MAE I TRUST MYSELF? SELF-EVALUATING VLA ACTION GENERATION WITH MARKOV ATTENTION ENTROPY

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action models (VLAs) integrate visual perception, language instruction, and action generation into end-to-end policies across heterogeneous architectures. However, enabling VLAs to self-evaluate their action generation reliability without external supervision remains a major challenge. Existing methods either rely on expert annotations or estimate uncertainty only from output statistics, largely ignoring internal signals. In this work, we observe that internal visual modality entropy exhibits consistent distinctions between successful and failed tasks across heterogeneous VLAs. Although VLAs' architectures differ in their action generation, we show that they share a common latent action generation abstraction evolving under visual perception, language instruction, and State Input, which we formulate as a Conditional Generative Markov Chain. Based on this formulation, we propose MAE (Markov Attention Entropy), a self-evaluation framework that directly converts internal attention signals into architecture-aware reliability scores, and introduce `LIBERO-Reflect`, a 4,000-episode benchmark combining 2,000 standard episodes and 2,000 challenging episodes across four subsets. Extensive experiments across heterogeneous VLA architectures and diverse scenarios show that MAE consistently outperforms state-of-the-art baselines on AUPR, AUROC, and FPR@95. Code is available in this repository.

1 INTRODUCTION

Vision-Language-Action models (VLAs) (Sapkota et al., 2026; Kim et al., 2024; 2025) translate visual observations and language instructions into executable robot actions by adapting pretrained Vision-Language Models to embodied control as illustrated in Figure 1. Their action-generation mechanisms, however, are heterogeneous. In this work, we distinguish two broad action-generation families: Latent-Readout VLAs and Latent-Refinement VLAs. The former evolves latent action representations and maps the final latent state to executable actions through a readout head; `OpenVLA` uses autoregressive discrete readout (Kim et al., 2024), while `OpenVLA-OFT` uses continuous readout (Kim et al., 2025). The latter maintains an evolving action or action-trajectory representation and refines it iteratively under learned flow- or diffusion-style dynamics, as in PI-family policies (Physical Intelligence et al., 2025). As VLAs generalize across robotic tasks (Liu et al., 2023), they also require reliable estimates of when their generated actions are likely to fail (Zhou et al., 2025).

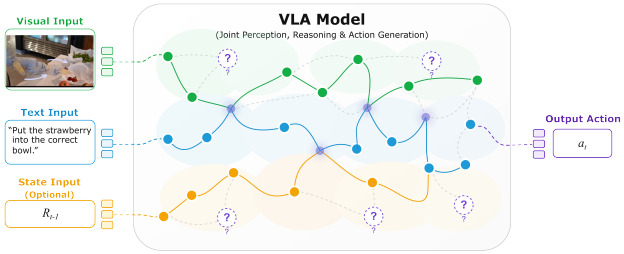

Figure 1: From multimodal conditioning to action generation in VLAs. We view VLA action generation as a conditional generative Markov chain: visual observations, language instructions, and State Input jointly condition a sequence of latent action states that evolves toward final action. Attention exposes how these latent states route information across modalities, and its entropy serves as a white-box signal for self-evaluating action-generation reliability.

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.

The first line of reliability work relies on **External Supervision**. Early studies introduced output-level uncertainty and quality metrics, often calibrated by expert annotations (Valle et al., 2025). Later work moved beyond passive evaluation by using auxiliary models to detect, explain, verify, and recover from failures (Duan et al., 2024; Qi et al., 2026; Lin et al., 2025; Kwok et al., 2025; Dai et al., 2025; Liu et al., 2025; Chen et al., 2026). These methods can improve VLA reliability, but their evaluation signal is supplied outside the policy. This can increase deployment cost, introduce sensitivity to evaluator or annotation shift, and give limited access to the internal state transitions that produced a failed action (Casper et al., 2023).

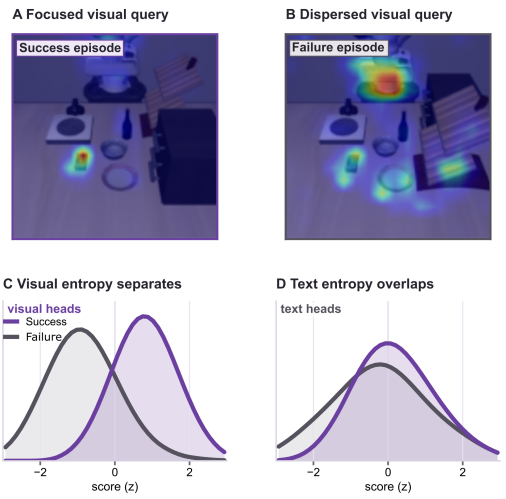

Figure 2: Attention entropy as an internal self-evaluation signal. Panels A–B show a successful visual query concentrated on the task-relevant object and a failed visual query dispersed across distractor regions. Panels C–D compare oriented episode-level score distributions from visual and text attention entropy on the same `Reflect-Goal` episodes. Visual entropy yields a much clearer separation between successful and failed episodes than text entropy, supporting visual attention entropy as the action-relevant self-evaluation signal.

A second line studies **Internal Self-evaluation**, where reliability is inferred from the VLA's own signals. Existing internal methods can be grouped into black-box and white-box approaches. Black-box methods rely on output statistics, such as multi-sample disagreement or self-uncertainty scores, to select actions without external supervision (Bi et al., 2025; Kadavath et al., 2022; Wang et al., 2022). However, they can be over-confident and provide limited interpretability (Bi et al., 2025). White-box self-evaluation has been studied more systematically in Large Language Models (LLMs), where internal signals such as softmax confidence, temperature scaling, or activation patterns are used to estimate generation reliability without external supervision (Bi et al., 2025; Si et al., 2023; Huang et al., 2023; Malinin & Gales, 2021). For VLAs, related evidence remains limited and fragmented. Existing studies show that internal signals can reflect path deviation, execution horizon, pathway specialization, or latent action bottlenecks (Jeong et al., 2026; Wang et al., 2026; Grant et al., 2026; Häon et al., 2025; Buurmeijer et al., 2026; Lian et al., 2026). What is still missing is a unified way to convert such internal signals into self-evaluation metrics that remain meaningful across heterogeneous VLA action-generation architectures. This gap motivates the following question:



Can VLA internal signals be transformed into self-evaluation metrics for action-generation reliability under a unified account of heterogeneous action-generation mechanisms, without relying on external supervision?

The Present Work. We propose `MAE` (Markov Attention Entropy), a white-box self-evaluation framework that converts a VLA's internal attention dynamics into reliability scores. Heterogeneous VLA architectures share a common action-generation structure: visual observations, language instructions, and State Input form the conditioning context, while latent action states evolve under this context toward the executable action. The architectural difference lies in how this evolution is implemented. In **Latent-Readout VLAs**, the policy evolves latent action representations and maps the final latent state to an executable action through a readout head, covering autoregressive discrete readout and continuous readout. In **Latent-Refinement VLAs**, the policy maintains an action or action-trajectory representation and refines it through repeated update steps, covering flow-style refinement and diffusion-style generation.

This conditioned state evolution is captured by the Conditional Generative Markov Chain view illustrated in Figure 3. The latent action state is the Markov state of the internal generation process. Conditioned on the current state and the conditioning context, the transition kernel determines how the next latent action state is formed, whether the update is implemented by autoregressive discrete

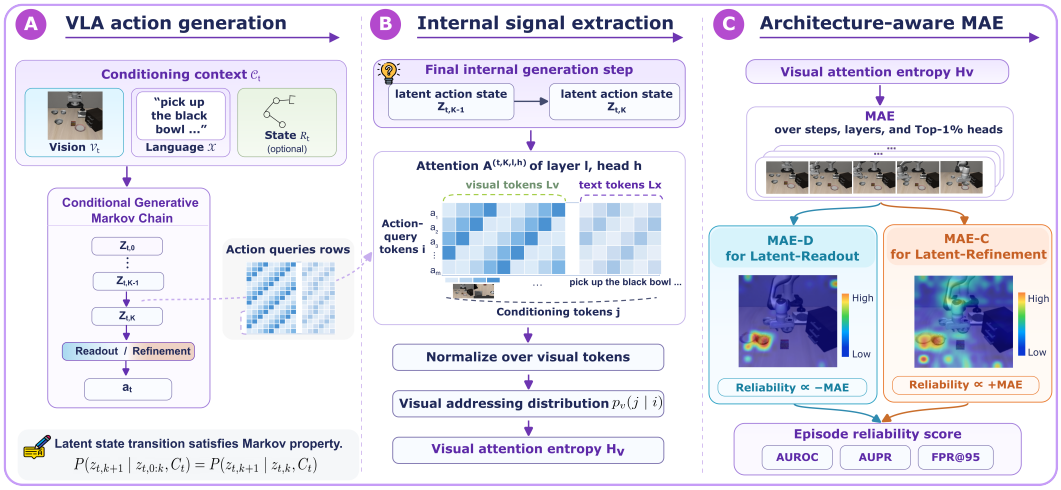

Figure 3: Unified Conditional Generative Markov Chain view of heterogeneous VLAs. Visual observations, language instructions, and State Input form the conditioning context, while latent action states evolve through architecture-specific transition kernels before the executable action is produced. MAE reads attention entropy from this internal transition process and converts it into architecture-aware self-evaluation scores.

readout, continuous readout, or flow-style refinement. Attention records which visual and language tokens the current latent action state consults during this transition process. Attention entropy measures this information routing and yields architecture-aware reliability scores.

In Figure 2, the paired attention maps show that successful and failed queries differ in how the latent action state addresses visual evidence, while the episode-level distributions show that visual-entropy scores separate reliable and unreliable executions more clearly than text-entropy scores on the same `Reflect-Goal` episodes. In the Markov formulation, visual attention entropy therefore measures how the transition kernel accesses visual perception when updating the latent action state. For Latent-Readout VLAs, failed episodes show more diffuse visual addressing, and MAE-D scores the corresponding loss of concentration. For Latent-Refinement VLAs, failed episodes show overly concentrated visual addressing at the final refinement step, and MAE-C scores the resulting loss of visual coverage before the action trajectory is returned. Both metrics are oriented so that larger values indicate higher estimated reliability.

