AMAP CV Lab | Alibaba Group

ABot-M0: VLA Foundation Model for Robotic Manipulation with Action Manifold Learning

AMAP CV Lab

See Contributions section for a full author list.

Abstract

Building general-purpose embodied agents across diverse hardware remains a central challenge in robotics, often framed as the "one-brain, many-forms" paradigm. Progress is hindered by fragmented data, inconsistent representations, and misaligned training objectives. We present **ABot-M0**, a framework that builds a systematic data curation pipeline while jointly optimizing model architecture and training strategies, enabling end-to-end transformation of heterogeneous raw data into unified, efficient representations. From six public datasets, we clean, standardize, and balance samples to construct **UniACT-dataset**, a large-scale dataset with over 6 million trajectories and 9,500 hours of data, covering diverse robot morphologies and task scenarios. Unified pre-training improves knowledge transfer and generalization across platforms and tasks, supporting general-purpose embodied intelligence. To improve action prediction efficiency and stability, we propose the *Action Manifold Hypothesis*: effective robot actions lie not in the full high-dimensional space but on a low-dimensional, smooth manifold governed by physical laws and task constraints. Based on this, we introduce **Action Manifold Learning (AML)**, which uses a DiT backbone to predict clean, continuous action sequences directly. This shifts learning from denoising to projection onto feasible manifolds, improving decoding speed and policy stability. ABot-M0 supports modular perception via a dual-stream mechanism that integrates VLM semantics with geometric priors and multi-view inputs from plug-and-play 3D modules such as VGGT and Qwen-Image-Edit, enhancing spatial understanding without modifying the backbone and mitigating standard VLM limitations in 3D reasoning. Experiments show components operate independently with additive benefits. We will release all code and pipelines for reproducibility and future research.

Date: February 11, 2026
Code: https://github.com/amap-cvlab/ABot-Manipulation
Project Page: https://amap-cvlab.github.io/ABot-Manipulation

Contents

1 Introduction

The ultimate goal of robotics is to build general embodied agents that operate across diverse robotic bodies, realizing the "one brain, many forms" vision [9, 29, 52]. Such an agent must perceive its environment, understand natural language instructions, reason about actions, and learn from interaction [9, 29, 48, 52]. While Large Language Models (LLMs) have achieved universality in the digital realm, embodied intelligence remains fragmented. Despite decades of progress, this vision remains unrealized. Policies trained on one robot rarely transfer to another, and high-level semantic reasoning struggles to translate into low-level physical precision. Embodied intelligence has yet to reach its transformative moment.

We argue that the field currently faces three fundamental barriers. First, data scale remains below critical mass. Unlike standard vision-language data, embodied data require precise action labels, which are expensive and slow to collect. Most current approaches rely on pre-training sets limited to a single robot type or platform. While effective for narrow deployment, this restricts the development of general policies by limiting exposure to diverse morphologies and task distributions. Second, data quality varies widely and lacks standardization. Action representations, coordinate systems, and control frequencies differ across datasets, making it difficult for models to extract common patterns across embodiments. The resulting fragmentation introduces noise and weakens learning, as models must allocate capacity to memorize idiosyncrasies rather than acquire transferable skills. Third, existing pre-training paradigms are mismatched. Most Vision-Language-Action (VLA) models start from Vision-Language Models (VLMs), whose visual encoders focus on semantic recognition rather than 3D structure or physical dynamics. Yet embodied behavior demands fine spatial understanding and causal reasoning, capabilities that cannot be acquired from 2D supervision alone, nor easily added through mechanisms such as Chain-of-Thought, because such methods operate at the reasoning level without enriching the underlying perceptual representation. Achieving general embodied intelligence therefore requires systematic improvements in data, representation, and training design.

We address a practical question: given the high cost and hardware dependence of robotic data collection, can we integrate global open-source datasets, currently isolated in silos, into a unified foundation for training VLA models? This points to a different direction: embodied intelligence need not emerge from closed, proprietary systems, but can instead develop through the aggregation of heterogeneous data and incremental capability building, with shared infrastructure enabling cumulative progress across research groups.

To this end, we introduce ABot-M0 (**A**map VLA Foundation Model for Ro**bot**ic **M**anipulation), a unified approach that jointly optimizes data processing and model architecture. We first construct the **UniACT-dataset**, combining six major open-source datasets to form a mixed training set of over 6 million trajectories spanning 9500+ hours and 20+ embodiments, which is currently the largest collection within the non-private domain. To resolve inconsistencies in action format, coordinate system, and sampling rate, we define a standardized preprocessing pipeline. All actions are converted to delta actions in the end-effector frame; rotations are encoded using rotation vectors to avoid singularities; and we apply a pad-to-dual strategy to support both single-arm and dual-arm tasks within a shared framework. During training, we conduct uniform sampling across datasets to balance the distribution of tasks and embodiments. This unified data foundation breaks down barriers between datasets and enables robust cross-embodiment generalization by aligning both geometric and temporal structures across sources.

