A2DB: AFFORDANCE AS A DISTILLATION BOTTLENECK FOR COMPRESSING VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

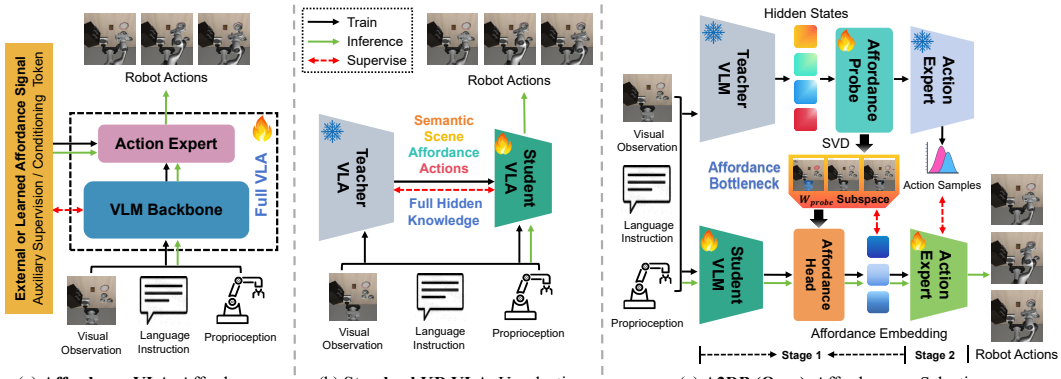
Vision-language-action (VLA) models offer broad manipulation capabilities but remain costly to train and deploy. Standard knowledge distillation requires compact students to reproduce dense teacher representations that contain control-irrelevant information. We introduce A2DB, an affordance-bottleneck VLA distillation framework that transfers only the interaction affordance information needed to determine where and how to act. Automatic affordance labels supervise the frozen π_0 teacher probe, whose scaled leading first-layer singular directions define a compact affordance subspace. A2DB follows a two-stage distillation strategy. Phase I distills an external-view student VLM through selective feature matching, affordance supervision, and language alignment. Phase II freezes this student VLM and trains an affordance-conditioned action expert with behavior cloning, sliced Wasserstein action distillation, and cycle consistency; deployment retains only the student VLM and action expert. Across four LIBERO suites, A2DB reaches 76.7% average success with 0.29 B parameters and 38.6 ms latency, outperforming standard knowledge distillation by 11.6 percentage points while using about 91% fewer parameters and 51% lower latency than the π_0 teacher. Phase I controls show that leading probe-weight directions and singular-value scaling outperform rank-matched and alternative extraction rules, with performance approaching a plateau at $d_A = 128$. Phase II controls identify a near-saturated sampling regime and show that action-distribution guidance and aligned cycle supervision are both important. Affordance predictions, attention maps, label-corruption tests, and failure analysis connect the gains to object, contact, and motion cues, while identifying contact and motion precision as the dominant residual errors and motivating evaluation across seeds, perceptual shifts, embodiments, and real robots. Code will be released after the paper is accepted.

1 INTRODUCTION

Robot manipulation remains challenging because robots must operate under contact-rich dynamics, embodiment mismatch, and diverse object configurations, which have driven a shift from task-specific controllers toward foundation-scale vision-language-action (VLA) models (Wu et al., 2024; Li et al., 2025b; Liu et al., 2025; Xiao et al., 2026), since a single VLA policy can in principle cover a broader range of tasks and embodiments. The same shift also raises the cost of training, storage, and deployment, which means that practical progress now depends not only on performance but also on whether the VLA model can be adapted and executed efficiently.

When compressing a large VLA model, a central question is what information should be preserved. Full feature knowledge distillation is inefficient because it supervises the student to match dense teacher activations even when only a small subset of that representation is truly responsible for control. Recent studies suggest that preserving the task-relevant part of the representation is more effective than retaining all visual features. Selective visual representations (Eftekhar et al., 2024) introduced a task-conditioned bottleneck for embodied learning, robot pretraining (Jiang et al., 2025) showed that manipulation-centric pretraining can improve downstream control, and later related work (Ye et al., 2026; Li et al., 2026) aligned intermediate visual tokens with geometry and action heads through spatially guided training and spatial forcing.

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



(a) **Affordance VLA**: Affordance as Auxiliary Supervision or Conditioning

(b) **Standard KD VLA**: Unselective Teacher-to-Student Knowledge Transfer

(c) **A2DB (Ours)**: Affordance as Selective Distillation Bottleneck

Figure 1: **Paradigm comparison.** Existing Affordance VLA uses affordance as auxiliary supervision or action conditioning, and trains full VLA models directly. Standard KD VLA transfers dense teacher knowledge without distinguishing between task-relevant and task-irrelevant information, whereas A2DB uses affordance as the selective distillation bottleneck, deciding what teacher knowledge is allowed to pass and selecting action-relevant knowledge for VLA model compression.

Notably, affordance provides a sharper information bottleneck for this knowledge transfer problem because it isolates the interaction structure that matters for control while abstracting away appearance variation and other nuisance factors. In robotics, affordance has served as a compact interaction prior, and recent works (Kroemer et al., 2021; Bahl et al., 2023; Li et al., 2025a; Xu et al., 2025) have further treated it as an intermediate representation for manipulation. From this view, the teacher does not need to transfer every latent detail; it only needs to transfer the spatial and functional cues that determine where and how interaction should occur, which makes affordance a more faithful and efficient bottleneck for VLA model compression.

Motivated by this insight, A2DB uses affordance as the selective bottleneck for VLA compression illustrated in Figure 1. Existing Affordance VLA methods treat affordance as auxiliary supervision or action conditioning and train a full VLA model directly. Standard KD VLA transfers dense teacher knowledge without distinguishing task-relevant information from task-irrelevant information. By contrast, our A2DB uses affordance as a distillation bottleneck, deciding what teacher knowledge is allowed to pass and retaining action-relevant information for VLA model compression.

In conclusion, the contributions of our work are summarized as follows:

- We propose A2DB, an affordance-bottleneck VLA distillation framework that replaces full feature transfer with task-relevant interaction knowledge transfer.
- We introduce automatic affordance label generation and a teacher-side affordance probe to extract an affordance subspace from demonstrations, so that the π_0 teacher explicitly controls what information passes to the lightweight student.
- We design a two-stage compression scheme with selective student VLM distillation and distillation of an affordance-conditioned action expert, enabling the compression of large VLA models while preserving action-relevant control cues.

2 Method: Affordance Bottleneck VLA Distillation

2.1 Problem Formulation

We consider language-conditioned robot manipulation from demonstrations, where each sample contains an external image $\mathbf{x}_b$, a wrist image $\mathbf{x}_w$, an instruction ℓ, a proprioceptive state $\mathbf{s} \in \mathbf{R}^{d_s}$, and an action chunk $\mathbf{a}_{1:H} \in \mathbf{R}^{H \times d_a}$. The frozen π_0 teacher encodes both images and instruction as tokens $\mathbf{Z}^T \in \mathbf{R}^{N_T \times d_T}$, while the compact student observes only $\mathbf{x}_b$ and ℓ to produce $\mathbf{Z}^S \in \mathbf{R}^{N_S \times d_S}$ and retain the affordance information required for control.

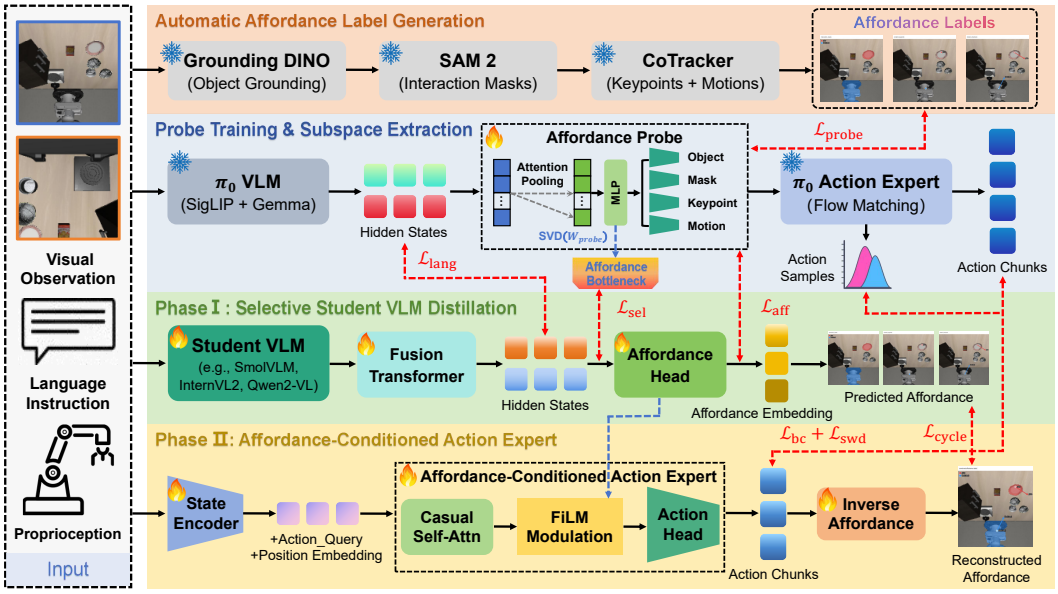


Figure 2: **Overview of A2DB.** A2DB first generates affordance labels from demonstrations, then trains a teacher-side probe to extract the affordance subspace. A2DB follows a two-stage distillation strategy: Phase I distills a compact student VLM through selective feature, affordance supervision, and language alignment. Phase II trains an affordance-conditioned action expert with behavior cloning, sliced Wasserstein action distillation, and cycle consistency.

A2DB decomposes this problem into two coupled learning phases, as shown in Figure 2. First, a teacher affordance probe identifies an affordance subspace $\mathbf{P}_A \in \mathbf{R}^{d_T \times d_A}$, and the student VLM is distilled only through this subspace rather than through the full teacher representation. Second, an affordance-conditioned action expert maps the student affordance embedding $\mathbf{e} \in \mathbf{R}^{d_A}$ and robot state $\mathbf{s}$ to an action chunk, yielding the following inference factorization:

$$\hat{\mathbf{a}}_{1:H} = g_\psi(\mathbf{s}, \mathbf{e}), \qquad \mathbf{e} = f_\theta(\mathbf{x}_b, \ell). \tag{1}$$

The frozen π_0 teacher defines the distillation bottleneck and action targets only during training, so deployment requires only the student VLM f_θ and action expert g_ψ.

2.2 Teacher Affordance Supervision

Teacher supervision converts demonstrations into interaction targets and trains an affordance probe on the frozen π_0, allowing the probe to expose compact representation directions that predict object masks, contact locations, and motion directions without becoming part of the deployed student VLA.

2.2.1 Automatic Affordance Label Generation

For each episode, the affordance annotator converts the task text into object prompts, detects relevant objects in the first frame, segments their boxes, and tracks points within each mask through the demonstration using Grounding DINO (Liu et al., 2024), SAM2 (Kirillov et al., 2023), and CoTracker (Karaev et al., 2024). Each valid object o receives an interaction mask $\mathbf{m}_o \in \{0,1\}^{h \times w}$, a normalized contact location $\mathbf{c}_o \in [0,1]^2$, and a unit motion direction $\mathbf{v}_o \in \mathbf{R}^2$. If $\mathbf{u}_{o,j,1}$ and $\mathbf{u}_{o,j,T}$ are the first and last visible positions of tracked point j, the targets are

$$\mathbf{c}_o = \frac{\text{median}_{j \in \mathcal{V}_o} \mathbf{u}_{o,j,1}}{\mathbf{r}}, \qquad \mathbf{v}_o = \frac{\text{median}_{j \in \mathcal{V}_o}(\mathbf{u}_{o,j,T} - \mathbf{u}_{o,j,1})}{\|\text{median}_{j \in \mathcal{V}_o}(\mathbf{u}_{o,j,T} - \mathbf{u}_{o,j,1})\|_2 + \varepsilon_0}, \tag{2}$$

where $\mathcal{V}_o$ contains points visible at both ends of the episode, $\mathbf{r} = [w-1, h-1]$ normalizes image coordinates, and ε_0 ensures numerical stability.