We evaluate MAE on three open-source VLAs spanning the two action-generation families: `OpenVLA` and `OpenVLA-OFT` as Latent-Readout VLAs, and `QwenPI-Flow` as a Latent-Refinement VLA. The evaluation uses `LIBERO-Reflect`, a 4,000-episode benchmark combining 2,000 standard episodes and 2,000 challenging episodes across four subsets. The benchmark covers four capability axes, Goal Semantics, Object Binding, Spatial Grounding, and Composite Generalization, and evaluates whether a self-evaluation score ranks successful episodes above failed episodes. Across `AUROC`, `AUPR`, and `FPR@95`, MAE improves reliability ranking over all baselines without an external evaluator, repeated rollouts, or auxiliary model passes. Our key contributions can be summarized as follows:

- ❶ We formulate heterogeneous VLA action generation as a Conditional Generative Markov Chain, making the transition kernel the common object that links Latent-Readout VLAs and Latent-Refinement VLAs.
- ❷ We introduce MAE, a white-box self-evaluation framework that scores the visual-attention entropy of these transitions with architecture-aware orientations, MAE-D for Latent-Readout VLAs and MAE-C for Latent-Refinement VLAs.
- ❸ We construct `LIBERO-Reflect`, a 4,000-episode benchmark combining 2,000 standard episodes and 2,000 challenging episodes across four subsets.
- ❹ We show across three open-source VLAs that MAE improves reliability ranking over black-box and white-box baselines without an external evaluator, repeated rollouts, or auxiliary model passes.

2 PRELIMINARY

2.1 THE CONDITIONAL GENERATIVE MARKOV CHAIN

We use a Conditional Generative Markov Chain to formalize the action-generation process of a VLA. The chain describes the model's latent state evolution during action generation, not the physical robot-environment dynamics. At timestep t, the VLA evolves a sequence of latent action states $z_{t,0:K} \in \mathcal{Z}$. These states are hidden representations used for action generation and are not directly executed by the robot; the executable action $a_t \in \mathcal{A}$, represented either as a single action or as an action chunk depending on the policy action head, is produced from the final latent action state, $a_t = g_\theta(z_{t,K})$. The chain is conditioned on the conditioning context $C_t = (V_t, X, R_t) \in \mathcal{C}$, where V_t denotes visual representations, X denotes the language instruction, and R_t denotes optional State Input when used by the policy. During the internal generation of a_t, C_t is fixed, while $z_{t,k}$ stores the complete action-generation state at internal generation step k: the generated action-token prefix for autoregressive discrete readout, the latent action representation for continuous readout, or the current action trajectory for flow-style refinement.

Formal definition. We define the internal action-generation process as a conditional generative Markov chain $\mathcal{M} = (\mathcal{Z}, \mathcal{C}, K, \mathcal{P}_\theta)$, where $\mathcal{Z}$ is the latent action-state space, $\mathcal{C}$ is the conditioning space, K is the number of internal generation steps, and $\mathcal{P}_\theta$ is the model's neural transition kernel. At timestep t, the internal process is $z_{t,0} \to z_{t,1} \to \cdots \to z_{t,K} \to a_t$, where $z_{t,k}$ is the latent action state at internal step k, and a_t is the executable action. The chain is specified by an initial distribution and an architecture-dependent transition kernel:

$$z_{t,0} \sim \rho_0(\cdot \mid C_t), \qquad z_{t,k+1} \sim \mathcal{P}_\theta(\cdot \mid z_{t,k}, C_t), \quad k = 0, \ldots, K-1. \tag{1}$$

Here ρ_0 is the initial latent action-state distribution, and $\mathcal{P}_\theta$ is the neural transition kernel induced by the VLA architecture, such as autoregressive action-token updates, continuous readout, or flow-style refinement. The probabilistic notation also covers deterministic updates as degenerate kernels. Equivalently, the conditional joint distribution factorizes as

$$P_\theta(z_{t,0:K} \mid C_t) = \rho_0(z_{t,0} \mid C_t) \prod_{k=0}^{K-1} \mathcal{P}_\theta(z_{t,k+1} \mid z_{t,k}, C_t), \tag{2}$$

The final action is $a_t = g_\theta(z_{t,K})$, where g_θ denotes the architecture-specific action head.

Markov property. The Markov property holds at the level of the *complete* latent action state: given the current latent action state $z_{t,k}$ and the conditioning context C_t, the next latent action state $z_{t,k+1}$ does not depend additionally on earlier states $z_{t,0:k-1}$. That is, $P(z_{t,k+1} \mid z_{t,0:k}, C_t) = P(z_{t,k+1} \mid z_{t,k}, C_t)$. Because $z_{t,k}$ denotes the complete action-generation state, this formulation identifies the transition kernel as the common object shared by heterogeneous VLA action-generation.

3 METHODOLOGY

3.1 MARKOV ATTENTION ENTROPY

Under the Conditional Generative Markov Chain interpretation, each transition $z_{t,k-1} \to z_{t,k}$ is an information-routing step. The transition kernel updates the latent action state by querying the conditioning context. We use transformer attention to measure this routing behavior. For internal generation step k, layer ℓ, and head h, let $\mathbf{A}^{(t,k,\ell,h)} \in \mathbb{R}^{S_a \times S_c}$ be the attention matrix from action-query tokens to conditioning context tokens, where S_a is the number of action-query tokens, and S_c is the number of conditioning context tokens. **MAE** evaluates attention at $k = K$, the final internal generation step before a_t is produced. We decompose the key side into modality-specific token sets $\mathcal{I}_v$ and $\mathcal{I}_x$, corresponding to vision and language tokens. For the visual modality, the normalized visual addressing distribution is

$$p_v^{(t,k,\ell,h)}(j \mid i) = \frac{A_{ij}^{(t,k,\ell,h)}}{\sum_{r \in \mathcal{I}_v} A_{ir}^{(t,k,\ell,h)} + \epsilon}, \qquad j \in \mathcal{I}_v. \tag{3}$$

The visual attention entropy is

$$H_v^{(t,k,\ell,h)}(i) = -\sum_{j \in \mathcal{I}_v} p_v^{(t,k,\ell,h)}(j \mid i) \log p_v^{(t,k,\ell,h)}(j \mid i). \tag{4}$$

This entropy measures the uncertainty of visual addressing. A high value indicates that the action state distributes its visual attention broadly across many patches, while a low value indicates concentrated visual addressing over fewer patches. In the Conditional Generative Markov Chain view, H_v measures how the transition kernel $\mathcal{P}_\theta$ accesses visual perception when updating the action state.

Episode MAE score We convert the token-level visual entropy into an episode-level score by first averaging over execution steps and action queries:

$$E_{\ell,h} = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{S_a}\sum_{i=1}^{S_a} H_v^{(t,K,\ell,h)}(i), \tag{5}$$

where N is the number of execution steps in the episode, and $E_{\ell,h}$ denotes the averaged visual entropy of head h in layer ℓ. Let m be the number of selected heads per layer. We write the episode-level MAE score with a single entropy orientation $\omega \in \{-1, +1\}$:

$$\texttt{MAE}_\omega^{(m)}(E) = \frac{1}{L}\sum_{\ell=1}^{L}\frac{1}{m}\sum_{h\in \texttt{TopM}_m(\omega E_{\ell,:})} \omega E_{\ell,h}. \tag{6}$$

Here $\texttt{TopM}_m(\omega E_{\ell,:})$ returns the m heads with the largest oriented entropy in layer ℓ. Larger $\texttt{MAE}_\omega^{(m)}(E)$ indicates a more reliable episode. The two MAE metrics used in the experiments are $\texttt{MAE-D}^{(m)} = \texttt{MAE}_{-1}^{(m)}(E)$ and $\texttt{MAE-C}^{(m)} = \texttt{MAE}_{+1}^{(m)}(E)$. MAE-D is used for Latent-Readout VLAs such as OpenVLA and OpenVLA-OFT, where reliable executions concentrate visual addressing before readout. MAE-C is used for Latent-Refinement VLAs such as QwenPI-Flow, where reliable executions retain broader visual addressing at the final refinement step.

4 LIBERO-Reflect Benchmark

Matched benchmark construction. We construct LIBERO-Reflect for episode-level VLA self-evaluation by pairing standard episodes from LIBERO with challenging episodes from LIBERO-PRO. For each suite, we use all 10 tasks and execute every task under 50 non-identical initializations with small scene-state differences, yielding 500 standard episodes per suite. We retain the same task and initialization counts on the challenging side, where task language remains fixed while the placements of target and surrounding objects are swapped. Each subset therefore contains 500 standard and 500 challenging episodes; across four subsets, the benchmark contains 4,000 episodes in total, split evenly between 2,000 standard and 2,000 challenging episodes.

Capability subsets. The four subsets isolate complementary capabilities while preserving the same 1,000-episode composition.

Goal Semantics `Reflect-Goal`
Goal-conditioned task understanding and action-type selection.
500 standard + 500 challenging

Object Binding `Reflect-Object`
Target-object grounding and object–action binding.
500 standard + 500 challenging

Spatial Grounding `Reflect-Spatial`
Spatial-relation reasoning and layout-sensitive action adaptation.
500 standard + 500 challenging

Composite Generalization `Reflect-10`
Mixed goals, objects, and spatial layouts from LIBERO-10.
500 standard + 500 challenging

Episode-level assessment. Each method assigns one scalar reliability score to each rollout. Evaluation uses the rollout's actual *simulator success flag* as the ground-truth label, rather than its nominal standard or challenging source. A stronger self-evaluation method should rank successful executions above failed ones.

Evaluation metrics. We report three complementary ranking metrics.

AUROC ↑	AUPR ↑	FPR@95 ↓
Global ranking quality across successful and failed episodes.	Precision-focused ranking quality under class imbalance.	False-positive rate when 95% of successful episodes are recalled.

Supporting construction statistics and nominal-to-actual label mappings are reported in Appendix C, with dataset-source considerations in Appendix H. The matched standard and challenging pools share suite organization, task count, initialization count, simulator, and rollout protocol. Because all metrics use realized simulator success rather than nominal source membership, `LIBERO-Reflect` evaluates reliability ranking within a matched mixed-difficulty pool rather than source classification.

5 Experiments

In this section, we conduct extensive experiments to answer the following research questions: (**RQ1**) Can `MAE` provide reliable self-evaluation across heterogeneous VLA architectures and various scenarios? (**RQ2**) Does `MAE` introduce significant computational overhead? (**RQ3**) How sensitive is `MAE` to its key components and hyperparameters?

5.1 Experimental Setup

VLA Backbones. We evaluate three representative open-source VLA policies on `LIBERO-Reflect`: `OpenVLA`, `OpenVLA-OFT`, and `QwenPI-Flow`. `OpenVLA` denotes the standard OpenVLA policy in Latent-Readout VLAs, with autoregressive discrete readout. `OpenVLA-OFT` keeps the same OpenVLA backbone family and uses the OFT adaptation recipe with continuous readout. `QwenPI-Flow` denotes a Qwen3-VL-based policy in Latent-Refinement VLAs, with a flow-style action module. Together, these policies pair two VLM backbone families with different action-generation mechanisms and training or adaptation regimes. This design tests whether `MAE` transfers across action-generation architectures, VLM backbone families, and training sources. Detailed checkpoint names, vision-language backbones, action heads, and training sources are provided in Appendix B.