Building on this unified data foundation, we construct the **ABot-M0** model based on a VLM followed by an action expert. We employ a two-stream feature interaction to enhance both semantic and 3D perception for action prediction and propose the Action Manifold Learning (AML) to improve the efficiency and stability of action prediction. First, most VLA models [24, 31, 37, 40] train the policy to predict noise, either ϵ-prediction or v-prediction, which is unstructured, meaningless, and often includes invalid actions such as jittering or discontinuities. These noisy targets spread across high-dimensional space, requiring large model capacity and leading to inefficient learning. Inspired by JiT [25] compilation, we propose the *"Action Manifold Hypothesis"*: successful actions do not scatter randomly but lie on a low-dimensional, smooth manifold shaped by physics, task goals, and environmental constraints. Based on this, we design the Action Manifold Learning (AML) mechanism, where the DiT backbone directly predicts clean action sequences. This shifts the learning objective from fitting noise to projecting onto feasible action manifolds, improving decoding speed and policy stability by reducing output uncertainty. Second, to improve feature interaction between action expert and VLM,

we design a dual-stream feature interaction. The action expert takes the feature of VLM's last layer to acquire visual and semantic reasoning. Additionally, a perception module injects geometry-aware features via optional models, such as VGGT [43] to compute scene-level structural features and Qwen-Image-Edit [45] to fuse features from multi-view observations. The perception module acts as a plug-and-play expert that compensates for the perceptual limitations of standard vision-language models.

We perform full ablation studies under a consistent training setup, showing that data standardization, architectural changes, and training redesign are orthogonal and their benefits accumulate. Our model achieves average success rates of 98.6%, 80.5%, 58.3% and 81.2% on benchmarks including LIBERO [28], LIBERO-Plus [15], RoboCasa GR1 Tabletop Tasks [31] and Robotwin2.0 [10], outperforming strong baselines such as $\pi_{0.5}$ [17], UniVLA [4], and OpenVLA-OFT [20]. These results validate a complete pipeline from data curation to architecture design to capability emergence, demonstrating that high-performance, generalizable embodied intelligence can be achieved through systematic engineering without reliance on proprietary data. The gains from each component prove consistent across platforms, indicating that learned representations transfer beyond dataset-specific biases. We will release our full data processing and training codebase to support open, collaborative progress toward general-purpose robotics.

2 Dataset

To support large-scale pre-training for general embodied intelligence, we conducted a comprehensive analysis of prevalent robotics datasets in Section 2.1. Then we undertake two systematic data-level initiatives. First, in Section 2.2 we perform comprehensive data cleaning and quality governance to address prevalent issues in multi-source datasets, including format inconsistencies, missing instructions, and semantic noise. And in Section 2.3 we establish a unified data representation to enable cross-platform and cross-embodiment alignment of manipulation semantics. These efforts together establish a reproducible, scalable, and high-purity foundation for embodied manipulation data, providing a robust basis for training high-performance Vision-Language-Action (VLA) models.

2.1 Analysis of Open-Source Datasets

With the rapid advancement of embodied intelligence, an increasing number of high-quality open-source datasets have become available. We systematically integrated the most representative Vision-Language-Action (VLA) data resources currently accessible and conducted a comprehensive evaluation of their characteristics to guide the curation and composition of large-scale pretraining datasets. Details are shown in Table 1.

Existing open-source Vision-Language-Action (VLA) datasets exhibit three key characteristics. **Large Scale:** Represented by the Open-X Embodiment (OXE) [32] and OXE-AugE [18] dataset, emphasizing massive data volume and broad coverage of scenes and tasks. These datasets serve as foundational pretraining corpora to foster strong generalization across diverse scenarios and tasks. **High Quality:** Exemplified by Agibot [13] and Galaxea [19], these datasets focus on structured task design, fine-grained annotations, and high physical fidelity. They are well-suited for capability refinement and downstream task adaptation. **Embodiment Diversity:** Represented by RoboCoin [47] and RoboMind [46], these datasets prioritize diversity in robot embodiments (e.g., different morphologies and kinematic structures), constructing data for multiple embodiment types on unified platforms with careful alignment to enhance cross-embodiment generalization.

A single dataset might possess one or two of the above characteristics. For instance, OXE [32] considers both large scale and diversity, Agibot-Beta [13] considers both high quality and large scale, and RoboCoin [47] involves both diversity and high quality. However, it is difficult for one dataset to simultaneously satisfy all three characteristics, and there is still much room for improvement.

To build the dataset for a general-purpose VLA (Vision-Language-Action) model, scale sets the floor, quality defines the ceiling, and diversity determines the scope. Data scale establishes the foundational lower bound of a model's generalization capability. Data quality defines the upper limit of achievable performance. And embodiment diversity determines the boundaries of cross-embodiment generalization. Furthermore, standardized data format such as LeRobot [5] and RLDS [36] plays a crucial role in lowering integration barriers and fostering collaborative dataset development within the research community.

