with a weakly trained decoder.

Figure 7 Qualitative Visualization of Where2Act Affordance Maps. The model demonstrates robust Vision-Language alignment, dynamically adjusting its affordance predictions based on varying language instructions given the identical visual observation.

this end, we design a decoupling experiment: we freeze our Stage II backbone (trained for 100k steps) and evaluate its feature extraction quality using decoders trained with varying, lower step counts (from 5k to 100k steps).

As shown in Fig. 6, even when paired with an under-trained decoder, the fully trained backbone still extracts highly meaningful affordance features. The performance exhibits a stable, monotonic increase as the decoder steps align with the backbone, rather than catastrophically failing under the weak decoder's evaluation. This confirms that the VLA backbone itself intrinsically assimilates the affordance representations to guide actions.

A.3 Qualitative Analysis of Affordance Grounding

To intuitively illustrate the model's grounding capabilities, we visualize the Where2Act Affordance Maps in Fig. 7.

We specifically design a Vision-Language (V-L) bridging experiment: given an identical visual observation, we alter the language instruction (L). The generated affordance map dynamically adapts to L, indicating that our model deeply comprehends V-L alignment rather than merely relying on visual salience biases. Furthermore, the model exhibits strong robustness; semantically similar commands consistently yield highly correlated affordance heatmaps, whereas divergent commands elicit clean and precise shifts in the affordance focus.

B Model Variants and Design Choice Details

B.1 Design Insight

A central **insight** of this work is that *blindly scaling up data fails to maximize the intrinsic power within the datasets*. Through extensive experiments, we observe that Vision-Language-Action (VLA) models are fundamentally representation learning systems—their performance ceiling is jointly determined by the quality of both the learned representations and the training data, not merely the quantity of data. This insight drives our design of AffordanceVLA, which centers on a conceptual bridge supported by two key technical design details:

1. **Affordance as an intermediate bridge:** At the conceptual level, we introduce affordance as a structural bridge to link visual perception with robotic execution.
2. **Design detail I: Architectural refinement:** To support high-fidelity affordance representations, we transition from a discrete VQ-VAE to a continuous Flux VAE for Which2Act supervision. This shift eliminates codebook quantization artifacts and enables the capture of finer-grained visual cues essential for precise manipulation.
3. **Design detail II: Data quality investment:** To ensure the architectural capacity is grounded in accurate knowledge, we curate high-quality datasets (InternData-A1 and DROID) and implement a rigorous automated annotation pipeline. This ensures near-perfect accuracy in affordance labels.

The synergy of this bridge-centric architecture and its supporting technical details is what enables AffordanceVLA to break through performance ceilings that data scaling alone cannot overcome (as validated by the data-efficiency analysis in the main text, Section 4.2).

B.2 From VQ-VAE to Flux VAE: Which2Act Redesign

In our preliminary version (denoted Prev), **Which2Act** supervision relied on a frozen multi-scale VQ-VAE encoder. The VQ-VAE's discrete codebook structure imposes a hard constraint on the token count N_{w2l}: specifically, $\sqrt{N_{w2l}}$ must lie within the predefined scale list $\{1, 2, 3, 4, 5, 6, 8, 10, 13, 16\}$. To achieve sufficient spatial resolution, we set $\sqrt{N_{w2l}} = 8$, requiring $N_{w2l} = 64$ tokens. However, with $N = 64$ tokens and a fixed total latent dimension of 16,384 floats (from the 256×256 input), each token's projection target is only $D_{\text{proj}} = 16{,}384/64 = 256$ dimensions—far below the Generation Expert's hidden size $D_{\text{gen}} = 1024$. This forces the projection head to compress each token by $4\times$, squandering the expert's representational capacity.

In AffordanceVLA, we replace VQ-VAE with a frozen Flux VAE encoder that produces continuous latents $z_q \in \mathbb{R}^{B\times 16\times 32\times 32}$ (16,384 floats total; 16 channels with $8\times$ spatial downsampling of the 256×256 crop). With continuous targets and MSE supervision, the token count N_{w2l} is freed from codebook constraints. We identify $N_{w2l} = 16$ as one choice because:

$$D_{\text{proj}} = \frac{16{,}384}{N_{w2l}} = \frac{16{,}384}{16} = 1024 = D_{\text{gen}} \tag{7}$$

At this configuration, the projection head performs an equal-dimension nonlinear transform ($D_{\text{gen}} \to D_{\text{proj}}$) with zero information compression or expansion, maximizing the per-token representational utility. Increasing the token count merely fragments the same information across more tokens while inflating the shared attention cost; decreasing it forces each token to carry a larger projection burden, requiring lossy expansion.