Baselines. We compare against seven baselines covering black-box and white-box self-evaluation. The black-box group includes Random (Bi et al., 2025), Verbal Confidence (Kadavath et al., 2022), and Self-Consistency (Wang et al., 2022). The white-box group includes Maximum Softmax Probability (Si et al., 2023), Perplexity (Si et al., 2023), Entropy (Huang et al., 2023), and Length-normalized Entropy (Malinin & Gales, 2021). For all methods, we compute `AUROC`, `AUPR`, and `FPR@95` under the same episode labels. Random serves as a uniform-ranking lower bound. Because the evaluated VLAs use different action-generation interfaces, some baselines are technically not applicable to specific action heads; baseline definitions, applicability, OpenVLA-specific protocols, and the Verbal Confidence prompt are detailed in Appendix F. The shared evaluation protocol is summarized in Appendix E.

Parameter Settings. We obtain attention from the same policy forward pass used to generate actions, then compute the visual-entropy matrix E defined in the Markov Attention Entropy subsection. The main results use all layers and `Top-1` head selection, selecting one oriented-entropy head per layer. We report `MAE-D` for Latent-Readout VLAs and `MAE-C` for Latent-Refinement VLAs according to the entropy orientations above. Appendix D, Appendix E, and Appendix G summarize exact ablation values, the common scoring protocol and model-specific layer/head settings, and policy input templates.

5.2 Main Results (**RQ1**)

Takeaway ❶: `MAE` improves over black-box and white-box self-evaluation baselines. Table 1 provides a comprehensive analysis of `MAE` under the full black-box and white-box baseline protocol across three VLA backbones and four `LIBERO-Reflect` subsets. `MAE` directly scores the internal visual-attention entropy of latent action generation. Against the Random lower bound, `MAE-D` improves `OpenVLA` by up to 90.1% `AUROC` on Object Binding and reduces `FPR@95` by 70.5%; `MAE-D` improves `OpenVLA-OFT` by 105.3% `AUROC` on Goal Semantics; and `MAE-C` improves `QwenPI-Flow` by 76.3% `AUROC` on Spatial Grounding. For the discrete `OpenVLA` policy, the output-statistic and

Model	Method	Goal Semantics Reflect-Goal			Object Binding Reflect-Object			Spatial Grounding Reflect-Spatial			Composite Reflect-10		
		AUROC ↑	AUPR ↑	FPR@95 ↓	AUROC ↑	AUPR ↑	FPR@95 ↓	AUROC ↑	AUPR ↑	FPR@95 ↓	AUROC ↑	AUPR ↑	FPR@95 ↓
❶ Latent-Readout VLAs													
OpenVLA	**Black-box baselines**												
	Random	48.55	39.27	95.36	47.85	35.98	95.87	48.97	39.49	94.82	52.51	28.88	93.92
	Verbal. Conf	50.36	40.82	92.91	49.74	37.21	93.64	50.88	41.03	92.47	53.69	30.04	91.86
	Self-Cons.†	56.82	41.28	84.90	58.41	42.76	82.35	57.94	44.15	87.80	53.64	27.20	90.64
	White-box baselines												
	MaxProb†	54.72	40.58	88.62	55.83	40.28	86.74	55.61	42.37	90.10	53.28	27.05	91.48
	Perplexity†	55.04	40.73	87.95	56.12	40.71	86.20	56.02	42.59	89.44	53.36	27.12	91.10
	Entropy†	56.10	41.05	86.80	57.48	41.64	84.75	57.17	43.26	88.37	53.82	27.30	90.38
	Length Ent.†	56.38	41.19	86.32	58.02	42.05	84.12	57.52	43.58	87.90	54.02	27.38	89.96
	MAE-D (Top-16)	59.56 +22.7%	40.67 +3.6%	71.52 ↓25.0%	79.64 +66.4%	58.98 +63.9%	48.81 ↓49.1%	63.99 +30.7%	49.67 +25.8%	84.81 ↓10.6%	53.21 +1.3%	26.56 -8.0%	87.43 ↓6.9%
	MAE-D (Top-1)	**63.94** +31.7%	**43.23** +10.1%	**61.92** ↓35.1%	**90.97** +90.1%	**75.88** +110.9%	**28.30** ↓70.5%	**66.86** +36.5%	**50.74** +28.5%	**75.79** ↓20.1%	**54.74** +4.2%	**30.45** +5.4%	**86.46** ↓7.9%
OpenVLA-OFT	**Black-box baselines**												
	Random	47.41	50.24	95.51	48.87	49.86	95.56	47.97	48.74	94.96	51.01	48.44	96.58
	Verbal. Conf	49.12	51.73	92.84	50.68	51.08	93.11	49.63	50.26	92.75	52.45	50.01	93.89
	MAE-D (Top-16)	91.84 +93.7%	94.50 +88.1%	7.94 ↓91.7%	75.96 +55.4%	82.86 +66.2%	64.03 ↓33.0%	91.71 +91.2%	89.85 +84.3%	42.74 ↓55.0%	77.13 +51.2%	62.10 +28.2%	45.06 ↓53.3%
	MAE-D (Top-1)	**97.34** +105.3%	**96.14** +91.4%	**7.14** ↓92.5%	**80.56** +64.8%	**81.10** +62.7%	**63.71** ↓33.3%	**92.63** +93.1%	**93.15** +91.1%	**39.31** ↓58.6%	**78.57** +54.0%	**64.08** +32.3%	**45.06** ↓53.3%
❷ Latent-Refinement VLAs													
QwenPI-Flow	**Black-box baselines**												
	Random	47.58	47.93	95.91	48.75	48.40	95.10	48.11	50.15	94.82	50.81	48.95	96.53
	Verbal. Conf	49.44	49.51	93.42	50.31	49.82	92.68	49.96	51.62	92.31	52.18	50.47	93.77
	MAE-C (Top-20)	62.63 +31.6%	59.16 +23.4%	80.93 ↓15.6%	65.92 +35.2%	63.43 +31.1%	81.59 ↓14.2%	66.89 +39.0%	64.70 +29.0%	73.50 ↓22.5%	68.91 +35.6%	61.92 +26.5%	74.52 ↓22.8%
	MAE-C (Top-1)	**80.57** +69.3%	**80.01** +66.9%	**60.12** ↓37.3%	**75.94** +55.8%	**76.48** +58.0%	**81.18** ↓14.6%	**84.80** +76.3%	**85.46** +70.4%	**68.53** ↓27.7%	**79.52** +56.5%	**76.24** +55.8%	**55.60** ↓42.4%

Table 1: Main self-evaluation results on LIBERO-Reflect. Applicable baselines are grouped as black-box or white-box methods. Purple rules highlight MAE rows, which report the score and relative change against Random; for FPR@95, the relative value is the reduction rate. Bold Top-1 values and underlined half-head values improve over Random within the same model, subset, and metric. † Token-statistic and Self-Consistency baselines are reported only for OpenVLA, whose autoregressive discrete action interface exposes action-token probabilities and sampled action-token sequences. Additional applicability details are provided in Appendix F.

Self-Consistency rows instantiate baselines that depend on action-token probabilities or sampled action-token agreement, while the continuous-action and flow-action policies cannot expose the same random variables without changing the estimator. The OpenVLA-specific baseline protocols are detailed in Appendix F.

Takeaway ❷: MAE transfers across model architectures, initialization families, and task families. The three evaluated policies differ in base VLMs, perception stacks, training data, and action-generation mechanisms. OpenVLA and OpenVLA-OFT instantiate Latent-Readout VLAs with different readout recipes. QwenPI-Flow instantiates Latent-Refinement VLAs with a flow-style action module. The gains on Goal Semantics, Object Binding, and Spatial Grounding show that the signal follows the action-condition coupling structure across distinct capability axes.

Takeaway ❸: MAE yields a shared execution-correctness signal across heterogeneous tasks. Reflect-10 pools diverse goals, objects, and layouts from LIBERO-10 into a single evaluation split. On this heterogeneous task mixture, OpenVLA-OFT reaches 78.57 AUROC and QwenPI-Flow reaches 79.52 AUROC. The entropy signal thus functions as a cross-task execution monitor, discriminating successful from failed rollouts as task semantics and scene configurations vary jointly.

5.3 Cost Analysis (RQ2)

Takeaway ❹: MAE achieves the highest performance in the high-efficiency zone, establishing itself as the Pareto-optimal solution. Figure 4a visualizes the efficiency-reliability Pareto view.

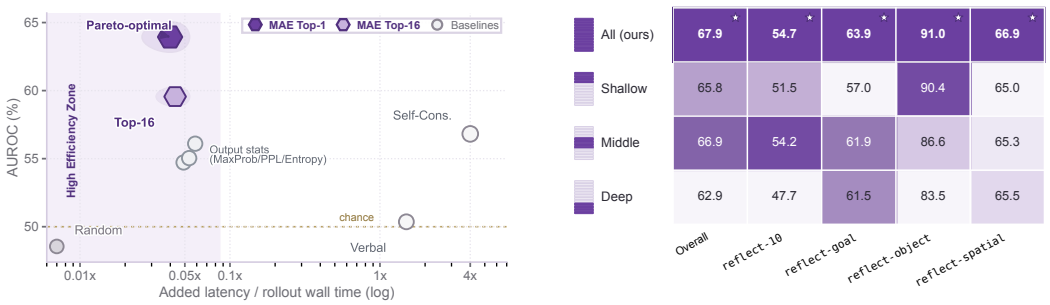

Figure 4: Efficiency–reliability Pareto analysis and layer-band ablation for MAE. The left panel reports added evaluation latency normalized by one rollout wall time. The right panel reports absolute AUROC for MAE-D with Top-1; colors are normalized within each column by closeness to the all-layer result, stars mark the best layer band, and the left schematic indicates the layer region.

While baselines like Self-Consistency or Verbal Confidence require multiple sampled action-token generations or external models (incurring $\geq 1\times$ latency overhead), MAE reuses internal attention maps and adds minimal computation ($< 0.1\times$ overhead; detailed hardware profiling on NVIDIA H100 is provided in Appendix J).

5.4 Framework Analysis (**RQ3**)

We ablate the key design choices in MAE from the following perspectives:

Text-head versus visual-head signals. We first test whether the reliability signal can be recovered from the language side alone. Figure 5 compares the text-head MAE Top-1 score against the visual-attention MAE Top-1 score used in Table 1; exact values are reported in Appendix D. The visual head consistently improves AUROC, with especially large gains on OpenVLA Object Binding and QwenPI-Flow Goal Semantics and Reflect-10. This shows that language-head entropy can reflect text-side decisiveness, but it does not reliably capture whether the action latent has gathered the visual evidence needed for manipulation. The visual head is therefore the appropriate signal for self-evaluating action generation.