Table 1 Analysis of Open-Source VLA Datasets. We compare scale, embodiment coverage, strengths, and limitations for commonly used dataset. Here Emb. Adopted only reflects the number of embodiments retained for pretraining after rigorous curation.

Dataset	Scale	Emb. Adopted	Strengths	Limitations
OXE [32]	>1.0M	11 Single	Large Scale Diverse scene	Low Complexity Short Trajectory Poor Quality
OXE-AugE [18]	4.4M	9 Single	Inherited from OXE Balanced Embodiment	Synthetic video Visual Artifacts Physical Unreality
AgiBot-Beta [13]	>1.0M	1 Dual	Long-Horizon Rich Atomic Skill High Quality	Single Embodiment
RoboCOIN [47]	180K	8 Single+Dual	Hierarchical label Diverse Dual-Arm	Limited Scale No Single-Arm
RoboMind [46]	107K	7 Single+Dual	Unified Protocol Long-Horizon Task Single & Dual-Arm	Nonstandard Data Format
Galaxea [19]	100K	1 Dual	Long-Horizon Subtask Annotation Rich Sensor Data	Single Embodiment
Others (LET [23], Robo360 [26], MolmoAct [22])	$\leq$ 100K	–	Task-Specific Design	Fixed Embodiment Inconsist Format Adaptation Cost

Our dataset currently satisfies these three principles simultaneously through the integration of multiple open-source datasets, data preprocessing (Section 2.2), and data format and action space alignment (Section 2.3). In the future, VLA datasets should strive to deliver higher standard of large scale, high quality, and rich diversity. In terms of scale, emphasis should be placed not only on raw data volume but also on effective interaction duration. Regarding embodiment diversity, datasets should encompass more types of embodiment, including single-arm, dual-arm, semi-humanoid, and full humanoid configurations, to foster cross-embodiment alignment and generalization. For data quality and richness, beyond providing multiple action representations (including both joint and end-effector trajectories) and fine-grained semantic annotations (covering task decomposition, key frames, and object relationships), datasets should also incorporate sensor and dynamics information (such as velocities, forces, and torques), high-definition synchronized multi-view video streams, and camera intrinsics and extrinsics to facilitate the computation of auxiliary 3D geometric labels. These metadata elements serve not only as sources of supervision signals but also as foundational pillars for achieving precise spatiotemporal alignment across vision, language, and action modalities.

2.2 Data Cleaning and Preprocessing

According to the analysis in Section 2.1, we take six open-source datasets, including OXE, OXE-AugE, Agibot-Beta, RoboCoin, RoboMind, and Galaxea. As shown in Table 1, the overall number of source trajectories is more than seven millions. To effectively leverage the unique characteristics and strengths of each dataset, we design dataset-specific curation strategies tailored to their structural and semantic properties. OXE serves as the foundational large-scale single-arm dataset, providing broad task coverage and baseline data diversity. Building upon this foundation, OXE-AugE is incorporated to augment embodiment variation and enrich morphological diversity within the single-arm domain. AgiBot-Beta and Galaxea contribute high-quality visual observations and temporally coherent action sequences. However, given that AgiBot-Beta, despite its

