TEACHING VISION-LANGUAGE MODELS TO EVALUATE ACTIONS LEADS TO BETTER MANIPULATION POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action models rely on pre-trained vision-language backbones, yet further pre-training with robot-specific descriptive objectives yields little downstream benefit. We therefore investigate whether an evaluative objective that distinguishes viable actions from near-misses and failures transfers more effectively to control. As a first instantiation, we consider object grasping and introduce Action Assessment Pre-Training (AAPT), which trains the backbone to autoregressively generate a physical reasoning trace followed by a scalar viability score for each candidate grasp. After pre-training, the assessment interface is discarded and only the backbone initializes the downstream policy. We separate the objective from both its data and its optimizer. On identical candidates, a predictive control supervised by simulated collision and stability outcomes already improves downstream control. Replacing its binary targets with graded evaluation under matched optimization improves further, and ranking refinement yields 88.8 on zero-shot LIBERO-Plus and 4.54 on CALVIN ABC→D. Permuting complete trace-and-score targets across candidates causes performance to fall below no pre-training, showing that candidate-quality alignment matters. AAPT's transfer advantage also grows from $+1.8$ to $+6.9$ points as demonstrations become scarce. Discarded alternatives are therefore useful supervision, and their graded relative ordering is what transfers to manipulation.

1 INTRODUCTION

Vision-language-action models attach action decoders to a pre-trained vision-language backbone (Brohan et al., 2023a; Kim et al., 2024; Black et al., 2024a; 2025). The premise is that web-scale pre-training gives the backbone knowledge a policy can reuse, so stronger backbones should give stronger policies. Yet two observations are difficult to reconcile with this. First, scaling the backbone does not monotonically improve manipulation (Appendix J), while Xie et al. (2026) obtain competitive results with a small model containing no language backbone. Second, further pre-training on embodied visual question answering gives little benefit (Zhang et al., 2026).

Together, these observations motivate us to examine the pre-training objective itself. *Descriptive* objectives report what is present, as in captioning and visual QA (Liu et al., 2024; Qu et al., 2025; Chen et al., 2025; Remyx AI, 2024). *Predictive* objectives produce the correct continuation, such as a future observation or the next action, as in future-frame synthesis, world models and latent action modeling (Ye et al., 2024; Zhang et al., 2025c; Liu et al., 2026). Neither explicitly requires comparing the physical viability of candidate actions. An *evaluative* objective instead makes that comparison explicit by ranking candidates. We exam-

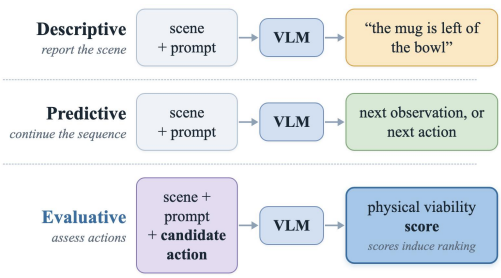

Figure 1. Three families of objectives. We study whether an evaluative objective that distinguishes viable actions from failures transfers to control.

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

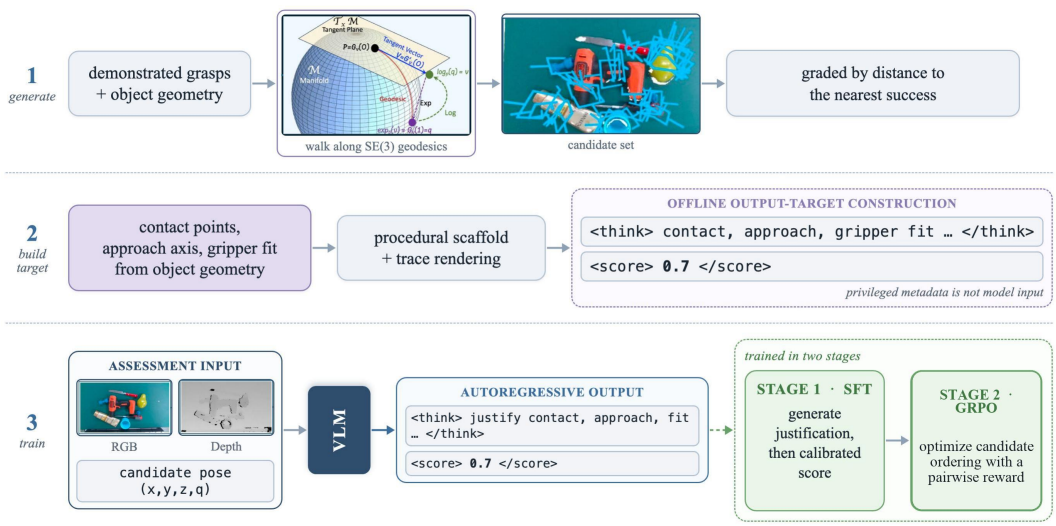

Figure 2. Action Assessment Pre-Training. Perturbed grasps are graded by proximity to the nearest success. Given RGB, optional depth and one candidate pose, the VLM autoregressively generates a physical reasoning trace followed by a score. Privileged geometry constructs targets but is never a model input. SFT calibrates scores, and GRPO with a ranking reward sharpens their ordering.

ine whether this graded relative judgment transfers more effectively than description, action generation or prediction of coarse outcomes.

Action Assessment Pre-Training (AAPT). As shown in Figure 2, AAPT pre-trains a VLM to compare candidate actions by physical viability. We instantiate this objective on object grasping, a tractable first case rather than a representative sample. In our setting, grasp quality can be assessed from the scene and pose without trajectory-level credit assignment, and object geometry often permits multiple valid grasps. Around each demonstrated success, AAPT synthesizes a graded continuum spanning near-misses at the success boundary and clear failures. Given the scene and one candidate pose, the VLM autoregressively generates a physical reasoning trace before estimating its viability score. After pre-training, the assessment interface is discarded and the backbone initializes the downstream policy. The same candidates also support a matched collision-and-stability predictor, which helps separate the effects of failure-aware data, graded evaluation and ranking refinement.

Our Contributions. We find that failures and near-misses discarded by demonstration pipelines can provide useful supervision for control when learned through an *evaluative* objective. Our contributions are fourfold.

- **An evaluative objective, AAPT.** It ranks perturbed grasp candidates by viability. Though pre-trained on grasp assessment, the resulting initialization improves policies on placing, stacking and instruction chaining.
- **Controlled objective and alignment comparisons.** Under matched SFT on identical candidates, graded assessment gives higher mean downstream performance than collision-and-stability prediction, and ranking refinement improves further. Permuting complete trace-and-score targets drops below no pre-training.
- **Transfer across pipelines, benchmarks and embodiments.** AAPT improves a minimal (Zhang et al., 2026) and a strong pipeline (Zhong et al., 2026) across LIBERO, LIBERO-Plus, CALVIN and SimplerEnv, reaching 4.54 on CALVIN and 88.8 on LIBERO-Plus.
- **A consistent gain profile.** Gains grow as demonstrations become scarce ($+1.8 \rightarrow +6.9$ points), at the hardest long-horizon steps and across held-out perturbations.

2 Related Work

Vision-language-action models. Most current policies place a pre-trained vision-language model between observation and action. Autoregressive designs emit discretized action tokens (Brohan

et al., 2023a; Kim et al., 2024), while continuous designs attach a regression head, flow-matching expert or diffusion decoder (Black et al., 2024a; 2025; Kim et al., 2025; Wang et al., 2026a). Others drop the language model (Xie et al., 2026) or the VLM entirely (Chi et al., 2023; Octo Model Team et al., 2024). We study how the backbone is trained before an action head is attached.

Pre-training objectives for control. Spatial grounding adds metric and relational structure (Qu et al., 2025), while embodied question answering turns teleoperation data or synthetic scenes into visual question-answer pairs (Chen et al., 2025; Remyx AI, 2024). Both are *descriptive*, and Zhang et al. (2026) report minimal downstream benefit. *Predictive* objectives instead produce a continuation, such as a sub-goal image (Zhao et al., 2025a), observation-action rollout (Cen et al., 2025b), enriched future observation (Zhang et al., 2025c), inverse dynamics (Tian et al., 2025; Wu et al., 2024), latent action vocabulary (Ye et al., 2024; Bu et al., 2025b) or unified observation-action denoising (Liu et al., 2026). Their gains often come with a jointly trained action pathway rather than the backbone alone. Prediction can also consume failed candidates. Our collision-and-stability control tests this case on the same data as AAPT.

Evaluative objectives. Robotics has long evaluated actions through grasp-quality models such as Contact-GraspNet (Sundermeyer et al., 2021), affordances in Where2Act (Mo et al., 2021), action values in QT-Opt (Kalashnikov et al., 2018) and energies in IBC (Florence et al., 2021). VIP (Ma et al., 2023b) and LIV (Ma et al., 2023a) are closer to our transfer setting. They learn value-oriented representations from action-free video and reuse them as features or rewards. AAPT instead conditions on an explicit 6-DoF candidate and learns the local ordering of successes, near-misses and failures. Existing evaluators also remain active during policy learning or deployment. Interactive reward functions (Huang et al., 2022) supervise policy learning, while V-GPS (Nakamoto et al., 2024) and RoVer (Dai et al., 2025) rerank actions at deployment. AAPT discards the assessor before downstream training and transfers only the backbone. It is therefore orthogonal to inference-time reasoning in language (Hancock et al., 2025), image space (Zhao et al., 2025a) or action space (Zhong et al., 2026). Concurrently, Peng et al. (2026) train a world model on failure rollouts and mask the imitation loss on failed actions rather than fitting them. AAPT instead synthesizes graded alternatives around successful actions and optimizes their relative viability without collecting rollouts.

3 METHODOLOGY

Our hypothesis is that a backbone may transfer more effectively to control when pre-training requires it to distinguish viable actions from near-misses and failures. AAPT tests this by constructing a graded candidate continuum, requiring the VLM to justify each assessment before scoring it. Concretely, the VLM receives an RGB image of the scene, a depth image when one is available, and a textual encoding of a candidate 6-DoF grasp pose as a position and quaternion. It then autoregressively generates a physical justification followed by a scalar score $s \in [0, 1]$. More specifically, it maps $x = (I_{\mathrm{RGB}}, I_{\mathrm{depth}}, g)$ to $y = (z_{\mathrm{phys}}, \texttt{<score>}, s)$, where z_{phys} must state the physical evidence supporting the assessment before the score token can appear. The trace is therefore an output, never an input. Appendix A details candidate construction, Appendix B details target generation, and Appendix B.4 gives the optimization and score-parsing protocol.

Step 1. Constructing and grading candidate sets. Each object contributes a set of successful grasps, $\mathrm{SE}(3)$ perturbations near the success boundary and clear failures. We construct these sets from HouseCat6D (Jung et al., 2024) and GraspNet-1Billion (Fang et al., 2020). This construction turns each annotated success into a local quality continuum. Near-boundary perturbations provide the hard comparisons that distinguish small contact errors, while dataset and uniform failures anchor the low-quality end. Binary labels would make subtle near-misses indistinguishable from gross failures, so we grade each candidate by its distance to the nearest known success. For $\mathbf{T} = (\mathbf{R}, \mathbf{t})$,

$$d_\tau(\mathbf{T}_1, \mathbf{T}_2) = \frac{1}{\tau_R} \arccos\left(\mathrm{clip}\left(\frac{\mathrm{tr}(\mathbf{R}_1^\top \mathbf{R}_2) - 1}{2}, -1, 1 \right) \right) + \frac{\|\mathbf{t}_1 - \mathbf{t}_2\|_2}{\tau_t},$$
$$r^{\mathrm{gt}} = \frac{1}{1 + d_{\min}/\tilde{d}}, \qquad \tau_R = 0.02 \text{ rad}, \quad \tau_t = 0.02 \text{ m}, \tag{1}$$

where $d_{\min}$ is the smallest distance from the candidate to any successful pose in the scene and $\tilde{d}$ is the median over that scene's failures. The minimum accommodates multiple valid grasps, while the median normalizes scene scale. The result is a relative quality target, not a calibrated success

probability. Section 4 validates it in simulation, and Appendix A gives the sampling schedule, dataset ratio and annotation limitations.