2.2.2 AFFORDANCE PROBE TRAINING AND SUBSPACE EXTRACTION

Given teacher tokens $\mathbf{Z}^T = \{\mathbf{z}_n^T\}_{n=1}^{N_T}$, the affordance probe uses a learned query $\mathbf{q}_T$ to pool the valid representation as:

$$\alpha_n^T = \frac{\exp((\mathbf{z}_n^T)^\top \mathbf{q}_T/\sqrt{d_T})}{\sum_j \exp((\mathbf{z}_j^T)^\top \mathbf{q}_T/\sqrt{d_T})}, \qquad \bar{\mathbf{z}}^T = \sum_{n=1}^{N_T} \alpha_n^T \mathbf{z}_n^T. \tag{3}$$

The pooled vector passes through a two-layer MLP and object-specific decoders that predict masks, normalized contact locations, and unit motion directions. With object validity indicator $v_{b,o}$, the probe combines mask binary cross entropy, contact smooth ℓ_1 error, and motion cosine error as:

$$\mathcal{L}_{\text{probe}} = \lambda_m \mathcal{L}_m + \lambda_c \mathcal{L}_c + \lambda_v \mathcal{L}_v, \tag{4}$$

$$\mathcal{L}_m = \frac{1}{C} \sum_{b,o} v_{b,o} \frac{1}{|\Omega|} \sum_{p \in \Omega} \text{BCE}(\hat{\mathbf{m}}_{b,o,p}, \mathbf{m}_{b,o,p}), \tag{5}$$

$$\mathcal{L}_c = \frac{1}{C} \sum_{b,o} v_{b,o} \rho(\hat{\mathbf{c}}_{b,o} - \mathbf{c}_{b,o}), \qquad \mathcal{L}_v = \frac{1}{C} \sum_{b,o} v_{b,o} \left(1 - \frac{\hat{\mathbf{v}}_{b,o}^\top \mathbf{v}_{b,o}}{\|\hat{\mathbf{v}}_{b,o}\|_2 \|\mathbf{v}_{b,o}\|_2 + \varepsilon_0}\right), \tag{6}$$

where $C = \max(1, \sum_{b,o} v_{b,o})$, Ω is the image grid, and ρ denotes the smooth ℓ_1 penalty.

After the teacher affordance probe training, singular value decomposition of the first probe layer selects the tested probe-derived directions associated with affordance prediction, and A2DB retains the leading d_A left singular vectors scaled by their singular values as:

$$\mathbf{W}_0 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \qquad \mathbf{P}_A = \mathbf{U}_{:,1:d_A} \mathbf{\Sigma}_{1:d_A,1:d_A}. \tag{7}$$

Projection $\mathbf{P}_A$ defines which teacher directions are considered affordance relevant during Phase I.

2.3 PHASE I: SELECTIVE STUDENT VLM DISTILLATION

2.3.1 LIGHTWEIGHT STUDENT VLM AND AFFORDANCE HEAD

The student VLM uses a lightweight vision-language backbone and a compact fusion transformer to combine only external image tokens with the task instruction. Each processor exposes vision and instruction masks plus its visual grid. The affordance head pools fused representations into a d_A dimensional embedding and predicts masks, contacts, and motions, while $\mathbf{W}_S \in \mathbf{R}^{d_S \times d_A}$ projects both modalities into the shared affordance subspace.

2.3.2 SELECTIVE FEATURE DISTILLATION

A2DB compares projected external image features within the affordance subspace. Let $\mathbf{G}_b^T \in \mathbb{R}^{16\times16\times d_T}$ and $\mathbf{G}_b^S \in \mathbb{R}^{h_b \times w_b \times d_S}$ denote the teacher and student external view grids, respectively. The selective distillation loss is

$$\mathcal{L}_{\text{sel}} = \frac{1}{C_v} \sum_{b=1}^{B} \sum_{i=1}^{16} \sum_{j=1}^{16} \mu_{b,i,j}^v \left\| \left[\mathcal{R}_{16\times16}\left[\mathbf{G}_b^S \mathbf{W}_S\right]\right]_{i,j} - \left[\mathbf{G}_b^T \mathbf{P}_A\right]_{i,j} \right\|_2^2. \tag{8}$$

where $C_v = \max\{1, \sum_{b,i,j} \mu_{b,i,j}^v\}$ counts valid teacher grid cells. Bilinear resizing maps the student grid onto the teacher external grid, so equal indices denote corresponding locations while wrist tokens remain excluded. Appendix E.2 details backbone grids and masks.

2.3.3 AFFORDANCE PREDICTION OBJECTIVE

The student affordance head uses the same automatically generated affordance targets as the teacher probe. Its loss retains the component structure of $\mathcal{L}_{\text{probe}}$:

$$\mathcal{L}_{\text{aff}} = \lambda_m \mathcal{L}_m^S + \lambda_c \mathcal{L}_c^S + \lambda_v \mathcal{L}_v^S. \tag{9}$$

The three terms measure mask binary cross entropy, contact smooth ℓ_1 error, and motion cosine error. Teacher instruction states reflect two views, while student states reflect the external view. Phase I projects, pools, and aligns these multimodal states through a cosine loss

$$\mathcal{L}_{\text{lang}} = 1 - \frac{(\bar{\mathbf{t}}^S)^\top \bar{\mathbf{t}}^T}{\|\bar{\mathbf{t}}^S\|_2 \|\bar{\mathbf{t}}^T\|_2 + \varepsilon_0}. \tag{10}$$

The total Phase I objective is

$$\mathcal{L}_{\text{I}} = \alpha \mathcal{L}_{\text{sel}} + \beta \mathcal{L}_{\text{aff}} + \gamma \mathcal{L}_{\text{lang}}. \tag{11}$$

2.4 Phase II: Affordance Conditioned Action Expert

2.4.1 Lightweight Action Decoder

Phase II freezes the distilled student VLM and trains a lightweight action expert that predicts an action chunk from the robot state and affordance embedding. The decoder projects the state to hidden dimension $d_h = 256$ and forms one token for each future step by adding a learned action query and sinusoidal position encoding, after which four affordance-conditioned decoder layers and a linear action head produce $\hat{\mathbf{a}}_{1:H}$.

2.4.2 Affordance Cross-Attention and FiLM Conditioning

Each decoder layer combines causal self-attention over action tokens, feature-wise affine conditioning from the affordance embedding, cross-attention to affordance tokens, and a feedforward block. For a layer input $\mathbf{Y}^l$, the layer applies

$$\tilde{\mathbf{Y}}^l = \mathbf{Y}^l + \text{MHA}(\text{LN}(\mathbf{Y}^l), \text{LN}(\mathbf{Y}^l), \text{LN}(\mathbf{Y}^l); \mathbf{M}_{\text{causal}}), \tag{12}$$

$$\bar{\mathbf{e}} = \frac{1}{K_A} \sum_{k=1}^{K_A} \mathbf{e}_k, \qquad \mathbf{Y}_F^l = \tilde{\mathbf{Y}}^l \odot (1 + \Gamma(\bar{\mathbf{e}})) + B(\bar{\mathbf{e}}), \tag{13}$$

$$\mathbf{Y}_C^l = \mathbf{Y}_F^l + \text{MHA}(\text{LN}(\mathbf{Y}_F^l), \mathbf{E}, \mathbf{E}), \qquad \mathbf{Y}^{l+1} = \mathbf{Y}_C^l + \text{FFN}(\text{LN}(\mathbf{Y}_C^l)). \tag{14}$$

Here $\mathbf{E}$ is the affordance embedding represented as a sequence, Γ and B are linear FiLM scale and shift maps, and $\mathbf{M}_{\text{causal}}$ is a lower triangular attention mask over the action horizon.

2.4.3 Action Expert Distillation and Cycle Consistency

The affordance-conditioned action expert is optimized with three losses. The first is behavior cloning on demonstration action chunks as:

$$\mathcal{L}_{\text{bc}} = \frac{1}{BHd_a} \sum_{b=1}^{B} \sum_{t=1}^{H} \sum_{i=1}^{d_a} (\hat{a}_{b,t,i} - a_{b,t,i})^2. \tag{15}$$

The second matches single-view student action samples to samples from the two-view teacher using sliced Wasserstein distillation. The student produces K samples by injecting Gaussian noise into its action token stream and decoding K chunks. After flattening every chunk, A2DB draws P random unit projections $\{\mathbf{u}_p\}_{p=1}^P$ and sorts the projected samples as:

$$\mathbf{d}_{b,p,:}^S = \text{sort}\left(\left\{ (\mathbf{u}_p)^\top \text{vec}(\hat{\mathbf{a}}_b^{S,k}) \right\}_{k=1}^K \right), \qquad \mathbf{d}_{b,p,:}^T = \text{sort}\left(\left\{ (\mathbf{u}_p)^\top \text{vec}(\mathbf{a}_b^{T,k}) \right\}_{k=1}^K \right). \tag{16}$$

$$\mathcal{L}_{\text{swd}} = \frac{1}{BPK} \sum_{b=1}^{B} \sum_{p=1}^{P} \sum_{k=1}^{K} \left(\mathbf{d}_{b,p,k}^S - \mathbf{d}_{b,p,k}^T \right)^2. \tag{17}$$

Finally, an inverse model r_ω reconstructs the conditioning affordance from the predicted action chunk, yielding the cycle loss as:

$$\mathcal{L}_{\text{cyc}} = \frac{1}{B} \sum_{b=1}^{B} \|r_\omega(\hat{\mathbf{a}}_{b,1:H}) - \mathbf{e}_b\|_2^2. \tag{18}$$

The total Phase II objective is

$$\mathcal{L}_{\text{II}} = \delta \mathcal{L}_{\text{bc}} + \epsilon \mathcal{L}_{\text{swd}} + \zeta \mathcal{L}_{\text{cyc}}. \tag{19}$$

The sampling distribution, projection count, inverse model architecture, and gradient path are reported in Appendix E.4.

2.5 End to End Inference

At inference time, A2DB removes all teacher-side components. Given an external image and task instruction, the student VLM computes an affordance embedding, and the action expert predicts the next action chunk from this embedding and the proprioceptive state as:

$$\mathbf{e} = f_\theta(\mathbf{x}_b, \ell), \qquad \hat{\mathbf{a}}_{1:H} = g_\psi(\mathbf{s}, \mathbf{e}). \tag{20}$$

3 Experiments

3.1 Main Results on LIBERO Benchmarks

Table 1 shows that the affordance-bottleneck distillation improves compact VLA policy learning across all four LIBERO suites. With SmolVLM, A2DB reaches 76.7% average success and exceeds behavior cloning, standard knowledge distillation, and SmolVLA by 14.5, 11.6, and 4.0 percentage points. Gains remain consistent across all four suites, including the longer-horizon `libero_long` tasks. A2DB remains 7.7 percentage points below the frozen π_0 teacher but reduces parameters from 3.3 B to 0.29 B and latency from 78.7 ms to 38.6 ms, which corresponds to about 91% fewer parameters and 51% lower latency. Larger students increase success to 78.8% with InternVL2-1B and 80.0% with Qwen2-VL-2B, although their costs rise to 0.97 B parameters with 55.8 ms latency and 2.24 B parameters with 72.7 ms latency. A2DB (SmolVLM) therefore provides the most efficient evaluated results and serves as the default student in subsequent analyses.