B.3 Model Variants

We have three model configurations, summarized in Table 6. At each scale, an optional set of wrist tokens—which shares the same decoder architecture and weights as Which2Act—can be appended, supervised by the full wrist camera image. In practice, the wrist token yields marginal improvement and occasionally weakens performance; we therefore treat it as optional and exclude it from the main results.

Table 6 AffordanceVLA model variants. N_{gen} is the total number of Affordance Generation expert tokens (excluding the optional Wrist group). Prev uses a frozen VQ-VAE with discrete codebook supervision (CrossEntropy), requiring $\sqrt{N_{w2l}} \in \{1, ..., 16\}$ (hard constraint). AffordanceVLA-fast and AffordanceVLA use a frozen Flux VAE with continuous latent supervision (MSE), freeing the token count from codebook constraints.

Variant	Which2Act	Where2Act	How2Act		N_{gen}	Which2Act Encoder
			Shape	Layout		
Prev	64	16	30	2	112	VQ-VAE
AffordanceVLA-fast	16	16	30	2	64	Flux VAE
AffordanceVLA	64	64	256	4	388	Flux VAE

Token allocation analysis. For **Where2Act**, tokens are decoded into a heatmap via Transformer cross-attention. For **How2Act Shape**, the decoder must condition a 3D diffusion process to reconstruct a 16^3 voxel latent. The conditioning pathway applies mean-pooling over shape tokens. While this bottleneck limits instance-level 3D fidelity, it suffices for encoding categorical shape priors ("cup-shaped" vs. "plate-shaped") that meaningfully guide downstream action generation. The AffordanceVLA variant allocates 256 shape tokens, providing a richer upstream representation that the mean-pooling aggregates more informatively. **How2Act Layout** regresses a compact 10D vector (rotation_4 + scale_3 + translation_3); even 2 or 4 tokens with mean-pooling are sufficient.

C Data-Centric Methodology

C.1 Data Quality as the Performance Ceiling

Since VLA is fundamentally representation learning, the quality of training data directly determines the model's performance ceiling: if the supervision signals are noisy or inconsistent, the learned representations will be fundamentally flawed regardless of model capacity. This principle guides our dataset selection and annotation strategy:

Dataset selection. For Stage II co-training, we only use the full InternData-A1 simulation dataset, renowned for its photorealistic pixel-level rendering quality across diverse manipulation tasks (basic, pick-and-place, and long-horizon categories). For real-world post-training, we select a curated subset of the DROID dataset, a large-scale in-the-wild robot manipulation corpus collected across diverse real-world environments with consistent high-quality demonstrations. Building upon the Stage II co-training, this phase serves as a critical intermediate step designed to enhance the model's generalization in diverse real-world settings before proceeding to final In-house trajectory fine-tuning. By exposing the model to the rich physical priors in DROID, we effectively bridge the sim-to-real gap, ensuring a more robust transition from large-scale pre-training to specific downstream tasks.

Annotation quality guarantee. Crucially, our affordance annotation pipeline (detailed in Section C.2) achieves near-perfect accuracy on both A1 and DROID, precisely because these datasets provide clean, high-resolution observations that enable reliable visual grounding. This creates a positive propagative effect: high-quality source data → high-quality annotations → high-quality representations → superior downstream performance.

C.2 Affordance Annotation Pipeline

Stage II (A1) and Stage III (LIBERO, CALVIN, DROID) robot datasets do not natively contain affordance annotations (bbox, heatmap, shape token, layout token). We design a fully automated annotation pipeline to generate high-quality affordance supervision for these datasets. For Stage I datasets, the derivation of affordance supervision varies according to their native annotations. AGD20K inherently provides highly accurate, manually annotated affordance heatmaps. In contrast, PRISM and RefSpatial lack native heatmaps. Since our downstream training requires heatmaps rather than explicit point coordinates, we synthesize them utilizing their available native labels. Specifically, for RefSpatial, we prompt SAM within its native bounding

boxes to obtain fine-grained segmentation masks, and subsequently generate Gaussian affordance heatmaps centered at the native interaction points, strictly bounded by these masks. For PRISM, which lacks bounding boxes but offers exceptionally precise interaction points derived from real-world robotic manipulation, we directly prompt SAM using these points to segment the interacted object parts, which serve as the masks for heatmap generation. Finally, we augment all Stage I data with SAM-3D-generated shape and layout tokens.