Step 2. Grounding the score in physical reasoning traces. To encourage the VLM to base its score on contact geometry, we supervise a short physical reasoning trace before every score. An offline procedural scaffold computes object shape, pose, approach, contact and gripper fit, and Qwen3-VL-8B renders it as z_{phys}. Because the justification is generated before the score, the score is autoregressively conditioned on the generated trace rather than followed by a post hoc explanation. The privileged geometry, scaffold and target reward construct the output target only. They are never VLM inputs and are unavailable to the downstream policy. Appendix B gives the dataset-specific scaffold, rendering prompt and complete autoregressive sequence.

Step 3. Two-stage training. The two stages separate calibration from ordering. SFT provides pointwise supervision on the graded scale, while GRPO directly optimizes relative order with a pairwise reward. This separation lets us test what ranking refinement adds beyond calibration and whether its contribution grows as candidate sets become denser.

Stage 1, calibration. The first stage is standard supervised fine-tuning. The model prompt contains only the scene and candidate pose. Under next-token prediction, the target sequence is z_{phys}, followed by `<score>` and r^{gt}. At inference no trace is supplied. The VLM must generate its justification first and then produce the score conditioned on that text.

Stage 2, sharpening the ordering. Pointwise calibration does not directly optimize candidate ordering. We therefore adopt a ranking reward with GRPO (Wu et al., 2025; Liu et al., 2025b) to complete generated sequences. Only the generated scalar score enters the reward. No reference trace enters the rollout context. Under the Thurstone-Gaussian model (Thurstone, 1927; Wu et al., 2025),

$$p_{\mathrm{gt}}(g_i \succ g_j) = \sigma\big(\kappa(r_i^{\mathrm{gt}} - r_j^{\mathrm{gt}})\big)\,,$$
$$p_{\mathrm{pred}}(g_i \succ g_j) = \Phi((s_i - \bar{s}_j)/\sigma_{\mathrm{comb}})\,, \qquad \sigma_{\mathrm{comb}} = \max\Big(\sqrt{\sigma_i^2 + \sigma_j^2 + \gamma}, \lambda\Big)\,. \tag{2}$$

Here $\bar{s}_j$ and σ_j are the rollout mean and standard deviation for g_j, σ_i is the standard deviation for g_i, and Φ is the standard normal CDF. The constants κ, γ and λ control preference sharpness and variance stabilization. The pairwise reward measures agreement between these preferences,

$$r_{\mathrm{pair}}(g_i, g_j) = \max\big(0,\ 2\big(p_{\mathrm{gt}}\, p_{\mathrm{pred}} + (1 - p_{\mathrm{gt}})(1 - p_{\mathrm{pred}})\big) - 1\big), \tag{3}$$

and each candidate is compared with its K nearest neighbors in ground-truth rank. Restricting comparisons to nearby ranks focuses GRPO on ambiguous candidates around the success boundary rather than spending most updates on obvious pairs. The rollout reward balances this ranking signal with pointwise calibration,

$$R(g_i) = \alpha\, R_{\mathrm{pair}}(g_i) + (1 - \alpha)\big(1 - |s_i - r_i^{\mathrm{gt}}|\big), \tag{4}$$

where R_{pair} averages over the selected comparisons. The second term discourages ranking refinement from drifting away from the graded scale learned by SFT. Appendix B.4 gives the score parser, rollout statistics, hyperparameters and checkpoint rule.

Transferring to a control policy. The AAPT checkpoint replaces the base VLM, while the assessment prompt, trace and score interface are discarded. Only the backbone transfers. We test this initialization in the minimal VLM4VLA pipeline (Zhang et al., 2026) and the strong ACoT-VLA pipeline (Zhong et al., 2026), both shown in Figure 3. The former provides a relatively direct test of backbone quality, while the latter tests whether the gain survives inside a stronger action architecture. AAPT's pre-training traces are not reused as downstream reasoning. VLM4VLA has no reasoning module, and ACoT-VLA trains separate explicit and implicit action reasoners from downstream demonstrations. Appendix E gives their architectures and losses.

4 Experimental setup

Evaluation design. The experiments ask three questions. Does the geometric reward reflect physical success, can AAPT recover the ordering it induces on unseen scenes, and does the resulting backbone improve control after the assessment interface is removed? The first two are diagnostics of the pre-training task. The downstream benchmarks provide the transfer test.

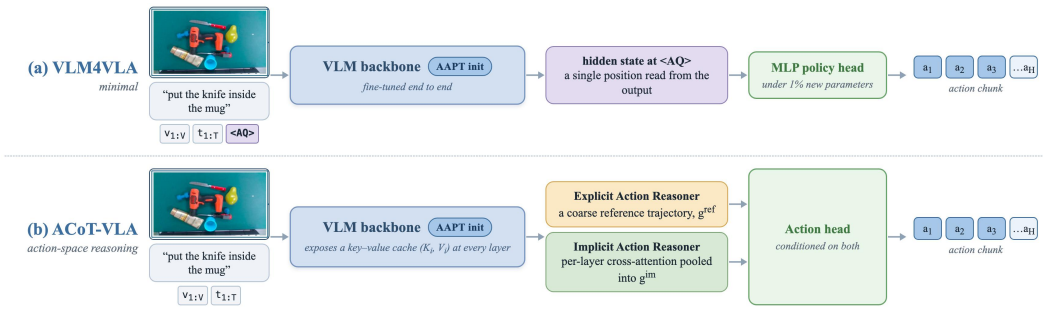

Figure 3. The two downstream pipelines, both initialized from the same backbone. **(a)** VLM4VLA (Zhang et al., 2026) appends one learnable action query token and decodes its hidden state through an MLP. **(b)** ACoT-VLA (Zhong et al., 2026) attaches two reasoners to the key-value cache, one proposing a coarse reference trajectory and one pooling per-layer cross-attention into a latent action prior, keeping the backbone representation directly involved even near ceiling.

Assessment diagnostics. We validate the target reward on 1000 held-out perturbations by comparing fixed geometric rewards with observed binary simulator success; no model prediction is used. We then evaluate the 500K-sample AAPT checkpoint on 200 held-out scenes with 20, 50 or 75 candidates per object. The main text reports pairwise ranking accuracy, while Appendix C gives the full ranking metrics and splits. Appendix A gives the reward-validation protocol.

Controlled transfer. Unless stated otherwise, AAPT uses Qwen3-VL-4B and the 500K corpus. The base and AAPT checkpoints initialize the same downstream architecture, and all policy parameters are then trained on the same demonstrations. Matched runs share the training seed and data order, so the controlled difference is the backbone initialization. We report means over three seeds and treat the training seed as the replication unit.

Downstream benchmarks. LIBERO (Liu et al., 2023) measures multi-task success. LIBERO-Plus (Fei et al., 2025) evaluates the same LIBERO checkpoints without fine-tuning under seven held-out perturbation factors. CALVIN ABC→D (Mees et al., 2022) measures the length of five-instruction chains completed in a held-out environment. SimplerEnv (Li et al., 2024b), reported in Appendix G, provides a descriptive second-embodiment check. Appendix F gives the training schedules and evaluation protocols. Appendix H gives the matched-seed results.

5 Results and Findings

Finding 1. The geometric reward tracks physical success. Before evaluating AAPT, we bin held-out candidates by their fixed geometric reward and execute each in simulation. A trial succeeds only with a 5–10 cm lift, stable hold and force-closure verification; no model prediction enters this analysis. Figure 4(a) shows that success rises monotonically across the five reward bins, from 2.9%

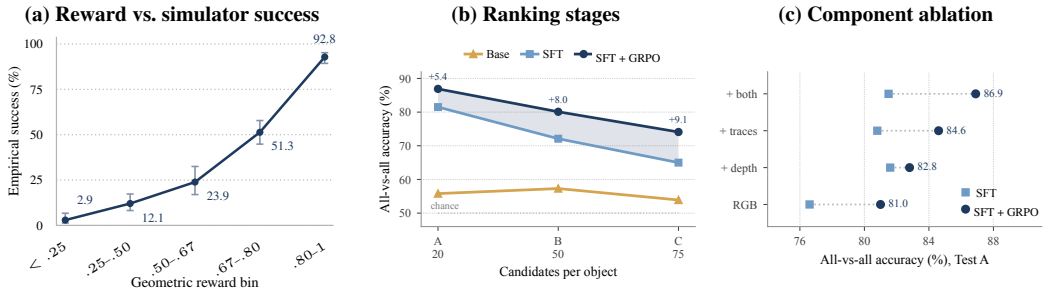

Figure 4. Assessment diagnostics. **(a)** Fixed geometric target reward r^{gt} versus empirical grasp success from 1000 stratified simulator executions; no model prediction is involved. **(b)** Ranking accuracy after SFT and GRPO as candidate sets grow. **(c)** AAPT component ablation on Test A. Appendix A gives the reward-validation protocol and Table 5 the exact values; Appendix C gives the complete metrics.

Table 1. LIBERO success rate (%). External rows are as published, grouped into direct policies and methods that add a prediction or world-model objective, and sorted by average within each group. Shaded rows are this work; the controlled AAPT gains are $+7.1$ points for VLM4VLA and $+0.4$ for ACoT-VLA. We re-implement both pipelines on Qwen3-VL-4B, and hence we include published results for ACoT-VLA, based on the SigLIP and Gemma-2B backbone of $\pi_{0.5}$. A subset of published methods is shown here; Table 12 in Appendix K gives the full comparison.

Method	Spatial	Object	Goal	Long	Avg
Direct policies					
SmolVLA (Shukor et al., 2025)	93.0	94.0	91.0	77.0	88.8
GR00T-N1 (Bjorck et al., 2025)	94.4	97.6	93.0	90.6	93.9
π_0 (Black et al., 2024a)	96.8	98.8	95.8	85.2	94.2
VLA-0 (Goyal et al., 2025)	97.0	97.8	96.2	87.6	94.7
GO-1 (Bu et al., 2025a)	96.2	97.8	96.0	89.2	94.8
Discrete Diffusion VLA (Liang et al., 2025)	97.2	98.6	97.4	92.0	96.3
MemoryVLA (Shi et al., 2026)	98.4	98.4	96.4	93.4	96.7
$\pi_{0.5}$ (Black et al., 2025)	98.8	98.2	98.0	92.4	96.9
OpenVLA-OFT (Kim et al., 2025)	97.6	98.4	97.9	94.5	97.1
VLA-Adapter (Wang et al., 2026a)	97.8	99.2	97.2	95.0	97.3
TurboVLA (Xie et al., 2026)	99.2	99.8	97.4	94.2	97.7
ACoT-VLA (Zhong et al., 2026)	98.6	99.0	99.4	97.0	98.5
Prediction & world models					
Seer (Tian et al., 2025)	–	–	–	87.7	87.7
FlowVLA (Zhong et al., 2025)	93.2	95.0	91.6	72.6	88.1
DreamVLA (Zhang et al., 2025c)	97.5	94.0	89.5	89.5	92.6
UD-VLA (Chen et al., 2026)	94.1	95.7	91.2	89.6	92.7
RynnVLA-002 (Cen et al., 2025a)	94.2	96.4	94.6	87.6	93.3
UniVLA (Bu et al., 2025b)	95.4	98.8	93.5	94.0	95.5
F1 (Lv et al., 2025)	98.2	97.8	95.4	91.3	95.7
GE-Act (Liao et al., 2026)	98.2	97.6	95.8	94.4	96.5
MMaDA-VLA (Liu et al., 2026)	98.8	99.8	98.0	95.2	98.0
Ours, on Qwen3-VL-4B					
VLM4VLA	80.2	87.2	72.4	44.4	71.1 $\pm$2.4
AAPT + VLM4VLA	84.0	90.2	77.6	60.8	**78.2** $\pm$2.1
ACoT-VLA	98.4	98.6	98.4	95.8	97.8 $\pm$0.6
AAPT + ACoT-VLA	98.8	99.4	98.8	96.0	**98.2** $\pm$0.4

for $r^{\mathrm{gt}} < 0.25$ to 92.8% for $r^{\mathrm{gt}} \geq 0.80$, and exceeds 50% in the $[0.67, 0.80)$ bin. Three of the four adjacent bin pairs have non-overlapping 95% intervals in Table 5, the exception being the $[0.25, 0.50)$ and $[0.50, 0.67)$ pair. The reward is therefore a graded proxy for physical viability under these perturbations, though not a calibrated success probability.