Table 1: **Main results on LIBERO benchmarks.** Success rate (SR, %) is reported under the same rollout protocol for every method.

Method	VLM Backbone	Params. B ↓	Latency (ms) ↓	LIBERO SR % ↑				
				Spatial	Object	Goal	Long	Avg.
π_0 Teacher	`SigLIP+Gemma`	3.3	78.7	86.8	88.6	85.8	76.4	84.4
SmolVLA	`SmolVLM`	0.45	46.3	72.6	75.8	73.4	68.8	72.7
BC only	`SmolVLM`	0.29	38.6	62.4	65.6	63.8	56.8	62.2
Standard KD	`SmolVLM`	0.29	38.6	65.6	67.8	64.4	62.6	65.1
A2DB	`SmolVLM`	0.29	38.6	76.4	79.8	77.2	73.4	76.7
A2DB	`InternVL2-1B`	0.97	55.8	78.6	81.2	79.4	75.8	78.8
A2DB	`Qwen2-VL-2B`	2.24	72.7	80.2	82.4	80.8	76.6	80.0

3.2 Ablation Studies

Table 2 isolates the Phase I components on `libero_spatial`. Removing selective feature distillation lowers success rate from 76.4% to 65.4%, while removing direct affordance supervision produces the largest decline to 63.6% and removing language alignment yields 67.4%. The larger drop without affordance supervision shows that the affordance bottleneck must stay aligned with interaction targets. Full hidden state distillation reaches 65.6%, whereas random and PCA subspaces reach 67.8% and 68.2%. Because all three alternatives remain at least 8.2 percentage points below A2DB, the improvement depends on selecting teacher directions that predict interaction affordance structure rather than reducing representation dimension alone.

Table 3 evaluates how the action expert uses the distilled affordance representation. Removing action distribution distillation reduces success rate from 76.4% to 63.8%, a decline of 12.6 percentage points. This result approaches the 62.4% achieved by behavior cloning alone, which is 14.0 percentage points below A2DB, and shows that teacher action samples provide information beyond the demonstrated target. This objective supplies distribution-level guidance beyond a single demonstrated action sequence. Removing cycle consistency yields 67.6%, an 8.8 percentage point decline, while replacing the student representation with label embeddings yields 64.8%, an 11.6 percentage point decline. These results show that the action expert must preserve a visually grounded affordance representation rather than depend on explicit labels alone.

Table 2: **Phase I ablations on `libero_spatial`.** The comparison isolates selective feature distillation, affordance supervision, language alignment, and teacher subspace selection.

Variant	$\mathcal{L}_{\text{sel}}$	$\mathcal{L}_{\text{aff}}$	$\mathcal{L}_{\text{lang}}$	Teacher subspace	SR % ↑	Δ pp
A2DB	✓	✓	✓	Probe-derived	76.4	0.0
No selective loss	✗	✓	✓	Probe-derived	65.4	-11.0
No affordance loss	✓	✗	✓	Probe-derived	63.6	-12.8
No language loss	✓	✓	✗	Probe-derived	67.4	-9.0
Standard KD	✗	✓	✓	Full hidden state	65.6	-10.8
Random subspace	✓	✓	✓	Random	67.8	-8.6
PCA subspace	✓	✓	✓	PCA	68.2	-8.2

Table 3: **Phase II ablations on `libero_spatial`.** The comparison isolates behavior cloning, sliced Wasserstein action distillation, cycle consistency, and the distilled student VLM.

Variant	$\mathcal{L}_{\text{bc}}$	$\mathcal{L}_{\text{swd}}$	$\mathcal{L}_{\text{cyc}}$	Distilled VLM	SR % ↑	Δ pp
A2DB	✓	✓	✓	✓	76.4	0.0
BC only	✓	✗	✗	✓	62.4	-14.0
No action distillation	✓	✗	✓	✓	63.8	-12.6
No cycle consistency	✓	✓	✗	✓	67.6	-8.8
Label embedding only	✓	✓	✓	✗	64.8	-11.6

3.3 Affordance Bottleneck Dimension Analysis

Figure 3 shows that the probe-derived representation forms a compact information bottleneck. Increasing its dimension from 16 to 32, 64, and 128 raises success rate from 69.6% to 73.8%, 75.4%, and 76.4%. Performance then improves by only 0.4 percentage points to 76.8% at dimension 256 and remains unchanged at dimension 512, indicating that the curve approaches a plateau at dimension 128. Standard KD reaches 60.2%, 61.8%, 63.4%, 65.6%, 66.4%, and 68.8% over the same sweep, leaving a 10.8 percentage point gap at dimension 128. Thus, increasing the width of an unstructured representation does not close the gap created by missing affordance alignment. The selected embedding also requires only 38.63 ms and 0.291 B parameters, compared with 40.33 ms and 0.293 B at dimension 512, which supports 128 as the affordance dimension because further expansion increases cost for only a marginal success-rate gain.

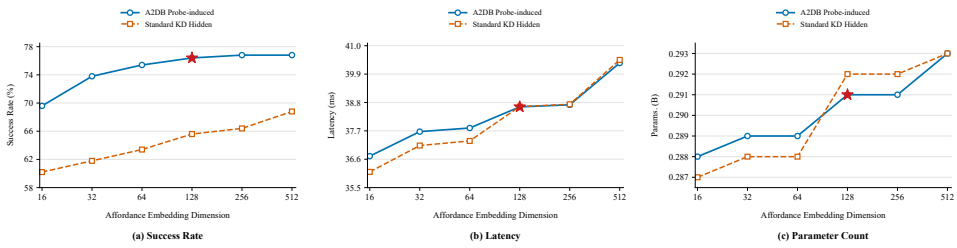


Figure 3: **Affordance bottleneck dimension analysis on `libero_spatial`.** Panels a, b, and c report success rate, inference latency, and parameter count across affordance dimensions. The blue solid curve denotes the probe-derived representation, the orange dashed curve denotes standard knowledge distillation, and the red star marks the selected affordance dimension $d_A = 128$.

3.4 Efficiency and Pareto Analysis

Figure 4 compares LIBERO success rate with per-chunk inference latency measured on NVIDIA RTX 4090. Under the same hardware and evaluation protocol, every A2DB variant lies on the favorable empirical frontier. At the same parameter and latency scale as behavior cloning and standard knowledge distillation, the default A2DB improves success by 14.5 and 11.6 percentage points, respectively. Compared with SmolVLA, it improves success by 4.0 percentage points while reducing

per-chunk latency from 46.3 to 38.6 ms and parameters from 0.45 to 0.29 B. Increasing student capacity raises success to 78.8% with InternVL2-1B and 80.0% with Qwen2-VL-2B, whereas the default A2DB model offers the strongest efficiency by using about 91% fewer parameters and 51% lower latency than the π_0 teacher. These comparisons show that the affordance bottleneck converts additional capacity into performance more effectively than the evaluated compact alternatives.

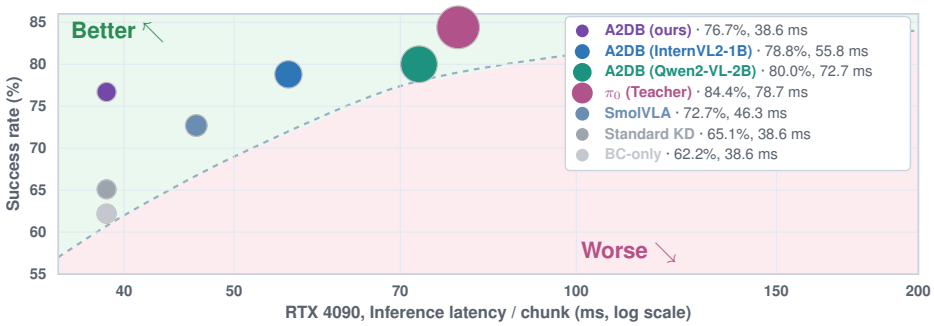


Figure 4: **Efficiency and Pareto analysis.** LIBERO success rate is plotted against per-chunk inference latency measured on NVIDIA RTX 4090, using a logarithmic horizontal axis. Circle area represents parameter count, and the shaded regions indicate favorable and unfavorable tradeoffs.

3.5 Qualitative Affordance and Attention Analysis

Figure 5 examines whether the distilled student preserves the structured interaction cues required for manipulation. In the successful rollout, it localizes the plate and black bowl while producing contact and motion predictions that follow the annotations, whereas the failed rollout retains plausible object regions but loses geometric precision, showing that semantic recognition alone does not ensure successful control. The contrast shows that small contact or motion errors can cause failure despite recognizable objects. In the ambiguous case, the distilled student separates the left and right plates and distinguishes visually similar mugs through their spatial roles. Together, these examples indicate that the affordance bottleneck transfers object identity, interaction region, contact location, and motion direction while exposing geometric prediction errors as a likely source of failure.

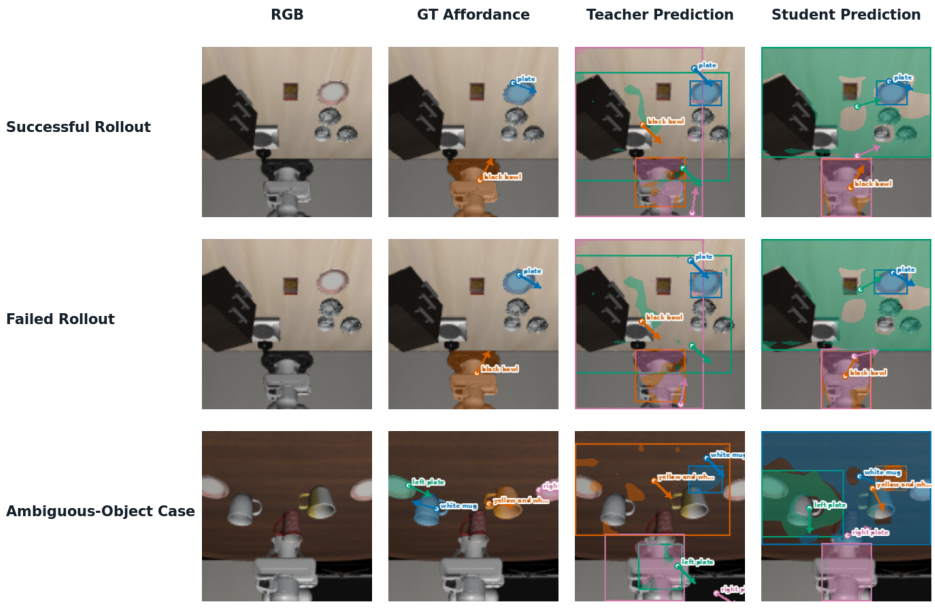


Figure 5: **Qualitative affordance predictions on `libero_spatial`.** Successful, failed, and ambiguous object cases compare RGB, GT affordances, teacher predictions, and student predictions.

Figure 6 shows how selective distillation reshapes visual attention across representative instructions. Teacher attention spreads across the scene and often responds to background structure, while student attention concentrates on the commanded object, destination, and robot interaction area, with high response regions that generally overlap the annotated affordance masks. This pattern persists across bowl placement, object relocation, and mug manipulation, indicating that A2DB filters diffuse context while preserving the visual evidence that links language to actionable regions and thereby supports affordance as a selective distillation bottleneck. This alignment indicates that selective transfer suppresses context while retaining regions needed for contact and placement. These heatmaps are interpretive evidence rather than a replacement for success measurements, but they provide a visual check that the learned transfer interface follows the intended task-related semantics.