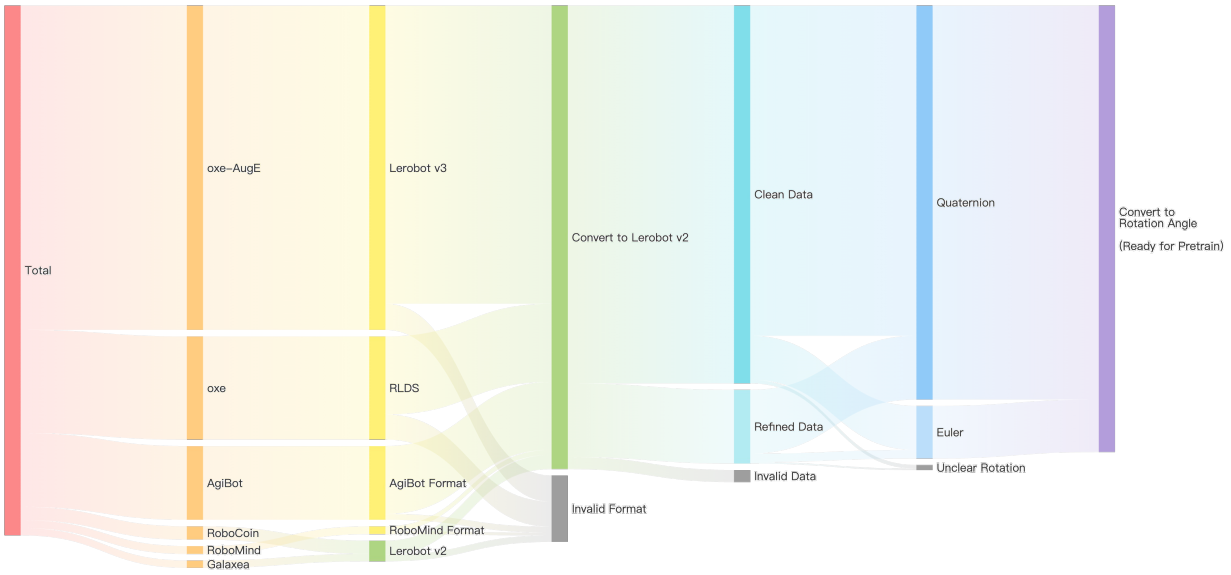


Figure 1 Data cleaning and preprocessing pipeline to construct the UniACT-dataset.

substantial scale, features only a single embodiment type, we deliberately reduce its sampling ratio during training to mitigate embodiment bias. Finally, since RoboCOIN and RoboMind are both annotated with fine-grained long-horizon task decompositions across multiple robot morphologies, they are prioritized to strengthen the model's capacity for complex task planning and cross-embodiment generalization for dual-arm robots. This principled integration balances scale, quality, and diversity while actively counteracting distributional imbalances inherent in the raw data sources.

Despite the abundance of manipulation trajectories provided by multiple open-source datasets, their raw versions often exhibit severe inconsistencies in data format. The raw data originates from heterogeneous sources—including LeRobot v2, LeRobot v3, RLDS (Robotics Learning Data Specification), and various dataset-native formats. To enable unified data processing and loading pipelines, we convert all trajectories into the LeRobot v2 format as a common standard.

Moreover, data missing, ambiguous language instructions, and semantic contamination might exist in these datasets. The direct use of such uncurated data risks training models on incorrect language-action mappings or ineffective behavioral policies. To address these challenges, we developed a systematic cleaning and processing pipeline. Through in-depth analysis of mainstream datasets, we identified several issues. For example, some task prompts contain non-English content (e.g., French, Spanish, and Chinese), nonsensical character sequences, redundant sentences, and even empty content, which severely impair intent parsing and language-action alignment. For long-horizon tasks, the original task files provide only high-level scene descriptions, while concrete subtask-level instructions are stored in separate files. These require additional processing to decompose trajectories into meaningful subtasks. We emphasize that neglecting proper text alignment will cause the trained model to effectively degenerate into a vision-action (VA) model lacking proper language grounding. Furthermore, significant frame rate discrepancies exist across OXE subsets—dropping as low as 5 FPS in some cases—leading to ambiguous time steps and increased difficulty in predicting actions for subsequent frames. Additionally, certain trajectories exhibit ambiguous or incomplete action annotations. For instance, the semantic meaning of individual action dimensions—such as whether a component encodes translation, rotation (and if so, in which representation), or gripper control—is often unspecified, and some action vectors lack critical components altogether. These deficiencies hinder the model's ability to generate complete and executable action predictions.

To mitigate these risks, we designed a multi-stage pipeline to filter out samples with typical issues and refine the remaining data to better align with our requirements. Specifically, for **invalid instructions**, we removed episodes with empty task fields, filtered out garbled instructions with nonsensical sequences, and normalized