C.2.1 Step 0: RexOmni Fine-Tuning.

While the open-source visual grounding model RexOmni natively possesses robust, general-purpose object detection and spatial pointing capabilities, we fine-tune it on the PRISM dataset to strictly adapt these skills for robotic manipulation scenarios. Originating from the GraspMolmo project, PRISM provides large-scale annotations of robotic grasp points and target bounding boxes across diverse tabletop settings. Rather than learning these concepts from scratch, this fine-tuning process specifically aligns RexOmni's strong visual grounding priors with the rigorous geometric and physical requirements of robotic affordance localization. Specifically, it enhances the model's ability to produce geometrically consistent outputs: the bounding box identifying the target object and the key interaction point indicating where to act.

C.2.2 Step 1: Rule-Based Keyframe Detection.

For each trajectory, we extract semantically meaningful keyframes from the robot state signal (joint positions, gripper states) using six complementary detection rules, inspired by BridgeVLA:

- **Start:** The first frame of the trajectory, capturing the initial scene configuration.
- **Pre-Action:** N frames before each gripper state change (N=30 by default). This "approach phase" captures the visual observation most indicative of where to act, as the robot is oriented toward but has not yet contacted the target.
- **Gripper:** The exact frame where the gripper's open/close state flips, marking the precise grasp or release event.
- **Stop:** Frames where all joint velocities fall below threshold ($\|\dot{q}_j\| < \epsilon$, ϵ=0.01) while the gripper state remains stable across neighboring frames. These correspond to mid-trajectory pauses, typically at sub-task transitions (e.g., stabilization after placement).
- **Apex:** Frames where the joint velocity norm is a local minimum with $\epsilon < \|\dot{q}\| < 5\epsilon$. These capture trajectory turning points where the motion direction reverses (e.g., transitioning from approach to retraction).
- **End:** The frame M steps before the trajectory end (M=25), avoiding potentially unstable or static trailing frames.

All detected keyframes are sorted chronologically, and pairs separated by fewer than 10 frames are merged (retaining the later index and aggregating reason labels) to eliminate redundancy.

C.2.3 Step 2a: Instruction Decomposition via a Text LLM (Claude Opus 4.5).

A long-horizon command (*e.g.* "Pick up the red cup and put it on the shelf") spans multiple primitive interactions, each aligned to a different keyframe. We therefore first decompose it using a text-only large language model, Claude Opus 4.5. The LLM receives the *original instruction* together with the *ordered keyframe-meaning sequence* from Step 1 (each keyframe tagged by its semantic role, *e.g. Pre-Action*, *Gripper*, *Stop*), and returns one atomic sub-instruction per keyframe that is temporally consistent with the keyframe roles. The exact template is shown in **Prompt A**.

Prompt A — Instruction Decomposition (Claude Opus 4.5, text-only LLM)

Role. You are an expert robotic-manipulation annotator.
Task. Decompose a long-horizon instruction into atomic, single-interaction sub-instructions—*one per keyframe*—that together accomplish the task. Each sub-instruction must describe exactly one primitive interaction (a single grasp, place, push, or release), respect the temporal order and the semantic role of its keyframe, and introduce no

step that is not implied by the original instruction.
Inputs.
• Original instruction: `[original_instruction]`
• Keyframe-meaning sequence: `[(index, role), ...]`
Output (strict JSON). A list aligned to the keyframes; each item is `{"keyframe_index": [int], "sub_instruction": [str]}`.

C.2.4 Step 2b: Per-Keyframe Affordance Annotation via a VLM (Qwen3-VL).

Given the per-keyframe sub-instructions, we then query our deployed **Qwen3-VL-235B** [74] multimodal model on each keyframe *image* to emit the two expressions that drive grounding. To remain compatible with the downstream grounding model, the prompt follows RexOmni [26] conventions: concise category names for detection and "where-to" referring expressions for pointing. Crucially, the VLM is conditioned on the full context—the *original instruction*, the *keyframe-meaning sequence*, the *current keyframe index*, the *current keyframe meaning*, and the *sub-instruction* from Step 2a—which disambiguates the intended target far more reliably than the keyframe image alone. The template is shown in **Prompt B**, and it produces:

1. **Detection category:** the single most specific target the gripper directly contacts at this keyframe (*e.g.* "drawer handle" rather than "drawer" for grasping, "shelf surface" for placing, or "button" for pushing), focusing on the localized interacting region.
2. **Affordance instruction:** a spatial "where" reformulation of the action intent (*e.g.* "Where to grasp the drawer handle to open it?", "Where to place the cup securely on the shelf?", or "Where to push the button?").