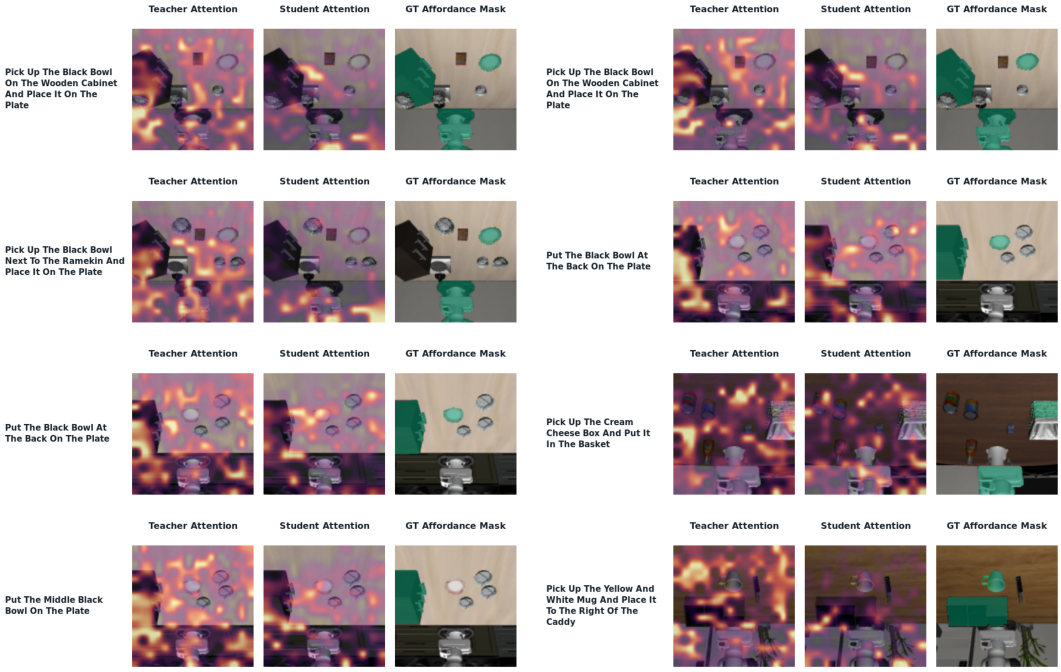


Figure 6: **Attention focus heatmaps on `libero_spatial`**. Each triplet compares teacher attention, student attention, and the GT affordance mask across representative manipulation tasks.

4 Conclusion

In this paper, A2DB treats affordance as a selective information bottleneck for VLA compression. A teacher-side probe extracts leading affordance-predictive directions, which Phase I transfers to a lightweight student VLM through selective feature matching, affordance supervision, and language alignment. Phase II freezes this distilled encoder and trains an affordance-conditioned action expert with behavior cloning, sliced Wasserstein action distillation, and cycle consistency, so deployment uses only the compact student VLM and action expert. Across four LIBERO suites, A2DB reaches 76.7% average success with 0.29 B parameters and 38.6 ms latency, improving over standard knowledge distillation by 11.6 percentage points while using about 91% fewer parameters and 51% lower latency than the π_0 teacher. Phase I controls support the leading probe directions, singular-value scaling, and $d_A = 128$ as the selected bottleneck; Phase II controls support the near-saturated $K = 4$, $P = 64$, and $\sigma = 0.05$ operating point. Together, these results support affordance as a practical interface for transferring control-relevant structure without reproducing the teacher's full representation. Removing action-distribution supervision, cycle consistency, or affordance-target alignment produces substantial drops, while affordance annotations, attention maps, deployment trajectories, and label-corruption tests reveal the intended transfer and its sensitivity to supervision quality. Contact localization and motion precision remain the dominant residual errors, motivating evaluation on real robots, broader embodiments, and shifted perceptual conditions.

AI USE STATEMENT

AI-based tools were used as assistants during the research and writing process. They supported the implementation and debugging of model code, drafting and language polishing of the manuscript, and the creation of visualizations and statistical plots from experimentally obtained data. All research decisions, experimental design, implementation validation, data analysis, result interpretation, and final manuscript revisions were conducted and verified by the authors. The authors take full responsibility for the content of this manuscript and the reported results.

REPRODUCIBILITY STATEMENT

We document the implemented data path, token serialization and matching rule, module dimensions, optimization settings, random seeds, and rollout protocol in Appendix E. The supplied artifacts record one training seed without a validation split. Claims about token matching, SWD, and cycle consistency are therefore limited to the implemented configuration rather than multi-seed or mechanism-level evidence.

ETHICS STATEMENT

A2DB is evaluated solely in the simulated LIBERO benchmark and involves no human participants, personal or sensitive data, or physical robot deployment. We report annotation-corruption sensitivity and failure modes, and do not claim real-world safety or transfer; physical deployment requires task-specific safety validation and human oversight.

REFERENCES

Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 01–13. IEEE, 2023.

Kewei Chen, Yayu Long, Shuai Li, and Mingsheng Shang. Ft-ncfm: An influence-aware data distillation framework for efficient vla models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18180–18188, 2026a.

Yuxuan Chen, Yixin Han, Yize Huang, and Xiao Li. Rlrc: Reinforcement learning-based recovery for compressed vision-language-action models. *IEEE Robotics and Automation Letters*, 2026b.

Shaoqi Dong, Chaoyou Fu, Haihan Gao, Yi-Fan Zhang, Chi Yan, Chu Wu, Xiaoyu Liu, Yunhang Shen, Jing Huo, Deqiang Jiang, et al. Vita-vla: Efficiently teaching vision-language models to act via action expert distillation. *arXiv preprint arXiv:2510.09607*, 2025.

Ainaz Eftekhar, Kuo-Hao Zeng, Jiafei Duan, Ali Farhadi, Aniruddha Kembhavi, and Ranjay Krishna. Selective visual representations improve convergence and generalization for embodied ai. In *International Conference on Learning Representations*, volume 2024, pp. 13256–13277, 2024.

Minghao Guo, Meng Cao, Jiachen Tao, Rongtao Xu, Yan Yan, Xiaodan Liang, Ivan Laptev, and Xiaojun Chang. Glad: Geometric latent distillation for vision-language-action models. *arXiv preprint arXiv:2512.09619*, 2025.

Xiangyu Huang, Zhenlin Hua, Han Zhou, Shounak Sural, and Ragunathan Rajkumar. Rt-vla: Real-time vision-language-action models via knowledge distillation. *arXiv preprint arXiv:2606.14010*, 2026.

Ingook Jang, Samyeul Noh, and Seonghyun Kim. Exploring policy distillation for vision-language-action generalist robotic policies. In *International Conference on Information and Communication Technology Convergence*, pp. 1017–1020. IEEE, 2025.

Boseong Jeon, Yunho Choi, and Taehan Kim. Shallow-{\pi}: Knowledge distillation for flow-based vlas. *arXiv preprint arXiv:2601.20262*, 2026.

Guangqi Jiang, Yifei Sun, Tao Huang, Huanyu Li, Yongyuan Liang, and Huazhe Xu. Robots pre-train robots: Manipulation-centric robotic representation from large-scale robot datasets. In *International Conference on Learning Representations*, volume 2025, pp. 81885–81905, 2025.

Tobias Jülg, Wolfram Burgard, and Florian Walter. Refined policy distillation: From vla generalists to rl experts. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 11677–11684. IEEE, 2025.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, pp. 18–35. Springer, 2024.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *IEEE/CVF International Conference on Computer Vision*, pp. 3992–4003. IEEE, 2023.

Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *Journal of Machine Learning Research*, 22(30): 1–82, 2021.

Minhyeok Lee, Chiyoung Kim, Chanhoe Gu, Seongrok Kim, Sanghyuk Roy Choi, Donghwan Hwang, Donghun Ryu, and Seokhyun Kim. Cotinyvla: Chain-of-thought distillation for a sub-billion-parameter vision-language-action model. *arXiv preprint arXiv:2607.25487*, 2026a.

Sangoh Lee, Sangwoo Mo, and Wook-Shin Han. Act with intent: Distilling behavior intent for vision-language-action models. *arXiv preprint arXiv:2608.23478*, 2026b.

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

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

Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memmel, Caelan Garrett, Fabio Ramos, Dieter Fox, Anqi Li, Abhishek Gupta, et al. Hamster: Hierarchical action models for open-world robot manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 24040–24068, 2025b.

Zuolei Li, Xingyu Gao, Xiaofan Wang, and Jianlong Fu. Latbot: Distilling universal latent actions for vision-language-action models. *arXiv preprint arXiv:2511.23034*, 2025c.

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

Litao Liu, Yifan Han, Pengfei Yi, Wenbo Yu, Hanqing Wang, Haoran Du, Enze Yuan, Zilin Yuan, Ruiding Feng, Michael Liu, et al. Affordance2action: Task-conditioned scene-level affordance grounding for real-time manipulation. *arXiv preprint arXiv:2606.04172*, 2026a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pp. 38–55. Springer, 2024.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025.

Yuanzhe Liu, Jingyuan Zhu, Yuchen Mo, Gen Li, Xu Cao, Jin Jin, Yifan Shen, Zhengyuan Li, Tianjiao Yu, Wenzhen Yuan, et al. Palm: Progress-aware policy learning via affordance reasoning for long-horizon robotic manipulation. *arXiv preprint arXiv:2601.07060*, 2026b.

Jihoon Oh, Kento Kawaharazuka, and Kei Okada. Vlaff: Vision-language-affordance model for unified actionable affordances. *arXiv preprint arXiv:2608.05215*, 2026.

Sangmim Song, Sarath Kodagoda, Marc Carmichael, and Karthick Thiyagarajan. Overcoming visual clutter in vision language action models via concept-gated visual distillation. *arXiv preprint arXiv:2603.10340*, 2026.

Wenxuan Song, Jiayi Chen, Pengxiang Ding, Yuxin Huang, Han Zhao, Donglin Wang, and Haoang Li. Ceed-vla: Consistency vision-language-action model with early-exit decoding. *arXiv preprint arXiv:2506.13725*, 2025.

Hongyu Wang, Chuyan Xiong, Ruiping Wang, and Xilin Chen. Bitvla: 1-bit vision-language-action models for robotics manipulation. *arXiv preprint arXiv:2506.07530*, 2025.

Runze Wang, Yuqian Fu, Yu Li, Tao Lin, Tianwen Qian, Mohamed Elhoseiny, Bo Zhao, Yanwei Fu, Yu-Gang Jiang, and Xiangyang Xue. Afford-vla: Action-aligned visual planning via internalized affordance. *arXiv preprint arXiv:2605.24203*, 2026a.

Zhaoning Wang, Yi Zhong, Jiawei Fu, Henrik I Christensen, and Jun Gao. Afun: Towards an affordance foundation model for functionality understanding. *arXiv preprint arXiv:2606.02551*, 2026b.

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. In *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024.

Wenli Xiao, Haotian Lin, Andy Peng, Haoru Xue, Tairan He, Zhengyi Luo, Yuqi Xie, Fengyuan Hu, Jim Fan, Guanya Shi, et al. Self-improving vision-language-action models with data generation via residual rl. In *International Conference on Learning Representations*, volume 2026, pp. 13209–13236, 2026.

Rongtao Xu, Jian Zhang, Minghao Guo, Youpeng Wen, Haoting Yang, Min Lin, Jianzheng Huang, Zhe Li, Kaidong Zhang, Liqiong Wang, et al. $a_\{0\}$: An affordance-aware hierarchical model for general robotic manipulation. In *IEEE/CVF International Conference on Computer Vision*, pp. 13491–13501. IEEE, 2025.

Jinhui Ye, Fangjing Wang, Ning Gao, Junqiu Yu, Yangkun Zhu, Bin Wang, Jinyu Zhang, Weiyang Jin, Yanwei Fu, Feng Zheng, et al. Spatially guided training for vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 32395–32434, 2026.