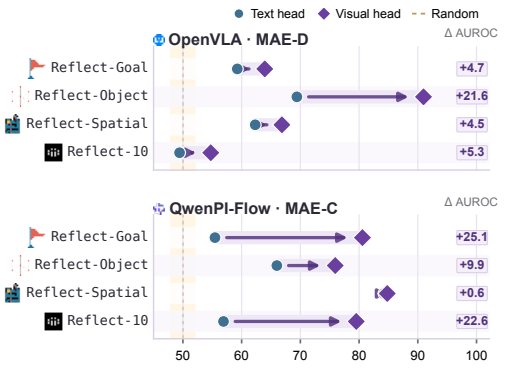

Figure 5: Text-head versus visual-head self-evaluation signals. OpenVLA uses MAE-D, and QwenPI-Flow uses MAE-C. Each connector reports ΔAUROC = V - T. Positive gaps across all subsets indicate that text-side entropy is not a stable substitute for visual information.

Impact of head selection. We evaluate how the number of selected heads affects MAE by varying Top-m. Figure 6 summarizes the full results by plotting each setting's drop from Top-1 in mean ranking quality, averaged over AUROC and AUPR across the four subsets; exact values are reported in Appendix D. Top-1 gives the most consistent ranking performance across all three models. It also uses the smallest selected-head set, reducing entropy aggregation cost and limiting the contribution of low-signal heads. This supports Top-1 as the default head-selection setting.

Impact of layer selection. To assess the effect of model depth, we compare the default all-layer aggregation with three layer-restricted variants: shallow, middle, and deep. The experiment keeps the same MAE-D score with Top-1 and changes only the layer band used to aggregate $E_{\ell,h}$. Figure 4b reports absolute AUROC values while coloring each column by closeness to the all-layer result, showing that all-layer aggregation gives the best overall AUROC and remains competitive across task families. This result is consistent with the layered function of transformer VLAs: earlier layers encode visual grounding and token-level perception, middle layers support cross-modal binding, and deeper layers are closer to action readout. Combining all layers provides a fuller internal signal and avoids introducing task- or architecture-specific layer-band hyperparameters.

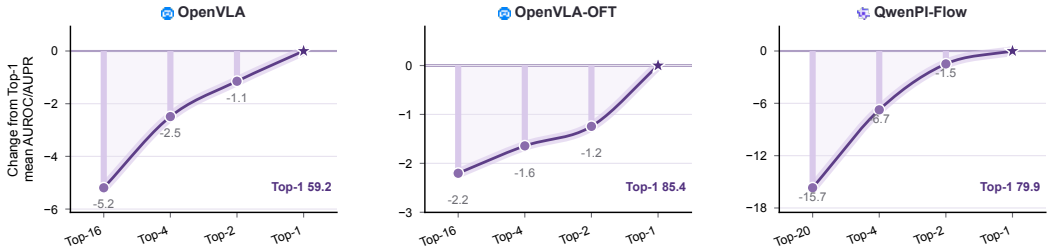

Figure 6: `Top-m` head selection ablation. Panels order selected-head settings from many heads to `Top-1` and use independent y-axis scales. Values are percentage-point changes from the `Top-1` ranking score, computed as the mean of `AUROC` and `AUPR` over the four `LIBERO-Reflect` subsets. Exact values are reported in Appendix D.

Run-to-run stability. We repeat the full `QwenPI-Flow` rollout evaluation using five independent simulator seeds. Each seed changes the scene initialization and stochastic rollout trajectory while keeping the policy checkpoint, task set, and evaluation protocol fixed. Table 2 shows consistent gains from `MAE-C Top-1` across all four subsets, with standard deviations no larger than 1.32 percentage points, including for the tail-sensitive `FPR@95`. The improvements are therefore stable across random trials rather than being driven by a favorable single run. Under a two-sided Student-t interval with four degrees of freedom, the largest 95% confidence-interval half-width is 1.64 percentage points. Because `AUPR` depends on positive prevalence, comparisons retain the same episode composition within each subset.

Benchmark subset	Method	AUROC↑	AUPR↑	FPR@95↓
`Reflect-Goal` Goal Semantics	Random	47.75±0.81	48.12±0.79	95.98±0.63
	`MAE-C Top-1`	**80.72±0.79**	**80.14±0.78**	**60.15±1.28**
`Reflect-Object` Object Binding	Random	48.91±0.79	48.56±0.79	95.18±0.64
	`MAE-C Top-1`	**76.12±0.85**	**76.64±0.83**	**81.13±1.27**
`Reflect-Spatial` Spatial Grounding	Random	48.30±0.86	50.31±0.86	94.90±0.62
	`MAE-C Top-1`	**84.95±0.77**	**85.60±0.77**	**68.46±1.32**
`Reflect-10` Composite Generalization	Random	50.99±0.82	49.13±0.83	96.63±0.65
	`MAE-C Top-1`	**79.71±0.80**	**76.41±0.80**	**55.59±1.26**

Table 2: Run-to-run stability of `QwenPI-Flow` with `MAE-C Top-1` across all four subsets. Values are percentages reported as sample mean ± sample standard deviation over five independent simulator seeds.

`MAE` retains within-domain execution reliability. To further assess execution-level reliability under a shared environment distribution, we evaluate `OpenVLA` using only standard LIBERO episodes. We retain all 558 natural policy failures and sample 558 subset-matched successes, yielding a balanced set of 1,116 episodes with 50% success prevalence. Under this setting, `MAE-D Top-1` improves over Length-normalized Entropy by 12.02 points in pooled `AUROC`, 11.36 points in `AUPR`, and 23.34 points in `FPR@95` reduction. The same ordering holds across all four subsets; on Object Binding, `MAE-D Top-1` reaches 88.75 `AUROC`, 86.30 `AUPR`, and 30.15 `FPR@95`. These results show that MAE preserves reliable execution ranking even when successful and failed episodes are drawn from the same standard-domain distribution. Full construction details and subset-level results are provided in Appendix H.1.

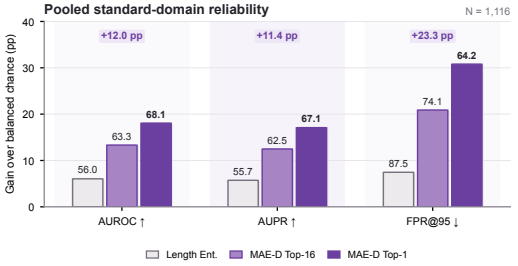

Figure 7: Pooled standard-only balanced results for `OpenVLA`. Bar heights report improvement over the balanced random expectation: score minus 50 for `AUROC`/`AUPR`, and 95 minus score for `FPR@95`. Labels above bars give the original metric values. Purple callouts report the gain of `MAE-D Top-1` over Length-normalized Entropy.

6 Conclusion

In this work, we observe that internal visual modality entropy exhibits consistent distinctions between successful and failed tasks across heterogeneous VLAs. We presented `MAE`, a white-box framework for VLA self-evaluation from internal attention dynamics. We introduce `LIBERO-Reflect`, a 4,000-episode benchmark combining 2,000 standard episodes and 2,000 challenging episodes across four subsets. Experiments on `LIBERO-Reflect` show that `MAE` improves reliability ranking without external evaluators, establishing itself as the Pareto-optimal solution.

AI USE STATEMENT

Generative AI tools were used as assistive tools during manuscript preparation and research development, including language refinement and limited support for routine research tasks. All research decisions, experimental design, implementation, analysis, and scientific conclusions were determined and verified by the authors. All AI-assisted outputs were independently reviewed, and the authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We have made efforts to support reproducibility throughout this work. The proposed method and scoring procedure are specified in Sections 2–3, while the benchmark construction and evaluation protocol are described in Sections 4–5. Additional implementation details, model configurations, baseline protocols, hyperparameter settings, benchmark statistics, and computational information are provided in the appendices and supplementary materials. Together, these descriptions are intended to facilitate independent reproduction of the reported experiments.

ETHICS STATEMENT

This work focuses on methodological research on self-evaluation for vision-language-action models using existing robotic simulation benchmarks. It does not involve human subjects or sensitive personal data. We acknowledge that reliability estimation alone does not constitute a guarantee of safe real-world robotic deployment, and practical deployment requires additional validation and appropriate safety measures.

REFERENCES

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying, 2023. URL https://arxiv.org/abs/2304.13734.

Jinhe Bi, Danqi Yan, Yifan Wang, Wenke Huang, Haokun Chen, Guancheng Wan, Mang Ye, Xun Xiao, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. Cot-kinetics: A theoretical modeling assessing lrm reasoning process. *ArXiv*, abs/2505.13408, 2025. URL https://api.semanticscholar.org/CorpusID:278769227.

Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X. Lee, Maria Bauza, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, Antoine Laurens, Claudio Fantacci, Valentin Dalibard, Martina Zambelli, Murilo Martins, Rugile Pevceviciute, Michiel Blokzijl, Misha Denil, Nathan Batchelor, Thomas Lampe, Emilio Parisotto, Konrad Żołna, Scott Reed, Sergio Gómez Colmenarejo, Jon Scholz, Abbas Abdolmaleki, Oliver Groth, Jean-Baptiste Regli, Oleg Sushkov, Tom Rothörl, José Enrique Chen, Yusuf Aytar, Dave Barker, Joy Ortiz, Martin Riedmiller, Jost Tobias Springenberg, Raia Hadsell, Francesco Nori, and Nicolas Heess. Robocat: A self-improving generalist agent for robotic manipulation, 2023. URL https://arxiv.org/abs/2306.11706.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei 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, 2023. URL https://arxiv.org/abs/2212.06817.

Hugo Buurmeijer, Carmen Amo Alonso, Aiden Swann, and Marco Pavone. Observing and controlling features in vision-language-action models, 2026. URL https://arxiv.org/abs/2603.05487.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Bıyık, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback, 2023. URL https://arxiv.org/abs/2307.15217.

Lingling Chen, Zongyao Lyu, and William J. Beksi. Reconvla: An uncertainty-guided and failure-aware vision-language-action framework for robotic control, 2026. URL https://arxiv.org/abs/2604.16677.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024. URL https://arxiv.org/abs/2303.04137.

Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard,

Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2025. URL https://arxiv.org/abs/2310.08864.

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, 2025. URL https://arxiv.org/abs/2510.10975.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-e: An embodied multimodal language model. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8469–8488. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/driess23a.html.

Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. Aha: A vision-language-model for detecting and reasoning over failures in robotic manipulation, 2024. URL https://arxiv.org/abs/2410.00371.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 06 2024. doi: 10.1038/s41586-024-07421-0.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.090.

Bryce Grant, Xijia Zhao, and Peng Wang. Not all features are created equal: A mechanistic study of vision-language-action models, 2026. URL https://arxiv.org/abs/2603.19233.

Yuheng Huang, Jiayang Song, Zhijie Wang, Shengming Zhao, Huaming Chen, Felix Juefei-Xu, and Lei Ma. Look before you leap: An exploratory study of uncertainty measurement for large language models. *arXiv preprint arXiv:2307.10236*, 2023.

Bear Häon, Kaylene Stocking, Ian Chuang, and Claire Tomlin. Mechanistic interpretability for steering vision-language-action models, 2025. URL https://arxiv.org/abs/2509.00328.

Jaehwan Jeong, Evelyn Zhu, Jinying Lin, Emmanuel Jaimes, Tuan-Anh Vu, Jungseock Joo, Sangpil Kim, and M. Khalid Jawed. Your vision-language-action model already has attention heads for path deviation detection, 2026. URL https://arxiv.org/abs/2603.13782.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts, 2023. URL https://arxiv.org/abs/2210.03094.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021. doi: 10.1162/tacl_a_00407. URL https://aclanthology.org/2021.tacl-1.57/.

Saurav Kadavath, Tom Conerly, Amanda Askell, Thomas Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zachary Dodds, Nova Dassarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, John Kernion, Shauna Kravec,

Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom B. Brown, Jack Clark, Nicholas Joseph, Benjamin Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *ArXiv*, abs/2207.05221, 2022. URL https://api.semanticscholar.org/CorpusID:250451161.

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

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

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms, 2024. URL https://arxiv.org/abs/2406.15927.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation, 2023. URL https://arxiv.org/abs/2302.09664.

Jacky Kwok, Christopher Agia, Rohan Sinha, Matt Foutter, Shulu Li, Ion Stoica, Azalia Mirhoseini, and Marco Pavone. Robomonkey: Scaling test-time sampling and verification for vision-language-action models, 2025. URL https://arxiv.org/abs/2506.17811.

Shijie Lian, Bin Yu, Xiaopeng Lin, Laurence T. Yang, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Cong Huang, and Kai Chen. Langforce: Bayesian decomposition of vision language action models via latent action queries, 2026. URL https://arxiv.org/abs/2601.15197.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words, 2022. URL https://arxiv.org/abs/2205.14334.

Zijun Lin, Jiafei Duan, Haoquan Fang, Dieter Fox, Ranjay Krishna, Cheston Tan, and Bihan Wen. Failsafe: Reasoning and recovery from failures in vision-language-action models, 2025. URL https://arxiv.org/abs/2510.01642.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL https://arxiv.org/abs/2306.03310.

Zeting Liu, Zida Yang, Zeyu Zhang, and Hao Tang. Evovla: Self-evolving vision-language-action model, 2025. URL https://arxiv.org/abs/2511.16166.

Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jN5y-zb5Q7m.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9004–9017, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.557. URL https://aclanthology.org/2023.emnlp-main.557/.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models, 2025. URL https://arxiv.org/abs/2501.09747.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z.

Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Carl Qi, Xiaojie Wang, Silong Yong, Stephen Sheng, Huitan Mao, Sriram Srinivasan, Manikantan Nambi, Amy Zhang, and Yesh Dattatreya. Self-refining vision language model for robotic failure detection and reasoning, 2026. URL https://arxiv.org/abs/2602.12405.

Ranjan Sapkota, Yang Cao, Konstantinos I. Roumeliotis, and Manoj Karkee. Vision-language-action (vla) models: Concepts, progress, applications and challenges, 2026. URL https://arxiv.org/abs/2505.04769.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Lee Boyd-Graber, and Lijuan Wang. Prompting GPT-3 to be reliable. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=98p5x51L5af.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback, 2023. URL https://arxiv.org/abs/2305.14975.

Pablo Valle, Chengjie Lu, Shaukat Ali, and Aitor Arrieta. Evaluating uncertainty and quality of visual language action-enabled robots, 2025. URL https://arxiv.org/abs/2507.17049.

Haoxuan Wang, Gengyu Zhang, Yan Yan, Ramana Rao Kompella, and Gaowen Liu. Vla knows its limits, 2026. URL https://arxiv.org/abs/2602.21445.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171, 2022. URL https://api.semanticscholar.org/CorpusID:247595263.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization, 2025. URL https://arxiv.org/abs/2510.03827.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

Thomas P Zollo and Richard Zemel. Confidence calibration in vision-language-action models, 2025. URL https://arxiv.org/abs/2507.17383.

A Related Work

Vision-Language-Action Models. VLAs translate visual observations and language instructions into output actions, building on progress in large-scale robot learning and multimodal modeling. Early generalist policies such as `RT-1` demonstrate scalable transformer-based control from real-world robot trajectories, while `VIMA` studies robot manipulation conditioned on multimodal task specifications (Brohan et al., 2023; Jiang et al., 2023). `PaLM-E` incorporates visual observations into a pretrained Large Language Model (LLM), and `RT-2` connects web-scale vision-language pretraining with action generation through action-token prediction (Driess et al., 2023; Zitkovich et al., 2023). Large-scale datasets and open policies further support generalization across tasks and robot embodiments, including `Open X-Embodiment`, `RoboCat`, and `Octo` (Collaboration et al., 2025; Bousmalis et al., 2023; Ghosh et al., 2024). Beyond these policy-scale developments, continuous generative policies such as `Diffusion Policy` and efficient action representations such as `FAST` expand the available designs for action generation (Chi et al., 2024; Pertsch et al., 2025). Together, these studies establish a broad family of policies that share visual observations and language instructions as conditions, but differ in how actions are generated.

Self-Evaluation in Large Language Models. Self-evaluation in LLMs studies whether a model can estimate the reliability of its own generated outputs. Calibration-based work evaluates whether model confidence is aligned with answer correctness, including probability-based calibration and uncertainty expressed through natural-language confidence statements (Jiang et al., 2021; Lin et al., 2022; Tian et al., 2023). A second line of work uses repeated generation to measure agreement among candidate responses: self-consistency improves reasoning by aggregating multiple sampled outputs, while `SelfCheckGPT` detects hallucinations from inconsistencies among sampled passages (Wang et al., 2022; Manakul et al., 2023). Moving beyond surface-level agreement, semantic uncertainty groups generations according to their meanings, and semantic entropy uses this semantic-level uncertainty to identify unreliable generations and confabulations (Kuhn et al., 2023; Farquhar et al., 2024). Recent probing studies additionally indicate that internal hidden representations can expose reliability-related signals without requiring full repeated sampling (Azaria & Mitchell, 2023; Kossen et al., 2024). These works provide important foundations for self-evaluation, but their outputs are language generations rather than VLAs' output actions conditioned on visual observations.

Self-Evaluation in Vision-Language-Action Models. Self-evaluation for VLAs must assess the reliability of action generation under visual observations and language instructions. Recent studies evaluate VLA uncertainty or confidence from generated output actions, including uncertainty-quality evaluation and calibration methods for VLA policies (Valle et al., 2025; Zollo & Zemel, 2025). Other work detects or reasons over failures through additional learned components: `AHA` uses a VLM to identify and explain robotic manipulation failures, and `RoboMonkey` introduces test-time sampling and verification for VLA action generation (Duan et al., 2024; Kwok et al., 2025). Complementary studies begin to examine reliability-related signals already present inside VLAs, showing that internal signals can reflect controllable behavior, path deviation, or model limitations (Häon et al., 2025; Jeong et al., 2026; Wang et al., 2026). Different from methods based on external evaluators, supervised failure detectors, or repeated action sampling, **`MAE`** directly converts internal visual attention entropy into episode-level reliability scores for heterogeneous VLA action-generation mechanisms.

B Model Configuration Details

Table 3 maps the model names used in the experiments to the concrete evaluated checkpoints and training configurations. The selected policies deliberately pair two VLM backbone families with different action-generation mechanisms and training or adaptation regimes. This design supports the claim that **`MAE`** is effective across action-generation architectures, VLM backbone families, and training sources.

Policy	Action-Generation Family	VLM and Perception Stack	Training / Adaptation Signal	Diversity Axis
OpenVLA	Latent-Readout VLAs	OpenVLA-7B VLM with prism-dinosiglip-224px; fused DINOv2 + SigLIP visual encoder; Llama 2 7B language backbone	official openvla/openvla-7b; tokenized action prediction trained on Open X-Embodiment	Llama/OpenVLA family; large-scale action-token training; tests autoregressive discrete readout behavior
OpenVLA-OFT	Latent-Readout VLAs	OpenVLA-7B VLM with the same DINOv2 + SigLIP visual base and Llama 2 7B language backbone	official OpenVLA-OFT checkpoints/code; efficient OFT adaptation with continuous actions, action chunking, and L1 regression	Same backbone family as OpenVLA with a continuous readout recipe; tests adaptation robustness
QwenPI-Flow	Latent-Refinement VLAs	Qwen3-VL-4B-Instruct with Qwen3-VL vision stack plus dinov2_vits14 in the StarVLA config	StarVLA/Qwen3-VL-PI-LIBERO-4in1; QwenPI policy with DiT-B, 7-DoF actions, horizon 8, and data_mix=libero_all	Qwen3-VL family; flow-style refinement; LIBERO-specific training mix; tests cross-family and cross-generation generality

Table 3: Detailed configurations for the three VLA policies evaluated in the main experiments. The table highlights architectural, family-level, and training-source diversity.

C LIBERO-Reflect Construction Details

This appendix documents the construction evidence for LIBERO-Reflect. Table 4 fixes the benchmark composition: each subset contains 500 standard episodes and 500 challenging episodes, yielding 1,000 episodes per subset and 4,000 episodes in total. The same table reports policy success rates on the standard side. Table 5 reports diagnostic success rates for the challenging episodes sampled from LIBERO-PRO. Figure 8 complements the tables with 16 visual case-study panels from all four subsets. The challenging episodes preserve the original task language while swapping the placements of target objects and surrounding objects.

Nominal splits versus actual evaluation labels. The 2,000 standard and 2,000 challenging episodes define the *nominal* composition of LIBERO-Reflect, designed to ensure sufficient exposure to both solvable standard scenes and difficult PRO scenes. However, depending on the evaluated policy's capability, nominal standard episodes may occasionally fail (as shown in Table 4), and nominal challenging episodes may occasionally succeed (as shown in Table 5). Because self-evaluation must assess whether the policy *actually* succeeded or failed on a given execution, we do not use the nominal dataset split as the ground-truth label for evaluation. Instead, the ground-truth label for computing all self-evaluation metrics (AUROC, AUPR, FPR@95) is strictly defined by the actual simulator success flag of that specific rollout. This ensures that our evaluation rigorously reflects the true correctness of the generated actions, avoiding any label noise introduced by policy capability variations.

