Libra-VLA: Achieving Learning Equilibrium via Asynchronous Coarse-to-Fine Dual-System

Yifei Wei[1,2], Linqing Zhong[1,2], Yi Liu[2], Yuxiang Lu[2], Xindong He[2], Maoqing Yao[2]*, Guanghui Ren[2]*

[1]Beihang University, [2]AgiBot

Abstract

Vision-Language-Action (VLA) models are a promising paradigm for generalist robotic manipulation by grounding high-level semantic instructions into executable physical actions. However, prevailing approaches typically adopt a *monolithic generation paradigm*, directly mapping visual-linguistic features to high-frequency motor commands in a flat, non-hierarchical fashion. This strategy overlooks the inherent hierarchy of robotic manipulation, where complex actions can be naturally modeled in a Hybrid Action Space, decomposing into *discrete* macro-directional reaching and *continuous* micro-pose alignment, severely widening the semantic-actuation gap and imposing a heavy representational burden on grounding high-level semantics to continuous actions. To address this, we introduce Libra-VLA, a novel Coarse-to-Fine Dual-System VLA architecture. We explicitly decouple the learning complexity into a coarse-to-fine hierarchy to strike a training equilibrium, while simultaneously leveraging this structural modularity to implement an asynchronous execution strategy. The Semantic Planner predicts discrete action tokens capturing macro-directional intent, while the Action Refiner conditions on coarse intent to generate high-frequency continuous actions for precise alignment. Crucially, our empirical analysis reveals that performance follows an inverted-U curve relative to action decomposition granularity, peaking exactly when the learning difficulty is balanced between the two sub-systems. With the asynchronous design, our approach offers a scalable, robust, and responsive solution for open-world manipulation.

1 Introduction

The pursuit of generalist robots capable of performing diverse manipulation tasks in open-world environments remains a central challenge in embodied intelligence (Zhao et al., 2023). Recent advancements in Large Vision-Language Models (VLMs), trained on internet-scale corpora, have endowed systems with unprecedented capabilities in visual understanding and semantic reasoning (Bai et al., 2025). Through extending these pre-trained backbones into the robotic domain, Vision-Language-Action (VLA) models have emerged as a dominant paradigm (Bjorck et al., 2025). Unlike traditional specialist policies limited to narrow tasks, VLAs demonstrate remarkable potential in grounding abstract natural language instructions into physical actions, enabling robots to generalize across novel objects and scenarios (Intelligence et al., 2025). However, bridging the gap between semantic intelligence and physical actuation requires more than a direct alignment. Naturally, physical manipulation is not a singular, atomic event, but a process with inherent structure—typically progressing from broad, semantic-driven approaches to precise, geometry-driven interactions. Despite this physical reality, the prevailing VLA paradigm typically employs a *monolithic generation strategy* that overlooks this hierarchical nature. Specifically, previous approaches often adopt direct mapping from high-level semantic features to low-level motor commands through a unified architecture, as

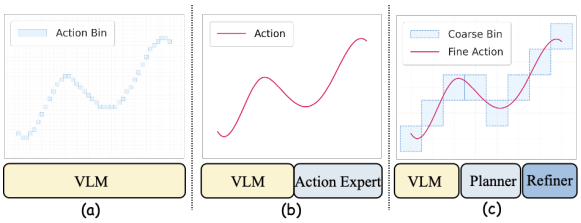

Figure 1: Comparison of action generation paradigms. (a) Discrete autoregressive approaches discretize actions into massive bins. (b) Continuous diffusion approaches directly predict continuous signals. (c) Our proposed Libra-VLA operates in a hybrid action space, where discrete coarse bins representing macro-intents serve as anchors for continuous fine actions, naturally aligning with the inherent hierarchical characteristics.

*Corresponding authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 39799–39815

shown in Fig. 1 (a) and (b).

These methods typically operate by either discretizing continuous actions into numerous action bins to approximate the continuous action space (Kim et al., 2024), or by attaching continuous diffusion heads to the VLM backbone to directly predict continuous signals (Black et al., 2024). Such enforced uniformity creates a massive semantic-actuation gap (Zhang et al., 2025a; Intelligence et al., 2025). The model is compelled to simultaneously grapple with high-level abstract reasoning and low-level high-frequency control within a flat, non-hierarchical process. A natural remedy is to introduce hierarchical control. Existing approaches primarily focus on *temporal decomposition*, guiding task execution through sequential milestones such as sub-instructions or goal images (Li et al., 2025; Shi et al., 2025). However, temporal decoupling only shortens the planning horizon. The model still performs a direct cross-modal translation to continuous actions, leaving the representational complexity of single-step generation unresolved.

In contrast to existing temporal decompositions, we advocate for a paradigm shift: extending the hierarchical philosophy to the representational action space. We propose modeling robotic action within a **Hybrid Action Space** (Fig. 1 (c)), decomposing complex behaviors into discrete macro-directional reaching for semantic grounding and continuous micro-pose alignment for geometric precision. Rather than striving to minimize quantization errors by indiscriminately increasing the number of action bins, we strategically leverage the inherent coarseness of this discrete space to represent abstract *macro-directional intents*. Intuitively, this hybrid decomposition acts as a progressive bridge across the modality gap: the discrete macro-intent resolves the high-level semantic ambiguity of *"where to go"*, providing a stable geometric anchor that constrains the search space. Conditioned on this anchor, the continuous micro-alignment is liberated to focus exclusively on *"how to interact"*, synthesizing the high-frequency residuals necessary for interaction-rich tasks. Ultimately, this formulation allows the discrete subspace to align with high-level linguistic semantics, while the continuous subspace ensures the fidelity required for complex physical execution.

Based on this insight, we introduce **Libra-VLA**, which instantiates the coarse-to-fine philosophy through a decoupled dual-system architecture. Instead of compelling a monolithic network to bridge the extensive semantic-actuation gap, our proposed framework distributes the computational complexity across two specialized modules: Semantic Planner and Action Refiner. Technically, Semantic Planner focuses on the discrete subspace, augmenting a general-purpose VLM with a parallel decoding transformer to predict coarse directional tokens. This design leverages the backbone's spatial reasoning to resolve semantic ambiguity without being burdened by metric precision. Conversely, the Action Refiner handles the continuous subspace. It employs a diffusion transformer equipped with an independent high-resolution visual encoder to capture local geometric details, synthesizing fine-grained residuals conditioned on the planner's intent. Beyond training stability, this structural modularity naturally facilitates an asynchronous execution strategy: Semantic Planner operates at a lower frequency to provide stable guidance, while Action Refiner executes at high frequency, ensuring real-time responsiveness.

To summarize, our main contributions are as follows:

- We propose a novel Coarse-to-Fine VLA paradigm grounded in a hybrid action space and identify the principle of learning complexity equipartition, demonstrating that model performance follows an inverted-U curve and peaks exactly when the learning difficulty is balanced between the two phases.

- Building on this hybrid paradigm, we implement a decoupled asynchronous dual-system architecture. This design allows the Semantic Planner to operate at a low frequency for stable discrete planning, while the Action Refiner executes at a high frequency for real-time continuous control.

- We develop Libra-VLA and demonstrate that it outperforms baselines by achieving higher success rates and lower inference latency.

2 Related works

2.1 Hierarchical Generation

Hierarchical control has a long-standing tradition in robotics. Option learning (Stolle and Precup, 2002) discovers temporally extended sub-policies by identifying useful subgoals, while hierarchical

reinforcement learning (Jiang et al., 2019) leverages language as the abstraction to decompose long-horizon tasks. Recent hierarchical VLA models inherit this temporal decomposition philosophy: HAMSTER (Li et al., 2025) and MOKA (Fang et al., 2024) predict keypoints or waypoints as intermediate sub-goals, while ViLA (Hu et al., 2023) and Hi Robot (Shi et al., 2025) generate step-by-step language sub-instructions to guide low-level policies. While effective for long-horizon planning, these approaches universally operate along the temporal axis. A common limitation is that the inter-level communication resides in a different modality from the final motor commands, forcing the low-level policy to perform a cross-modal translation that introduces a severe modality gap.

Our work circumvents this bottleneck by introducing the coarse action as an intra-modal intermediate state that progressively bridges the modality gap through two simplified mappings. First, mapping VLM features to coarse actions with a small vocabulary size is formulated as a low-cardinality discrete classification task, which substantially reduces the alignment difficulty. Second, mapping coarse actions to fine actions operates entirely within the same physical modality, where the coarse intent serves as a geometric anchor that drastically narrows the search space for final continuous action generation. We note that HybridVLA (Liu et al., 2025) also models actions in a hybrid space, yet both of its branches independently predict fine-grained actions and are fused via arithmetic averaging, constituting a flat parallel architecture without hierarchical structure.