Wencheng Ye, Tianshi Wang, Lei Zhu, Fengling Li, Guoli Yang, and Hengtao Shen. Actdistill: General action-guided self-derived distillation for efficient vision-language-action models. *arXiv preprint arXiv:2511.18082*, 2025.

Qize Yu, Jiadi You, Yuran Wang, Jiaqi Liang, Bowen Ping, Yang Tian, Yue Chen, Minghong Cai, Zeying Gong, Ruihai Wu, et al. Affordancevla: A vision-language-action model empowering action generation through affordance-aware understanding. *arXiv preprint arXiv:2606.06155*, 2026.

Ruiteng Zhao, Wenshuo Wang, Yicheng Ma, Xiaocong Li, Francis EH Tay, Marcelo H Ang Jr, and Haiyue Zhu. Fd-vla: Force-distilled vision-language-action model for contact-rich manipulation. *arXiv preprint arXiv:2602.02142*, 2026.

Zhide Zhong, Haodong Yan, Junfeng Li, Junjie He, Tianran Zhang, and Haoang Li. Vla-opd: Bridging offline sft and online rl for vision-language-action models via on-policy distillation. *arXiv preprint arXiv:2603.26666*, 2026.

A2DB: AFFORDANCE-BOTTLENECK DISTILLATION FOR EFFICIENT VISION-LANGUAGE-ACTION MODELS

APPENDIX CONTENTS

A Related Work **14**

A.1 Vision-Language-Action Models 14

A.2 Affordance-Centric Robot Learning 14

A.3 Knowledge Distillation and Efficient VLA Models 14

A.4 Spatial Intermediate Representations for Action Generation 14

B Algorithm Pseudo-Code **15**

C Discussion and Limitations **15**

D Affordance Annotation Visualization **16**

E Experimental Setup and Module Details **17**

E.1 Benchmark, Affordance Label Generation, and Evaluation Protocol 17

E.2 Teacher Prefix and Cross Architecture Token Matching 17

E.3 Module Architecture and Object Slots 18

E.4 Phase II Sampling and Cycle Optimization 18

E.5 Model Optimization and Seeds 19

F Additional Experiments **19**

F.1 SVD-based Affordance Subspace Verification 19

F.2 Phase II Mechanism Analysis 20

F.3 Failure Case Taxonomy 22

F.4 Affordance Label Corruption 22

G Simulation Deployment on Libero Benchmarks **23**

A Related Work

A.1 Vision-Language-Action Models

Vision-Language-Action (VLA) models connect visual observations, language instructions, and robot actions within a unified policy, making them a natural foundation for generalist manipulation. Large-scale video pretraining improves visual robot control by exposing policies to diverse interaction traces, hierarchical action modeling extends open-world manipulation with structured action generation, and diffusion-based action foundations increase the capacity of robotic policies for continuous control (Wu et al., 2024; Li et al., 2025b; Liu et al., 2025). Recent self-improving VLA models further show that post-training and generated interaction data can refine action policies beyond the initial supervised model (Xiao et al., 2026). These works mainly study how to increase the coverage, capacity, or adaptability of full VLA systems. A2DB addresses a complementary problem: how to compress a capable teacher VLA into a compact student by transferring only the interaction knowledge needed for downstream action.

A.2 Affordance-Centric Robot Learning

Affordance has provided a compact way to describe the actions that an object, scene, or state makes possible, which is why it remains central to robotic manipulation (Kroemer et al., 2021). Recent VLA research has moved affordance from an external cue toward an internal representation: CoA-VLA (Li et al., 2025a) uses visual-textual affordance chains, Affordance2Action (Liu et al., 2026a) grounds task-conditioned scene-level affordances, AffordanceVLA (Yu et al., 2026) and Afford-VLA (Wang et al., 2026a) use affordance-aware understanding or internalized affordance for action generation and planning, PALM (Liu et al., 2026b) combines affordance reasoning with task progress, and AFUN (Wang et al., 2026b) extends affordance modeling toward broad functionality understanding. VLAff further learns a unified set of visual, grasp, and trajectory affordances from egocentric human videos, emphasizing embodiment-agnostic actionable structure for downstream manipulation (Oh et al., 2026). These methods establish affordance as a useful representation for policy reasoning, human-to-robot transfer, or action generation. In contrast, A2DB uses affordance as a distillation bottleneck: the affordance variables are decoded from a frozen teacher, compressed into a low-dimensional subspace, and used to determine which teacher knowledge is transferred during VLA model compression.

A.3 Knowledge Distillation and Efficient VLA Models

Efficient VLA deployment has motivated distillation over actions, intermediate features, reasoning traces, and decoding depth. ActDistill (Ye et al., 2025), CEED-VLA (Song et al., 2025), GLaD (Guo et al., 2025), LatBot (Li et al., 2025c), and VITA-VLA (Dong et al., 2025) respectively target action-guided, consistency or early-exit, geometric-latent, universal-action, and action-expert transfer. A comparison of VLA policy distillation further distinguishes VLA-to-RL refinement, RL-generated supervision, and privileged-teacher transfer, exposing a generalization–reliability trade-off (Jang et al., 2025). Recent methods target complementary bottlenecks: BitVLA uses ternary weights with quantize-then-distill training (Wang et al., 2025); RLRC recovers pruned VLAs through supervised fine-tuning and reinforcement learning (Chen et al., 2026b); RT-VLA distills visual, query, waypoint, and language outputs for real-time driving (Huang et al., 2026); CoTinyVLA distills episode-level Plan and chunk-level Think traces (Lee et al., 2026a); and Act with Intent distills behavior-level objectives into the action decoder (Lee et al., 2026b). Other works (Jülg et al., 2025; Chen et al., 2026a; Zhao et al., 2026; Song et al., 2026; Jeon et al., 2026; Zhong et al., 2026) refine VLA generalists with on-policy RL, influential-data selection, force distillation, concept gating, shallow flow decoding, or offline-to-online distillation. Unlike these methods that preserve selected outputs, representations, reasoning traces, or compression properties, A2DB uses affordance as the transfer interface, restricting teacher knowledge to action-relevant interaction structure.

A.4 Spatial Intermediate Representations for Action Generation

Spatial intermediate representations are important for VLA policies because action generation depends on where interaction should occur, which object parts are functionally relevant, and how visual

tokens align with control outputs. Selective visual representations (Eftekhar et al., 2024) show that embodied policies benefit from task-conditioned feature filtering, manipulation-centric pretraining (Jiang et al., 2025) emphasizes robot-relevant representation learning, and spatially guided training (Ye et al., 2026) or spatial forcing (Li et al., 2026) aligns intermediate visual tokens with geometry and action heads. Affordance-based manipulation methods (Bahl et al., 2023; Li et al., 2025a; Xu et al., 2025) also indicate that spatial and functional interaction cues can serve as intermediate representations before action prediction. These studies demonstrate the value of spatial structure for action generation, but they mainly use such representations to improve training or policy reasoning. A2DB instead uses a teacher-side affordance subspace as a selective distillation channel and couples it with an affordance-conditioned action expert for compact VLA model compression.

B Algorithm Pseudo-Code

Algorithm 1 summarizes the affordance-bottleneck two-stage distillation paradigm of A2DB. Sub-algorithm (a) creates affordance interaction targets, trains the teacher probe on the frozen two-view π_0 teacher, extracts the scaled leading SVD directions, and distills a student VLM that receives only the external view and instruction. Sub-algorithm (b) freezes that student VLM, samples teacher and noisy student action chunks, and jointly optimizes behavior cloning, sliced Wasserstein distillation, and the deterministic cycle consistency loss. At deployment, A2DB retains only the distilled student VLM and affordance-conditioned action expert.

Algorithm 1 A2DB: affordance-bottleneck two-stage VLA model distillation.

(a) Teacher probe and Student VLM distillation

Require: Demonstrations $\mathcal{D} = \{(\mathbf{x}_b, \mathbf{x}_w, \ell, \mathbf{s}, \mathbf{a}_{1:H})\}$; frozen teacher π_0; affordance annotator $\mathcal{A}$; bottleneck d_A

Ensure: Trained student f_{θ^*} and projection $\mathbf{P}_A$

$\mathcal{Y} \leftarrow \mathcal{A}(\mathcal{D})$ ▷ masks, contacts, motion directions
Freeze π_0; initialize teacher probe parameters and affordance decoders
for each minibatch $\mathcal{B} \subset \mathcal{D}$ **do**
  $\mathbf{Z}^T \leftarrow \pi_0(\mathbf{x}_b, \mathbf{x}_w, \ell)$; $\widehat{\mathcal{Y}} \leftarrow \mathrm{Probe}(\mathbf{Z}^T)$
  $\mathcal{L}_{\mathrm{probe}} \leftarrow \lambda_m \mathcal{L}_m + \lambda_c \mathcal{L}_c + \lambda_v \mathcal{L}_v$
  Update probe parameters using $\nabla \mathcal{L}_{\mathrm{probe}}$
end for
$\mathbf{W}_0 \leftarrow$ first probe-layer weights; $\mathbf{W}_0 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \leftarrow \mathrm{SVD}(\mathbf{W}_0)$
$\mathbf{P}_A \leftarrow \mathbf{U}_{:,1:d_A}\mathbf{\Sigma}_{1:d_A,1:d_A}$
Initialize student f_θ, affordance head, and token map $\mathbf{W}_S$
for each minibatch $\mathcal{B} \subset \mathcal{D}$ **do**
  $\mathbf{Z}^T \leftarrow \pi_0(\mathbf{x}_b, \mathbf{x}_w, \ell)$; $(\mathbf{Z}^S, \mathbf{e}) \leftarrow f_\theta(\mathbf{x}_b, \ell)$
  $\mathcal{L}_{\mathrm{I}} \leftarrow \alpha \mathcal{L}_{\mathrm{sel}}(\mathbf{Z}^S\mathbf{W}_S, \mathbf{Z}^T\mathbf{P}_A) + \beta \mathcal{L}_{\mathrm{aff}}(\mathbf{e}, \mathcal{Y}) + \gamma \mathcal{L}_{\mathrm{lang}}$
  Update $(\theta, \mathbf{W}_S)$ using $\nabla \mathcal{L}_{\mathrm{I}}$
end for
return $\theta^*, \mathbf{P}_A$

(b) Affordance-conditioned action expert training

Require: Frozen f_{θ^*} and π_0; demonstrations $\mathcal{D}$; horizon H; $K = 4$, $P = 64$, $\sigma = 0.05$