Finding 2. Ranking refinement matters more as candidate sets become denser. Without assessment training, the base backbone ranks candidates barely above chance in Figure 4(b). SFT establishes a useful ordering, and GRPO improves all-versus-all ranking accuracy by 5.4, 8.0 and 9.1 percentage points on Tests A, B and C as the candidate set grows from 20 to 75. The increasing gain is consistent with direct ranking optimization becoming more useful as the problem becomes denser. The component ablation in Figure 4(c) shows complementary effects. Depth improves global ordering, whereas the autoregressive physical reasoning trace improves discrimination near the success boundary. Together they add 4.1 points over depth alone and 2.3 over reasoning supervision alone. These gains carry to control in Table 7, linking the assessment diagnostic to downstream behavior, though depth helps downstream only once the justification target is present.

Finding 3. The procedural scaffold carries most of the ranking gain. Table 6 separates physical content from its linguistic rendering. The procedural scaffold already recovers most of the gain over score-only supervision, improving Test A all-versus-all ranking accuracy by 2.90 percentage points and LIBERO-Plus zero-shot success by 0.9 points. Rewriting that same content in language aligned with the backbone adds a further 1.21 points of ranking accuracy and 0.7 points of LIBERO-Plus success. Most of the measured benefit therefore comes from the physical evidence the trace states rather than from its phrasing, so the objective does not depend on a strong external rewriter. Since

Table 2. CALVIN ABC→D. Columns give the success rate (%) for completing at least k consecutive instructions, and Avg. Len. the mean number completed out of five. The controlled AAPT gains in Avg. Len. are $+0.20$ for VLM4VLA and $+0.18$ for ACoT-VLA. A subset of methods is shown here; Table 13 in Appendix K gives the full comparison.

Method	1/5	2/5	3/5	4/5	5/5	Avg. Len.
Direct policies						
π_0 (Black et al., 2024a)	96.8	85.0	76.7	68.1	59.9	3.92
LLaDA-VLA (Wen et al., 2025)	95.6	87.7	79.5	73.9	64.5	4.01
OpenHelix (Cui et al., 2025)	97.1	91.4	82.8	72.6	64.1	4.08
OpenVLA-OFT (Kim et al., 2025)	96.3	89.1	82.4	75.8	66.5	4.10
RoboVLMs (Liu et al., 2025a)	98.0	93.6	85.2	77.9	70.4	4.25
VLA-Adapter (Wang et al., 2026a)	99.1	94.6	88.8	82.8	76.5	4.42
Prediction & world models						
UniVLA (Wang et al., 2026b)	95.5	85.8	75.4	66.9	56.5	3.80
ReconVLA (Song et al., 2026)	95.6	87.6	76.9	69.3	64.1	3.95
UP-VLA (Zhang et al., 2025b)	92.8	86.5	81.5	76.9	69.9	4.08
Seer (Tian et al., 2025)	96.3	91.6	86.1	80.3	74.0	4.28
VPP (Hu et al., 2025)	95.7	91.2	86.3	81.0	75.0	4.33
UniVLA (Bu et al., 2025b)	98.9	94.8	89.0	82.6	75.1	4.41
DreamVLA (Zhang et al., 2025c)	98.2	94.6	89.5	83.4	78.1	4.44
MMaDA-VLA (Liu et al., 2026)	99.8	98.6	96.3	93.5	89.7	4.78
Ours, on Qwen3-VL-4B						
VLM4VLA	93.3	85.7	79.0	71.9	64.4	3.94 $\pm$0.08
AAPT + VLM4VLA	94.4	88.1	83.2	77.4	70.9	**4.14** $\pm$0.06
ACoT-VLA	96.1	92.1	87.3	82.8	77.6	4.36 $\pm$0.05
AAPT + ACoT-VLA	97.2	94.9	90.3	86.7	84.8	**4.54** $\pm$0.03

the assessment output is discarded before policy training, this benefit must be carried by the learned backbone rather than by an extra downstream input.

Finding 4. Graded evaluation and ranking refinement each add beyond failure-aware prediction. Table 3 separates these effects with a fixed Qwen3-VL-4B backbone and downstream recipe. First, a same-data collision-and-stability predictor retains AAPT's scenes, candidates, successes, failures, inputs, physical-reasoning supervision and SFT schedule, but replaces the graded score with binary simulator outcomes and omits GRPO. Even this predictive objective improves both benchmarks, reaching 87.5 on LIBERO-Plus and 4.42 on CALVIN. Second, replacing only those binary targets with graded assessment under the same SFT schedule reaches 88.1 ± 0.5 and 4.45 ± 0.06. The evaluative target therefore adds $+0.6$ points on LIBERO-Plus and $+0.03$ average sequence length on CALVIN despite the predictive control's label advantage. Finally, ranking refinement adds a further $+0.7$ and $+0.09$, respectively, to give full AAPT 88.8 and 4.54. Descriptive objectives and success-only grasp-pose prediction do not improve both benchmarks. Table 8 audits the remaining differences in data, targets and optimization.

Finding 5. Destroying candidate–quality alignment reverses the gain. The permuted-target control keeps the scenes, candidate poses, output format, sequence lengths and compute fixed, but assigns each candidate another candidate's complete trace-and-score target from the same scene. It therefore breaks only candidate–quality alignment. Performance falls below the base VLM, from 87.0 to 82.6 on LIBERO-Plus and from 4.36 to 4.27 on CALVIN. Correct alignment is therefore necessary for the gain, while training toward a misaligned ordering is harmful.

Finding 6. The gain survives both minimal and strong control pipelines. AAPT improves the minimal VLM4VLA pipeline by 7.1 points on LIBERO and from 3.94 to 4.14 on CALVIN. This provides the cleaner attribution test because fewer than 1% of VLM4VLA's parameters are new, it has no diffusion or flow-matching decoder, and the controlled rows differ only in backbone initialization. Its lower absolute performance reflects the deliberately minimal action decoder. The gain also survives in the stronger ACoT-VLA architecture. LIBERO is nearly saturated, where AAPT adds 0.4 points, mainly serving as a parity check near the benchmark ceiling. On CALVIN, which has more headroom, AAPT reaches 4.54, behind only MMaDA-VLA, which uses 61M steps of cross-embodiment VLA pre-training. Together, VLM4VLA provides relatively direct attribution

Table 3. Components of each controlled pre-training objective using Qwen3-VL-4B and ACoT-VLA. Checks mark included components; aligned means each input retains its own target. LIBERO-Plus is success rate (%) and CALVIN is average sequence length. Table 8 audits data/optimization.

Pre-training	Action input	Failure examples	Physical reasoning	Graded score	Ranking refinement	Target aligned	LIBERO-Plus	CALVIN
Base VLM, no pre-training	–	–	–	–	–	–	87.0	4.36
Alternative objectives								
Robo2VLM (Chen et al., 2025)	×	×	×	×	×	✓	86.4	4.38
VQASynth (Remyx AI, 2024)	×	×	×	×	×	✓	87.3	4.34
Grasp-pose prediction	×	×	×	×	×	✓	85.9	4.32
Matched AAPT progression								
Collision + stability prediction	✓	✓	✓	×	×	✓	87.5	4.42
AAPT, no ranking refinement	✓	✓	✓	✓	×	✓	88.1	4.45
AAPT	✓	✓	✓	✓	✓	✓	**88.8**	**4.54**
Target-alignment control								
AAPT, permuted targets	✓	✓	✓	✓	✓	×	82.6	4.27

Table 4. Zero-shot LIBERO-Plus success rate (%) across seven held-out perturbation factors, evaluated on the LIBERO checkpoints of Table 1 without fine-tuning. The controlled AAPT gain is $+1.8$ points. * reproduced from Zhong et al. (2026). Table 14 in Appendix K gives the full comparison.

Method	Camera	Robot	Lang.	Light	Backgr.	Noise	Layout	Avg
UniVLA (Bu et al., 2025b)	1.8	46.2	69.6	69.0	81.0	21.2	31.9	45.8
π_0-FAST (Black et al., 2024a)	65.1	21.6	61.0	73.2	73.2	74.4	68.8	62.5
RIPT-VLA (Tan et al., 2025)	55.2	31.2	77.6	88.4	91.6	73.5	74.2	70.2
π_0^* (Black et al., 2024a)	61.0	40.8	63.5	89.3	84.1	80.1	76.4	70.7
OpenVLA-OFT (Kim et al., 2025)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	71.4
$\pi_{0.5}^*$ (Black et al., 2025)	75.8	79.4	83.3	95.5	95.0	89.6	87.0	86.5
ACoT-VLA (Zhong et al., 2026)	72.6	82.6	87.5	97.7	96.5	87.8	88.1	87.5
Ours, on Qwen3-VL-4B								
ACoT-VLA	72.4	82.2	88.0	96.1	95.4	87.2	87.9	87.0 ±0.7
AAPT + ACoT-VLA	**75.4**	**83.4**	**88.8**	**97.5**	**97.2**	**89.6**	**89.5**	**88.8** ±0.8

to the backbone initialization, while ACoT-VLA shows that the same initialization remains useful inside a substantially stronger action architecture. The matched-seed results in Table 10 show paired gains of $+0.4 \pm 0.3$ points on LIBERO, $+1.8 \pm 0.2$ on LIBERO-Plus and $+0.18 \pm 0.03$ average sequence length on CALVIN. All nine paired differences are positive, but with three seeds we treat this as descriptive evidence rather than a significance claim. Appendix I reports conditional CALVIN success, where the gain is largest at the hardest steps.

Finding 7. The gains carry to a second embodiment. LIBERO and CALVIN both use the Franka arm. SimplerEnv provides a descriptive check on WidowX and Google Robot (Table 9). Across both pipelines, AAPT improves every task except the one already at ceiling. The largest gain is on stack-cube with ACoT-VLA, from 33.3 to 45.8. These results lack training-seed uncertainty, so they support transfer beyond Franka but not a stand-alone embodiment claim.

Finding 8. Zero-shot robustness improves on every perturbation factor. LIBERO-Plus evaluates the LIBERO checkpoints without additional training. It therefore changes the observation distribution while holding the learned policy fixed. AAPT reaches 88.8% across the seven factors, a controlled gain of $+1.8$ points over the identical base. Unlike the nearly saturated LIBERO result, this evaluation retains enough headroom to expose differences in robustness. We do not use the heterogeneous external rows to support the controlled gain. All seven factor means improve, though they are correlated evaluations of the same checkpoints and are therefore descriptive rather than independent replications. Camera viewpoint improves most at $+3.0$

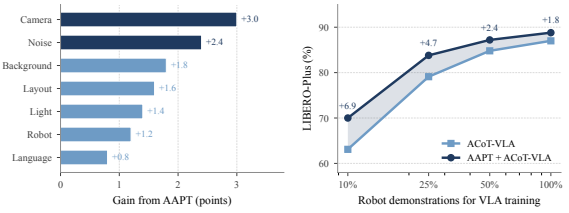

Figure 5. AAPT gains with ACoT-VLA. The left panel reports each LIBERO-Plus perturbation factor. The right panel shows that the advantage widens as robot demonstrations are withdrawn.

points, followed by sensor noise at $+2.4$. Language phrasing improves least at $+0.8$, consistent with an objective that does not alter instruction semantics. The larger gains under viewpoint and sensor changes suggest that AAPT improves how the policy reads the scene, rather than merely memorizing the appearance of successful grasps.

Finding 9. AAPT gains widen as downstream data decreases. Figure 5 (right) trains ACoT-VLA on subsets of the LIBERO demonstrations and evaluates it zero-shot on LIBERO-Plus. The gain grows monotonically as demonstrations are removed, from $+1.8$ points with all data to $+2.4$ at 50%, $+4.7$ at 25% and $+6.9$ at 10%. This pattern is consistent with a stronger initialization, although final performance alone cannot separate representation quality from faster or better optimization.

Analysis. More assessment data helps, with diminishing returns. Figure 6 scales AAPT to 1M samples with ACoT-VLA. Performance improves at every tested corpus size on both benchmarks. The agreement across LIBERO-Plus and CALVIN makes a single-benchmark fluctuation less likely, while the progressively smaller increments indicate diminishing returns. The final increase to 1M adds $+0.2$ points on LIBERO-Plus and $+0.04$ on CALVIN. The trend supports continued gains within the tested range, but does not establish scaling beyond 1M samples.