2.2 Dual-System VLA Architectures

Inspired by cognitive dual-process theory (Kahneman, 2011; Evans, 2008), several recent works (Cui et al., 2025; Bjorck et al., 2025; Chen et al., 2025; Shentu et al., 2024) adopt a Dual-System architecture that decouples manipulation into a slow, deliberate System 2 for high-level reasoning and a fast, intuitive System 1 for low-level execution. However, current implementations exhibit shared structural bottlenecks. Models such as GR00T N1 (Bjorck et al., 2025) rely on static latent embeddings as the inter-system communication bridge. During asynchronous generation, these features lack future temporal context and become increasingly lagging as the environmental states change. In contrast, our planner generates a predictive sequence of coarse actions to cover the upcoming execution horizon. This enables effective asynchronous execution via a predictive intent buffer, providing temporally synchronized guidance for each time step. Meanwhile, architectures like FiS-VLA (Chen et al., 2025) force feature coupling within a single backbone, imposing a heavy representational burden by forcing shared network weights to simultaneously encode high-level semantics and low-level fine-grained features. We instead equip the fast system with an independent visual encoder, achieving structural decoupling that eliminates this feature-squeezing bottleneck. Moreover, existing dual systems such as OpenHelix (Cui et al., 2025) universally employ high-dimensional black-box latent vectors for inter-system communication, rendering the information flow between subsystems opaque and difficult to interpret. Our framework replaces these implicit latents with explicit, directly executable coarse actions that carry clear physical semantics as macro-directional intents, yielding a transparent and interpretable communication protocol between the planner and the refiner.

3 Methodology

3.1 Problem Formulation

We formulate the VLA problem as learning a policy π that maps instructions L and observations $\mathbf{o}_t$ to continuous actions $\mathbf{a}_t \in \mathcal{A}$. Specifically, we decompose the action generation into a discrete coarse-grained intention and a continuous fine-grained action:

$$\mathbf{a}_t = \Phi(\mathbf{a}_t^c, \mathbf{a}_t^f), \tag{1}$$

where:

- $\mathbf{a}_t^c \in \mathcal{V}_{act}$ resides in the discrete semantic subspace, capturing the broad directional intent to align with VLM reasoning.
- $\mathbf{a}_t^f \in \mathbb{R}^d$ resides in the continuous geometric subspace, responsible for precise geometric adjustments.

Based on this decomposition, we factorize the joint policy distribution using the probabilistic chain rule. The probability of generating the final action $\mathbf{a}_t$ is decoupled into two conditional distributions:

$$P(\mathbf{a}_t|\mathbf{o}_t, L) \approx \underbrace{P(\mathbf{a}_t^f|\mathbf{a}_t^c, \mathbf{o}_t)}_{\text{Action Refiner}} \cdot \underbrace{P(\mathbf{a}_t^c|\mathbf{o}_t, L).}_{\text{Semantic Planner}} \tag{2}$$

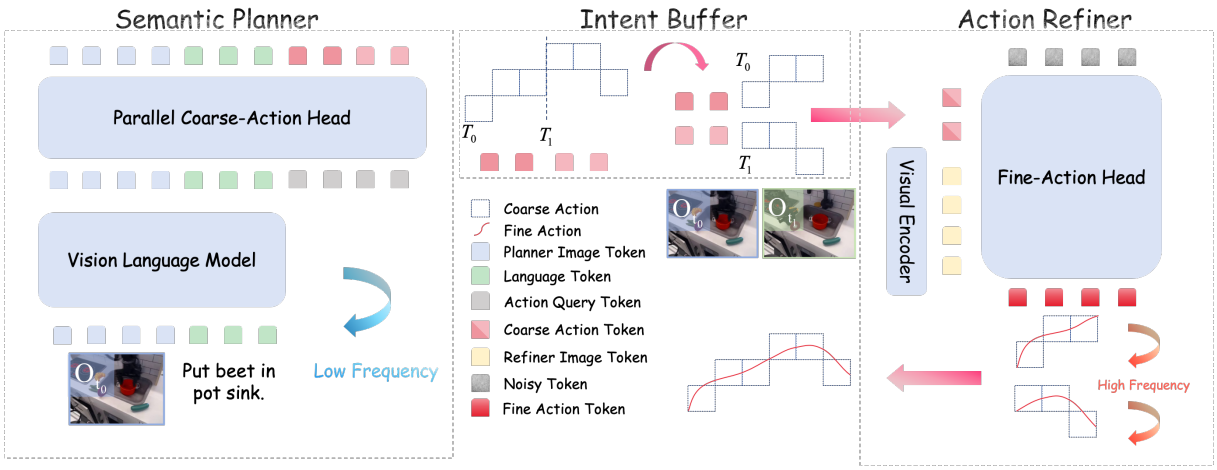

Figure 2: Architectural Overview of Libra-VLA. The framework adopts a Coarse-to-Fine generation paradigm, explicitly modeling actions within a **Hybrid Action Space** across two specialized phases. (Left) System 2: Semantic Planner runs at a low frequency, employing a VLM backbone augmented with a Parallel Coarse-Action Head to predict discrete macro-directional intents (coarse action tokens) within the discrete semantic subspace. (Right) System 1: Action Refiner runs at a high frequency, utilizing a diffusion transformer equipped with an independent visual encoder to synthesize continuous micro-pose alignments (fine action tokens) within the continuous geometric subspace. Crucially, the two systems are bridged via an asynchronous execution strategy managed by an intent buffer. Controlled by the Horizon Expansion Factor (M), the Semantic Planner anticipates an extended coarse action chunk in a single inference pass. Subsequently, the Action Refiner iteratively retrieves the specific coarse action slice corresponding to the current timestep from the buffer as the condition for its high-frequency generation.

This factorization formally underpins our dual-system architecture: System 2 (Semantic Planner) is optimized to maximize the likelihood of the coarse intent $P(\mathbf{a}_t^c|\mathbf{o}_t, L)$ by leveraging the reasoning capabilities of VLMs, while System 1 (Action Refiner) models the conditional distribution $P(\mathbf{a}_t^f|\mathbf{a}_t^c, \mathbf{o}_t)$ via a generative diffusion process to synthesize precise actions. The overall framework of Libra-VLA is shown in Fig. 2.

3.2 System 2: Semantic Planner

Coarse-Grained Directional Discretization. To cast physical actions into a format natively compatible with the VLM's discrete output space, we discretize the normalized continuous actions into N uniform bins. Specifically, let $\mathbf{a}_t \in [-1, 1]^D$ denote the normalized continuous action vector at timestep t, where D is the dimensionality of the action space. The ground truth discrete index for the i-th dimension, $y_{t,i}^{gt} \in \{0, 1, \ldots, N-1\}$, is obtained via uniform quantization:

$$y_{t,i}^{gt} = \text{clip}\left(\left\lfloor \frac{a_{t,i}+1}{2} \times N \right\rfloor, \ 0, \ N-1\right), \quad (3)$$

where $(a_{t,i}+1)/2$ maps the action value from $[-1, 1]$ to $[0, 1]$, the multiplication by N scales it to the bin range, $\lfloor \cdot \rfloor$ assigns the integer bin index, and clip handles boundary conditions. Distinct from previous VLA works (Kim et al., 2024) that typically utilize fine-grained binning ($N = 256$) to approximate continuous control with high precision, we deliberately employ a significantly smaller number of bins ($N \ll 256$). Rather than pursuing precise kinematics, this coarse discretization abstracts actions into macro-directional intents that naturally align with the VLM's semantic reasoning capabilities. Meanwhile, the significantly reduced number of bins narrows the discrete action vocabulary, effectively alleviating the learning burden on Semantic Planner. A detailed quantitative analysis is presented in the ablation studies.

Parallel Coarse-Action Head. The primary role of Semantic Planner is to generate a *coarse-grained directional intent* $\mathbf{a}_t^c$ via a **Parallel Coarse-Action Head** to accelerate inference. We introduce learnable query tokens $\mathbf{Q}_{act} \in \mathbb{R}^{K \times D}$, which are concatenated with the VLM features $\mathbf{H}_t$ and fed into a bidirectional transformer. The output tokens corresponding to the query positions are sliced to obtain the refined features $\mathbf{Z}_{act}$, which are then projected to probability distributions over the N discrete bins:

$$\begin{aligned} \mathbf{Z}_{act} &= \text{Self-Attention}([\mathbf{Q}_{act}; \mathbf{H}_t])_{0:K}, \\ P(\mathbf{a}_t^c) &= \text{Softmax}\left(\text{Linear}\left(\mathbf{Z}_{act}\right)\right). \end{aligned} \quad (4)$$

The training objective of Semantic Planner is to minimize the standard Cross-Entropy loss between

the predicted probability distribution $P(\mathbf{a}_t^c)$ and the ground truth discrete action indices $\mathbf{y}_t^{gt}$ derived from the quantization strategy. Formally, the loss function is defined as:

$$\mathcal{L}_{plan} = \mathcal{L}_{CE}\left(P(\mathbf{a}_t^c), \mathbf{y}_t^{gt}\right). \tag{5}$$

To summarize, Semantic Planner yields a probability distribution over discrete bins representing the macro-directional intent, which subsequently serves as the conditional geometric anchor for the fine-grained action synthesis in the next stage.