Ensure: Trained action expert g_{ψ^*} and inverse model r_{ω^*}

Freeze f_{θ^*}; initialize g_ψ and r_ω
for each minibatch $\mathcal{B} \subset \mathcal{D}$ **do**
  $\mathbf{e} \leftarrow f_{\theta^*}(\mathbf{x}_b, \ell)$; $\widehat{\mathbf{a}}^{S,0} \leftarrow g_\psi(\mathbf{s}, \mathbf{e}; \mathbf{0})$
  $\{\mathbf{a}^{T,k}\}_{k=1}^K \leftarrow \pi_0(\mathbf{x}_b, \mathbf{x}_w, \ell, \mathbf{s})$ ▷ online teacher samples
  $\{\widehat{\mathbf{a}}^{S,k}\}_{k=1}^K \leftarrow g_\psi(\mathbf{s}, \mathbf{e}; \boldsymbol{\xi}_k)$, $\boldsymbol{\xi}_k \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$
  $d^S_{b,p,:} \leftarrow \mathrm{sort}_k[\mathbf{u}_p^\top \mathrm{vec}(\widehat{\mathbf{a}}_b^{S,k})]$ and analogously $d^T_{b,p,:}$
  $\mathcal{L}_{\mathrm{swd}} \leftarrow (BPK)^{-1} \sum_{b,p,k} (d^S_{b,p,k} - d^T_{b,p,k})^2$ ▷ $\mathbf{u}_p$ are normalized Gaussian directions
  $\mathcal{L}_{\mathrm{bc}} \leftarrow \|\widehat{\mathbf{a}}^{S,0} - \mathbf{a}\|_2^2/(Hd_a)$
  $\mathcal{L}_{\mathrm{cyc}} \leftarrow \|r_\omega(\widehat{\mathbf{a}}^{S,0}) - \mathbf{e}\|_2^2$ ▷ backpropagate through r_ω and g_ψ
  $\mathcal{L}_{\mathrm{II}} \leftarrow \delta \mathcal{L}_{\mathrm{bc}} + \epsilon \mathcal{L}_{\mathrm{swd}} + \zeta \mathcal{L}_{\mathrm{cyc}}$
  Update (ψ, ω) using $\nabla \mathcal{L}_{\mathrm{II}}$
end for
Deploy $\widehat{\mathbf{a}}_{1:H} = g_{\psi^*}(\mathbf{s}, f_{\theta^*}(\mathbf{x}_b, \ell))$
return f_{θ^*}, g_{ψ^*} ▷ teacher, probe, and inverse model are unused

C Discussion and Limitations

Discussion A2DB's evaluation results support affordance as a selective transfer interface rather than merely an auxiliary prediction target. At the same student scale, A2DB improves average success from 65.1% under standard knowledge distillation to 76.7% across the four LIBERO suites, with 0.29 B parameters and 38.6 ms latency (Table 1). Phase I analyses narrow this gain to the structure of the transferred subspace: leading probe-weight directions outperform trailing, orthogonal-complement, random, Jacobian, PLS, CCA, and PCA-based alternatives (Tables 7 and 8), while removing singular-value scaling costs 3.8 percentage points (Table 9) and performance approaches a plateau at $d_A = 128$ (Figure 3). Phase II analyses provide complementary evidence that this

representation must be translated through distribution-aware action learning: the selected $K = 4$, $P = 64$, and $\sigma = 0.05$ regime is near saturation, whereas removing action distillation, cycle consistency, or affordance-target alignment causes marked degradation (Figure 8; Tables 10 and 11). Affordance predictions, attention maps, and simulated trajectories further connect these quantitative gains to object selection, contact, and motion cues.

Limitations Current evidence remains confined to simulated LIBERO tasks, and most mechanism controls are reported only on `libero_spatial`; the results therefore do not establish transfer to real robots or unseen embodiments, nor robustness to appearance, viewpoint, clutter, and spatial shifts. The affordance subspace controls show that probe-weight SVD is the strongest tested extraction rule, but they do not prove that first-layer singular directions define a unique, invariant, or causal affordance subspace. Likewise, the inverse-model controls support aligned cycle supervision, yet reconstructing the affordance embedding does not guarantee semantic identifiability or action diversity. Training also depends on a large frozen π_0 teacher and automatic detection, segmentation, and tracking, although deployment removes these components. Label-corruption experiments expose this dependence, with severe perturbations causing large drops and the strongest random-mask and object-swap settings reaching 0.0% success (Figure 10). Finally, the π_0 teacher and A2DB student's trajectory comparisons are qualitative, and the 50-episode failure review does not measure temporal deviation, recovery, or physical safety. Future work should add broader simulated and real-world shifts, and calibrated recovery and safety tests.

D Affordance Annotation Visualization

Figure 7 visualizes the automatic affordance annotation pipeline on `libero_spatial`. Across object transfer, drawer closing, and multi-object placement instructions, Grounding DINO identifies the task-relevant objects and SAM2 provides their spatial supports. The derived contact keypoints and motion arrows encode the corresponding interaction locations and directions, including distinct object-specific motions in scenes with several referents. These signals provide the mask, contact, and motion targets used to train the teacher probe and the student affordance head.

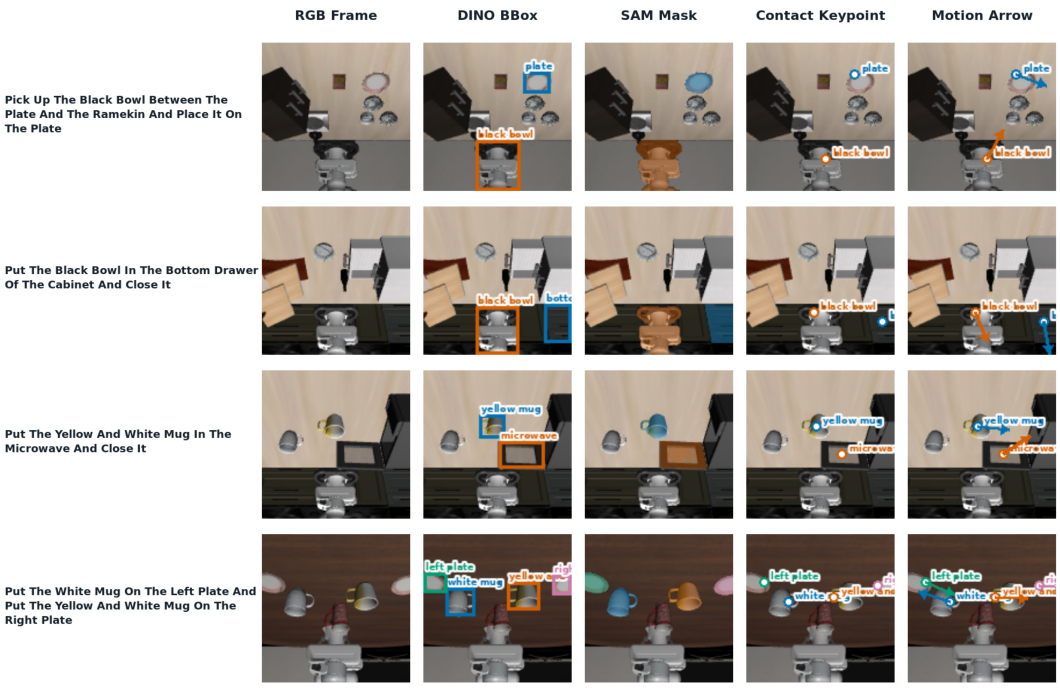


Figure 7: **Affordance annotation visualization on `libero_spatial`**. Each row shows an RGB frame followed by Grounding DINO bounding boxes, SAM2 masks, Contact Keypoints, and Motion Arrows for the task-relevant objects.

E Experimental Setup and Module Details

E.1 Benchmark, Affordance Label Generation, and Evaluation Protocol

Benchmarks We evaluate A2DB on LIBERO (Liu et al., 2023), a simulated robot manipulation benchmark built on the MuJoCo physics simulator. The four reported suites are `libero_spatial`, `libero_object`, `libero_goal`, and `libero_long`, which cover spatial, object, goal, and long-horizon tasks. Each suite contains 10 tasks with 50 demonstration episodes per task. We train and evaluate a separate distilled student model for each suite using its 500 demonstrations; across the four independent runs, this amounts to 2,000 demonstrations in aggregate. Task instructions are derived from LIBERO filenames by removing every leading uppercase scene identifier before replacing underscores with spaces.

Affordance Label Generation We use every `agentview_rgb` frame in an episode for tracking but store one target set anchored at the first frame. Grounding DINO uses box and text thresholds 0.30 and 0.25, respectively; duplicate detections of the same name are using an IoU threshold of 0.8, and the detection with highest confidence is retained for each object phrase. SAM2 produces the first frame masks, and CoTracker tracks eight deterministic mask interior points per object. Masks are resized to 224×224, contacts are normalized by $[w-1, h-1]$, and motion is stored as a two dimensional unit vector.

Evaluation Protocol Each suite uses its official initial states with environment seed 42 and 50 rollouts per task. Episodes begin with 10 idle steps. `libero_spatial`, `libero_object`, `libero_goal`, and `libero_long` suites allow 220, 280, 300, and 520 control steps, respectively. LIBERO provides one external view and one wrist view. The teacher receives both valid views during training, whereas the student receives only the external view in Phase I, Phase II, and evaluation. Both views are resized with zero padding that preserves aspect ratio to 224×224 before they are passed to the corresponding model path. The eight-dimensional state concatenates end effector position, axis angle orientation, and two gripper coordinates. The policy predicts $H = 50$ seven-dimensional actions, executes the first five, and then replans.

Baselines and Metrics We compare A2DB with the frozen π_0 teacher, SmolVLA, behavior cloning, and standard knowledge distillation, then isolate its components through loss removal and subspace replacement. The primary metric is task success rate, computed as:

$$\text{Success} = \frac{1}{N_{\text{ep}}} \sum_{i=1}^{N_{\text{ep}}} \mathbf{1}\left[\text{done}_i = 1\right], \tag{21}$$

where N_{ep} counts evaluated episodes and done_i is the environment success signal.

E.2 Teacher Prefix and Cross Architecture Token Matching

Teacher Prefix The frozen π_0 teacher serializes 256 external-image tokens, 256 wrist-image tokens, 256 zero tokens for the absent second wrist view, and 48 padded PaliGemma instruction positions, giving 816 positions of width 2048. Masks retain the two valid images and instruction positions; spatial distillation uses the external grid, while language alignment uses the teacher instruction span reported in Table 4.

Student Tokenization Each student processes one external image and the complete task instruction with its native processor. Features are projected to width 512 and passed through the common four-layer fusion transformer. Image identifiers define the visual mask; retokenizing the raw task and selecting its final serialized occurrence defines the instruction mask, excluding markers, padding, and template text. Table 4 summarizes the backbone interfaces.

Spatial and Language Matching Spatial matching selects the final global image group when a processor emits several groups, projects it with $\mathbf{W}_S$, resizes it to the teacher 16×16 grid, and computes masked MSE under the teacher external-view mask. Language matching projects student instruction tokens and teacher positions 768–815 with $\mathbf{W}_S$ and $\mathbf{P}_A$, pools them with validity masks, and uses cosine loss. Teacher instruction states use both views, whereas student states use the external view and task text.

Table 4: Token interfaces used for cross-architecture matching.

VLM	Width	Input serialization	Mapping
π_0 (`SigLIP+Gemma`)	2048	External + wrist images, absent-wrist placeholder, and 48 PaliGemma instruction tokens; $N_T = 816$.	External $16 \times 16 \to \mathbf{P}_A$; pool positions 768–815.
`SmolVLM`	576	Idefics3: 17 groups $\times$ 64 image tokens; task tokens follow the final global group.	Final 8×8 global grid $\to 16 \times 16$; pool task tokens.
`InternVL2-1B`	896	448-pixel base image, 256 visual tokens, then task instruction.	Native 16×16 grid; pool task tokens.
`Qwen2-VL-2B`	1536	Native chat template with vision markers and dynamic patch-14 grid (merge 2).	Use `image_grid_thw`; resize to 16×16; drop system text.

Table 5: A2DB core modules for the default student VLM (`SmolVLM`) with $d_A = 128$.