D Ablation Details and Exact Values

This appendix provides the exact values for the ablation figures and explains how to read the trends. The ablations are not additional methods; they test whether MAE depends on broad head averaging or manually selected layer ranges. The main configuration deliberately uses all layers with Top-1 head selection, because this setting preserves the strongest and most stable architecture-aware signal while adding the least aggregation overhead.

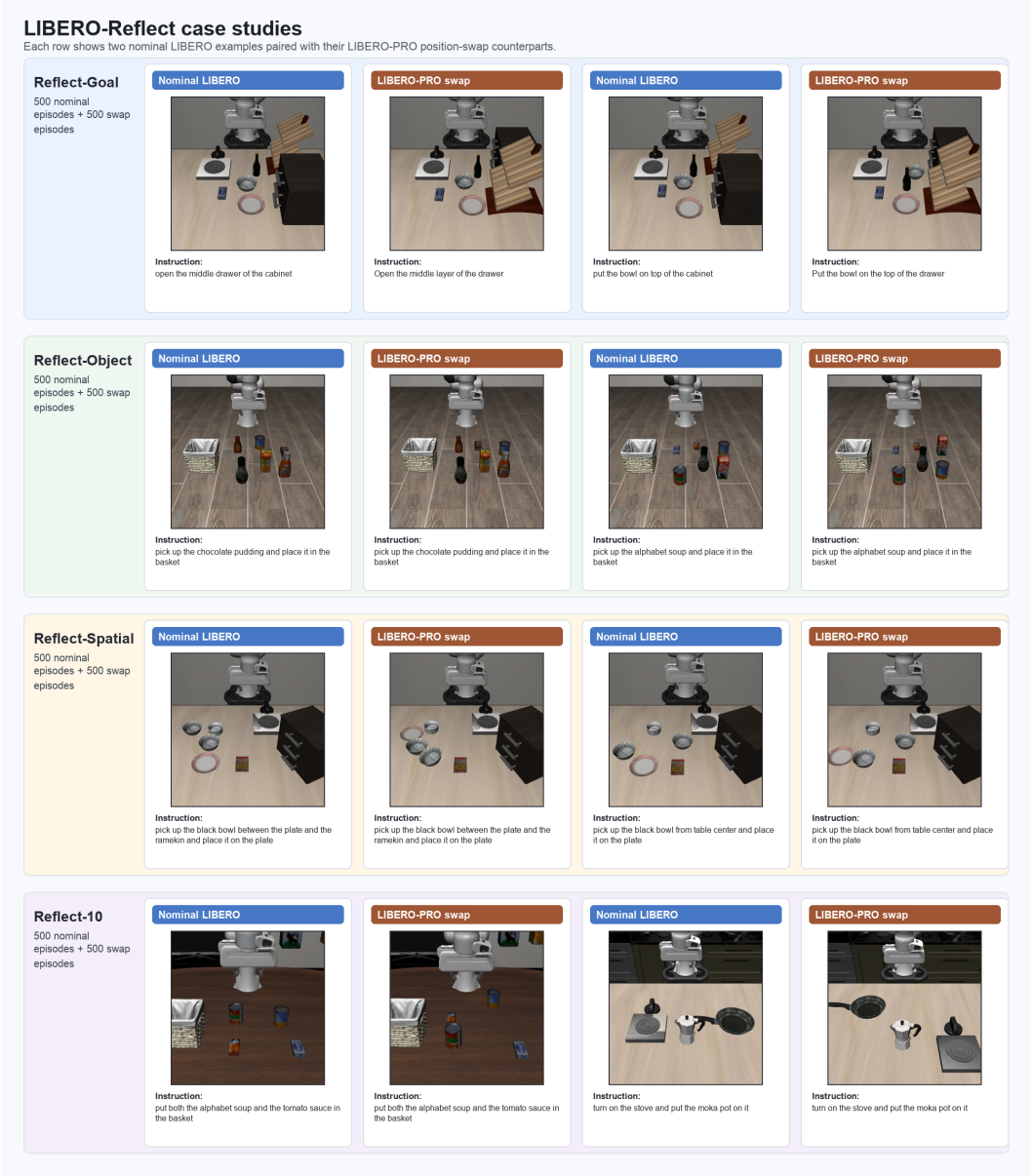

Figure 8: Case studies from the `LIBERO-Reflect` construction. Each row corresponds to one subset and shows representative standard episodes paired with challenging episodes. The instruction is printed below each panel, yielding 16 data points across the four subsets.

Component	Goal Semantics Reflect-Goal	Spatial Grounding Reflect-Spatial	Composite Reflect-10	Object Binding Reflect-Object	Overall
Task count	10	10	10	10	40
Standard episodes	500	500	500	500	2,000
Challenging episodes	500	500	500	500	2,000
Total episodes	1,000	1,000	1,000	1,000	4,000
❶ Standard-side policy success rate (%)					
QwenPI-Flow	96.8	93.8	94.4	98.0	95.8
OpenVLA-OFT	97.2	93.6	94.8	99.6	96.3
OpenVLA	79.2	79.8	55.2	74.2	72.1

Table 4: LIBERO-Reflect composition and standard-side policy success rates. The upper block reports the number of tasks and episodes used to form the benchmark. The lower block reports policy success rates on the standard side. Overall success is the mean over four equally sized suites. QwenPI-Flow corresponds to the Qwen3-VL-PI-LIBERO-4in1 checkpoint described in Table 3.

Policy	Goal Semantics Reflect-Goal	Spatial Grounding Reflect-Spatial	Composite Reflect-10	Object Binding Reflect-Object	Mean
QwenPI-Flow	0.0	7.2	1.8	0.0	2.25
OpenVLA-OFT	4.6	7.2	0.0	1.2	3.25
OpenVLA	0.0	0.4	0.0	0.0	0.10

Table 5: Diagnostic success rates for the challenging episodes sampled from LIBERO-PRO and retained in LIBERO-Reflect. Values are percentages and summarize the challenging side used by the benchmark.

Model	Subset	Text Head	Text AUROC	Visual Head	Visual AUROC	ΔAUROC
❶ Latent-Readout VLA						
OpenVLA	Reflect-Goal	MAE-D Top-1	59.26	MAE-D Top-	**63.94**	**+4.68**
	Reflect-Object	MAE-D Top-1	69.41	MAE-D Top-	**90.97**	**+21.56**
	Reflect-Spatial	MAE-D Top-1	62.31	MAE-D Top-	**66.86**	**+4.55**
	Reflect-10	MAE-D Top-1	49.43	MAE-D Top-	**54.74**	**+5.31**
❷ Latent-Refinement VLA						
QwenPI-Flow	Reflect-Goal	MAE-C Top-1	55.46	MAE-C Top-	**80.57**	**+25.11**
	Reflect-Object	MAE-C Top-1	66.00	MAE-C Top-	**75.94**	**+9.94**
	Reflect-Spatial	MAE-C Top-1	84.25	MAE-C Top-	**84.80**	**+0.55**
	Reflect-10	MAE-C Top-1	56.91	MAE-C Top-	**79.52**	**+22.61**

Table 6: Exact AUROC values for the text-head versus visual-head comparison in the main paper. Purple cells mark the visual-head signal used by MAE, and purple deltas report visual-head MAE minus text-head MAE under the same Top-1 setting. Subset names use the LIBERO-Reflect split identifiers.

Text versus visual heads. Table 6 reports the exact values for the text-head versus visual-head comparison in the main paper. We keep only the text-head Top-1 counterpart and use the same orientation terminology as the main method: MAE-D for OpenVLA and MAE-C for QwenPI-Flow. The comparison shows why text-side entropy is insufficient: it can be high on subsets where language-side uncertainty is predictive, but the large visual advantages on OpenVLA Object Binding and QwenPI-Flow Goal Semantics/Reflect-10 show that it misses visually grounded action failures.

Head selection. Table 7 reports the full Top-m head-selection sweep used in the main paper. Across all three models, Top-1 is the most reliable default when considering both AUROC and AUPR.

Model	Top-m Setting	Goal Semantics Reflect-Goal			Object Binding Reflect-Object			Spatial Grounding Reflect-Spatial			Composite Reflect-10		
		AUROC	AUPR	FPR @95	AUROC	AUPR	FPR @95	AUROC	AUPR	FPR @95	AUROC	AUPR	FPR @95
❶ Latent-Readout VLAs													
OpenVLA	MAE-D Top-1	**63.94**	**43.23**	**61.92**	**90.97**	**75.88**	**28.30**	**66.86**	**50.74**	**75.79**	**54.74**	**30.45**	**86.46**
	MAE-D Top-2	62.41	42.33	68.21	89.14	72.76	30.05	64.72	52.29	77.13	54.01	26.97	85.91
	MAE-D Top-4	60.36	41.01	70.03	87.13	69.26	34.18	64.25	52.14	78.80	53.20	26.55	86.88
	MAE-D Top-16	59.56	40.67	71.52	79.64	58.98	48.81	63.99	49.67	84.81	53.21	26.56	87.43
OpenVLA-OFT	MAE-D Top-1	**97.34**	**96.14**	**7.14**	**80.56**	**81.10**	**63.71**	**92.63**	**93.15**	**39.31**	**78.57**	**64.08**	**45.06**
	MAE-D Top-2	96.86	95.26	7.76	76.74	77.29	65.32	93.10	93.59	38.10	77.86	62.92	43.73
	MAE-D Top-4	94.63	94.38	8.37	78.42	79.10	63.71	90.25	93.01	35.48	77.79	62.85	46.39
	MAE-D Top-16	91.84	94.50	7.94	75.96	82.86	64.03	91.71	89.85	42.74	77.13	62.10	45.06
❷ Latent-Refinement VLAs													
QwenPI-Flow	MAE-C Top-1	**80.57**	**80.01**	**60.12**	**75.94**	**76.48**	**81.18**	**84.80**	**85.46**	**68.53**	**79.52**	**76.24**	**55.60**
	MAE-C Top-2	78.75	84.28	55.45	73.01	72.27	82.35	85.73	82.67	53.00	77.75	72.72	56.18
	MAE-C Top-4	74.16	81.17	64.79	63.42	59.80	86.27	79.63	79.03	68.12	77.47	70.39	59.27
	MAE-C Top-20	62.63	59.16	80.93	65.92	63.43	81.59	66.89	64.70	73.50	68.91	61.92	74.52

Table 7: Top-m head-selection ablation values for the main paper. Bold values mark the Top-1 setting used in the main results, while underlined values mark the half-head comparison: Top-16 for 32-head OpenVLA-family models and Top-20 for the 40-head QwenPI-Flow model. Top-1 gives the most stable ranking quality while avoiding noisy aggregation over many heads.