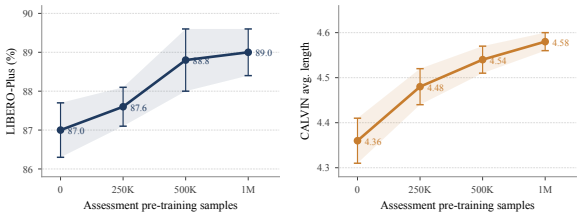

Figure 6. Downstream performance versus assessment-corpus size with ACoT-VLA.

6 Conclusion

AAPT asks whether a backbone intended for control should learn not only what is present or what action follows, but also which candidate action is physically viable and why. We instantiate this idea on grasp poses by turning successful grasps, near-misses and failures into a graded local ordering.

The experiments support two conclusions. First, discarded alternatives are useful supervision. The geometric target tracks simulated success, destroying candidate-to-quality alignment reverses the downstream gain, and a same-data predictor of collision and stability improves both benchmarks. Second, graded assessment gives the strongest transfer in our comparisons. AAPT improves both a minimal action head and a strong action-reasoning pipeline, shows gains on a second embodiment, and helps most under distribution shift, long-horizon difficulty and scarce demonstrations. With the optimization held fixed, the graded evaluative target already exceeds a predictive control supervised by simulated outcomes, and ranking refinement adds a further gain, so both the target form and its refinement contribute rather than either alone. The broader result is that pre-training becomes more useful for control when it represents the physical boundary between a viable action and a nearby failure. Demonstration pipelines already generate many such alternatives and then discard them. Preserving and structuring those alternatives may be as important as collecting more successes.

Limitations and outlook. The study is deliberately narrow in three respects. First, AAPT is constructed on parallel-jaw grasps, where viability can be judged from the scene and pose without trajectory-level credit assignment. The downstream policies place, stack and execute instruction chains, but this does not establish that the pre-training objective itself extends to placement, pushing or mid-trajectory decisions. A direct second action class is the clearest next test. Second, candidate generation and trace supervision use object geometry, 3D models and annotated successful grasps. Collections without these assets will require alternative sources of supervision, such as learned geometry, multi-view reconstruction or action-conditioned outcome logs. The reward is also relative to the available success set, so an unrecorded valid grasp can be assigned too little credit. Third, all experiments are in simulation. Simulation makes controlled perturbations and physical validation possible, but it does not capture the perception errors, contact stochasticity and hardware variation of a real system. We next plan to extend the construction to other state-determined actions such as placement, then to trajectory-dependent actions where assessment must predict future consequences and assign credit across time, and finally test whether the resulting representation transfers to a real robot.

ETHICS STATEMENT

This work uses only public simulation benchmarks and grasp datasets, namely LIBERO, LIBERO-Plus, CALVIN, SimplerEnv, HouseCat6D and GraspNet-1Billion, and involves no human subjects, private data or physical robot deployment. Its subject is manipulation policy learning, which carries the general dual-use considerations of robotics research; our contribution is a pre-training objective and introduces no risks beyond those. We foresee no specific ethical concerns.

REPRODUCIBILITY STATEMENT

The assessment task, the graded reward and the two-stage recipe are specified in Section 3, with the full candidate-set construction and perturbation schedule in Appendix A and the procedural scaffold, trace-rendering prompt and exact autoregressive training sequence in Appendix B. Downstream training settings for both pipelines, the paired three-seed protocol and the evaluation protocols are summarized in Section 4 and specified in Appendices F and H. The assessment-task metrics and splits are given in Appendix C. We build the pre-training corpus from public benchmarks and will release the corpus generator, generated candidate-and-target manifests, trace-generation prompts, fixed data splits, training configurations and the reported checkpoints upon publication.

USE OF LARGE LANGUAGE MODELS

Large language models appear in this work in two roles. As a component of the method, we prompt Qwen3-VL-8B to render the procedural scaffolds into the *output targets* described in Section 3 and Appendix B, so part of our pre-training supervision is synthetic text generated by a model. Qwen3-VL-8B acts only as an offline trace generator. Its privileged prompt is never shown to the pre-trained VLM. The scaffolds are procedural and derive from ground-truth geometry, so the generator rewrites supplied content rather than inventing the physical analysis. As a writing aid, we used a general-purpose assistant to copy-edit prose and LaTeX and to draft and revise passages of the introduction, abstract and results section. All technical content, the objective, the experiments and every claim are the authors' own, and the authors verified each number, citation and statement in the submitted text.

REFERENCES

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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. *arXiv preprint arXiv:2410.24164*, 2024a.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *International Conference on Learning Representations (ICLR)*, pp. 33431–33452, 2024b.

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. *arXiv preprint arXiv:2504.16054*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander

Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023a.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabber, Chelsea Finn, Keerthana Gober, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems (RSS)*, 2023b.

Qingwen Bu, Jia Zeng, Li Chen, Yanchao Yang, Guyue Zhou, Junchi Yan, Ping Luo, Heming Cui, Yi Ma, and Hongyang Li. Clover: Closed-loop visuomotor control with generative expectation for robotic manipulation. *Advances in Neural Information Processing Systems (NeurIPS)*, 37: 139002–139029, 2024.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025a.

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. In *Robotics: Science and Systems*, 2025b.

Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Bohan Hou, Yuming Jiang, Jiayan Guo, Xin Li, et al. Rynnvla-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025a.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025b.

Jiayi Chen, Wenxuan Song, Pengxiang Ding, Ziyang Zhou, Han Zhao, Barrett Tang, Donglin Wang, and Haoang Li. Unified diffusion vla: Vision-language-action model via joint discrete denoising diffusion process. In *International Conference on Learning Representations (ICLR)*, pp. 139291–139311, 2026.

Kaiyuan Chen, Shuangyu Xie, Zehan Ma, Pannag R Sanketi, and Ken Goldberg. Robo2vlm: Visual question answering from large-scale in-the-wild robot manipulation datasets. *arXiv preprint arXiv:2505.15517*, 2025.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

Can Cui, Pengxiang Ding, Wenxuan Song, Shuanghao Bai, Xinyang Tong, Zirui Ge, Runze Suo, Wanqi Zhou, Yang Liu, Bofang Jia, et al. Openhelix: A short survey, empirical analysis, and open-source dual-system vla model for robotic manipulation. *arXiv preprint arXiv:2505.03912*, 2025.

Mingtong Dai, Lingbo Liu, Yongjie Bai, Yang Liu, Zhouxia Wang, Rui Su, Chunjie Chen, Liang Lin, and Xinyu Wu. RoVer: Robot reward model as test-time verifier for vision-language-action model. *arXiv preprint arXiv:2510.10975*, 2025.

Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. GraspNet-1Billion: A large-scale benchmark for general object grasping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A. Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

Ankit Goyal, Hugo Hadfield, Xuning Yang, Valts Blukis, and Fabio Ramos. Vla-0: Building state-of-the-art vlas with zero modification. *arXiv preprint arXiv:2510.13054*, 2025.

Asher J. Hancock, Allen Z. Ren, and Anirudha Majumdar. Actions as language: Fine-tuning VLMs into VLAs without catastrophic forgetting. *arXiv preprint arXiv:2509.22195*, 2025.

Zhi Hou, Tianyi Zhang, Yuwen Xiong, Hengjun Pu, Chengyang Zhao, Ronglei Tong, Yu Qiao, Jifeng Dai, and Yuntao Chen. Diffusion transformer policy. *arXiv preprint arXiv:2410.15959*, 2024.

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. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24328–24346. PMLR, 2025.

Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:82782–82802, 2026.

Kun Huang, Edward S. Hu, and Dinesh Jayaraman. Training robots to evaluate robots: Example-based interactive reward functions for policy learning. In *Conference on Robot Learning*, 2022.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U Tan, Navonil Majumder, Soujanya Poria, et al. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*, 2025.

HyunJun Jung, Patrick Braun, Niklas Groth, Volker Krueger, and Ales Leonardis. HouseCat6D: A large-scale multi-modal category level 6D object perception dataset with household objects in realistic scenarios. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. In *Proceedings of the 2nd Conference on Robot Learning (CoRL)*, 2018.

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. In *Proceedings of the 8th Conference on Robot Learning (CoRL)*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Robotics: Science and Systems*, 2025.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. In *International Conference on Learning Representations (ICLR)*, pp. 26703–26721, 2024a.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. In *Proceedings of the 8th Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*. PMLR, 2024b.

Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Liuao Pei, Tian Nian, Shunbo Zhou, Xiaokang Yang, Jiangmiao Pang, et al. Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint arXiv:2508.20072*, 2025.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Si Liu, Jianlan Luo, Liliang Chen, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. In *International Conference on Learning Representations (ICLR)*, pp.

88446–88463, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Huaping Liu, Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, and Hanbo Zhang. Towards generalist robot policies: what matters in building vision-language-action models. *Research Square preprint*, 2025a.

Yang Liu, Pengxiang Ding, Tengyue Jiang, Xudong Wang, Minghui Lin, Wenxuan Song, Hongyin Zhang, Zifeng Zhuang, Han Zhao, Wei Zhao, Siteng Huang, Jinkui Shi, and Donglin Wang. MMaDA-VLA: Large diffusion vision-language-action model with unified multi-modal instruction and generation. *arXiv preprint arXiv:2603.25406*, 2026.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025b.

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. *arXiv preprint arXiv:2509.06951*, 2025.

Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. LIV: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning*, 2023a.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations*, 2023b.

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*, 7(3):7327–7334, 2022.

Kaichun Mo, Yuzhe Qin, Fanbo Xiang, Hao Su, and Leonidas Guibas. Where2Act: From pixels to actions for articulated 3d objects. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. Steering your generalists: Improving robotic foundation models via value guidance. In *Conference on Robot Learning*, 2024.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Quanquan Peng, Yutong Liang, Rui Yan, Nicklas Hansen, and Xiaolong Wang. FACT: Failure-aware causal training for world-action models. *arXiv preprint arXiv:2608.10232*, 2026.

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. *arXiv preprint arXiv:2501.15830*, 2025.

Remyx AI. VQASynth: A pipeline for synthesizing embodied visual question answering data. `https://github.com/remyxai/VQASynth`, 2024. Open-source toolkit for generating embodied VQA training data from RGB-D scenes.

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

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. Reconvla: Reconstructive vision-language-action model as effective robot perceiver. In *Proceedings of the AAAI Conference on Artificial Intelli-*

gence, volume 40, pp. 18549–18557, 2026.

Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-GraspNet: Efficient 6-DoF grasp generation in cluttered scenes. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Kraehenbuehl. Interactive post-training for vision-language-action models. In *CVPR Workshop on Foundation Models Meet Embodied Agents*, 2025.

Louis Leon Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273–286, 1927.

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. In *International Conference on Learning Representations*, 2025.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. VLA-Adapter: An effective paradigm for tiny-scale vision-language-action model. In *AAAI Conference on Artificial Intelligence*, 2026a.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. In *International Conference on Learning Representations (ICLR)*, pp. 80929–80944, 2026b.

Yuqing Wen, Hebei Li, Kefan Gu, Yucheng Zhao, Tiancai Wang, and Xiaoyan Sun. Llada-vla: Vision language diffusion action models. *arXiv preprint arXiv:2509.06932*, 2025.

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 (ICLR)*, pp. 10641–10662, 2024.

Tianhe Wu, Jian Zou, Jie Liang, Lei Zhang, and Kede Ma. VisualQuality-R1: Reasoning-induced image quality assessment via reinforcement learning to rank. *arXiv preprint arXiv:2505.14460*, 2025.

Hengyi Xie, Chenfei Yao, Xianjin Wu, Xuanyang Xi, Yiping Tang, Di Xu, Yingying Zhu, Dingkang Liang, Xiang Bai, and Han Ding. TurboVLA: Real-time vision-language-action model at 32 hz on an RTX 4090 with $<$1 gb VRAM. *arXiv preprint arXiv:2607.27205*, 2026.