3.3 System 1: Action Refiner

While System 2 provides the high-level semantic roadmap, the execution of manipulation tasks requires continuous and precise motor commands. To achieve this, System 1 operates as a conditional diffusion policy that refines the coarse intent into executable precise motions.

Adaptive Intent Injection. The core function of System 1 is to synthesize fine-grained actions conditioned on macro-intent provided by System 2. To instantiate this embedding, we maintain a learnable codebook $\mathbf{E} \in \mathbb{R}^{N \times D}$, where N corresponds to the bin size defined in the quantization strategy. Distinct from standard teacher-forcing methods, we implement a dynamic curriculum strategy for determining the source of $\mathbf{e}_{intent}$ during training, which evolves based on the performance of System 2.

System 1 synthesizes fine-grained actions conditioned on the macro-intent embedding $\mathbf{e}_{intent}$ retrieved from the codebook $\mathbf{E}$. To bridge the training-inference gap, we implement a dynamic curriculum strategy. In early stages (when the prediction success rate of System 2 is below a given threshold τ), $\mathbf{e}_{intent}$ is retrieved using the ground-truth discrete tokens to stabilize training. As the planner improves, we switch to obtaining $\mathbf{e}_{intent}$ by sampling from the predicted distribution $P(\mathbf{a}_t^c)$. This strategy exposes the refiner to planning noise, effectively fostering an inherent error-correction capability against minor directional deviations.

Precise Action Generation. We model the precise action generation as a conditional denoising process. The core architecture of Action Refiner is implemented as a diffusion transformer, fully utilizing its ability to model complex multi-modal distributions. To furnish sufficiently fine-grained visual representations for precise actuation while achieving structural decoupling from Semantic Planner, we augment the Fine-Action Head with an auxiliary visual encoder $\mathcal{E}_{vis}$ to extract geometric features $\mathbf{F}_t^{geo} = \mathcal{E}_{vis}(\mathbf{o}_t)$. Subsequently, the Fine-Action Head conditions on the composite input of the noisy action $\mathbf{x}_k$, the geometric features $\mathbf{F}_t^{geo}$, and the macro-intent $\mathbf{e}_{intent}$ to predict the noise $\epsilon_\theta(\mathbf{x}_k, \mathbf{F}_t^{geo}, \mathbf{e}_{intent})$, thereby iteratively reversing the diffusion process to recover the robot action $\mathbf{a}_t^f$.

The entire System 1 is trained to minimize the standard Mean Squared Error between the predicted noise and the actual noise added during the forward diffusion process. The loss function is defined as:

$$\mathcal{L}_{diff} = \mathbb{E}_{k,\mathbf{x}_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_k, \mathbf{F}_t^{geo}, \mathbf{e}_{intent})\|^2\right]. \tag{6}$$

To jointly optimize the hierarchical architecture, the overall loss function is defined as follows:

$$\mathcal{L}_{total} = \lambda_{diff}\mathcal{L}_{diff} + \lambda_{plan}\mathcal{L}_{plan}, \tag{7}$$

where λ_{diff} and λ_{plan} are calibrated to balance the loss magnitudes, thereby preventing gradient dominance.

3.4 Asynchronous Execution Strategy

To mitigate the computational burden of executing the expensive VLM at every inference cycle, we explicitly decouple high-level reasoning from low-level control via an asynchronous execution strategy managed by a *semantic intent buffer*. This mechanism allows System 2 to plan periodically in bursts while System 1 operates continuously at high frequency.

Planning Horizon Expansion. To enable asynchronous execution, we configure Semantic Planner to predict a longer macro-horizon $L_{macro} = M \times H_{chunk}$ in a single inference pass, where H_{chunk} is the execution horizon of Action Refiner. Here, M serves as the *Horizon Expansion Factor*, representing the ratio of control to planning frequency. This allows Semantic Planner to encapsulate the directional intent for M subsequent execution chunks, reducing the planning frequency.

Intent Buffering and Consumption. We build a First-In-First-Out (FIFO) queue, denoted as the Intent Buffer $\mathcal{Q}$, to bridge Semantic Planner and Action Refiner. The execution workflow proceeds as follows:

Buffer Refill (Low Frequency): At the beginning of a cycle, if the buffer $\mathcal{Q}$ is empty, System

Methods	Action Space	Spatial	Object	Goal	Long	Avg.
CoT-VLA (Zhao et al., 2025)	Discrete	87.5	91.6	87.6	69.0	81.1
WorldVLA (Cen et al., 2025)	Discrete	87.6	96.2	83.4	60.0	81.8
DD-VLA (Liang et al., 2025)	Discrete	97.2	98.6	97.4	92.0	96.3
OpenVLA (Kim et al., 2024)	Discrete	84.7	88.4	79.2	53.7	76.5
π_0-FAST (Pertsch et al., 2025)	Discrete	96.4	96.8	88.6	60.2	85.5
Diffusion Policy (Chi et al., 2025)	Continuous	78.3	92.5	68.3	50.5	72.4
Octo (Team et al., 2024)	Continuous	78.9	85.7	84.6	51.1	75.1
DreamVLA (Zhang et al., 2025b)	Continuous	97.5	94.0	89.5	89.5	92.6
F1 (Lv et al., 2025)	Continuous	98.2	97.8	95.4	91.3	95.7
GR00T-N1 (Bjorck et al., 2025)	Continuous	94.4	97.6	93.0	90.6	93.9
GO-1 (Bu et al., 2025a)	Continuous	96.2	97.8	96.0	89.2	94.8
GE-Act (Liao et al., 2025)	Continuous	98.2	97.6	95.8	**94.4**	96.5
π_0 (Black et al., 2024)	Continuous	96.8	98.8	95.8	85.2	94.1
$\pi_{0.5}$ (Intelligence et al., 2025)	Continuous	**98.8**	98.2	**98.0**	92.4	96.9
Ours	**Hybrid**	98.6	**99.4**	**98.0**	92.8	**97.2**

Table 1: Comparison on the LIBERO benchmark. The best results are highlighted in **bold**, and the second-best results are underlined.

2 performs a forward pass given the current observation $\mathbf{o}_t$ and instruction L. It generates L_{macro} coarse tokens, which are immediately pushed into $\mathcal{Q}$. Crucially, during the subsequent remaining $M-1$ control steps, System 2 remains dormant, bypassing the time-consuming VLM inference.

Conditional Consumption (High Frequency): System 1 operates at the robot's control frequency. At each step k, instead of querying the VLM, System 1 retrieves the corresponding slice of coarse tokens from the buffer:

$$\mathbf{a}^c_{slice} = \mathcal{Q}.\text{pop}(H_{chunk}). \tag{8}$$

This retrieved slice $\mathbf{a}^c_{slice}$ serves as the condition $\mathbf{e}_{intent}$ for precise action generation. System 1 then denoises the fine-grained actions for current timestep.

4 Experiments

In this section, we present a comprehensive empirical evaluation of our proposed Libra-VLA. To rigorously evaluate the model's performance in terms of both precise manipulation and robustness, we utilize two simulation benchmarks: LIBERO (Liu et al., 2023) for assessing standard capabilities, and LIBERO-Plus (Fei et al., 2025) for conducting an in-depth analysis. Furthermore, to demonstrate the effectiveness of Libra-VLA in physical world, we also conducted a series of real-world experiments. More details about these simulation benchmarks and real-world tasks are introduced in Appendix A.

4.1 Experimental Setup

Model Implementation. We build Libra-VLA based on GO-1 (Bu et al., 2025a) and initialize the VLM backbone with InternVL2.5-2B. Structurally, both the Parallel Coarse-Action Head and the Fine-Action Head are implemented as transformer blocks comprising $N = 12$ attention layers, the hidden state dimension of which is set to 1024, corresponding to half of the VLM backbone's hidden size. We employ SigLIP (Zhai et al., 2023) as the visual encoder within the Action Refiner. Unless explicitly stated otherwise, all results on simulation benchmarks are obtained with a Horizon Expansion Factor of $M = 2$ and an action chunk size of $H_{chunk} = 5$, resulting in a macro-horizon of $L_{macro} = 10$. More details about model implementation and training configuration are introduced in the Appendix C and Appendix D respectively.

4.2 Simulation Experiments

LIBERO Benchmark. LIBERO serves as the primary test for evaluating the capabilities of generalist robot policies. It provides a procedural generation pipeline comprising multiple diverse manipulation tasks, categorized into four distinct task suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. The benchmark is designed to provide a comprehensive, multifaceted evaluation of the model's capabilities, encompassing spatial understanding, object manipulation, instruction following, and long-horizon task execution. Following standard evaluation protocols in prior works, each task is evaluated 50 times independently, 500 rollouts total for the task suites. We report the success rates on the four task suites, as well as the average success rate across all four tasks.