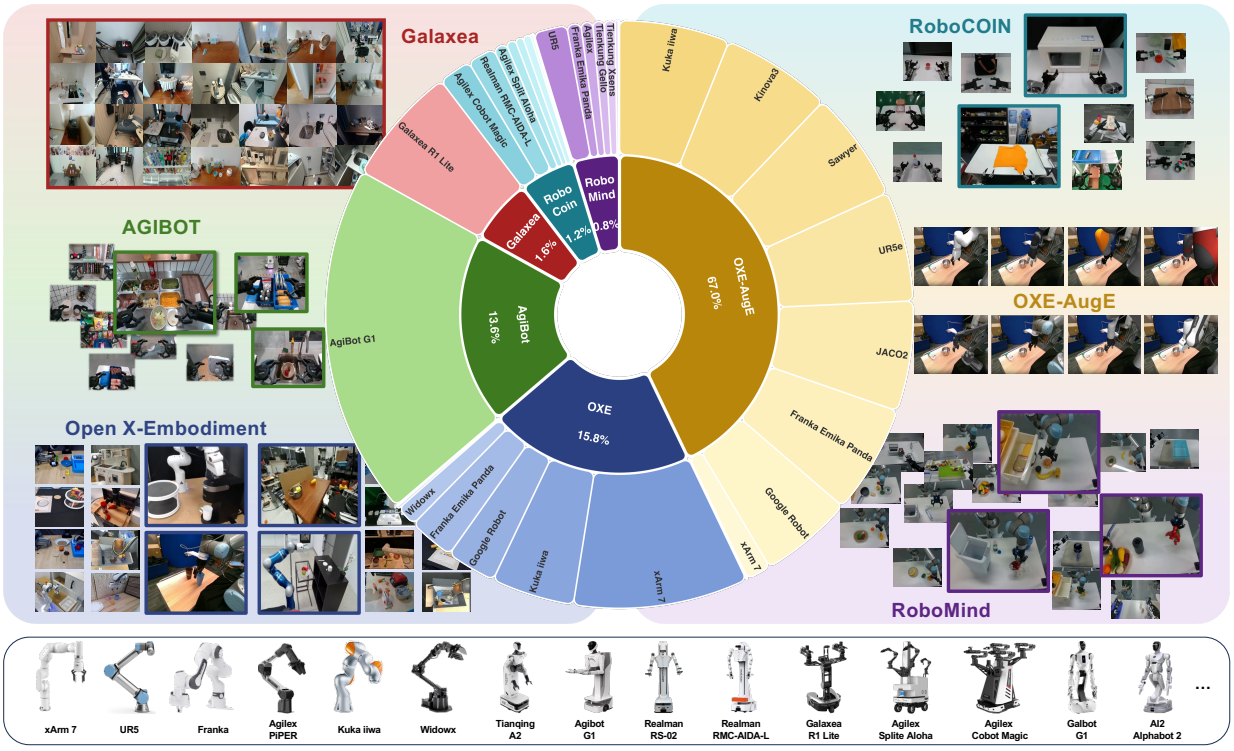


Figure 2 Overview of the integrated UniACT-dataset, which contains more than six million trajectories in 9500+ hours with 20+ unique robot embodiments.

mixed-language instructions via machine translation to ensure instruction validity and linguistic consistency. To address **frame-instruction misalignment**, we resolve index mismatches by recomputing temporal alignment across heterogeneous data files. Furthermore, we perform subtask decomposition and insert frame-aligned subtask instructions into the language stream to recover missing granular guidance. Moreover, we filter out **visual anomalies**. Specifically, trajectory segments containing visually degraded frames (such as completely black images, severe motion blur, or heavy occlusions) were discarded. Additionally, records captured from ineffective camera viewpoints (e.g. wrist cameras with insufficient field of view to observe the manipulation workspace) were excluded to ensure visual fidelity and task relevance. We also check and discard **abnormal action sequences**. Trajectories exhibiting abnormal length were excluded to avoid failure of original data collection process. And those with large consecutive action deltas were filtered to suppress jitter-induced noise. Furthermore, samples with severe mismatches between action update frequency and video frame rate were eliminated to maintain consistent temporal alignment between modalities. For trajectories with incomplete or **ambiguous actions**—such as missing action dimensions or unspecified rotation representations (e.g., unclear whether axis-angle, Euler angles, or quaternions are used)—we adopt a strict curation policy and discard such samples to ensure annotation integrity and enable reliable action prediction.

The overall data processing pipeline is illustrated in Figure 1. After applying this procedure, approximately 16% trajectories were discarded. Other trajectories were refined and merged into the final dataset. The remaining high-confidence samples constitute the **UniACT-dataset**, which serves as the primary source for pretraining our general-purpose VLA model. UniACT-dataset contains more than six million trajectories in 9500+ hours with 20+ embodiments, which is currently the largest collection within the non-private system.

2.3 Standardization of Data Formats

After Section 2.2, we have cleaned and preprocessed the trajectories from all sources. Here in this section we conduct data alignment to ensure full consistency in action space, observation inputs, and training interfaces. This unified framework is built upon three core design principles:

- **Action Representation: Delta Actions in End-Effector Position with Rotation Vectors.** All tasks are standardized into delta actions with end-effector (EEF) position. Converting absolute actions to relative (delta) actions simplifies and improves the efficiency of model training. And taking EEF rather than joint bridges the embodiment gap to facilitate cross-embodiment generalization. Moreover, all orientation representations (e.g., Euler angles, rotation matrices, quaternions) are transformed into rotation vectors (axis-angle representation), which yield greater stability in action prediction, particularly for fine-grained rotational manipulation. The 3D rotation vector of the end-effector is defined as $\boldsymbol{r} = \theta \boldsymbol{k}$, where $\boldsymbol{r} \in \mathbb{R}^3$, $\theta \in [0, \pi]$ is the rotation angle, and $\boldsymbol{k} \in \mathbb{R}^3$ (with $\|\boldsymbol{k}\| = 1$) is the unit axis about which the rotation occurs. At each timestep, the policy outputs two 7-dimensional action vectors for both left and right arm, where for each arm the action vector is represented as $[\Delta x, \Delta y, \Delta z, \boldsymbol{r}, gripper]$. This representation enhances controller robustness and ensures cross-platform compatibility.
- **Unified Single- and Dual-Arm Training via Zero Padding.** To allow a single policy network to handle both single-arm and dual-arm tasks, we adopt a pad-to-dual-arm strategy: for single-arm trajectories, the action dimensions corresponding to the unused arm are filled with zeros, and all single-arm data are uniformly treated as right-arm executions within a dual-arm configuration. The model consistently generates dual-arm action sequences but activates only the relevant arm channels during execution. This design enables parameter sharing across tasks while allowing the policy to implicitly learn when to employ one arm versus coordinated bimanual interaction. Guided by task instructions, the model can autonomously determine the appropriate interaction mode during inference, thereby achieving true unification of single- and dual-arm control.
- **Data Distribution: Diverse Embodiments and Task Scenarios.** The composition of the integrated dataset, including scale proportions and embodiment distribution, is illustrated in Figure 2. The scale and predominant robot embodiments for each dataset are visualized as a pie chart in the center panel. The left and right side illustrate the heterogeneity across datasets in terms of robots, scenes, tasks, camera viewpoints, and visual styles. The integrated dataset with more than six million trajectories spans more than twenty unique robot embodiments and 9500+ hours. At the bottom, we showcase representative robot embodiments including both single-arm and dual-arm. In terms of data distribution, the single-arm dataset OXE-AugE dominates with 67% of the total volume, followed by OXE as the second largest contributor. The rest four dual-arm datasets collectively account for approximately 17.2%. Given the inherent scale imbalance and uneven embodiment distribution in the raw data, we employ multi-granularity uniform sampling during training (discussed in Section 4) to balance embodiment coverage with skill learning efficiency.

3 The ABot-M0 Model

This section presents the detail of ABot-M0 model. The overall structure is shown in Figure 3. To achieve an end-to-end mapping from multimodal perception to robot action generation, as discussed in Section 3.1, we employ a two-component architecture consisting of a VLM and an action expert. In addition to the basic model, we utilize *action manifold learning* to directly predict actions. Section 3.2 introduces the two-stage training paradigm. The first stage performs unified pre-training on large-scale heterogeneous data to learn general, cross-task and cross-embodiment manipulation policies. The second stage introduces fine-grained spatial priors through Supervised Fine-Tuning (SFT) to enhance execution accuracy on complex tasks while preserving the model's generalization capabilities. We further introduce a feature collaboration architecture in Section 5 with an optional 3D spatial module to enhance spatial reasoning and dynamic coordination.

3.1 Model Architecture

Visual Language Model Backbone. Our model separates vision-language understanding from action generation into two specialized components, including a Visual Language Model (VLM) as the perception engine and an action expert as the decision module. The VLM processes stacked multi-view image sequences, typically captured from front-facing, wrist-mounted, and top-down cameras, together with natural language instructions. Both modalities are independently tokenized and fused into a unified token sequence to enable cross-modal reasoning. We use Qwen3-VL as the backbone VLM due to its strong image-text alignment and

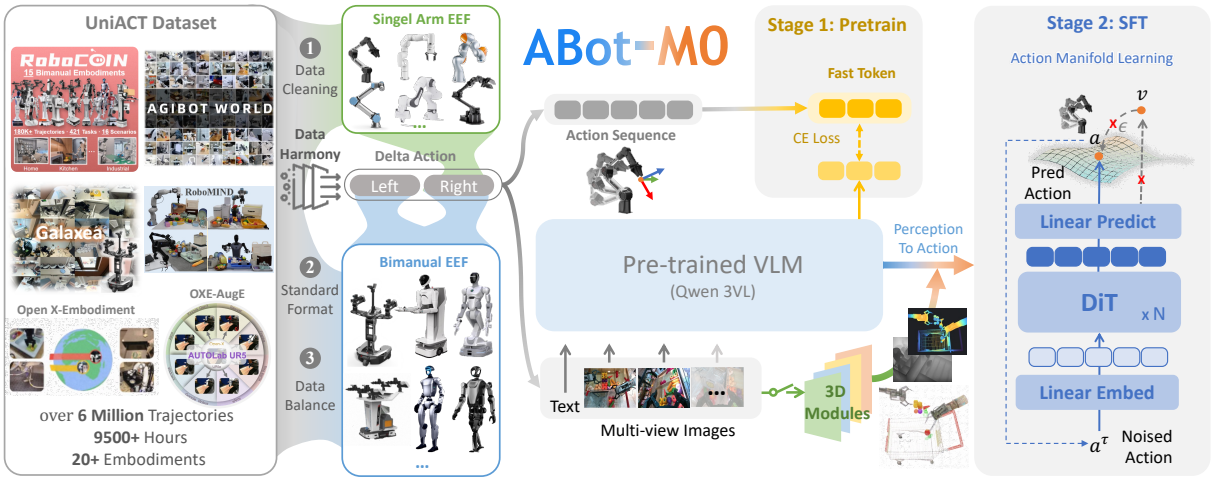


Figure 3 Model architecture of ABot-M0. We employ a two-component architecture consisting of a VLM and an action expert. In addition, we utilize action manifold learning to predict actions with two-stage training paradigm. We then carefully select VLM features and further introduce an optional 3D module and to enhance spatial reasoning.

long-sequence modeling capability, which is one of the few open-source models that combine high performance with broad applicability. By jointly processing visual and linguistic inputs, the VLM produces spatially aligned multimodal representations, including rich visual features and textual embeddings, which are passed to the action expert as context for action prediction.