Yifan Yang, Zhixiang Duan, Tianshi Xie, Fuyu Cao, Pinxi Shen, Peili Song, Chenyang Zhao, Piaopiao Jin, Guokang Sun, Shaoqing Xu, et al. Fpc-vla: A vision-language-action framework with a supervisor for failure prediction and correction. *Expert Systems with Applications*, pp. 131742, 2026.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

Yang Yue, Yulin Wang, Bingyi Kang, Yizeng Han, Shenzhi Wang, Shiji Song, Jiashi Feng, and Gao Huang. Deer-vla: Dynamic inference of multimodal large language models for efficient robot execution. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:56619–56643, 2024.

Hongyin Zhang, Pengxiang Ding, Shangke Lyu, Ying Peng, and Donglin Wang. Gevrm: Goal-expressive video generation model for robust visual manipulation. In *International Conference on Learning Representations (ICLR)*, pp. 91789–91811, 2025a.

Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74911–74922. PMLR, 2025b.

Jianke Zhang, Xiaoyu Chen, Qiuyue Wang, Mingsheng Li, Yanjiang Guo, Yucheng Hu, Jiajun Zhang, Shuai Bai, Junyang Lin, and Jianyu Chen. VLM4VLA: Revisiting vision-language-models in vision-language-action models. *arXiv preprint arXiv:2601.03309*, 2026.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. In *Advances in Neural Information Processing Systems*, 2025c.

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. *arXiv preprint arXiv:2503.22020*, 2025a.

Wei Zhao, Pengxiang Ding, Zhang Min, Zhefei Gong, Shuanghao Bai, Han Zhao, and Donglin Wang. Vlas: Vision-language-action model with speech instructions for customized robot manipulation. In *International Conference on Learning Representations (ICLR)*, pp. 51676–51693, 2025b.

Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22508–22519. IEEE, 2025.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Maoqing Yao, Si Liu, and Guanghui Ren. ACoT-VLA: Action chain-of-thought for vision-language-action models. *arXiv preprint arXiv:2601.11404*, 2026.

Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Tianran Zhang, Wenxuan Song, Jiayi Chen, Xinhu Zheng, Hesheng Wang, et al. Flowvla: Visual chain of thought-based motion reasoning for vision-language-action models. *arXiv preprint arXiv:2508.18269*, 2025.

A CANDIDATE SET GENERATION

Each object instance yields a candidate set containing an annotated success, controlled perturbations of that success and clear failures. The construction provides both difficult comparisons near a known success and unambiguous low-quality candidates.

Successful poses. We sample one annotated successful grasp as the seed for each candidate set and assign the unperturbed seed $r^{\mathrm{gt}} = 1$. HouseCat6D supplies binary success labels directly. GraspNet-1Billion supplies continuous force-closure scores, which we threshold at 0.8 to define the annotated success set. Scores below the threshold are treated as failures. An object can have thousands of valid grasps, so we do not use every success as a perturbation seed. Seeds are sampled randomly subject to diversity over $\mathrm{SE}(3)$, which avoids dense clusters of near-duplicate poses and covers distinct approach and contact configurations. All annotated successes remain available when computing the reward, even though only one is selected as the seed of a given set.

Near-successful perturbations. We generate four $\mathrm{SE}(3)$-perturbed variants of that grasp at increasing noise levels $\sigma_R \in \{0.02, 0.05, 0.08, 0.12\}$ radians and $\sigma_t \in \{0.02, 0.05, 0.08, 0.12\}$ meters. Rotation uses zero-mean isotropic Gaussian axis-angle noise and translation uses an independent zero-mean isotropic Gaussian in $\mathbb{R}^3$; the four rungs pair the listed rotation and translation standard deviations. These are the candidates that straddle the success boundary, and their rewards decay smoothly with $\mathrm{SE}(3)$ distance to the nearest success.

Failure poses. The remaining candidates are sampled from the dataset's annotated failures. If an object does not provide enough failure poses, we fill the remaining slots with uniformly sampled poses. This fallback ensures that every set also contains candidates far from any annotated success.

Training-time composition. Every training candidate set contains $N = 20$ poses. It includes one unperturbed seed, one perturbation from each of the four noise rungs and 15 failure poses from the dataset or the uniform fallback. Corpus sizes in the paper count individual candidate examples rather than 20-candidate sets. Across the corpus, 40% of examples come from HouseCat6D and 60% from GraspNet-1Billion.

Reward assignment. Every candidate is scored by Equation 1 using its minimum $\mathrm{SE}(3)$ distance to any annotated success for that object in the scene. The rotational and translational reference scales first make the two distance terms dimensionless. Both scales equal the standard deviation of the smallest perturbation rung, so a reference-scale rotational error and a reference-scale translational error contribute equally. Dividing by the median failure distance then adapts the reward range to the scene. A candidate at the median failure distance has $r^{\mathrm{gt}} = 0.5$, while more distant failures approach zero.

Taking the minimum over the complete annotated success set accommodates multiple valid and symmetry-equivalent grasps, including successes not selected as the perturbation seed. It cannot account for missing annotations, so an unrecorded valid grasp may receive an artificially low reward. The target is therefore a relative geometric quality score, not a calibrated success probability. Annotated successes and nearby perturbations occupy the high-reward end, while distant failures anchor the low-reward end. This mixture yields easy comparisons against gross failures and difficult comparisons among geometrically similar poses near the success boundary. The GRPO stage therefore compares each candidate only with its nearest K neighbors in ground-truth rank.

Evaluation splits. We hold out 200 scenes for assessment evaluation and construct three candidate-set sizes per object instance. Test A contains 20 candidates with 5 successes and 15 failures. Test B contains 50 with 20 successes and 30 failures. Test C contains 75 with 25 successes and 50 failures. These successes are annotated dataset grasps at $r^{\mathrm{gt}} = 1$, not synthesized perturbations. The evaluation sets therefore draw several annotated successes per object instance, whereas each training set is seeded from a single one, and no perturbed pose is ever counted as a success at test time. The larger sets contain more plausible alternatives and therefore more difficult pairwise decisions. The split is scene-disjoint but not object-instance- disjoint or category-disjoint. These diagnostics measure generalization to unseen scenes, not to unseen object identities.

Physical validation of the graded reward. We execute 1000 synthetic candidates in simulation and measure empirical grasp success within fixed r^{gt} bins. These are observed simulator outcomes, not predictions from AAPT or any control. The binary success label requires the object to be lifted 5–10 cm, remain stably held over the fixed evaluation window and pass force-closure verification. The candidates are drawn from a set held out entirely from the pre-training corpus, so no executed candidate contributes to any AAPT, control or baseline training run. The validation therefore tests the target construction itself and cannot leak simulator outcomes into a trained model. Annotated seed successes at $r^{\mathrm{gt}} = 1$ are excluded. Ordinary candidate generation does not populate every interval equally, so we resample the generation process until all bins have adequate coverage. This stratification changes the reward distribution and makes a pooled success rate uninterpretable. Table 5 therefore reports the number of executions and conditional success rate within each bin. Every candidate uses the same joint 6-DoF perturbation process as pre-training. Translation and rotation are not varied separately, and the analysis is not stratified by object symmetry.

Table 5. Physical validation of the geometric reward over 1000 synthetic simulator executions. Success increases monotonically with r^{gt}. Sampling was stratified to cover all bins; consequently, the per-bin rates, not their pooled average, are the relevant quantities. Intervals are 95% Wilson score intervals on the per-bin rate.

Reward bin	Candidates	Successes	Empirical success (%)	95% CI
$0 < r^{\mathrm{gt}} < 0.25$	170	5	2.94	[1.3, 6.7]
$0.25 \le r^{\mathrm{gt}} < 0.50$	199	24	12.06	[8.2, 17.3]
$0.50 \le r^{\mathrm{gt}} < 0.67$	113	27	23.89	[17.0, 32.5]
$0.67 \le r^{\mathrm{gt}} < 0.80$	226	116	51.33	[44.8, 57.8]
$0.80 \le r^{\mathrm{gt}} < 1.00$	292	271	92.81	[89.3, 95.2]
Total	**1000**	**443**	–	–

B PHYSICAL REASONING TRACE GENERATION

The physical reasoning trace target is constructed offline in two steps. A deterministic generator first converts available ground-truth geometry into a structured scaffold. Qwen3-VL-8B then rewrites that scaffold as concise natural language. The two datasets expose different geometric metadata, so their scaffold generators differ.

Target construction and model training

Model input in SFT, GRPO and ranking evaluation. scene RGB, optional depth and the candidate grasp pose.
Autoregressive model output. physical reasoning trace $z_{\mathrm{phys}} \rightarrow$ <score> $\rightarrow$ scalar score s.
Optimization. SFT applies next-token loss to the complete output sequence. GRPO samples complete sequences, parses only the generated scalar score and applies the pairwise-plus-calibration reward; no reference trace enters the rollout context.
Offline target-construction information. object mesh, dimensions, principal axes, object pose, procedural scaffold and target reward. These construct the target z_{phys} only; none is inserted into the model prompt.
Downstream transfer. the assessment prompt, trace and score interface are discarded, and only the trained backbone initializes the VLA.

B.1 HOUSECAT6D

HouseCat6D provides binary success labels and a categorical object name such as *bottle*, *cup* or *teapot*. Given the object's camera-frame pose $\mathbf{T}_{\mathrm{obj}}$, its scale $\mathbf{s} = (w, h, d)$, the candidate grasp transform $\mathbf{T}_{\mathrm{grasp}}$ and optionally the depth image, the generator computes

1. **Grasp-center direction**, by classifying the unit vector from the object centroid to the grasp center as above, below, left, right, front, behind or angled.

2. **Grasp type**, from the dot product of the gripper approach axis $\mathbf{R}_{\text{grasp}}\hat{\mathbf{z}}$ with the object's up axis, giving top-down, bottom-up, side or angled.
3. **Shape descriptor**, from the aspect ratio $\max(\mathbf{s})/\min(\mathbf{s})$, giving elongated, compact or intermediate-aspect-ratio.
4. **A category-specific part vocabulary**, where the object name selects a target part such as the neck, the rim, the handle or the toe box, together with a grasp-strategy phrase consistent with the grasp-center direction.
5. **Relative height and lateral offset**, namely where along the object's vertical axis the grasp lands and how far it sits off the central axis.
6. **Depth analysis**, when the depth image is available, reporting the object's distance from the camera, the clearance between approach and contact, and a finger-spacing hint of narrow, wide or optimal.

B.2 GRASPNET-1BILLION

GraspNet-1Billion has no complete category vocabulary because only 34 of its 88 objects carry semantic YCB names. Every object does provide a mesh. Once per object identifier, we compute its axis-aligned extents, PCA principal axes, oriented extents and a cross-section similarity score $\text{sym} = \min(e_2, e_3)/\max(e_2, e_3)$. Values near one indicate similar extents along the two minor principal axes and serve as a limited proxy for axial symmetry around the longest axis. They are not a complete test of object symmetry. We also assign one of six shape tags, namely elongated-cylindrical, elongated, flat, roughly-spherical, compact or intermediate-aspect-ratio. For each candidate, the generator reports

1. **The object's pose in the camera frame**, classified as upright, upside-down, tilted on its side or tilted at an angle by projecting the model-up axis into the camera frame.
2. **The camera-frame approach direction**, classified as top-down, bottom-up or from the left, right, front or rear using the camera-frame gripper $+\mathbf{z}$ axis.
3. **The object-frame approach alignment**, obtained by re-expressing the approach direction in the object frame and testing it against the longest, intermediate and shortest principal axes.
4. **Grasp type relative to the object's up axis**, giving top-down, bottom-up, side or angled.
5. **Contact location**, by transforming the grasp center into the object frame and reporting relative height along with the lateral offset from the vertical axis in millimeters.
6. **Gripper fit**, by comparing the opening width against the object's thinnest axis to give too narrow, well matched, reasonable or too wide, with the depth the gripper is translated into the object reported in millimeters.
7. **A quality tier**, obtained by binning the target geometric reward as high, medium, marginal or weak. This tier is a textual summary of r^{gt}, not an additional simulator label.

Where the object identifier appears in the known YCB mapping we prepend the name. Otherwise the object is referred to only by its shape and dimensions, which avoids confabulating names for the 54 objects that have none.