As shown in Table 1, Libra-VLA establishes a new state-of-the-art with an average success rate of 97.2%, significantly outperforming baselines. Specifically, the framework achieves dominant performance on precision-critical tasks (99.4% on Object), validating the fine-grained geometric control of our Action Refiner. It also excels in complex long-horizon tasks (92.8% on Long), underscoring the robust macro-directional guidance provided by the Semantic Planner.

LIBERO-Plus Benchmark. LIBERO-Plus introduces controlled perturbations across seven distinct dimensions, including variations in camera viewpoints, lighting conditions, background textures, object layouts, and robot initial states. We evaluate under both *Zero-Shot Transfer*, where models trained on LIBERO are directly tested on LIBERO-Plus, and *Supervised Fine-Tuning* on the LIBERO-Plus training set. We report the success rate across each of the seven perturbation dimensions and the overall average.

Methods	Action Space	Camera	Robot	Language	Light	Background	Noise	Layout	Avg.
Zero-Shot Transfer									
WorldVLA (Cen et al., 2025)	Discrete	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
OpenVLA (Kim et al., 2024)	Discrete	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
NORA (Hung et al., 2025)	Discrete	2.2	37.0	65.1	45.7	58.6	12.8	62.1	39.0
UniVLA (Bu et al., 2025b)	Continuous	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
π_0-Fast (Pertsch et al., 2025)	Discrete	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
OpenVLA-OFT (Kim et al., 2025)	Continuous	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
Ours	**Hybrid**	**68.9**	**48.8**	**92.7**	**97.9**	**93.4**	**86.3**	**77.5**	**79.5**
Supervised Fine-Tuning									
π_0^* (Black et al., 2024)	Continuous	79.6	21.1	72.5	84.7	86.2	68.3	69.4	67.4
$\pi_{0.5}^*$ (Intelligence et al., 2025)	Continuous	70.3	41.7	81.1	**97.3**	**94.6**	71.8	**84.9**	75.7
OpenVLA-OFT+ (Fei et al., 2025)	Continuous	92.8	30.3	**85.8**	94.9	93.9	89.3	77.6	79.6
Ours	**Hybrid**	**94.5**	**41.8**	83.2	95.3	94.3	**93.7**	75.3	**82.3**

Table 2: Results on the LIBERO-Plus benchmark under two settings: *Zero-Shot Transfer*, where models trained on LIBERO are tested on LIBERO-Plus without fine-tuning, and *Supervised Fine-Tuning*, where models are trained on the LIBERO-Plus training set. An asterisk (*) denotes results reproduced by us. The best results are highlighted in **bold**, and the second-best results are underlined.

As detailed in Table 2, Libra-VLA achieves state-of-the-art performance under both settings. Under Zero-Shot Transfer, our model attains 79.5% average success rate, demonstrating strong robustness against diverse perturbations without explicit adaptation. Under Supervised Fine-Tuning, Libra-VLA further improves to 82.3%, surpassing baselines. Regarding resilience to visual domain shifts, results validate that the Semantic Planner maintains robust intent guidance, ensuring the Action Refiner generates effective precise actions. Furthermore, for robot state initialization errors, the stable macro-directional guidance provided by Semantic Planner enables the model to dynamically adjust action directions, demonstrating error recovery capabilities.

4.3 Ablation Studies

In this section, we conduct extensive ablation studies on the LIBERO benchmark to provide a comprehensive analysis of Libra-VLA. Specifically, our investigations focus on the following topics:

1. **Architectural Effectiveness:** Dissect the contribution of individual model components.
2. **Intent Granularity:** Analyze the impact of action intent granularity on model performance.
3. **Training Strategy:** Validate the effectiveness of the dynamic curriculum training strategy.
4. **Asynchronous Execution:** Investigate the influence of the *Horizon Expansion Factor* (M).

Model	VE	Refine	Spatial	Object	Goal	Long	Avg.
Libra-Base	✗	✗	95.8	95.4	86.0	76.0	88.3
Libra-VE	✓	✗	94.8	94.6	69.2	89.4	87.0
Libra-Refinement	✗	✓	**98.6**	98.6	96.0	87.0	95.1
Full	✓	✓	**98.6**	**99.4**	**98.0**	**92.8**	**97.2**

Table 3: Ablation study on model components.

Architectural Effectiveness. To validate the effectiveness of individual model components, we evaluate three variants: Libra-Base (a standard monolithic VLA), Libra-VE (the baseline augmented with an auxiliary visual encoder), and Libra-Refinement (adopting the Coarse-to-Fine generation paradigm but without the auxiliary visual encoder). Detailed architectural specifications are provided in Appendix C.

The results in Table 3 reveal three key insights into the architectural rationale of Libra-VLA.

Comparing Libra-VE against the baseline Libra-Base, we observe that simply appending a visual encoder does not guarantee improvement. In fact, the average success rate declines to 87.0%. Notably, performance on the Goal task plummets from 86.0% to 69.2%. This suggests that without structural constraints, the incorporation of dense visual features induces the model to learn visual shortcuts.

In contrast, while maintaining the same trainable parameters as Libra-Base, Libra-Refinement

VE	Refine	Bin (N)	Spatial	Object	Goal	Long	Avg.
✗	✓	2	95.8	97.6	95.4	80.2	92.3
		10	**98.6**	**98.6**	**96.0**	**87.0**	**95.1**
		50	96.0	96.4	78.2	79.4	87.5
		100	94.0	95.0	70.8	75.6	83.9
✓	✓	2	97.0	98.6	38.6	81.8	79.0
		10	**98.6**	**99.4**	**98.0**	**92.8**	**97.2**
		50	96.8	98.4	95.0	89.2	94.9
		100	95.4	96.8	92.8	90.4	93.9

Table 4: Ablation study on coarse bin sizes (N).

achieves a substantial leap in performance to 95.1%. This validates that the core advantage stems from the Coarse-to-Fine generation paradigm, which lowers the overall learning difficulty, enabling the model to synthesize robot actions more effectively.

Moreover, full model surpasses Libra-Refinement, particularly in long-horizon tasks, by resolving the feature coupling bottleneck where the single VLM backbone struggles to balance the competing demands of high-level semantic reasoning and low-level geometric feature extraction. The introduction of an independent Visual Encoder achieves *structural decoupling*: it specifically extracts geometric features for the refinement phase, effectively offloading the VLM. This allows the VLM to focus solely on semantic planning while the encoder handles precise actuation, resulting in a synergistic boost in robustness.

Intent Granularity. A critical hyperparameter in our architecture is the number of quantization bins (N) for the coarse action, which defines the granularity of the intent. We posit that an appropriate N exists. If N is too small, it fails to provide informative guidance, while an excessively large N turns the coarse-level planning task into an intractable fine-grained classification problem, which fundamentally undermines our core motivation of decomposing the learning complexity via a coarse-to-fine hierarchical action generation strategy. To verify the hypothesis, we conduct experiments across two model configurations: the Libra-Refinement variant and our full model.

As shown in Table 4, we observe a consistent inverted-U performance trend in both settings. We provide a visualization of this trend in Fig. 9 in Appendix G. This phenomenon can be fundamentally interpreted through the principle of *complexity decomposition*. The bin size N acts as a lever that shifts the distribution of learning difficulty between the two subsystems.

At extremely low granularity ($N = 2$), the coarse action tokens suffer from insufficient information density. Due to the lack of informative guidance, the system effectively degenerates into a pure diffusion paradigm, forcing the Action Refiner to shoulder the entire burden of trajectory synthesis. The Semantic Planner fails to provide a meaningful geometric anchor, leaving the refiner to solve the complex manipulation task almost single-handedly, resulting in suboptimal performance.

Conversely, employing excessive bin sizes ($N \geq 50$) shifts the paradigm towards high-precision discrete autoregression. This configuration drastically increases the number of discrete categories, disproportionately overwhelming the Semantic Planner with the complexity of precise metric prediction. The resulting degradation in coarse prediction accuracy feeds erroneous directional guidance to the subsequent Action Refiner, triggering a cascading error that ultimately compromises the overall task success rate.

The "Libra point" is thus found at $N = 10$, where the model achieves peak performance. This setting represents a learning equilibrium. The coarse tokens provide sufficiently informative guidance to effectively alleviate the learning load on Action Refiner, yet remain abstract enough to avoid imposing an excessive learning burden on the planner. By effectively distributing the workload, assigning broad intent modeling to the planner and local refinement to the refiner, this configuration maximizes the efficacy of the hierarchical architecture. A detailed analysis of the resulting training dynamics and convergence behavior is further provided in Appendix E.