Module	Input	Operations	Output
Teacher probe	$B \times 816 \times 2048$ + mask	Masked query pooling; $2048 \to 256 \to 128$ GELU MLP; expand to 4×128.	Global embedding + four slots.
Probe SVD	$\mathbf{W}_0 \in \mathbb{R}^{2048\times256}$	Reduced float32 SVD; retain and scale 128 leading vectors.	$\mathbf{P}_A \in \mathbb{R}^{2048\times128}$.
Student fusion	Base image + task tokens + masks	Project to 512; four pre-norm Transformer layers (8 heads, FFN 2048, GELU, dropout 0.1); backbone frozen.	$B \times N_S \times 512$ tokens.
Student affordance head	Fused tokens + mask	Query pooling; $512 \to 256 \to 128$ GELU MLP; expand to 4×128.	128-D global embedding + four features.
Object decoders	128-D object feature	Mask $128 \to 14 \times 14 \to 224 \times 224$; contact $128 \to 2$ sigmoid; motion $128 \to 2$ tanh + ℓ_2 norm.	Mask logits, normalized contact, unit motion.
Action expert	State $B \times 8$ + affordance $B \times 128$	State $8 \to 256$; 50 queries + sinusoidal positions; four causal self-attention layers, FiLM, one-token cross-attention, $256 \to 1024 \to 256$ GELU FFN.	$B \times 50 \times 7$ chunk.
Inverse model	Flattened chunk $B \times 350$	$350 \to 256 \to 256 \to 128$ MLP; GELU after the first two-layers.	128-D reconstructed affordance.

E.3 Module Architecture and Object Slots

Module Architecture Table 5 summarizes the omitted dimensions. The teacher probe uses float32 masked query pooling, a $2048 \to 256 \to 128$ GELU MLP, and one reduced float32 SVD to obtain $\mathbf{P}_A = \mathbf{U}_{:,1:128}\,\mathrm{diag}(\boldsymbol{\sigma}_{1:128})$. No LayerNorm or explicit normalization precedes the first projection. The retained singular values range from 0.9821 to 1.3507 for the recorded run.

Object Slots Object phrases follow instruction order; detections are deduplicated, ranked by Grounding DINO confidence, copied to four slots, and zero padded. Validity masks exclude padded slots from all affordance losses, and decoder weights are shared across slots. The mask, contact, and motion weights are $(\lambda_m, \lambda_c, \lambda_v) = (1.0, 0.5, 0.3)$.

E.4 Phase II Sampling and Cycle Optimization

Teacher Sampling For each episode, the teacher generates $K = 4$ online π_0 action chunks with 10 flow-integration steps and keys $s + i \cdot 1009 + k$ ($s = 0$). Output transforms and LIBERO unnormalization are applied in the seven-dimensional action space.

Student Sampling and SWD The student also draws $K = 4$ samples by adding Gaussian noise ($\sigma = 0.05$) to each of the 50×256 action-token entries. SWD flattens each chunk to $\mathbb{R}^{350}$, uses $P = 64$ normalized Gaussian directions, sorts the four projections, and averages their squared difference; noise and projection keys are resampled each step.

Cycle Optimization Cycle consistency uses deterministic actions; the inverse MLP and action expert are trained jointly from scratch, with gradients through both r_ω and g_ψ. Thus it enforces recoverability under the learned inverse map but does not guarantee semantic identifiability or action diversity. Phase II evidence is provided by the sampling-sensitivity, action-distribution, and inverse-model controls summarized in Figure 8 and Tables 10 and 11.

Table 6: Default optimization settings.

Stage	Data	Optimizer	Details
Teacher probe	500 training episodes per suite	AdamW: lr 10^{-4}, $\beta = (0.9, 0.999)$, $\varepsilon = 10^{-8}$, wd 10^{-4}	$\lambda_{m,c,v} = (1.0, 0.5, 0.3)$.
Phase I	500 training demonstrations per suite	AdamW: lr 10^{-4}, $\beta = (0.9, 0.999)$, $\varepsilon = 10^{-6}$, wd 10^{-5}; clip 1.0	$(\alpha, \beta, \gamma) = (1.0, 0.5, 0.1)$.
Phase II	500 training demonstrations per suite	AdamW: lr 3×10^{-4}, $\beta = (0.9, 0.999)$, $\varepsilon = 10^{-8}$, wd 10^{-5}	$(\delta, \epsilon, \zeta) = (1.0, 0.3, 0.1)$; $K = 4$, $P = 64$.
Evaluation	50 rollouts / task	None	Seed 42; 5 actions replan.

E.5 Model Optimization and Seeds

Table 6 summarizes the default student VLM (`SmolVLM`) and teacher probe configurations. All optimizers use constant learning rates without a scheduler or warmup. The training seed is 0 for probe initialization, Phase I, Phase II, and teacher sampling; the simulator seed is 42. Only one seed is used, so success rates do not include uncertainty across seeds.

The A2DB code environment uses Python 3.11, PyTorch 2.7.1, JAX 0.5.3, Flax 0.10.2, Optax 0.2.8, Transformers 4.53.2, and Orbax Checkpoint 0.11.13.

F Additional Experiments

F.1 SVD-based Affordance Subspace Verification

The affordance subspace analyses test whether the affordance transfer benefit is tied to the leading, probe-derived teacher directions rather than to rank matching alone. Across the subspace controls, Top-d SVD reaches 76.4% success, compared with 71.6% for Bottom-d SVD, 69.4% for the orthogonal complement, and 67.8% for a random subspace. Alternative extraction rules show the same qualitative conclusion: Weight-SVD remains at 76.4%, whereas CCA, PLS, Jacobian, supervised PCA, PCA, and random selection range from 71.4% to 67.8%. Finally, retaining the leading directions but removing their singular-value weights reduces success to 72.6%, confirming that both direction selection and singular-value scaling contribute to the SVD-based affordance representation.

F.1.1 Teacher Subspace Controls

Table 7 tests whether the Phase I gains depend on the leading probe-derived singular directions rather than an arbitrary rank-matched projection. All variants use the same student architecture, training data, optimization schedule, and Phase I objectives; only the teacher projection changes. Top-d SVD achieves 76.4% success, exceeding Bottom-d SVD, the orthogonal-complement control, and the random subspace by 4.8, 7.0, and 8.6 percentage points, respectively. The ordering among these controls shows that rank matching alone is insufficient: trailing singular directions preserve more task-relevant structure than either a rank-matched basis constrained outside the top-d span or a random orthonormal basis, yet they remain less effective than the leading directions. The orthogonal complement exceeds the random subspace by only 1.6 percentage points, indicating that excluding the top-d span leaves only a modest advantage over an unstructured projection. Overall, Top-d SVD is the strongest evaluated teacher projection among these subspace variants.

Table 7: **Teacher subspace controls on `libero_spatial`.** All projections use $d_A = 128$ under the same Phase I training protocol.

Subspace Variant	Construction	d_A	SR % ↑	Δ pp
Top-d SVD (A2DB)	Leading singular directions	128	**76.4**	0.0
Bottom-d SVD	Trailing singular directions	128	71.6	-4.8
Orthogonal complement	Rank-matched basis orthogonal to top-d	128	69.4	-7.0
Random subspace	Random orthonormal basis	128	67.8	-8.6

F.1.2 Subspace Extraction Method Comparison

Table 8 compares alternative teacher-subspace extraction rules at the fixed dimension $d_A = 128$ under the same Phase I training protocol. Weight-SVD, which uses the leading singular directions of the probe weights, achieves the best success rate at 76.4%. The strongest alternative is CCA at 71.4% (-5.0 pp), followed by PLS at 71.2% (-5.2 pp) and Jacobian at 70.6% (-5.8 pp). Target-aware variance selection and unsupervised controls perform worse: supervised PCA reaches 69.2% (-7.2 pp), PCA 68.2% (-8.2 pp), and a random orthonormal basis 67.8% (-8.6 pp). Thus, under an identical training protocol, the probe-weight spectrum provides a more effective control subspace than local sensitivity, feature–affordance association, variance, or random selection criteria.

Table 8: **Subspace extraction methods on `libero_spatial`.** All methods use $d_A = 128$ under the same Phase I training protocol.

Extraction method	Signal used	d_A	SR % ↑	Δ pp
Weight-SVD (A2DB)	Probe-weight spectrum	128	**76.4**	0.0
Jacobian	Probe input sensitivity	128	70.6	-5.8
PLS	Feature–affordance covariance	128	71.2	-5.2
CCA	Feature–affordance correlation	128	71.4	-5.0
Supervised PCA	Target-aware feature variance	128	69.2	-7.2
PCA	Teacher-feature variance	128	68.2	-8.2
Random	Random orthonormal basis	128	67.8	-8.6

F.1.3 SVD Scaling Ablation

To separate the contribution of the selected teacher directions from the contribution of singular-value weighting, Table 9 compares the default scaled projection $\mathbf{U}_d\mathbf{\Sigma}_d$ with the unscaled projection $\mathbf{U}_d$. Both variants retain the same top-d_A directions and use the same data, optimization schedule, and Phase I objectives. The scaled projection reaches 76.4% success, whereas removing the singular-value scaling reduces success to 72.6%, a 3.8-percentage-point drop. Thus, the leading directions account for most of the subspace benefit, while singular-value weighting provides a further measurable improvement by preserving the relative strength of those directions.

Table 9: **SVD scaling ablation on `libero_spatial`.** Both variants use $d_A = 128$ and the same Phase I training protocol; only singular-value scaling differs.

SVD variant	Teacher projection	d_A	SR % ↑	Δ pp
Scaled (A2DB)	$\mathbf{U}_d\mathbf{\Sigma}_d$	128	**76.4**	0.0
Unscaled	$\mathbf{U}_d$	128	72.6	-3.8

F.2 Phase II Mechanism Analysis

The Phase II mechanism analyses decompose the student action expert pipeline into sampling, action-distribution distillation, and cycle-consistent inverse-model controls. Figure 8 identifies a compact operating regime at $K = 4$, $P = 64$, and $\sigma = 0.05$, where performance is at or within 0.2 percentage points of the best value observed in each sweep. Table 10 shows that removing action-distribution supervision or stochastic sampling reduces success from 76.4% to 63.8% or 70.6%,

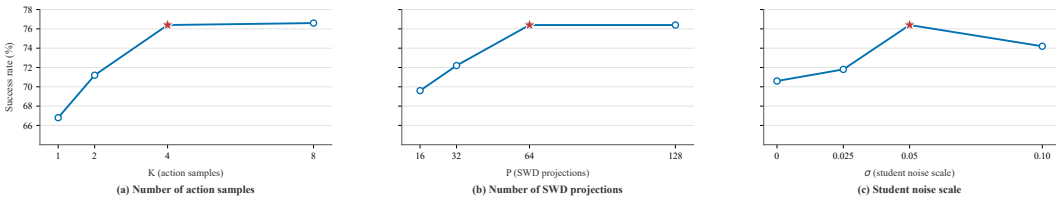


Figure 8: **Phase II sampling sensitivity on `libero_spatial`.** The three panels vary the number of student action samples K, the number of sliced-Wasserstein projections P, and the student noise scale σ, respectively, while holding the other quantities at their default values. The red star marks the default configuration $K = 4$, $P = 64$, and $\sigma = 0.05$.

respectively, while Table 11 shows further drops when cycle consistency or affordance-target alignment is disrupted, reaching 56.4% for shuffled targets. Taken together, the results attribute Phase II performance to the interaction of a well-calibrated sampler, teacher action-distribution guidance, and correctly aligned cycle-consistency supervision.

F.2.1 Sampling Sensitivity

Figure 8 shows that the three Phase II sampling controls have distinct saturation or optimum regimes. Increasing the number of student action samples improves success from 66.8% at $K = 1$ to 76.4% at $K = 4$, while $K = 8$ yields only a marginal increase to 76.6%, indicating that four samples capture nearly all of the available gain. The projection sweep follows the same pattern: success increases from 69.6% at $P = 16$ to 72.2% at $P = 32$ and 76.4% at $P = 64$, then remains unchanged at $P = 128$. In contrast, the noise sweep has a clear optimum at $\sigma = 0.05$ (76.4%); setting σ to 0, 0.025, or 0.10 lowers success to 70.6%, 71.8%, and 74.2%, respectively. Thus, the default $K = 4$, $P = 64$, and $\sigma = 0.05$ setting coincides with the projection and noise optima and is within 0.2 percentage points of the best observed sample-count result, providing a compact operating point for the remaining Phase II analyses.