These two expressions respectively drive the Which2Act (*what object*) and Where2Act (*where to interact*) supervision signals.

Prompt B — Per-Keyframe Affordance Annotation (Qwen3-VL, RexOmni-style)

Role. You prepare prompts for RexOmni, a detection-and-pointing model driven by category names and referring expressions.
Task. For the current keyframe, output (i) the single most specific target part the gripper directly interacts with, and (ii) a spatial "where-to" affordance query for that interaction. Follow RexOmni conventions: the detection category is a concise noun phrase (*e.g.* "drawer handle", not "drawer"); the affordance instruction is a referring expression of the form "Where to [verb] the [target] to [goal]?".
Inputs.
• Original instruction: `[original_instruction]`
• Keyframe-meaning sequence: `[(index, role), ...]`
• Current keyframe index: `[int]`
• Current keyframe meaning: `[role]`
• Sub-instruction: `[sub_instruction]`
• Current keyframe image: `[image]`
Output (strict JSON). `{"detection_category": [str], "affordance_instruction": [str]}`.

Note. The prompts shown above are illustrative examples; the complete dataset-specific prompt templates are implemented in the data-preprocessing module of our codebase, and the resulting preprocessed annotations will also be made publicly available.

C.2.5 Step 3: Visual Grounding and Affordance Generation.

The two instruction types from Step 2 are fed to the fine-tuned RexOmni in dual modes: *detection categories* → *detection mode* → bbox; *affordance instructions* → *pointing mode* → affordance point. From the resulting bbox and point, we generate the full affordance annotation:

1. **Visual latent (Which2Act GT):** We crop the observation according to the target bbox and extract a continuous visual latent.
2. **Heatmap (Where2Act GT):** We run SAM within the bbox region to obtain a fine-grained segmentation mask, then generate a Gaussian affordance heatmap centered at the affordance point and constrained

by the mask boundary.

3. **Shape token & Layout token (How2Act GT):** The bbox crop and SAM mask are input to a SAM-3D service, which produces a shape token $\in \mathbb{R}^{4096\times 8}$ (3D voxel latent) and a layout token $\in \mathbb{R}^{10}$ (rotation_4 + scale_3 + translation_3).

C.2.6 Step 4: Rigorous Quality Verification.

We implement multi-level quality control to ensure annotation reliability:

1. **Pipeline consistency — Point-in-Bbox verification:** After running the full pipeline, we verify that 100% of generated affordance points fall strictly within their corresponding bounding boxes. Only when this spatial consistency check passes do we certify the pipeline as operationally correct, fundamentally ensuring that the detection (bbox) and pointing (point) outputs are geometrically coherent. Successfully passing this rigorous geometric coherence check not only certifies the operational correctness of the pipeline but also serves as the definitive standard for verifying the RexOmni fine-tuning, ensuring that its decoupled detection (bbox) and pointing (point) outputs are perfectly aligned for downstream manipulation tasks.
2. **Human audit — 100-round random sampling:** We conduct 100 independent random sampling rounds, each drawing 30 annotation samples for manual inspection. Only when all 100 rounds (totaling 3,000 inspected samples) achieve a 100% pass rate is the batch admitted into model training. This stringent audit protocol eliminates systematic errors and edge cases, providing strong empirical evidence of annotation quality.

D Dataset Details

Table 7 summarizes all datasets across training stages.

E Training Details

All training is conducted on a cluster of 16× NVIDIA H200 GPUs (141 GB HBM3e per card). Table 8 summarizes the hyperparameters across all stages. In Stage I, only the Affordance Generation expert, Affordance Query, and all decoders are trained; the Understanding and Action experts remain frozen. In Stage II and III, all experts and decoders are jointly trained end-to-end, with the vision encoder fine-tuned at a lower learning rate.