Training Strategy	Spatial	Object	Goal	Long	Avg.
Pure Teacher Forcing	96.6	99.2	95.8	92.4	96.0
No Teacher Forcing	97.4	99.0	95.0	90.4	95.5
Dynamic Curriculum (Ours)	**98.6**	**99.4**	**98.0**	**92.8**	**97.2**

Table 5: Ablation study on the training strategy.

Training Strategy. As described in Section 3.3, we employ a dynamic curriculum strategy to stabilize training and enhance the robustness of the Action Refiner. To validate its effectiveness, we compare against two baselines: (1) *Pure Teacher Forcing*, which always conditions on ground-truth coarse actions, and (2) *No Teacher Forcing*, which always conditions on predicted coarse actions from Semantic Planner.

As shown in Table 5, our dynamic curriculum strategy achieves the highest average success rate. Pure Teacher Forcing, while providing stable training gradients, suffers from a significant training-

Factor (M)	Spatial	Object	Goal	Long	Avg.	Latency (ms)	Reduction
2	**98.6**	99.4	**98.0**	92.8	**97.2**	122	44.5%
3	97.6	99.2	94.2	**93.8**	96.2	112	49.1%
4	97.4	**99.8**	93.2	**93.8**	96.1	107	51.4%
5	97.8	98.8	92.0	92.4	95.3	104	52.7%

Table 6: Ablation study on the *Horizon Expansion Factor* (M).

inference gap: the Action Refiner becomes over-reliant on perfect coarse inputs and fails to develop error-correction capabilities, leading to degraded performance when conditioned on imperfect predictions during inference. Conversely, No Teacher Forcing yields the lowest performance, particularly on long-horizon tasks, as the early-stage planning noise from the unconverged Semantic Planner severely destabilizes the Action Refiner optimization. Our strategy effectively balances these two extremes: initial ground-truth forcing ensures stable early convergence, while the subsequent transition to predicted anchors exposes the Action Refiner to realistic planning noise, fostering robustness and consistency between training and inference distributions.

Asynchronous Execution. We further investigate the impact of the *Horizon Expansion Factor* (M), which governs the frequency of the asynchronous semantic planning updates. The results are summarized in Table 6. Serving as the baseline, Libra-Base records an inference time of 220 ms on Nvidia RTX 4090. A detailed comparison of the generation paradigms is provided in Appendix C to ensure fair evaluation.

As the expansion factor M increases from 2 to 5, we observe a slight downward trend in the average success rate, declining from 97.2% to 95.3%. However, this performance degradation is not drastic. Even at $M = 5$, the model maintains a high success rate of over 95%. We attribute the robustness against higher expansion factors (M) to the spatial tolerance of our coarse quantization ($N = 10$), which effectively accommodates the accumulated trajectory errors and state drift inherent in prolonged open-loop execution. Increasing M substantially reduces the average inference latency by amortizing the heavy computational cost of the VLM-based planning over more execution steps.

4.4 Real-World Experiments

We conducted several real-world experiments to further evaluate the effectiveness and robustness of our proposed Libra-VLA in unstructured physical environments. We conducted three long-horizon

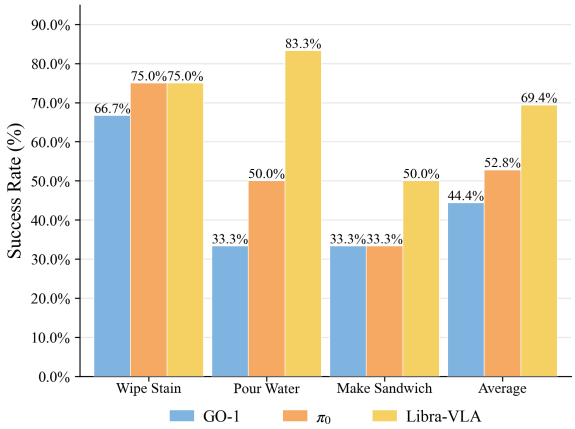

Figure 3: Results of real-world experiments.

tasks, e.g., "Wipe Stain", "Pour Water", and "Make Sandwich", which necessitate sustained temporal coherence and high-precision spatial grounding. Detailed execution protocols for these tasks are as follows. "Wipe Stain" requires the robot to first identify and grasp a sponge randomly positioned on the table, then clean the target stain randomly scattered. "Pour Water" task requires the robot to grasp a kettle positioned on the table, localize the target cup, pour an appropriate amount of water, and subsequently return the kettle to its designated coaster. "Make Sandwich" is a complex, multi-stage manipulation task comprising four distinct sub-tasks. This task entails sequentially stacking ingredients (e.g., bread, meat, lettuce) onto a plate to assemble a complete sandwich. More details about real-world tasks can be found in Appendix A.3.

To comprehensively assess performance, we compare our method against competitive baselines π_0 and Go-1, reporting both the *Single-task Success Rate* and the *Average Success Rate*. The result is shown in Fig. 3.

5 Conclusion

In this paper, we presented Libra-VLA, which pioneers a novel *Coarse-to-Fine generation paradigm* for action generation. Grounded in a Hybrid Action Space, this framework decomposes manipulation into discrete macro-intents and continuous micro-residuals. Through extensive analysis, we identified the "Libra Point", a balanced granularity equilibrium that effectively balances the learning complexity between two generation phases. Empirical results demonstrate that this decoupled architecture not only achieves state-of-the-art success rates on complex benchmarks but also significantly reduces inference latency for real-time control.

Limitations

While Libra-VLA demonstrates strong robustness in asynchronous manipulation, we observe a marginal attenuation in performance as the frequency of asynchronous execution increases. Although this performance decline is slight rather than drastic, it indicates a limitation of the current asynchronous strategy. Our current asynchronous strategy adopts a relatively straightforward protocol where all predicted coarse macro-intents are utilized sequentially. This approach lacks a dynamic verification mechanism to filter out potentially suboptimal anchors during long-horizon execution. To address this, our future work aims to integrate a real-time confidence estimation mechanism. This will allow the system to dynamically assess the reliability of the current macro-intent and trigger a regeneration of the coarse action if the confidence falls below a critical threshold, thereby further enhancing adaptability in complex scenarios.

Ethical Considerations

We acknowledge several ethical considerations and potential risks associated with our research.

Physical Safety. The primary risk in deploying VLA models lies in the potential for unpredictable physical actions, which could lead to hardware damage or safety hazards in unstructured environments. To mitigate this, all real-world experiments in this study were conducted in a controlled laboratory setting under strict human supervision, with immediate emergency stop mechanisms in place. We emphasize that future deployment of such models in open-ended environments requires rigorous safety testing and fail-safe protocols.

Data Privacy. We strictly adhere to ethical data usage standards. While the dataset encompasses real-world scenarios, rigorous filtering protocols were applied to protect privacy. We ensured that personally identifiable information (PII), particularly human faces, has been anonymized or excluded from the training and evaluation sets. No offensive content is present in the data.

Model Bias. As our model builds upon pre-trained Vision-Language Models (VLMs), it may inherit biases present in the large-scale pre-training data. While we focus on manipulation tasks, users should be aware of these potential biases when interpreting the model's high-level reasoning capabilities.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, and 22 others. 2025. GR00T N1: an open foundation model for generalist humanoid robots. *CoRR*, abs/2503.14734.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, and 5 others. 2024. π_0: A vision-language-action flow model for general robot control. *CoRR*, abs/2410.24164.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, Jianlan Luo, Ping Luo, and 31 others. 2025a. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. 2025b. Univla: Learning to act anywhere with task-centric latent actions. *CoRR*, abs/2505.06111.

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

Hao Chen, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Renrui Zhang, Xiaoqi Li, Xiao He, Yandong Guo, Chi-Wing Fu, Shanghang Zhang, and Pheng-Ann Heng. 2025. Fast-in-slow: A dual-system foundation model unifying fast manipulation within slow reasoning. *CoRR*, abs/2506.01953.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. 2025. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704.

Can Cui, Pengxiang Ding, Wenxuan Song, Shuanghao Bai, Xinyang Tong, Zirui Ge, Runze Suo, Wanqi

Zhou, Yang Liu, Bofang Jia, Han Zhao, Siteng Huang, and Donglin Wang. 2025. Openhelix: A short survey, empirical analysis, and open-source dual-system vla model for robotic manipulation. *Preprint*, arXiv:2505.03912.

Jonathan St. B. T. Evans. 2008. Dual-processing accounts of reasoning, judgment, and social cognition. *Annu. Rev. Psychol.*, 59(1):255–278.

Kuan Fang, Fangchen Liu, Pieter Abbeel, and Sergey Levine. 2024. MOKA: open-world robotic manipulation through mark-based visual prompting. In *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. 2025. Libero-plus: In-depth robustness analysis of vision-language-action models. *CoRR*, abs/2510.13626.

Yingdong Hu, Fanqi Lin, Tong Zhang, Li Yi, and Yang Gao. 2023. Look before you leap: Unveiling the power of GPT-4V in robotic vision-language planning. *CoRR*, abs/2311.17842.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. 2025. NORA: A small open-sourced generalist vision language action model for embodied tasks. *CoRR*, abs/2504.19854.