F.2.2 Action-Sample Distillation Ablation

Table 10 compares the default A2DB student with controls that remove action-distribution supervision, remove stochasticity, or retain only behavior-cloning supervision. The default A2DB student reaches 76.4% success. Removing action distillation reduces success to 63.8%, a 12.6 percentage-point drop, while BC-only training reaches 62.4%, a 14.0 percentage-point drop. Setting the student noise scale to $\sigma = 0$ gives 70.6%, indicating that stochastic action sampling contributes 5.8 percentage points under the same protocol. The π_0 teacher reaches 86.8% and serves as a teacher reference, leaving a 10.4 percentage-point gap to the default student. Together, these results show that both teacher action-distribution guidance and stochastic sampling are important for transferring the teacher's behavior into the compact student.

Table 10: **Action-sample distillation ablation on `libero_spatial`.** All variants use the same matched states and instructions; the π_0 teacher is included as a reference.

Variant	SR % ↑
A2DB (default)	76.4
No action distillation	63.8
BC only	62.4
A2DB ($\sigma = 0$)	70.6
π_0 Teacher	86.8

F.2.3 Inverse-Model and Affordance-Target Controls

Table 11 isolates how inverse-model initialization, gradient routing, and affordance-target alignment affect cycle-consistent training. The joint-from-scratch variant (A2DB) achieves the highest success among the controllable variants at 76.4%. Removing cycle consistency lowers success to 67.6%,

an 8.8 percentage-point drop, while restricting gradients to the inverse model or using a pretrained inverse model gives 70.6%, 70.4%, and 71.2% for stop-gradient, frozen, and finetuned controls, respectively. The largest degradation occurs when affordance targets are shuffled, reducing success to 56.4%, which indicates that the cycle objective depends on a correctly aligned affordance representation rather than on inverse-model supervision alone.

Table 11: **Inverse-model and affordance-target controls for cycle consistency on `libero_spatial`.** All variants use the same Phase II data and objective weights. The controls vary inverse-model initialization, the gradient path from the cycle loss, or the alignment of the affordance target; the final column reports task success rate.

Variant	Inverse control	SR % ↑
No cycle consistency	Loss removed	67.6
Joint from scratch (A2DB)	Gradients through g_ψ and r_ω	76.4
Stop-gradient	Gradients to r_ω only	70.6
Pretrained, frozen	GT-action pretraining; freeze r_ω	70.4
Pretrained, finetuned	GT-action pretraining; joint finetuning	71.2
Shuffled affordance	Batch-shuffled $\mathbf{e}$ target	56.4

F.3 Failure Case Taxonomy

Figure 9 summarizes the manually reviewed failure modes from 50 failed episodes. Wrong motion prediction (28%), contact localization (24%), and object selection (22%) together account for 74% of the reviewed failures, concentrating the dominant residual errors in task-relevant interaction geometry. Action expert failures represent 16%, while simulator or control failures contribute 10%, separating downstream execution errors from failures in affordance prediction. This distribution identifies contact and motion precision as the principal remaining targets for improving the VLA.

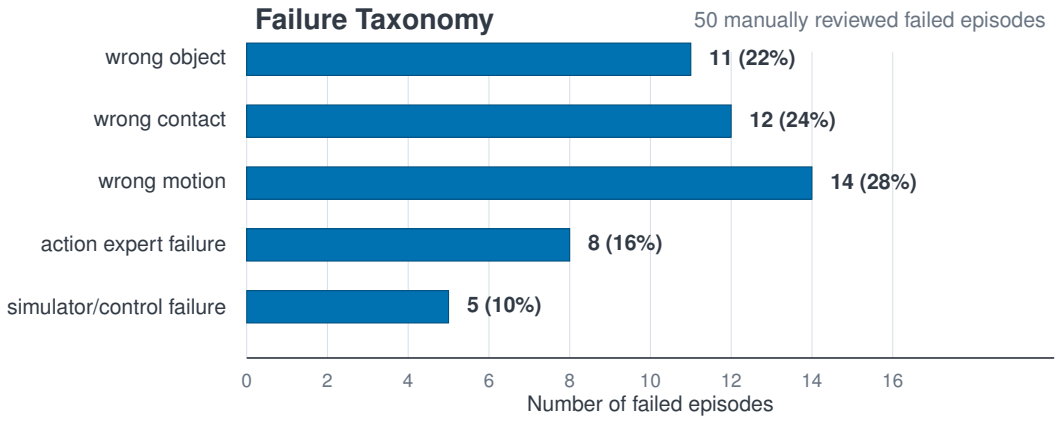


Figure 9: **Failure taxonomy over 50 manually reviewed failed episodes.** Bars report counts and shares for wrong object selection, wrong contact localization, wrong motion prediction, action expert failure, and simulator or control failure.

F.4 Affordance Label Corruption

Figure 10 measures the sensitivity of A2DB to controlled corruptions in automatically generated affordance labels on `libero_spatial`. Across all five perturbation types, success decreases from the uncorrupted 76.4% baseline as corruption strength increases, showing that the compact student policy remains sensitive to the spatial and semantic affordance supervision used during training. At the highest strengths, random masks and object-label swaps reduce success to 0.0%, while mask shifts, keypoint noise, and motion rotation reduce success to 19.4%, 29.7%, and 23.6%, respectively. These trends highlight the importance of consistent object identity, spatial support, contact localization, and motion direction for distilled robot manipulation behavior.

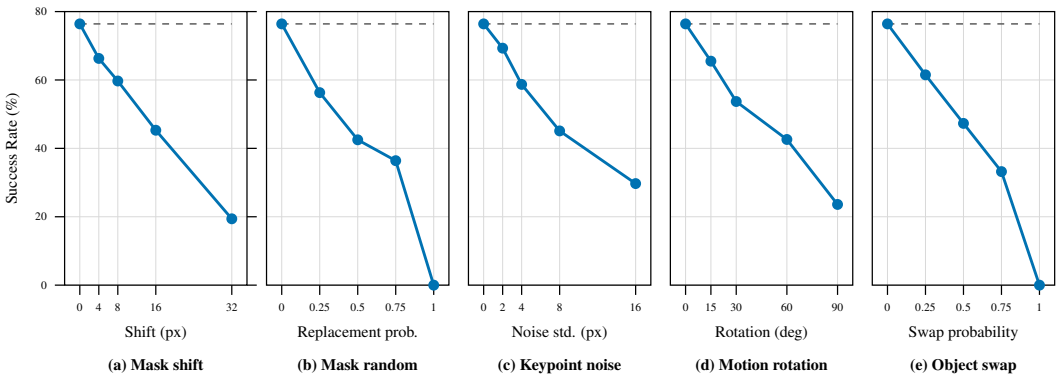


Figure 10: **Robustness to automatic affordance-label corruption on `libero_spatial`.** Each panel varies the strength of one corruption type and reports average task success rates. The horizontal axes retain their corresponding physical units or replacement probabilities, and the gray dashed line marks the 76.4% uncorrupted A2DB student baseline.

G Simulation Deployment on Libero Benchmarks

Figure 11 compares normalized execution snapshots of the π_0 teacher and A2DB student on representative tasks from the four LIBERO suites. From 0% to 100% rollout progress, the paired trajectories depict the main interaction sequence required by spatial placement, object selection, goal-conditioned manipulation, and long-horizon tasks. The student rollouts qualitatively follow the teacher's broad reaching, grasping, transport, and task-completion progression while retaining the compact deployment model, which provides qualitative context for the aggregate success rates reported in the main paper.

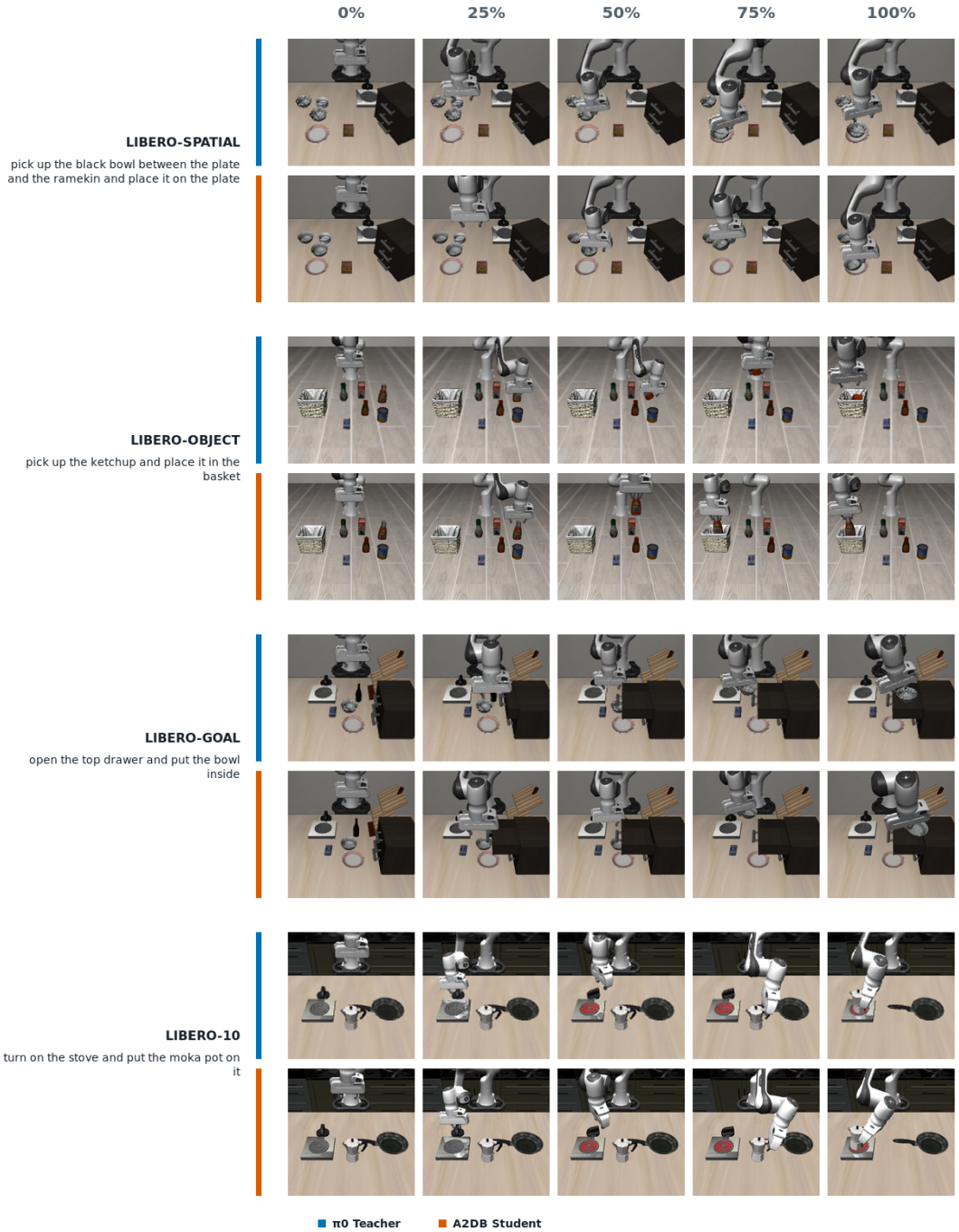


Figure 11: **Simulation deployment trajectories of the π_0 teacher and A2DB student on the four LIBERO suites**. Columns show normalized execution progress from 0% to 100%; blue and orange strips indicate teacher and student rollouts, respectively.