Layer Band	Overall All subsets	Composite Reflect-10	Goal Semantics Reflect-Goal	Object Binding Reflect-Object	Spatial Grounding Reflect-Spatial
All layers	**67.88**	**54.74**	**63.94**	**90.97**	**66.86**
Shallow	65.82	51.49	57.00	90.43	64.97
Middle	66.88	54.25	61.93	86.59	65.28
Deep	62.91	47.71	61.49	83.50	65.55

Table 8: OpenVLA layer-band ablation values for the main paper. Values are MAE-D with Top-1 AUROC percentages. The all-layer setting is the default configuration because it is strongest overall and avoids task-specific layer tuning.

Increasing m admits more heads, but the additional heads are not guaranteed to be action-relevant; in practice they often dilute the oriented entropy signal. The half-head setting remains competitive in some subsets, yet it is less consistent and costs more aggregation, so we use Top-1 in the main protocol.

Layer bands. Table 8 reports the exact values for the layer-band ablation in the main paper. Restricting the score to shallow, middle, or deep layers can preserve parts of the signal, especially on subsets where object grounding is already strongly localized. However, no restricted band dominates across task families. The all-layer score gives the best overall AUROC and avoids tuning a layer range per architecture or per task, which is important for a self-evaluation method intended to transfer across VLA backbones.

E Evaluation Protocol

All self-evaluation methods are compared under the same episode-level protocol. A policy first executes a LIBERO-Reflect episode, and the simulator success flag defines the binary label. The self-evaluation method then assigns a scalar reliability score to that episode without using the label. We evaluate whether the score ranks successful executions above failed executions using AUROC,

Policy	Action-generation family	Attention depth	Main score	Head-selection settings	Evaluation role
OpenVLA	Latent-Readout VLAs	32 layers / 32 heads	MAE-D Top-1	Top-1, Top-2, Top-4, Top-16	Autoregressive discrete readout; supports token-statistic baselines
OpenVLA-OFT	Latent-Readout VLAs	32 layers / 32 heads	MAE-D Top-1	Top-1, Top-2, Top-4, Top-16	Same VLA family with continuous readout
QwenPI-Flow	Latent-Refinement VLAs	36 layers / 40 heads	MAE-C Top-1	Top-1, Top-2, Top-4, Top-20	Cross-family flow-style refinement policy

Table 9: Evaluation settings for MAE across the three VLA policies. The main score uses all layers with Top-1 head selection. Larger Top-m settings are reported only for the head-selection ablation; the half-head setting is Top-16 for 32-head OpenVLA-family policies and Top-20 for the 40-head QwenPI-Flow policy.

Policy	Internal generation step k	Final internal step K used by MAE	K in our implementation
OpenVLA	One autoregressive action-token generation step	The last action-token generation step	Number of generated action tokens
OpenVLA-OFT	One continuous-readout step	The continuous-readout step before the action head produces the action chunk	1
QwenPI-Flow	One flow-style refinement step	The last refinement step before the action trajectory is returned	Number of inference refinement steps

Table 10: Model-specific meaning of the internal generation step k and the final internal step K used by MAE.

AUPR, and FPR@95. For MAE, attention maps are taken from the same policy forward passes that generate the robot actions; no auxiliary model, additional rollout, or external verifier is required. Table 9 summarizes the model-specific attention dimensions and the score orientation used for each policy.

The transformer layer index is denoted by ℓ, while k denotes the internal action-generation step. Table 10 summarizes the model-specific correspondence between k and the final internal step K used by MAE.

Layer-band analysis. The main protocol aggregates visual attention entropy over all layers. For the layer-band ablation, we additionally divide the network depth into shallow, middle, and deep regions and recompute the same MAE-D or MAE-C score within each region. This isolates whether the reliability signal is concentrated at a particular depth or benefits from integrating the full action-generation process. The corresponding exact values are reported in Appendix D.

Random baseline. The random baseline assigns an independent uniform score to every episode under the same labels over successful and failed episodes. It is used only as a ranking lower bound and is not tuned per subset or per model.

F Baseline Protocols and Applicability

This appendix defines the baselines used in the main results table and clarifies their applicability to heterogeneous VLA action heads. We only discuss baselines that are part of the reported

Verbal Confidence Prompt

System context: You are given a robot manipulation episode.

Image input: The attached image is a stitched contact sheet of observation frames sampled from the episode. Frames are ordered chronologically from left to right and top to bottom. Each frame is labeled with its timestep.

Task instruction:
{TASK_INSTRUCTION}

Evaluation request: Based on the task instruction and the attached contact-sheet image, estimate whether the robot successfully completed the task.

Output format: Return only one number between 0 and 1: 0 means definitely failed, 0.5 means uncertain, and 1 means definitely succeeded. Return only the number and nothing else.

Figure 9: Prompt used for the Verbal Confidence baseline. The stitched contact sheet is provided as an image input in the same multimodal API request.

protocol. The main compatibility issue is whether a policy exposes autoregressive discrete action-token logits or sampled action-token sequences. Token-statistic baselines are meaningful for discrete `OpenVLA`, but they are not defined for `OpenVLA-OFT`'s continuous action head or `QwenPI-Flow`'s flow-matching action head without adding a separate likelihood model over executed actions. Token-level Self-Consistency has the same interface requirement because it measures sampled action-token agreement. The main results table omits technically not-applicable rows for continuous-action and flow-action policies, while this appendix documents the interface mismatch behind those omissions.

Random. Random assigns an i.i.d. uniform score to each episode and serves as a lower-bound ranking baseline under the same labels over successful and failed episodes.

Verbal Confidence. Verbal Confidence adapts p(True)-style verbal self-checking to VLA evaluation. Since most VLA policies do not naturally output a calibrated verbal probability during action generation, we use an external multimodal evaluator as a proxy. For each episode, we query gpt-4.1 with the task instruction and a stitched contact-sheet image of sampled observations. The model returns a scalar confidence in $[0, 1]$, which is used directly as the episode-level reliability score. The evaluator is not given the ground-truth success label, simulator success state, object poses, perturbation metadata, robot state trajectories, generated action vectors, or oracle information.

Self-Consistency. Self-Consistency is implemented as token-level sampled action-token agreement for discrete `OpenVLA`. At each execution step, we sample n action-token sequences from the same observation and task condition. For each action-token dimension, the score is the frequency of the most common sampled token divided by n; the step score averages this value over action-token dimensions, and the episode score averages over execution steps. The first sampled action is executed, while the additional samples are used only to compute the self-evaluation score.

This baseline requires discrete sampled action tokens. `OpenVLA-OFT` predicts executed actions through a continuous action head, and `QwenPI-Flow` produces continuous action trajectories through flow-matching refinement. Their action heads do not expose sampled categorical action-token sequences for the executed action, so token-level Self-Consistency is not defined for these two policies in the reported protocol.

Maximum Softmax Probability. Maximum Softmax Probability measures the sharpness of autoregressive action-token predictions. For a generated action-token sequence of length T, with categorical distribution y_t over the vocabulary at generation step t, the episode score is

$$\frac{1}{T}\sum_{t=1}^{T}\max_{i} y_{t,i}.$$

This requires a vocabulary-level distribution for each generated action token, so it applies to discrete OpenVLA but not to continuous OFT or flow-matching PI action heads.

Perplexity. Perplexity-style confidence uses the negative log confidence of the selected action-token distribution:

$$\frac{1}{T}\sum_{t=1}^{T} -\log\left(\max_i y_{t,i}\right).$$

We invert the direction when necessary so that larger reported scores indicate higher estimated reliability. As with Maximum Softmax Probability, this score is only defined when action generation exposes token-level categorical probabilities.

Entropy. Entropy measures the uncertainty of the full output distribution:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i} -y_{t,i}\log y_{t,i}.$$

Lower entropy indicates a sharper token distribution, so the reliability score uses the sign convention that larger is better. This is a token-distribution baseline and is not comparable for continuous action regression or flow integration without an additional probabilistic action model.

Length-normalized Entropy. Length-normalized entropy generates n candidate outputs $\mathcal{Y} = \{Y_1, \ldots, Y_n\}$, then averages token entropy across the sampled outputs:

$$\frac{1}{n}\sum_{Y\in\mathcal{Y}}\frac{1}{T_Y}\sum_{t=1}^{T_Y}\sum_{i} -y_{t,i}\log y_{t,i}.$$

We set $n = 5$.

OpenVLA output-statistic values. We report reliability-oriented scores: Perplexity is represented by inverse PPL, Entropy by inverse token entropy, and Length-normalized Entropy by inverse length-normalized entropy, so larger values always indicate higher estimated reliability. These baselines can be computed for `OpenVLA` because it generates discrete action tokens with vocabulary-level probabilities; applying them to `OpenVLA-OFT` or `QwenPI-Flow` would require adding a separate likelihood model over continuous executed actions.

Applicability to heterogeneous VLAs. Discrete OpenVLA exposes autoregressive action-token distributions, so token-probability baselines such as Maximum Softmax Probability, Perplexity, Entropy, and Length-normalized Entropy are conceptually defined for that model family. OpenVLA-OFT changes the action interface: it keeps the OpenVLA backbone but predicts continuous actions through a regression head, so the placeholder action slots are not generated action tokens and their logits do not define the executed action. QwenPI-Flow uses a flow-matching action head that maps visual-language hidden states and noise through iterative velocity prediction; its output is a continuous action trajectory rather than a categorical token sequence. Applying token-probability baselines to these models would evaluate a different random variable from the executed action. By contrast, `MAE` reads internal attention entropy during the same policy forward process and then uses architecture-aware aggregation, `MAE-D` for Latent-Readout VLAs and `MAE-C` for Latent-Refinement VLAs, which is why it remains comparable across the heterogeneous action heads in the main results table.

G Model Input Templates

Table 11 reports the policy input formats used to condition the three evaluated VLA backbones. The OpenVLA-family policies use the same action-query prompt form with lower-cased LIBERO instructions, while `OpenVLA-OFT` additionally uses proprioceptive State Input for continuous readout. `QwenPI-Flow` uses a Qwen3-VL-style multimodal message and appends the grounding text expected by the StarVLA policy. We place these templates after the experimental tables because they are protocol details rather than additional results.