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, and 17 others. 2025. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *CoRR*, abs/2504.16054.

Yiding Jiang, Shixiang Gu, Kevin Murphy, and Chelsea Finn. 2019. Language as an abstraction for hierarchical deep reinforcement learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9414–9426.

Daniel Kahneman. 2011. *Thinking, fast and slow*. Farrar, Straus and Giroux.

Moo Jin Kim, Chelsea Finn, and Percy Liang. 2025. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*.

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

Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memmel, Raymond Yu, Caelan Reed Garrett, Fabio Ramos, Dieter Fox, Anqi Li, Abhishek Gupta, and Ankit Goyal. 2025. HAMSTER: hierarchical action models for open-world robot manipulation. *CoRR*, abs/2502.05485.

Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Liuao Pei, Xiaokang Yang, Jiangmiao Pang, Yao Mu, and Ping Luo. 2025. Discrete diffusion VLA: bringing discrete diffusion to action decoding in vision-language-action policies. *CoRR*, abs/2508.20072.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng Yan, Maoqing Yao, and Guanghui Ren. 2025. Genie envisioner: A unified world foundation platform for robotic manipulation. *CoRR*, abs/2508.05635.

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

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, Chengkai Hou, Mengdi Zhao, KC alex Zhou, Pheng-Ann Heng, and Shanghang Zhang. 2025. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *Preprint*, arXiv:2503.10631.

Qi Lv, Weijie Kong, Hao Li, Jia Zeng, Zherui Qiu, Delin Qu, Haoming Song, Qizhi Chen, Xiang Deng, and Jiangmiao Pang. 2025. F1: A vision-language-action model bridging understanding and generation to actions. *arXiv preprint arXiv:2509.06951*.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. 2025. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*.

Yide Shentu, Philipp Wu, Aravind Rajeswaran, and Pieter Abbeel. 2024. From llms to actions: Latent codes as bridges in hierarchical robot control. *CoRR*, abs/2405.04798.

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

Martin Stolle and Doina Precup. 2002. Learning options in reinforcement learning. In *Abstraction, Reformulation, and Approximation*, pages 212–223, Berlin, Heidelberg. Springer Berlin Heidelberg.

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

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986.

Chuheng Zhang, Rushuai Yang, Xiaoyu Chen, Kaixin Wang, Li Zhao, Yi Chen, and Jiang Bian. 2025a. How do vlas effectively inherit from vlms? *Preprint*, arXiv:2511.06619.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. 2025b. Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge. *CoRR*, abs/2507.04447.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Tsung-Yi Lin, Gordon Wetzstein, Ming-Yu Liu, and Donglai Xiang. 2025. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 1702–1713. Computer Vision Foundation / IEEE.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. 2023. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*.

A Training Data

In this section, we present a comprehensive introduction of the simulation benchmarks and real-world tasks in this work.

A.1 LIBERO Benchmark Suite

LIBERO is a large-scale, procedurally generated benchmark designed to assess knowledge transfer and lifelong learning capabilities in robot manipulation. It provides a diverse set of tasks that require the agent to master both declarative knowledge and procedural knowledge.

The benchmark consists of four distinct task suites, each curating specific distribution shifts to evaluate different facets of the model's generalization ability:

- **LIBERO-Spatial**: This suite contains 10 tasks where the robot must perform the same manipulation primitive but operates under varying spatial layouts. The objects remain consistent, but their relative positions change, requiring the agent to possess robust spatial reasoning capabilities.

- **LIBERO-Object**: Comprising 10 tasks, this suite fixes the spatial layout and task structure but introduces diverse object instances. The agent must generalize its manipulation skills across different visual textures and geometries, testing its object-centric visual grounding.

- **LIBERO-Goal**: This suite includes 10 tasks that share the same workspace and object arrangement but differ in the semantic goals. This evaluates the agent's ability to follow and execute distinct instructions within an identical visual context.

- **LIBERO-Long**: As the most challenging suite, it consists of 10 long-horizon tasks. Each task requires the sequential execution of multiple primitives to complete a complex objective. This suite rigorously tests the model's ability to handle temporal dependencies and multi-stage planning.

For the training data, LIBERO dataset contains 1,693 episodes and 273,465 frames, recorded at a fixed 10 Hz. Our model is trained for 30k steps with global batch size 128. Examples of the LIBERO benchmark are shown in Fig. 4.

A.2 LIBERO-Plus Benchmark Suite

While the standard LIBERO benchmark evaluates the agent's ability to transfer knowledge across varying task semantics and spatial layouts, it operates within a relatively "clean" and idealized visual environment. To rigorously assess the robustness of our proposed model against real-world nuisance variables, we further employ the LIBERO-Plus benchmark.

LIBERO-Plus extends the evaluation protocol by performing a systematic vulnerability analysis across seven distinct perturbation dimensions. These perturbations are designed to mimic distribution shifts and uncertainties inherent in unstructured real-world deployments:

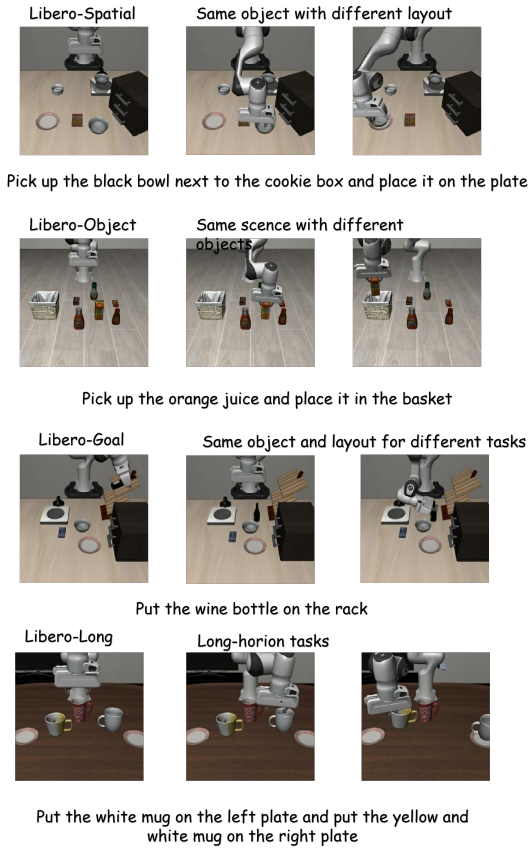

Figure 4: Examples of the LIBERO benchmark.

- **Object Layouts**: This perturbation introduces variations in the initial spatial configuration of task-relevant objects, testing the model's ability to adapt to unseen object arrangements beyond the training distribution.

- **Camera Viewpoints**: This perturbation introduces random jitter and offsets to the camera's extrinsic parameters (position and orientation). This evaluates the model's tolerance to viewpoint shifts and its ability to maintain spatial consistency without overfitting to a fixed camera pose.

- **Robot Initial States**: The initial joint configuration or end-effector pose of the robot is randomized. This challenges the policy to recover from diverse starting conditions and successfully plan trajectories to the target.

- **Language Instructions**: To assess semantic robustness, the task instructions are rephrased using synonyms or different sentence structures while preserving the original semantic meaning (e.g., changing "pick up" to "grasp").

- **Light Conditions**: This perturbation simulates uncontrolled illumination environments by altering the intensity, direction, and color temperature of the scene's lighting, testing the model's invariance to photometric shifts.

- **Background Textures**: The visual appearance of the workspace (e.g., table surface, background walls) is randomized with diverse textures, evaluating the model's ability to perform figure-ground separation and ignore background distractors.

- **Sensor Noise**: Gaussian noise or other forms of signal corruption are injected into the visual observations or proprioceptive states, mimicking high-ISO camera noise or sensor degradation in physical hardware.

LIBERO-Plus dataset provides 14,347 episodes and 2,238,036 frames, captured at 20 Hz. Our model is trained for 50k steps with global batch size 128. Examples of the LIBERO-Plus benchmark are shown in Fig. 5.

A.3 Real-World Tasks

All real-world data collection and evaluation experiments were conducted on AgiBot G1 robot platform. Real-world tasks evaluated in our experiments are shown in Fig. 6.

In the "Wipe Stain" task, the robot is required to first visually localize and grasp a sponge placed on the tabletop. Following the grasp, the agent must identify the specific location of the stain and manipulate the sponge to perform the erasing action. The "Wipe Stain" task consists of 177 episodes totaling $356,316$ frames.

For the "Pour Water" task, the robot is tasked with initially grasping a kettle, maneuvering it to an appropriate position above a target cup, dispensing an appropriate amount of water, and subsequently returning the kettle to its original location. This task comprises $1,821$ episodes with $5,062,506$ frames.