B.3 RENDERING THE SCAFFOLD AS TEXT

The deterministic templates above produce structured language grounded in geometry, but we do not use that text directly as the training target. We provide the scaffold and its supporting metadata to Qwen3-VL-8B for one offline generation per example. The rewrite converts the fixed fields into concise natural language while preserving their physical content. The resulting text is the VLM's autoregressive *output target*, not an input feature. Figure 8 gives the prompt that turns a scaffold such as the one in Figure 7 into that target trace.

Procedural scaffold output for two candidates on the same object

Object. Peach, roughly-spherical, $6.2 \times 6.3 \times 5.9$ cm, symmetry score 1.00.

Highest-reward candidate, grasp width 6.3 cm. *The target is a peach, classified as roughly-spherical (dimensions 6.2 x 6.3 x 5.9 cm, oriented-extent 6.4/5.9/5.9 cm, symmetry score 1.00). In this view the object sits tilted at an angle at 0.48 m depth from the camera. The candidate gripper approaches top-down, aligned with the object's longest axis. This is an angled grasp, placed near the top of the object, 55 mm off-center laterally. The gripper opening (6.3 cm) is well matched to the object's thinnest axis (5.9 cm); the gripper is translated 30 mm into the object along the approach axis. The target geometric reward places this candidate in the high-quality tier.*

Lowest-reward candidate, same instance, grasp width 9.4 cm. . . . *The candidate gripper approaches from below or rear, aligned with the object's intermediate axis. This is a side grasp, perpendicular to the object's up axis, placed near the top of the object, slightly off-center laterally (28 mm). The gripper opening (9.4 cm) is reasonable for the object's thinnest axis (5.9 cm); the gripper is translated 10 mm into the object along the approach axis. The target geometric reward places this candidate in the weak tier.*

Figure 7. What the procedural generator emits for two candidates on the same object instance. This scaffold is used offline to construct the model's output target; it is never included in the model prompt. Appendix B.3 rewrites it into the target reasoning trace. The entries share the same object description, while their candidate-specific approach, contact and gripper-fit fields differ. Object identity and appearance alone therefore cannot determine their ordering.

Trace-rendering prompt for assessment reasoning

System. You are an expert in robotic grasping. You reason about grasp candidates based on object geometry, gripper pose, and contact quality.

User.
Image 1 (RGB). `[scene RGB observation]`
Image 2 (Depth). `[scene depth image]`
Object description. `[category, dimensions, principal axes from CAD / PCA]`
Candidate grasp pose. `[position, quaternion, gripper width]`
Target reward $r^{\text{gt}} \in [0, 1]$**.** `[ground-truth quality score]`
Procedural scaffold. `[structured analysis from Appendices B.1/B.2 covering approach direction, grasp type, contact location, gripper fit, quality tier]`

Produce a concise natural-language reasoning trace that implicitly justifies the target reward without stating the score itself. First establish the physical evidence by covering:
(i) the object and its pose in the scene,
(ii) the candidate grasp in terms of approach direction and grasp type,
(iii) the contact location and gripper-object fit.
Only after this justification, conclude with (iv) a qualitative assessment (high / medium / marginal / weak) consistent with the evidence and target reward.

Stay faithful to the procedural scaffold. Never give a numerical score or an unsupported qualitative rating, and never state the assessment before its physical justification.

Figure 8. Offline trace-generation prompt used to construct assessment-reasoning targets. Qwen3-VL-8B receives privileged metadata, the scaffold and target reward, and must justify the assessment physically before expressing a verdict. The VLM receives none of those privileged fields. Its prompt contains only the scene and candidate pose, and it must autoregressively generate the resulting trace before emitting its score.

Exact autoregressive training sequence. The offline procedure above produces a fixed target trace z_{phys} for each candidate, but that trace is never concatenated to the model prompt. In SFT, the model input is

$$x = [I_{\text{RGB}},\, I_{\text{depth}},\, \text{candidate position},\, \text{candidate quaternion}],$$

and the next-token target is

$$y = [z_{\text{phys}},\, \texttt{<score>},\, r^{\text{gt}}].$$

During next-token training, earlier tokens of z_{phys} appear only as preceding target tokens in the usual autoregressive sequence. They are never placed in the prompt. During GRPO and assessment evaluation, the model receives only x, generates $\hat{z}_{\text{phys}}$ itself, and then emits <score> and $\hat{s}$. The ranking metrics and GRPO reward use $\hat{s}$. During downstream VLA training, neither x nor this output format is retained. Only the learned backbone parameters initialize the policy.

B.4 Optimization and score parsing

For SFT, we update the language model at full rank while freezing the vision encoder. Training runs for one epoch with learning rate $5\text{e}{-6}$, batch size 32, cosine decay and 10% warmup. The prompt requests a decimal score in $[0, 1]$ after <score>. The parser reads the first numeric value after that sentinel. It does not clip the parsed value. The output instruction is the only mechanism requesting the $[0, 1]$ range. If a completion contains no parseable value after the sentinel, it is discarded and regenerated with the same decoding settings.

The GRPO stage runs for 500 steps at learning rate $3\text{e}{-6}$ with five rollouts per pose, $\kappa = 8$, $\gamma = 0.01$, $\lambda = 0.25$, $K = 4$ and $\alpha = 0.6$. We use a KL penalty of $5\text{e}{-3}$, weight decay $1\text{e}{-2}$ and gradient clipping at 1.0. Each GRPO group contains five complete sequences sampled from the same scene-and-pose prompt, and its rewards are normalized within that group. When a rollout for candidate g_i is compared with candidate g_j, $\bar{s}_j$ and σ_j are computed from the five separately generated completions for g_j. The uncertainty term σ_i is computed from all five completions for g_i, including the current rollout score s_i. Thus, the current rollout contributes to σ_i but never to the comparison mean $\bar{s}_j$. We transfer the final checkpoint after the fixed SFT and GRPO budgets and do not select it using downstream policy performance.

C Assessment task evaluation

All results in this section use the 500K-sample pre-training corpus and the 200 held-out scenes defined in Appendix A. We report three metrics. *All-versus-all accuracy* is the fraction of valid candidate pairs whose predicted ordering agrees with the ground-truth ordering. *Top-M accuracy* uses only pairs containing at least one of the $M = 5$ highest-reward candidates, emphasizing decisions that involve poses near the top of the set. Chance accuracy is 50.0% for either pairwise metric at every candidate-set size.

MRR@5 uses the candidate with the highest ground-truth reward as the target. If that candidate appears at predicted rank $r \leq 5$, the example receives reciprocal rank $1/r$ and otherwise receives zero. Under a uniformly random ranking, the expected value is $\left(\sum_{r=1}^{5} 1/r\right)/N$. The corresponding chance levels are 0.11, 0.046 and 0.030 for Tests A, B and C. Exact ties do not contribute to pairwise metrics. We exclude a pair if its ground-truth rewards are equal or its predicted scalar scores are equal.

Figure 9 expands the two ranking panels of Figure 4 to every metric, which shows two things the aggregate hides.

First, the metrics respond differently as candidate sets grow. GRPO improves all-versus-all accuracy by 5.4 to 9.1 points and MRR@5 by 0.15 to 0.21, with both gains increasing from Test A to Test C. Top-M improves least on Test A because SFT already reaches 92.0% there. The clearest effect is on MRR@5, which rises from 0.26 to 0.47 on Test C, compared with a chance level of 0.030. The GRPO stage therefore has a larger effect on recovering the best candidate in dense sets than all-pairs accuracy alone indicates.

Second, depth and the autoregressive physical reasoning trace target affect different parts of the ordering. After SFT, the depth-only and justification-supervised variants have similar all-versus-all accuracy, 81.6% and 80.8%, while justification supervision is five points higher on Top-M, 90.6% against 85.6%. Depth primarily improves the global ordering. Requiring the model to generate a physical reasoning trace before its score primarily improves discrimination involving the best candidates, with most of that Top-M gain already present after SFT. After GRPO, combining both components adds 4.1 points over depth alone and 2.3 over justification supervision alone on all-versus-all accuracy. The full model reaches 95.6% on Top-M and 0.64 MRR@5.

Table 6. Effect of rendering the physical reasoning target. All variants use the same RGB + depth inputs and the final SFT+GRPO checkpoint. Ranking metrics are on Test A; LIBERO-Plus is zero-shot success rate (%, mean over three seeds). The deterministic template supplies the same procedural physical content without the Qwen rewrite.

Autoregressive output target	**All-vs-all**	**Top-M**	**MRR@5**	**LIBERO-Plus**
Score only	82.78	89.81	0.55	87.2 $\pm$0.6
Procedural template $\rightarrow$ score	85.68	94.54	0.61	88.1 $\pm$0.5
Qwen rewrite $\rightarrow$ score	**86.89**	**95.56**	**0.64**	**88.8** $\pm$0.8

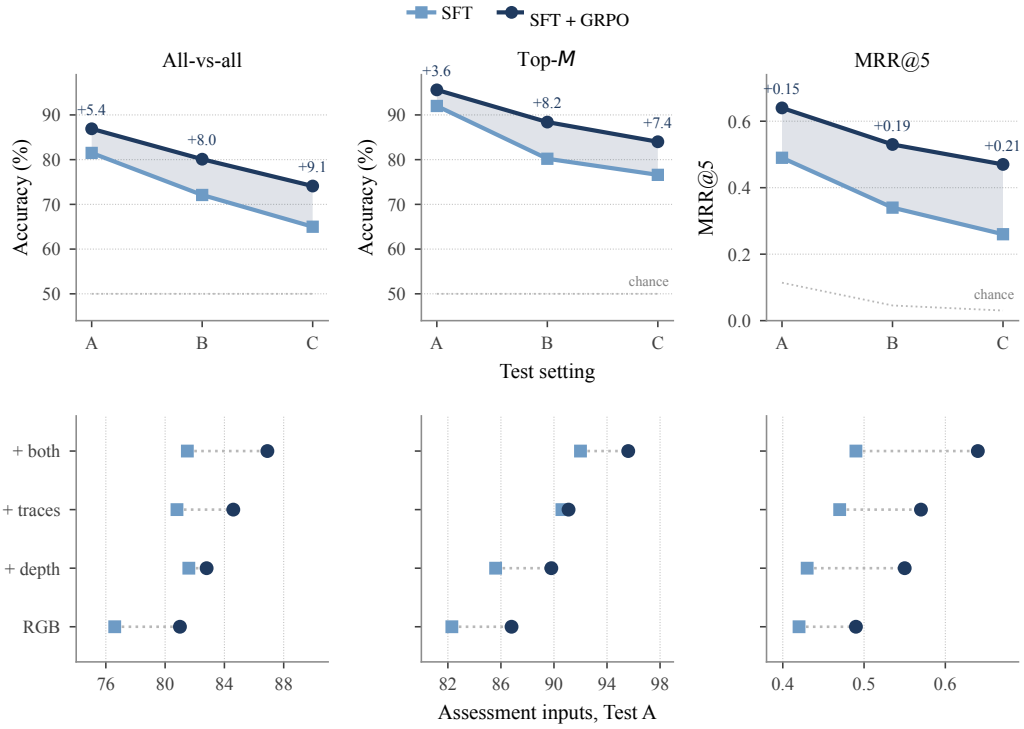

Figure 9. Assessment accuracy on 200 held-out scenes after training on 500K samples, on all three metrics. Top row, the three test settings, which present 20, 50 and 75 candidates per object. Bottom row, the component ablation on Test A, where PR denotes supervision to generate a procedural physical-reasoning trace before the score. The trace is an output target, not an input. Columns share a metric. Dotted lines in the top row mark chance, which is 50 for the two pairwise metrics at any number of candidates and decays with the candidate count for MRR@5.

The ranking gains carry over downstream. Table 7 transfers the same assessment configurations to ACoT-VLA and evaluates the resulting policies zero-shot on LIBERO-Plus. Components that improve the assessment diagnostic also improve downstream success, with depth the one exception noted below. RGB score-only AAPT exceeds the base VLM by 0.8 points. Generating a physical reasoning trace before the score increases the gain to 1.5 points, and adding depth produces the full 1.8-point improvement. These results link the assessment behavior to transfer, but they do not make the output format alone sufficient. The permuted-target control in Table 3 retains the full RGB, depth and justification-to-score format while destroying candidate-quality alignment, and its downstream performance falls below the base VLM.