F Inference Latency

Table 9 reports the inference latency breakdown for the AffordanceVLA model deployed on an NVIDIA RTX 5090 GPU (32 GB GDDR7).

G Real-World Experiments Details and Extra Experiments

Our real-world experiments are conducted on a 7-DoF Franka Emika Panda manipulator equipped with a DH-Robotics PGC-140 parallel jaw gripper (50 mm stroke, 140 N grip force). Visual observations are captured by two Intel RealSense D435 cameras (one fixed third-person view, one wrist-mounted) streaming RGB images at 15 Hz. The robot connects to the host workstation via wired Ethernet.

To bridge the sim-to-real gap, the model is first pre-trained on a curated subset of the DROID dataset, then fine-tuned on in-house demonstrations. During deployment, the model processes single-frame observations, executing the first 5 steps from each predicted action chunk before re-planning.

To comprehensively evaluate our method, we design a diverse set of manipulation tasks. To highlight the generalizability and robustness of our learned policy, we present a more complex task, “Clean all the rubbish”,

Dataset	Source	Stage	Embodiment	FPS	# Samples	Type
PRISM	GraspMolmo	I	–	–	412K	VQA (point + bbox)
AGD20K	Public	I	–	–	20K	VQA (heatmap)
RefSpatial	Public	I	–	–	2500K	VQA (bbox + heatmap)
A1 (InternData)	In-house	II	Franka	30	149K	Trajectory
LIBERO	Public	III	Franka	20	–	Trajectory
CALVIN	Public	III	Franka	30	–	Trajectory
DROID (subset)	Public	Real	Franka	15	∼150K	Trajectory
Close microwave	In-house	Real	Franka	15	80	Trajectory
Close safe	In-house	Real	Franka	15	80	Trajectory
Pick up red cup	In-house	Real	Franka	15	80	Trajectory
Pick up green cup	In-house	Real	Franka	15	80	Trajectory
Pick up duck	In-house	Real	Franka	15	100	Trajectory
Pick up banana	In-house	Real	Franka	15	80	Trajectory
Pick up flower	In-house	Real	Franka	15	100	Trajectory
Pick up bear	In-house	Real	Franka	15	100	Trajectory
Pick The Item in Drawer	In-house	Real	Franka	15	100	Trajectory
Close the drawer	In-house	Real	Franka	15	100	Trajectory
Pick the bread	In-house	Real	Franka	15	100	Trajectory
Toast the bread	In-house	Real	Franka	15	150	Trajectory
Pick all the rubbish	In-house	Real	Franka	15	200	Trajectory

Table 7 Data statistics. Stage I datasets provide visual affordance annotations for grounding pre-training. Stage II uses the full A1 simulation dataset exclusively. Stage III fine-tunes on the target benchmark (LIBERO or CALVIN). All Stage I–III robot data are augmented with automatically generated affordance annotations via our pipeline (Section C.2); Stage I data are additionally augmented with SAM-3D-generated shape and layout tokens. The full command for "Toast the bread" is "Toast the bread with toaster button".

as shown in Figure 8. Notably, this figure also features a trial with explicit human intervention. The successful recovery from dynamic external disturbances demonstrates that our closed-loop re-planning strategy enables the robot to seamlessly adapt to unexpected changes in the environment.