"Make Sandwich" presents a long-horizon assembly challenge, which demands the robot to sequentially retrieve distinct ingredients—specifically bread, meat, and lettuce—and vertically stack them onto a serving plate to assemble a complete sandwich. The "Make Sandwich" dataset contains 452 episodes comprising $1,222,087$ frames.

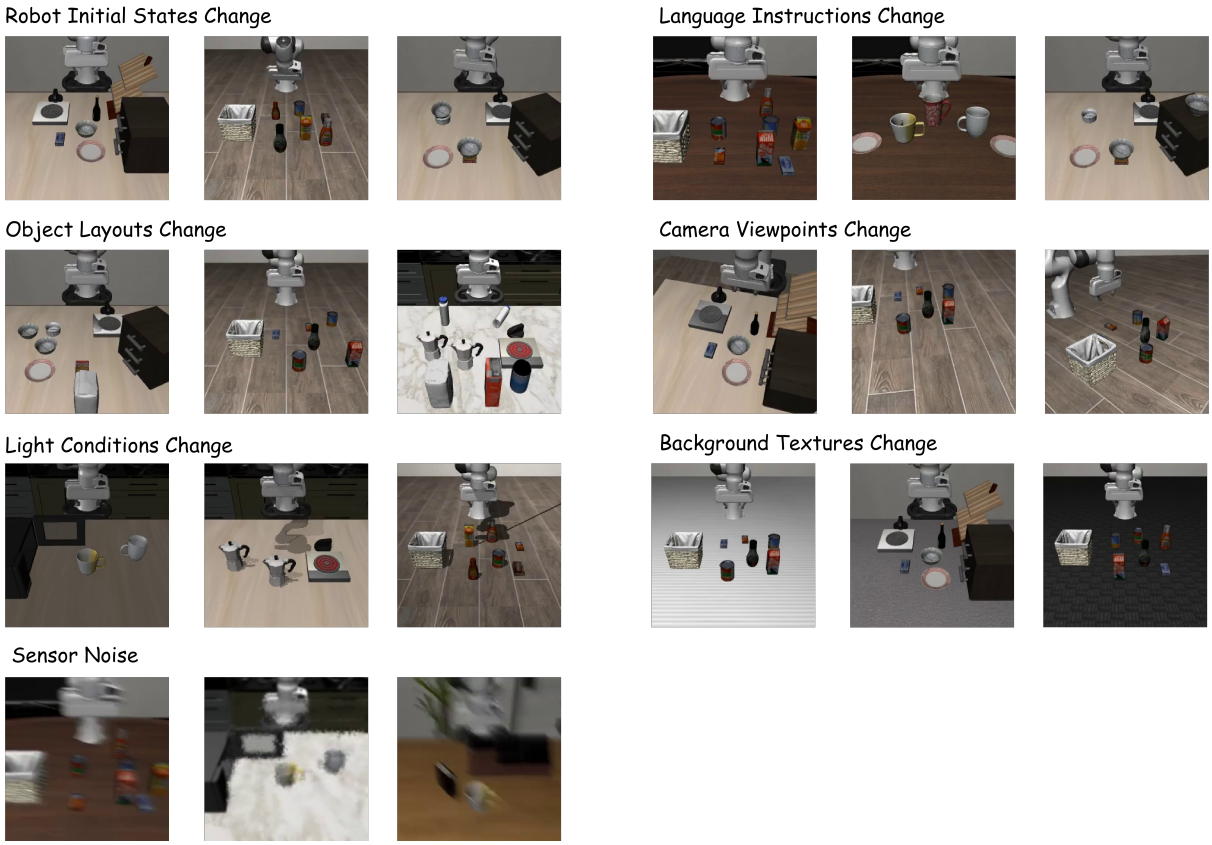

Figure 5: Examples of the LIBERO-Plus benchmark.

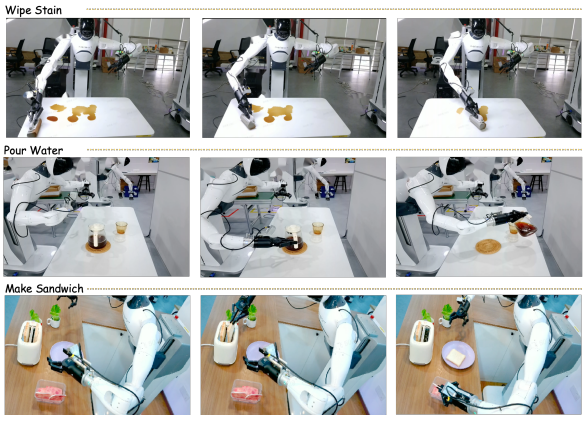

Figure 6: Visualization of real-world tasks.

B Evaluation Metrics

To comprehensively assess the performance of our method, we employ rigorous evaluation protocols. For real-world comparisons in particular, all compared models are evaluated under identical physical setups with pre-marked target positional configurations on the workspace. All rollouts for different models are further conducted consecutively within a short time window by the same human testers, robot, and props, so that lighting and other environmental conditions remain as consistent as possible across models.

B.1 Simulation Benchmarks

For the LIBERO and LIBERO-Plus simulation benchmarks, we strictly adhere to the official evaluation protocols to ensure a fair and consistent comparison with prior works. Specifically, we utilize the official testing scripts and maintain identical parameter settings as defined in the standard benchmark suite. This ensures that all reported results reflect the model's true capability under standardized conditions.

Furthermore, to rigorously evaluate real-time performance of Libra-VLA, we analyze the average inference latency across different Horizon Expansion Factor M. We define the average latency as the total duration of a complete asynchronous execution cycle divided by the number of execution steps within that cycle.

B.2 Real-World Experiments

For real-world evaluations, we define specific experimental setups and success criteria for each task as follows.

Wipe Stain. To test the model's generalization to spatial variations, the sponge is initialized at three distinct initial positions, while the target stain is randomly located at one of four different positions

on the table. A single trial is considered successful if the robot successfully grasps the sponge and completely wipes the target stain.

Pour Water. In this task, we vary the initial position of the kettle across two locations and the target cup across three different locations. A trial is considered successful if the robot securely grasps the kettle and successfully pours an adequate amount of water into the cup.

Make Sandwich. This is a long-horizon task where all ingredients are placed at fixed initial positions. The robot acts as a chef and is required to sequentially grasp and place four ingredients onto a plate to assemble a complete sandwich. The entire task is considered successful only if all four sequential sub-tasks are correctly executed. Given the complexity of this long-horizon process, we allow a maximum of one retry for a specific sub-task if a failure occurs during execution.

C Implementation Details

In this section, we provide a more comprehensive description of the implementation details for our proposed Libra-VLA architecture, alongside the specific configurations of the variant models employed in our comparative analysis.

We instantiate the VLM backbone of Libra-VLA with InternVL2.5-2B. Structurally, both the Parallel Coarse-Action Head and the Fine-Action Head are implemented as transformer blocks comprising $N = 12$ attention layers, the hidden state dimension of which is set to 1024, corresponding to half of the VLM backbone's hidden size. We employ SigLIP as the visual encoder within the Action Refiner. Unless explicitly stated otherwise, all results on simulation benchmarks are obtained with a Horizon Expansion Factor of $M = 2$ and an action chunk size of $H_{chunk} = 5$, resulting in a macro-horizon of $L_{macro} = 10$. For real-world evaluations, we maintain the Horizon Expansion Factor at $M = 2$ while increasing the action chunk size to $H_{chunk} = 20$, resulting in a macro-horizon of $L_{macro} = 40$.

Furthermore, we clarify the structural configurations of the ablation variants. Libra-Base adopts a standard VLA architecture like π_0, employing a monolithic Action Expert comprising 24 stacked attention layers attached to the VLM backbone. For Libra-Refinement, we maintain the fundamental model structure but allocate 12 attention layers each to the Parallel Coarse-Action Head and the Fine-Action Head. Consequently, this architecture essentially decouples the monolithic expert into two specialized modules, ensuring that Libra-Base and Libra-Refinement possess an identical number of trainable parameters. Libra-VE builds upon the Libra-Base baseline by incorporating an additional visual encoder (SigLIP) to provide auxiliary visual inputs, while retaining the monolithic 24-layer Action Expert configuration.

The Libra-Base, Libra-VE, and Libra-Refinement models all adopt a synchronous inference mode, where perceptual processing and action generation occur sequentially within a single control cycle. In contrast, the full Libra-VLA architecture employs an asynchronous execution strategy. The specific structural configurations and parameter statistics for Libra-VLA and its ablation variants are detailed in Table 7.