Depth does not transfer monotonically. Added to score-only supervision it slightly reduces downstream success, from 87.8 to 87.2, even though it improves global ordering on the assessment task itself. It helps only once the model must generate a physical reasoning trace, where it adds a further 0.3 points. Depth is also a pre-training input only: both downstream pipelines consume RGB and language alone, so any benefit must be carried by the learned backbone rather than by a depth

channel available at policy time. We therefore read depth as sharpening the contact geometry that the justification has to account for, rather than as an input the policy inherits. A pre-training signal that is useful for assessment is consequently not guaranteed to be useful for control on its own.

Table 7. Sensitivity of downstream transfer to the assessment configuration. AAPT is pre-trained with each input/output-target combination and evaluated with ACoT-VLA, zero-shot on LIBERO-Plus (%, mean over three seeds). "Trace → score" means that the trace is generated autoregressively before the score; it is never an input.

Assessment pre-training configuration	**LIBERO-Plus**
Base VLM (no AAPT)	87.0 ±0.7
AAPT, RGB input; score only	87.8 ±0.4
AAPT, RGB + depth inputs; score only	87.2 ±0.6
AAPT, RGB input; trace → score	88.5 ±0.7
AAPT, RGB + depth inputs; trace → score	**88.8** ±0.8

D OBJECTIVE-SWAP AUDIT

Table 8 separates exact controls from broader objective comparisons. Every row uses the same Qwen3-VL-4B backbone and downstream ACoT-VLA recipe. AAPT and its permuted-target control are matched in data, inputs, output format and optimization, differing only in the assignment of targets to candidates. Collision-and-stability prediction uses the same candidate examples, model inputs, physical reasoning trace targets and SFT stage as AAPT. It replaces the scalar assessment score with two categorical outcomes and omits GRPO. Its two indicators are measured in the simulator for every candidate in the corpus, whereas AAPT never queries the simulator and derives its target from annotated grasp geometry alone. The control is therefore not disadvantaged by weaker labels: it is supervised by physical outcomes, while AAPT is supervised by a geometric proxy for them. We do not substitute these simulated outcomes for the AAPT target because they are binary. As argued in Section 3, binary supervision makes a near-miss indistinguishable from a gross failure, and the graded ordering that distinguishes them is the object of study. The two rows therefore contrast a predictive objective on true outcomes against an evaluative objective on graded proxies, rather than contrasting label quality. The SFT-only AAPT row completes this comparison. It differs from the collision-and-stability row in the target alone, sharing its scenes, candidates, inputs, justification supervision, example count and SFT budget, and omitting GRPO exactly as that row does. The objective family is therefore identified without the confound of an additional optimization stage, and the full AAPT row measures what ranking refinement adds on top. Grasp-pose prediction shares the source scenes and successful grasps but does not receive failures or candidate-conditioned targets. The VQA objectives use their own descriptive corpora and textual targets. Thus, collision-and-stability prediction is the strongest same-data predictive control, while the other rows test representative alternative pre-training objectives.

Table 8. Detailed audit of the pre-training objectives in Table 3, using the same row groups and order. "Privileged" geometric fields are used only by the offline trace generator to construct output targets; they are never model inputs.

Objective	Data and composition	Model input	Target and supervision	Budget and optimization
Base VLM, no pre-training	No additional pre-training data	–	–	–
Alternative objectives				
Robo2VLM	Robo2VLM-1 teleoperation-derived robot-manipulation VQA (Chen et al., 2025); no success/failure structure	RGB	Textual answer to an embodied-VQA prompt; no AAPT geometry or physical-reasoning trace	684K examples; 15K SFT steps
VQASynth	Synthetic embodied VQA generated from BridgeData V2 with VQASynth (Remyx AI, 2024); no success/failure structure	RGB	Textual answer to a synthetic embodied-VQA prompt; no AAPT geometry or physical-reasoning trace	500K examples; 10K SFT steps
Grasp-pose prediction	The same HouseCat6D and GraspNet-1Billion scenes and successful grasps as AAPT; positives only	RGB and depth; no candidate pose	The successful grasp pose; no generated reasoning trace, failures, scores or candidate comparisons	500K examples; 10K SFT steps
Matched AAPT progression				
Collision + stability prediction	Exactly the same scenes, candidates and success/failure composition as AAPT	RGB, depth and candidate pose	The same autoregressive physical reasoning trace target as AAPT, followed by binary indicators for environment collision and grasp stability instead of a scalar score; both indicators are measured in the simulator for every candidate; no pairwise ranking target	500K examples; 10K SFT steps; no GRPO
AAPT, no ranking refinement	Exactly the same scenes, candidates and success/failure composition as AAPT	RGB, depth and candidate pose	Identical to AAPT: the same physical reasoning trace → `<score>` → graded scalar score. Differs from collision-and-stability prediction in the target alone	500K examples; 10K SFT steps; no GRPO
AAPT	HouseCat6D and GraspNet-1Billion; successes, four $\mathrm{SE}(3)$ perturbation levels and dataset or uniform failures; graded rewards	RGB, depth and candidate pose	Qwen3-VL-8B rewrites a privileged geometric scaffold into a physical reasoning trace → `<score>` → scalar score; privileged fields are absent from the model prompt	500K examples; 10K SFT + 500 GRPO steps
Target-alignment control				
AAPT, permuted targets	Exactly the same scenes, candidates and success/failure composition as AAPT	RGB, depth and candidate pose	The same trace → score sequence as AAPT, but each candidate receives another candidate's paired trace and reward from the same scene. SFT uses the permuted sequence and GRPO uses the corresponding permuted reward	500K examples; 10K SFT + 500 GRPO steps

E DOWNSTREAM PIPELINE DETAILS

Both pipelines take the current RGB observation and language instruction, with no proprioceptive or privileged state. Each policy is initialized from the relevant pre-trained checkpoint and then trained on its downstream benchmark.

VLM4VLA. A single learnable action-query token is appended to the token sequence. A two-layer MLP decodes the hidden state at that position into the next H end-effector actions, each represented by position, orientation and gripper state. Training updates the complete model, including the backbone. Position and orientation use Huber regression, while gripper state uses binary cross-entropy. The MLP produces the action chunk directly without a diffusion or flow-matching decoder.

ACoT-VLA. The Explicit Action Reasoner is a compact Transformer with one reasoner layer for each backbone layer. Reasoner layer i cross-attends to the key-value cache of backbone layer i and denoises a reference action sequence of horizon H^{ref} into a coarse trajectory. The reference target

is a temporally downsampled copy of the demonstrated action chunk, so it requires no separately annotated reasoning labels. During training, the action head is conditioned on explicit guidance computed from the ground-truth reference trajectory. At inference, the reasoner instead conditions on its own predicted trajectory, following the stabilization protocol of the original method.

The Implicit Action Reasoner attaches a learnable query to every backbone layer, pools the resulting cross-attention output and projects it through an MLP. The projected outputs are aggregated into one guidance vector. This vector has no separate supervision and is learned only through the downstream action loss. The explicit trajectory and implicit vector jointly condition the action head.

We make two architectural adaptations for Qwen3-VL-4B. The published $\pi_{0.5}$-derived configuration pairs SigLIP with an 18-layer Gemma-2B language model at 224×224 resolution. We instead use Qwen3-VL-4B at its native dynamic resolution. The explicit reasoner therefore follows Qwen's layer count, and its cross-attention projections match Qwen's grouped-query key-value layout.

F Downstream training and evaluation protocols

Replication. All experiments use Qwen3-VL-4B unless stated otherwise, and AAPT uses the 500K-sample corpus outside the corpus-size sweep. Every baseline and AAPT comparison shares the downstream seed and data order. We use the same three seeds for both pipelines and treat the training seed, not rollout episodes or perturbation factors, as the replication unit. Since three seeds do not support strong inferential claims, we report paired differences descriptively without a formal significance test. Appendix H gives the exact matched values.

LIBERO and LIBERO-Plus. LIBERO (Liu et al., 2023) contains Spatial, Object, Goal and Long suites, each with ten tasks and fifty human demonstrations per task. We train one policy per suite and report mean success over fifty rollouts per task. LIBERO-Plus (Fei et al., 2025) re-renders these tasks under seven held-out factors covering camera pose, robot configuration, language phrasing, lighting, background, sensor noise and object layout. It contains 10,030 episodes. We evaluate the LIBERO checkpoints without additional fine-tuning.

CALVIN. CALVIN ABC→D (Mees et al., 2022) trains on environments A, B and C and evaluates in held-out environment D. The protocol contains 1000 chains of five language instructions. We report the rate of completing at least each chain position and the mean number completed.

VLM4VLA training. The action horizon is $H = 4$ on LIBERO and SimplerEnv and $H = 10$ on CALVIN. On LIBERO and SimplerEnv, we fine-tune all parameters with AdamW for 50K steps using learning rate $5\mathrm{e}{-5}$ and batch size 512. CALVIN uses 60K steps, learning rate $2\mathrm{e}{-5}$ and batch size 128.

ACoT-VLA training. The reference horizon is $H^{\mathrm{ref}} = 15$ and the action horizon is $H = 10$. We train with AdamW for 60K steps using peak learning rate $5\mathrm{e}{-5}$, a 10K-step warmup, cosine decay and batch size 128. Checkpoint selection follows the original ACoT-VLA protocol (Zhong et al., 2026).

G SimplerEnv

Table 9 reports the SimplerEnv visual-matching protocol (Li et al., 2024b). Its WidowX and Google Robot embodiments differ from the Franka arm used in LIBERO and CALVIN, providing a second-embodiment check. We report each task separately because their operating points differ substantially. Put eggplant in basket is saturated for our policies, while stack cube is near zero for VLM4VLA but improves with ACoT-VLA. A four-task mean would obscure both effects. The recent methods compared in the main LIBERO and CALVIN tables do not report this protocol, so we keep the result in the appendix.

Table 9. SimplerEnv success rate (%), visual-matching protocol, Qwen3-VL-4B. These are descriptive point estimates without training-seed uncertainty.

Method	Carrot	Eggplant	Spoon	Cube
OpenVLA (Kim et al., 2024)	4.2	0.0	0.0	12.5
Octo (Octo Model Team et al., 2024)	8.3	43.1	12.5	0.0
DiT Policy (Hou et al., 2024)	29.4	65.5	34.4	0.0
ThinkAct (Huang et al., 2026)	37.5	70.8	58.3	8.7
π_0 (Black et al., 2024a)	62.5	100.0	54.2	25.0
Ours, on Qwen3-VL-4B				
VLM4VLA	54.2	95.8	75.0	0.0
AAPT + VLM4VLA	**58.4**	**100.0**	**79.2**	**4.2**
ACoT-VLA	62.5	100.0	83.3	33.3
AAPT + ACoT-VLA	**70.8**	**100.0**	**87.5**	**45.8**

H SEED-WISE PAIRED RESULTS

Table 10 gives the matched-seed values behind the main ACoT-VLA comparisons. Within each row, the Base and AAPT policies share the training seed and data order, and Δ is computed before aggregation. Summary rows report the mean and sample standard deviation across three training seeds. The standard deviation in the final column is computed directly from the three paired differences, not from the marginal standard deviations. Rollout episodes estimate a checkpoint's success rate, and the seven LIBERO-Plus factors are correlated evaluations of the same checkpoints. Neither is treated as an independent training replicate.

Table 10. Seed-wise paired ACoT-VLA results. LIBERO and LIBERO-Plus report success rate (%); CALVIN reports average sequence length. $\Delta = \text{AAPT} - \text{Base}$. Summary rows give mean $\pm$ sample SD, including the SD of the within-seed differences.