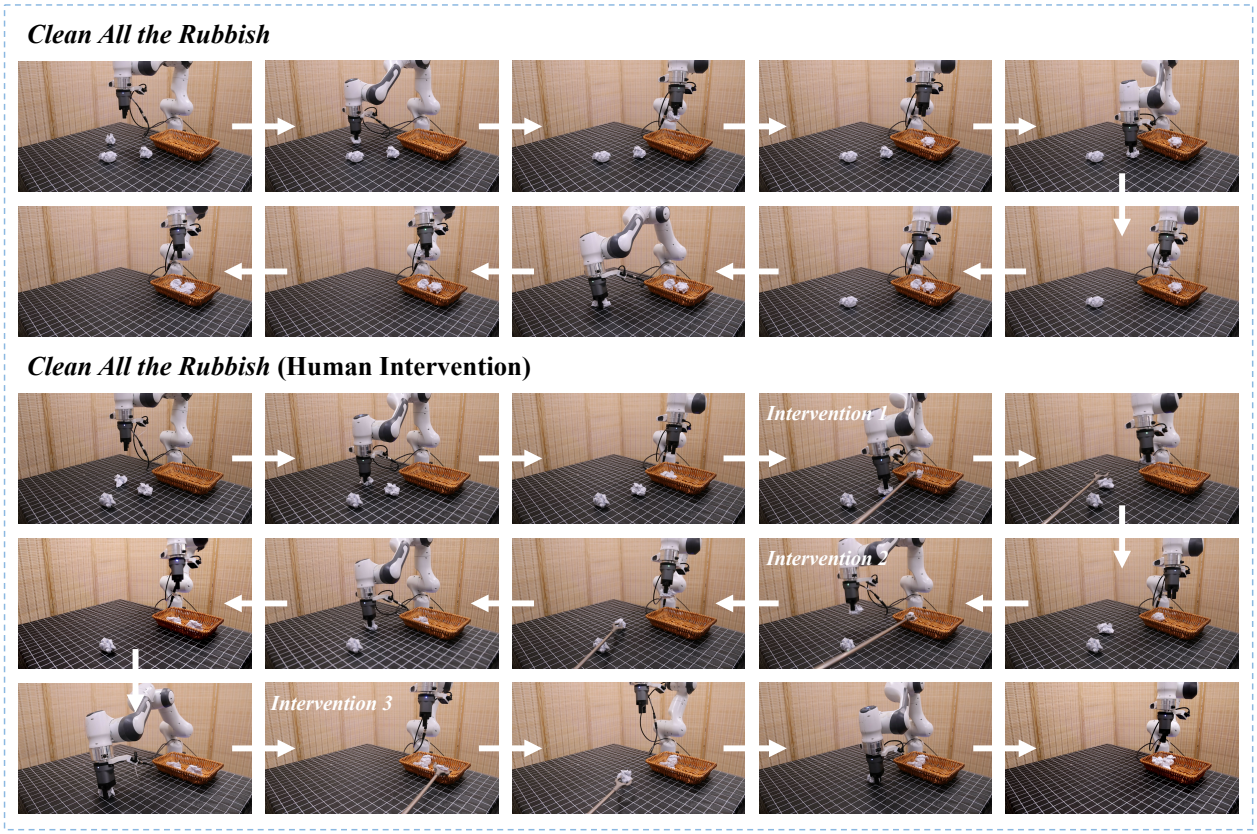


Figure 8 Sequential execution of the "Clean all the rubbish" task, alongside a demonstration of human intervention. The policy's ability to recover from external disturbances highlights its strong generalizability and robustness.

Hyperparameters	Stage I	Stage II	Stage III			Real-World
			LIBERO	CALVIN	DROID	
Batch Size (effective)	3,072	1024	256	256	1024	128
Learning Rate	5×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-5}
LR Scheduler	Cosine	Constant	Cosine	Cosine	Constant	Cosine
Loss Weight (Afd : Act)	–	0.5 : 1	0.15 : 1	0.15 : 1	0.15 : 1	0.15 : 1
Training Steps	300K	230K	–	–	200K	–
Training Epochs	–	–	20	20	–	40
Input Resolution	224×224	224×224	224×224	224×224	224×224	224×224
Which2Act Crop Size	256×256	256×256	256×256	256×256	256×256	256×256
Action Chunk Size	–	30	6	6	30	50
Denoise Steps	–	10	10	10	10	10

Table 8 Training recipe of AffordanceVLA on 16× NVIDIA H200 GPUs. The "Afd : Act" row reports the *task-level* weight ratio λ_{afd}:λ_{act} between the aggregated affordance loss ("Afd") and the action flow-matching loss ("Act"), with λ_{act} normalized to 1. The internal weighting among the four affordance sub-objectives is held fixed across all stages at λ_{which}:λ_{where}:λ_{shape}:λ_{layout} = 0.1:0.1:0.1:0.04 (equivalently 5:5:5:2). In Stage I, the action loss is inactive (no ground-truth actions).

Component	Latency (ms)
Image preprocessing (resize + normalize)	∼6
Phase 1: Image encoder (SigLIP)	∼22
Phase 1: Understanding + Affordance Generation	∼52
Phase 2: ×10 Action denoising (Euler flow matching)	∼92
Misc (tokenization, state preparation)	∼4
Total inference	**∼176**

Table 9 Inference latency of AffordanceVLA on RTX 5090. The total inference time of ∼176 ms enables real-time control at ∼5.7 Hz. The robot is connected via wired Ethernet; reported numbers exclude communication delays.