Policy	Input convention	Episode payload	Rendered prompt form
OpenVLA	Pure action-prompt format used by the OpenVLA policy.	One RGB observation image and the lower-cased LIBERO task instruction.	`In: What action should the robot take to {instruction.lower()}? Out:`
OpenVLA-OFT	OpenVLA-family action prompt with the OFT continuous readout.	Primary image, wrist image, proprioceptive State Input, and the lower-cased task label.	`In: What action should the robot take to {task_label.lower()}? Out:`
QwenPI-Flow	Qwen3-VL multimodal message format followed by the StarVLA grounding request.	One or more image placeholders followed by the LIBERO instruction and object-localization request.	`<|im_start|>user <|vision_start|><|image_pad|><|vision_end|> Your task is {instruction}. To identify the key objects for your task. Locate their bounding boxes in [x1,y1,x2,y2] format. <|im_end|> <|im_start|>assistant`

Table 11: Model input templates used for policy conditioning. The table records the prompt forms and non-text policy inputs needed to reproduce the action-generation inputs; benchmark construction and experimental results are reported in Appendices C–D.

Category	Method	Reflect-Goal			Reflect-Object			Reflect-Spatial			Reflect-10			Overall (N = 1,116)		
		AUROC↑	AUPR↑	FPR@95↓	AUROC↑	AUPR↑	FPR@95↓	AUROC↑	AUPR↑	FPR@95↓	AUROC↑	AUPR↑	FPR@95↓	AUROC↑	AUPR↑	FPR@95↓
Black-box	Random	49.82	50.11	95.12	50.15	49.85	95.30	49.90	50.05	94.90	50.20	49.90	94.80	50.02	49.98	95.02
	Verbal Conf.	50.45	50.62	93.20	50.10	50.30	93.80	51.10	51.40	92.10	52.80	52.40	92.50	51.11	51.18	92.90
	Self-Cons.	56.12	55.80	85.30	57.80	57.20	83.10	57.30	56.90	88.20	53.20	53.00	91.20	56.11	55.72	86.95
White-box	MaxProb	54.30	54.10	89.10	55.20	54.90	87.30	55.10	54.70	90.40	52.90	52.60	92.00	54.38	54.08	89.70
	Perplexity	54.80	54.55	88.40	55.60	55.20	86.80	55.50	55.10	89.80	53.05	52.80	91.60	54.74	54.41	89.15
	Entropy	55.70	55.40	87.20	56.80	56.40	85.30	56.70	56.30	88.70	53.40	53.10	90.90	55.65	55.30	88.04
	Length Ent.	56.05	55.80	86.80	57.40	57.00	84.70	57.10	56.80	88.20	53.60	53.30	90.50	56.04	55.72	87.55
Ours	MAE-D Top-16	58.90	58.20	72.80	78.40	76.50	50.20	63.20	62.70	85.40	52.80	52.50	88.10	63.32	62.48	74.12
	MAE-D Top-1	**63.15**	**62.40**	**63.20**	**88.75**	**86.30**	**30.15**	**66.15**	**65.80**	**76.40**	**54.20**	**53.80**	**87.10**	**68.06**	**67.08**	**64.21**

Table 12: Standard-only balanced evaluation for OpenVLA. Every positive and negative episode comes from the same standard LIBERO environment, and each subset is balanced 1:1 by matching every natural failure with a randomly sampled success from that subset. Values are percentages. Overall reports global ranking over the pooled 558 successes and 558 failures.

H Within-Domain Robustness Evaluation

The benchmark combines standard and challenging episodes to evaluate reliability across a broad range of execution conditions. All metrics use the actual simulator success flag for each rollout, rather than the nominal episode source, so naturally failed standard episodes and occasionally successful challenging episodes remain part of the evaluation. This label definition directly measures execution outcome while preserving the intended mixed-difficulty coverage.

The construction statistics characterize the complementary difficulty regimes. Table 4 shows high standard-side success, with overall rates of 95.8% for QwenPI-Flow, 96.3% for OpenVLA-OFT, and 72.1% for OpenVLA; Table 5 reports the challenging-side diagnostic success rates. To verify that the reliability signal is not tied to the mixed-difficulty composition, we additionally evaluate a balanced standard-only protocol below.

This standard-only evaluation provides a controlled within-domain test: every positive and negative episode is drawn from the same standard LIBERO distribution, while the 1:1 class balance fixes

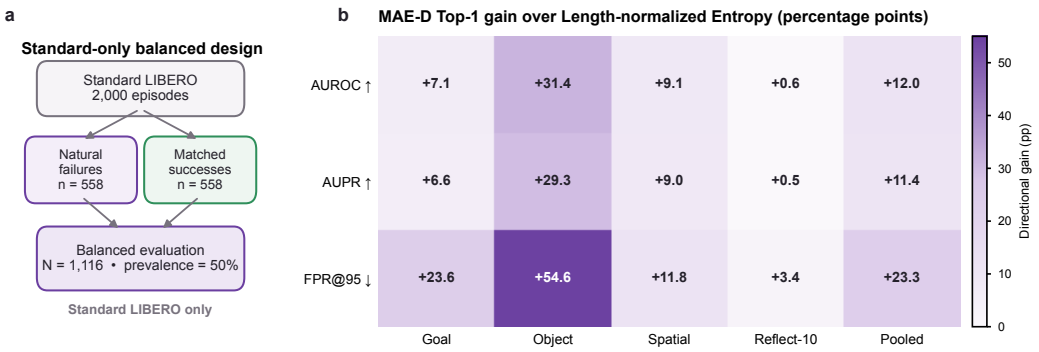

Figure 10: Standard-only balanced evaluation details. **a**, All 558 natural failures from 2,000 standard LIBERO episodes are paired with 558 subset-matched successes, yielding 1,116 standard-domain episodes at 50% success prevalence. **b**, Direction-aware percentage-point gains of `MAE-D Top-1` over Length-normalized Entropy across each subset and the pooled set; positive values denote higher `AUROC`/`AUPR` or lower `FPR@95`.

the positive prevalence for `AUPR`. Consistent performance under this protocol demonstrates that the method ranks successful and failed executions within a shared environment distribution.

H.1 Standard-Only Balanced Evaluation

We construct a complementary evaluation entirely within the standard LIBERO environment. From the 2,000 standard `OpenVLA` episodes reported in Table 4, we retain every natural policy failure: 104 from `Reflect-Goal`, 101 from `Reflect-Object`, 224 from `Reflect-Spatial`, and 129 from `Reflect-10`, for 558 failures in total. We then sample an equal number of successful episodes within each subset, without introducing placement swaps or any LIBERO-PRO episode. The resulting set contains 558 successes and 558 failures ($N = 1{,}116$), fixing success prevalence at 50%. Under this balanced protocol, the random expectations are 50% `AUROC`, 50% `AUPR`, and 95% `FPR@95`. Figure 10 summarizes the sampling construction and the directional gains over the strongest white-box baseline.

Table 12 reports the full baseline comparison. `MAE-D Top-1` is strongest on every subset and pooled metric. Relative to Length-normalized Entropy, the strongest white-box baseline, it improves pooled `AUROC` from 56.04 to 68.06 and pooled `AUPR` from 55.72 to 67.08, while reducing pooled `FPR@95` from 87.55 to 64.21. The largest subset-level separation occurs on `Reflect-Object`, where `MAE-D Top-1` reaches 88.75 `AUROC`, 86.30 `AUPR`, and 30.15 `FPR@95`. These results establish reliable within-domain ranking among executions drawn from the same standard environment and balanced class prior.

I Architecture-Determined Entropy Orientation

The opposite entropy orientations used by `MAE-D` and `MAE-C` are determined by the action-generation interface of the evaluated policy before any episode-level reliability metric is computed. The distinction follows from the role played by the visual-attention distribution in the transition kernel of the conditional generative Markov chain.

For Latent-Readout VLAs, the internal generation process first builds a latent action representation and then maps the final latent state to an executable action through a readout head. In this family, the transition kernel must progressively consolidate task-relevant visual evidence into the latent state before the final readout. A reliable transition therefore tends to concentrate action-query attention on the relevant object, region, or spatial relation needed for the action. Conversely, diffuse visual addressing indicates that the latent action state has not localized the necessary visual evidence, which often corresponds to ambiguous grounding or incorrect object-action binding. For this reason, lower visual entropy is assigned higher reliability, yielding the decreasing-entropy orientation `MAE-D`.

Component	Detail
GPU Hardware	NVIDIA H100
Model Evaluated	QwenPI-Flow
Avg. Rollout Time	∼14.0 s / episode
Avg. MAE Extra Time	**0.57 s / episode**
MAE Latency Overhead	**4.09%** ($< 0.1\times$)
Memory Overhead	Negligible (Reuses internal attention)

Table 13: Empirical cost analysis of MAE. The overhead strictly satisfies the $< 0.1\times$ boundary highlighted in the Pareto-optimal zone of the main paper.

For Latent-Refinement VLAs, the policy maintains an explicit action or trajectory variable and repeatedly refines it under visual-language conditioning, as in flow- or denoising-style generation. Here the transition kernel corrects an evolving action trajectory by re-querying the condition across refinement steps. MAE uses the attention at the final internal generation step $k = K$, before the action trajectory is returned. Reliable executions retain broader visual addressing at this final refinement step, especially when the correct action depends on object relations, spatial constraints, or multi-step manipulation context. Overly low visual entropy indicates that the final action trajectory is determined with too narrow a visual context. Thus, in this family, higher visual entropy at the final refinement step is assigned higher reliability, yielding the increasing-entropy orientation MAE-C.

This orientation rule is architecture-level rather than data-fitted. A model is assigned to Latent-Readout VLAs if its action is produced by reading out a final latent state without iterative refinement of an explicit action variable. It is assigned to Latent-Refinement VLAs if it maintains an action or trajectory representation that is updated across multiple refinement or denoising steps under repeated conditioning. This decision can be made from the model's inference computation graph alone and does not require labels over successful and failed episodes, external supervision, or validation-set optimization. Accordingly, MAE-D is fixed for Latent-Readout VLAs and MAE-C is fixed for Latent-Refinement VLAs before evaluating AUROC, AUPR, or FPR@95.

The same principle also separates orientation from head and layer aggregation. The Top-1 operation used in the main protocol is a deterministic per-episode reduction over oriented entropy values, not a fixed attention head selected by test labels. Likewise, all-layer aggregation is used as the default rule to avoid selecting a task-specific or model-specific layer band. The Top-m and layer-band experiments are therefore sensitivity analyses of a frozen scoring rule rather than procedures for choosing the reported orientation.

J Cost Analysis Details

To substantiate the efficiency claims in the main experiments, we profile the wall-clock latency of MAE during evaluation. Table 13 reports the average execution time per episode and the extra latency incurred by MAE. The overhead is strictly bounded because MAE reuses the internal attention matrices from the same forward pass and only performs entropy computation and head aggregation.