Benchmark	Seed	Base	AAPT	Paired Δ
LIBERO	1	97.1	97.8	+0.7
LIBERO	2	98.3	98.6	+0.3
LIBERO	3	98.0	98.2	+0.2
LIBERO summary	–	97.8 ± 0.6	98.2 ± 0.4	$+0.4 \pm 0.3$
LIBERO-Plus	1	86.4	88.0	+1.6
LIBERO-Plus	2	87.7	89.6	+1.9
LIBERO-Plus	3	86.9	88.8	+1.9
LIBERO-Plus summary	–	87.0 ± 0.7	88.8 ± 0.8	$+1.8 \pm 0.2$
CALVIN	1	4.35	4.52	+0.17
CALVIN	2	4.32	4.53	+0.21
CALVIN	3	4.41	4.57	+0.16
CALVIN summary	–	4.36 ± 0.05	4.54 ± 0.03	$+0.18 \pm 0.03$

I CONDITIONAL PER-STEP RELIABILITY ON CALVIN

The columns of Table 2 are cumulative and report the probability of completing at least k instructions. They therefore decay multiplicatively even when conditional per-step reliability is constant. Table 11 removes this compounding. For step one, it reports the original one-step success rate. For each later step k, it reports $P(\text{complete } k \mid \text{completed } k-1) = C_k/C_{k-1}$, where C_k is the cumulative rate in Table 2. Published-method values are computed from their reported cumulative columns in the same way.

Table 11. Conditional per-step success on CALVIN ABC→D. Δ is in percentage points.

	VLM4VLA			ACoT-VLA			Published		
Step	Base	+ AAPT	Δ	Base	+ AAPT	Δ	Seer	VLA-Adapter	MMaDA-VLA
1	0.933	0.944	+1.1	0.961	0.972	+1.1	0.963	0.991	0.998
2	0.919	0.933	+1.5	0.958	0.976	+1.8	0.951	0.955	0.988
3	0.922	0.944	+2.3	0.948	0.952	+0.4	0.940	0.939	0.977
4	0.910	0.930	+2.0	0.948	0.960	+1.2	0.933	0.932	0.971
5	0.896	0.916	+2.0	0.937	0.978	+4.1	0.922	0.924	0.959

Conditional reliability varies much less across chain position than the cumulative columns suggest, including for the strongest published methods. In our pipelines, however, the AAPT advantage is larger toward the end of the chain. With VLM4VLA, the conditional gain rises from 1.1 points at step one to 2.0 points at step five, with a maximum of 2.3 at step three. With ACoT-VLA, the gain stays within 1.8 points through step four and reaches 4.1 points at the final step. This is also where the base model has its lowest conditional reliability, 0.937. The widening cumulative AAPT gap is therefore not explained only by repeatedly compounding a constant advantage.

J BACKBONE-SIZE SCALING

The backbone-size sweep in Figure 10 uses VLM4VLA because this minimal pipeline introduces fewer additional learned components than ACoT-VLA. Across the tested 2B to 8B range, increasing backbone size leaves LIBERO success nearly flat. AAPT with the 2B backbone already exceeds the base 8B backbone, indicating that the pre-training objective has a larger effect than model size within this range. This result does not imply that capacity is irrelevant outside the tested range. It complements the corpus-size result in Figure 6, where additional assessment data continues to improve transfer.

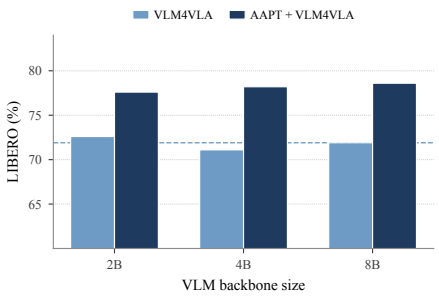

Figure 10. Backbone size with VLM4VLA. Success is flat across the range; the dashed line is the base VLM at 8B.

K FULL BENCHMARK COMPARISONS

The main text abbreviates the published comparisons for readability. Tables 12, 13 and 14 restore all collected methods. Our rows, evaluation protocols and method groupings are unchanged. Published rows are reproduced from the cited papers and may differ in training data, backbone size and policy architecture, so these tables provide context rather than controlled attribution.

Table 12. LIBERO success rate (%), full comparison. Grouping and protocol are as in Table 1.

Method	Spatial	Object	Goal	Long	Avg
Direct policies					
Diffusion Policy (Chi et al., 2023)	78.3	92.5	68.3	50.5	72.4
Octo (Octo Model Team et al., 2024)	78.9	85.7	84.6	51.1	75.1
OpenVLA (Kim et al., 2024)	84.9	88.4	79.2	53.7	76.5
UniAct (Zheng et al., 2025)	77.0	87.0	77.0	70.0	77.8
SpatialVLA (Qu et al., 2025)	88.2	89.9	78.6	55.5	78.1
DiT Policy (Hou et al., 2024)	84.2	96.3	85.4	63.8	82.4
ThinkAct (Huang et al., 2026)	88.3	91.4	87.1	70.9	84.4
π_0-FAST (Black et al., 2024a)	96.4	96.8	88.6	60.2	85.5
FPC-VLA (Yang et al., 2026)	87.0	92.0	86.2	82.2	86.9
SmolVLA (Shukor et al., 2025)	93.0	94.0	91.0	77.0	88.8
GR00T-N1 (Bjorck et al., 2025)	94.4	97.6	93.0	90.6	93.9
π_0 (Black et al., 2024a)	96.8	98.8	95.8	85.2	94.2
VLA-0 (Goyal et al., 2025)	97.0	97.8	96.2	87.6	94.7
GO-1 (Bu et al., 2025a)	96.2	97.8	96.0	89.2	94.8
Discrete Diffusion VLA (Liang et al., 2025)	97.2	98.6	97.4	92.0	96.3
MemoryVLA (Shi et al., 2026)	98.4	98.4	96.4	93.4	96.7
$\pi_{0.5}$ (Black et al., 2025)	98.8	98.2	98.0	92.4	96.9
OpenVLA-OFT (Kim et al., 2025)	97.6	98.4	97.9	94.5	97.1
VLA-Adapter (Wang et al., 2026a)	97.8	99.2	97.2	95.0	97.3
TurboVLA (Xie et al., 2026)	99.2	99.8	97.4	94.2	97.7
ACoT-VLA (Zhong et al., 2026)	98.6	99.0	99.4	97.0	98.5
Prediction & world models					
CoT-VLA (Zhao et al., 2025a)	87.5	91.6	87.6	69.0	81.1
WorldVLA (Cen et al., 2025b)	87.6	96.2	83.4	60.0	81.8
Seer (Tian et al., 2025)	–	–	–	87.7	87.7
FlowVLA (Zhong et al., 2025)	93.2	95.0	91.6	72.6	88.1
DreamVLA (Zhang et al., 2025c)	97.5	94.0	89.5	89.5	92.6
UD-VLA (Chen et al., 2026)	94.1	95.7	91.2	89.6	92.7
RynnVLA-002 (Cen et al., 2025a)	94.2	96.4	94.6	87.6	93.3
UniVLA (Bu et al., 2025b)	95.4	98.8	93.5	94.0	95.5
F1 (Lv et al., 2025)	98.2	97.8	95.4	91.3	95.7
GE-Act (Liao et al., 2026)	98.2	97.6	95.8	94.4	96.5
MMaDA-VLA (Liu et al., 2026)	98.8	99.8	98.0	95.2	98.0
Ours, on Qwen3-VL-4B					
VLM4VLA	80.2	87.2	72.4	44.4	71.1 $\pm$2.4
AAPT + VLM4VLA	84.0	90.2	77.6	60.8	**78.2** $\pm$2.1
ACoT-VLA	98.4	98.6	98.4	95.8	97.8 $\pm$0.6
AAPT + ACoT-VLA	98.8	99.4	98.8	96.0	**98.2** $\pm$0.4

Table 13. CALVIN ABC→D, full comparison. Grouping and columns are as in Table 2.

Method	1/5	2/5	3/5	4/5	5/5	Avg. Len.
Direct policies						
Diffusion Policy (Chi et al., 2023)	40.2	12.3	3.2	1.3	0.0	0.56
RT-1 (Brohan et al., 2023b)	53.3	22.2	9.4	3.8	1.3	0.90
VLAS (Zhao et al., 2025b)	87.2	64.2	40.9	28.1	19.6	2.40
RoboFlamingo (Li et al., 2024a)	82.4	61.9	46.6	33.1	23.5	2.47
DeeR-VLA (Yue et al., 2024)	84.8	72.3	54.9	44.6	33.5	2.90
OpenVLA (Kim et al., 2024)	91.3	77.8	62.0	52.1	43.5	3.27
π_0 (Black et al., 2024a)	96.8	85.0	76.7	68.1	59.9	3.92
LLaDA-VLA (Wen et al., 2025)	95.6	87.7	79.5	73.9	64.5	4.01
OpenHelix (Cui et al., 2025)	97.1	91.4	82.8	72.6	64.1	4.08
OpenVLA-OFT (Kim et al., 2025)	96.3	89.1	82.4	75.8	66.5	4.10
RoboVLMs (Liu et al., 2025a)	98.0	93.6	85.2	77.9	70.4	4.25
VLA-Adapter (Wang et al., 2026a)	99.1	94.6	88.8	82.8	76.5	4.42
Prediction & world models						
SuSIE (Black et al., 2024b)	87.0	69.0	49.0	38.0	26.0	2.69
GEVRM (Zhang et al., 2025a)	92.0	70.0	54.0	41.0	26.0	2.83
GR-1 (Wu et al., 2024)	85.4	71.2	59.6	49.7	40.1	3.06
CLOVER (Bu et al., 2024)	96.0	83.5	70.8	57.5	45.4	3.53
UniVLA (Wang et al., 2026b)	95.5	85.8	75.4	66.9	56.5	3.80
ReconVLA (Song et al., 2026)	95.6	87.6	76.9	69.3	64.1	3.95
UP-VLA (Zhang et al., 2025b)	92.8	86.5	81.5	76.9	69.9	4.08
Seer (Tian et al., 2025)	96.3	91.6	86.1	80.3	74.0	4.28
VPP (Hu et al., 2025)	95.7	91.2	86.3	81.0	75.0	4.33
UniVLA (Bu et al., 2025b)	98.9	94.8	89.0	82.6	75.1	4.41
DreamVLA (Zhang et al., 2025c)	98.2	94.6	89.5	83.4	78.1	4.44
MMaDA-VLA (Liu et al., 2026)	99.8	98.6	96.3	93.5	89.7	4.78
Ours, on Qwen3-VL-4B						
VLM4VLA	93.3	85.7	79.0	71.9	64.4	3.94 $\pm$0.08
AAPT + VLM4VLA	94.4	88.1	83.2	77.4	70.9	**4.14** $\pm$0.06
ACoT-VLA	96.1	92.1	87.3	82.8	77.6	4.36 $\pm$0.05
AAPT + ACoT-VLA	97.2	94.9	90.3	86.7	84.8	**4.54** $\pm$0.03

Table 14. Zero-shot LIBERO-Plus success rate (%), full comparison. Protocol is as in Table 4; * reproduced from released checkpoints by Zhong et al. (2026).

Method	Camera	Robot	Lang.	Light	Backgr.	Noise	Layout	Avg
OpenVLA (Kim et al., 2024)	0.8	3.5	23.0	8.1	34.8	15.2	28.5	16.3
WorldVLA (Cen et al., 2025b)	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.6
NORA (Hung et al., 2025)	2.2	37.0	65.1	45.7	58.6	12.8	62.1	40.5
UniVLA (Bu et al., 2025b)	1.8	46.2	69.6	69.0	81.0	21.2	31.9	45.8
π_0-FAST (Black et al., 2024a)	65.1	21.6	61.0	73.2	73.2	74.4	68.8	62.5
RIPT-VLA (Tan et al., 2025)	55.2	31.2	77.6	88.4	91.6	73.5	74.2	70.2
π_0^* (Black et al., 2024a)	61.0	40.8	63.5	89.3	84.1	80.1	76.4	70.7
OpenVLA-OFT (Kim et al., 2025)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	71.4
$\pi_{0.5}^*$ (Black et al., 2025)	75.8	79.4	83.3	95.5	95.0	89.6	87.0	86.5
ACoT-VLA (Zhong et al., 2026)	72.6	82.6	87.5	97.7	96.5	87.8	88.1	87.5
Ours, on Qwen3-VL-4B								
ACoT-VLA	72.4	82.2	88.0	96.1	95.4	87.2	87.9	87.0 $\pm$0.7
AAPT + ACoT-VLA	**75.4**	**83.4**	**88.8**	**97.5**	**97.2**	**89.6**	**89.5**	**88.8** $\pm$0.8