We further clarify the generation paradigms of Libra-Base and Libra-VLA to facilitate fair inference latency comparison. Libra-Base adopts the same architecture as π_0, utilizing a monolithic diffusion-based action generation process rather than autoregressive decoding. In contrast, the generation process in Libra-VLA is divided into two stages: (1) *Coarse Action Generation*, where the Semantic Planner employs bidirectional parallel decoding to generate all coarse action tokens simultaneously in a single forward pass; and (2) *Fine Action Generation*, where the Action Refiner generates fine-grained actions using diffusion, iteratively refining Gaussian noise into precise robot control commands conditioned on the coarse intents. The faster inference speed of Libra-VLA is primarily attributed to three factors: the asynchronous execution strategy amortizes the expensive VLM forward pass over multiple control steps; the Action Refiner contains fewer attention layers than the monolithic expert, reducing per-step computation; and the bidirectional parallel decoding in the Semantic Planner avoids the sequential bottleneck of autoregressive generation.

D Training Details

In this section, we provide a comprehensive description of the training implementation for the proposed framework. Unless explicitly stated otherwise, the training of all model variants adheres to the standardized protocols and hyperparameter settings described below.

For the simulation benchmarks, we train the

Model	VLM Backbone	Action Expert Layers	Extra VE	Trainable Params	Total Params
Libra-Base	InternVL2.5-2B	$N = 24$	–	2287M	2591M
Libra-VE	InternVL2.5-2B	$N = 24$	SigLIP	2738M	3042M
Libra-Refinement	InternVL2.5-2B	$N = 12$ / $N = 12$	–	2287M	2591M
Libra-VLA (Full)	InternVL2.5-2B	$N = 12$ / $N = 12$	SigLIP	2738M	3042M

Table 7: Detailed architectural specifications and parameter statistics.

model for 30k steps on LIBERO and 50k steps on LIBERO-Plus, employing a consistent global batch size of 128.

For real-world tasks, the training configurations are specified as follows: the "Wipe Stain" task is trained for 40k steps with a global batch size of 96, the "Pour Water" task for 30k steps with a global batch size of 128, and the "Make Sandwich" task for 50k steps with a global batch size of 96.

To ensure training stability and convergence, we utilize a peak learning rate of 2×10^{-5}. The learning rate is dynamically adjusted using a cosine-decay scheduler, accompanied by a linear warmup phase spanning the initial 1,000 steps. To regularize the model and mitigate overfitting, we apply a weight decay of 0.01. For Libra-VLA and its ablation variants, the vision encoder within the VLM remains frozen during training, while all other model components are trainable. For the reproduced π_0 and $\pi_{0.5}$ on LIBERO-Plus, we train the vision encoder and the action expert. Regarding the computational infrastructure, all models are trained on a cluster equipped with 8 NVIDIA H100 GPUs. Furthermore, we employ bfloat16 mixed-precision training to optimize memory efficiency and computational throughput without compromising numerical stability.

E Training Convergence Analysis

To further examine the convergence behavior of our coarse-to-fine design, we compare the intermediate rollout performance of Libra-VLA against the monolithic baseline Libra-Base at 10,000 training steps, i.e., one-third of the total training budget. Both models are trained on the same LIBERO data with the identical batch size and optimizer configuration described in Appendix D. We focus on intermediate rollout success rates to directly reflect task-level behavior.

As shown in Table 8, at 10k steps Libra-VLA reaches an average success rate of 88.4%, exceeding Libra-Base by 16.3 points under identical training conditions. The gap is most pronounced on the Long suite, where the longer-horizon planning is more sensitive to the optimization load of grounding high-level semantics to continuous actions.

Method	Spatial	Object	Goal	Long	Avg.
Libra-Base	81.0	86.8	66.0	54.4	72.1
Libra-VLA (Ours)	**97.2**	**99.2**	**75.0**	**82.2**	**88.4**

Table 8: Success rates (%) at 10,000 training steps on the LIBERO benchmark. Results are averaged over 500 independent rollouts per task suite, following the same evaluation protocol as Appendix F.

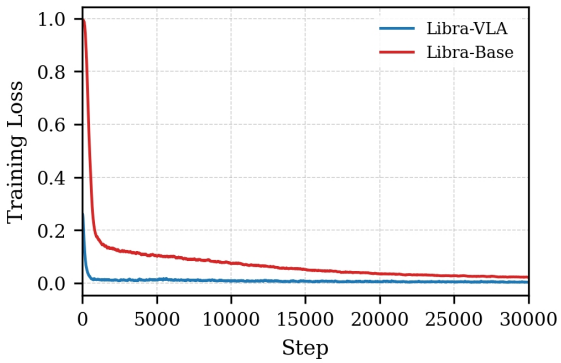

Figure 7: Training loss curves of Libra-VLA and Libra-Base on LIBERO.

We attribute this intermediate gap to the workload decoupling inherent in our design. The Semantic Planner focuses on low-frequency macro-intent prediction, while the Action Refiner performs continuous micro-alignment conditioned on the predicted coarse anchors. These anchors narrow the effective search space of the Action Refiner, which alleviates the optimization burden observed in monolithic generation. This workload decoupling is directly reflected in the training loss curves shown in Figs. 7 and 8, which plot the MSE loss on the continuous action outputs, corresponding to the fine-action generation in Libra-VLA and the monolithic action generation in Libra-Base. The log-scale view more clearly reveals the low-loss regime where the linear-scale view in Fig. 7 saturates visually. At 10k training steps, this continuous-action MSE loss of Libra-VLA decreases to ap-

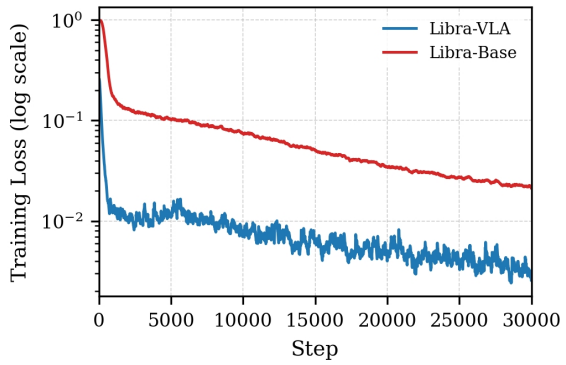

Figure 8: Training loss curves plotted on a logarithmic scale.

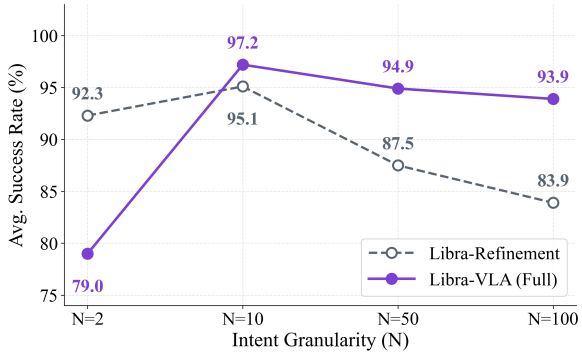

Figure 9: Performance trend with respect to the coarse bin size N.

proximately 0.01, while that of Libra-Base remains noticeably higher at around 0.07, consistent with the workload-balance argument discussed in Section 4.3.

We further note a short-lived rise in the Libra-VLA curve around step 5,000. This corresponds to the transition point of our dynamic curriculum strategy (Section 4.3): once the coarse-action prediction accuracy of the Semantic Planner surpasses a preset threshold, the input to the Action Refiner is switched from ground-truth coarse actions to those predicted by the Semantic Planner, which briefly exposes the refiner to imperfect anchors. The loss soon resumes its downward trend within a limited number of steps, indicating that the Action Refiner gradually learns to compensate for noisy coarse inputs, which is in line with the error-correction behavior discussed in Section 4.3.

F Further Comparison

To empirically validate the advantage of our hierarchical action generation over flat hybrid approaches, we conduct a direct quantitative comparison against HybridVLA on the LIBERO benchmark. We use the official open-source codebase of HybridVLA without any modification to its training or inference logic. To ensure fairness, both models are trained under identical configurations: the same LIBERO training data, an identical global batch size of 128, and a fixed training budget of 30,000 steps. We perform 500 independent rollout evaluations for each task suite.

Method	Spatial	Object	Goal	Long	Avg.
HybridVLA	24.4	48.6	38.4	20.4	32.9
Libra-VLA (Ours)	**98.6**	**99.4**	**98.0**	**92.8**	**97.2**

Table 9: Comparison with HybridVLA on LIBERO.

As shown in Table 9, under aligned training overheads, Libra-VLA comprehensively outperforms HybridVLA. Due to the training burden imposed by its flat architecture and high-precision hybrid generation, HybridVLA exhibits a significantly slower convergence rate, achieving an average success rate of only 32.9% at 30k steps. In contrast, Libra-VLA achieves 97.2% by decomposing the learning complexity via the coarse-to-fine hierarchy, demonstrating substantially faster convergence and higher performance.

G Intent Granularity Ablation Curve

Fig. 9 visualizes the ablation results on coarse bin size N from Section 4.3 as line plots.

H LLM Usage Statement

In this paper, Large Language Models (LLMs) were used exclusively for linguistic polishing and grammatical correction to enhance readability. None of the technical methodology, implementation details, or experimental results were generated by LLM.