Action Manifold Learning. Conventional diffusion or flow-based generative models are typically trained to predict noise (ϵ-pred) or velocity (v-pred), as illustrated in Figure 4. We argue that this poses a fundamental limitation for robot learning. According to the manifold hypothesis [6, 8], real-world data such as natural images and human language do not scatter randomly across their high-dimensional representation space but rather lie on an intrinsic low-dimensional manifold. We extend this insight to robotics, positing that an effective, coherent, and meaningful sequence of robot actions is also a highly structured entity residing on a low-dimensional *action manifold*. In stark contrast, the prediction targets of noise or velocity are inherently high-dimensional and off-manifold [25, 41]. Forcing a network with finite capacity to directly regress these unstructured, high-dimensional targets is inefficient. The challenge escalates significantly as the embodiment complexity grows from a single robotic arm to a full-body humanoid with two dexterous hands, which results in a substantially expanded action space and increased degrees of freedom. To relieve the pressure of model learning, we propose to predict the action directly rather than velocity, marked as a-pred. This shift enables our model, particularly the action expert, to focus on learning the intrinsic structure and semantics of actions, rather than expending its capacity on the inefficient task of filtering out noise from the vast ambient space.

We employ the Diffusion Transformer[33] (DiT) as the action generator, marked as V_θ. We apply flowing matching for action learning. Instead of predicting flow velocity, the action generator predicts the denoised action chunk $\hat{A}_t = [\hat{a}_t, \hat{a}_{t+1}, \ldots, \hat{a}_{t+H-1}]$ in length of H. Given a ground-truth action chunk A_t, a diffusion timestep $\tau \in [0, 1]$, and noise ϵ sampled from a standard normal distribution, we first construct the noisy action $A_t^\tau = \tau A_t + (1-\tau)\epsilon$. The network V_θ then takes the features ϕ_t from previous models including VLM and 3D modules, the current robot state q_t, and the noisy action A_t^τ as input and directly predict the action chunk $\hat{A}_t$ as the estimation of ground-truth action A_t:

$$\hat{A}_t = V_\theta(\phi_t, A_t^\tau, q_t). \tag{1}$$

Although the model directly predicts action chunk A_t, here we conduct loss on velocity, which yields better performance than on action both in our experiment and JiT [25]. The estimated velocity $\hat{v}$ and ground-truth velocity v is derived from

$$\begin{aligned} \hat{v} &= (\hat{A}_t - A_t^\tau)/(1-\tau), \\ v &= (A_t - A_t^\tau)/(1-\tau) \end{aligned} \tag{2}$$

Then we calculate the mean squared error (MSE) loss on velocity, which equals to a reweighted form of the action loss:

$$\mathcal{L}(\theta) = \mathbb{E}\|v_{\text{pred}} - v_{\text{target}}\|^2 = \mathbb{E}\left[w(\tau)\|V_\theta(\phi_t, A_t^\tau, q_t) - A_t\|^2\right], \tag{3}$$

where the weight is $w(\tau) = \frac{1}{(1-\tau)^2}$. This weighting function stems from the Jacobian of the transformation from the action space to the velocity space. It elegantly preserves the advantage of the flow matching method, which is to dynamically adjust the learning signal's strength across different noise levels. Intuitively, as τ approaches 1 (low noise), the weight tends to infinity, compelling the model to make fine-grained refinements in the final steps. Conversely, for small τ, the smaller weight allows the model to perform more substantial denoising.

Inference Process. During inference, we follow an Ordinary Differential Equation (ODE) solving trajectory to generate the action. Starting from pure noise $A_t^0 \sim \mathcal{N}(0, \mathbf{I})$, we perform iterative denoising over multiple steps. At each timestep τ, we first use the model to predict the estimated pure action $\hat{A}_t = V_\theta(\phi_t, A_t^\tau, q_t)$. And then compute the corresponding instantaneous flow velocity $\hat{v}$ according to Equation (3). Finally, we employ a numerical integrator (such as the Euler method or a higher-order solver) to update the action:

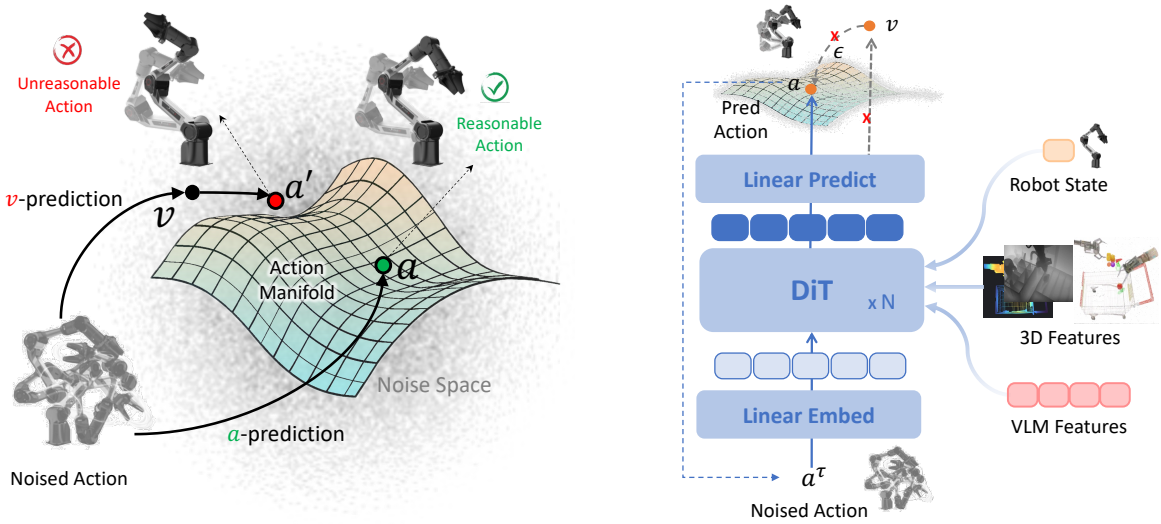
$$A_t^{\tau+\Delta\tau} = A_t^\tau + \Delta\tau \cdot \hat{v}. \tag{4}$$

This update cycle allows us to achieve an elegant action prediction at the model level, while still retaining the smooth and stable trajectory generation capabilities of flow models in the underlying dynamics.

3.2 Two-Stage Training Paradigm

Stage 1: Large-Scale Pre-training for Generalizable Action Priors. This stage aims to train the model on massive heterogeneous data so it can capture common manipulation patterns across different tasks and robot embodiments, forming a strong and transferable action prior. We use the **UniACT-dataset**, which contains approximately 6 million trajectories covering single-arm, dual-arm, and various robotic configurations, ranging from basic grasping to complex interactive behaviors. As discussed in Section 2.3, actions are represented as delta actions in the end-effector (EEF) frame. For each arm, the action vector is represented as $[\Delta x, \Delta y, \Delta z, \boldsymbol{r}, gripper] \in \mathbb{R}^7$, where $\boldsymbol{r}$ is a three-dimensional rotation vector. For dual-arm format, the action vector is in length of 14. The model takes multi-view image sequences and natural language instructions as input and outputs action sequences, aiming to map perceptual signals into executable motor commands. To improve convergence and stability, we apply a discretized modeling strategy using an action classification loss with a fast token head, which quantizes continuous actions in a way that preserves gradient flow during training.

To support both single-arm and dual-arm tasks within the same network, we use a *pad-to-dual-arm* strategy: for single-arm tasks, the unused arm's action dimensions are zero-padded, and all such operations are treated uniformly as right-arm actions. The model always generates full dual-arm outputs in length of 14, but only activates the relevant arm-channels during execution. This allows the action expert to share parameters across different embodiments, improving parameter efficiency and enabling the model to infer whether bimanual coordination is needed based on the instruction. Furthermore, to address imbalances in task and embodiment distributions, we apply a dual-weighted sampling strategy based on task category and robot morphology (detailed in Section 4). Rare tasks are oversampled with probability inversely proportional to their data ratio, while different robot configurations are balanced at the batch level. Experiments show this reduces long-tail bias and improves zero-shot generalization on challenging benchmarks such as RoboCasa and Libero-Plus, providing a solid base for downstream tasks.



(a) Action Manifold Hypothesis

(b) Action Manifold Learning

Figure 4 Action Manifold. (a) We posit that a meaningful action sequence is a highly structured entity residing on a low-dimensional action manifold. The conventional prediction targets of noise or velocity are inherently high-dimensional and off-manifold, which increase the burden of model learning and lead to unreasonble action. (b) We propose to predict the action directly rather than velocity, which enables the model to focus on learning the intrinsic structure and semantics of actions.

Stage 2: Space-Aware Supervised Fine-Tuning via Knowledge Injection. Although the pre-trained model generalizes well across many tasks, it still exhibits accumulated errors and unstable spatial alignment in high-precision scenarios such as fine insertion, cloth folding, or bimanual cooperation. These tasks require not only semantic reasoning but also accurate modeling of 3D spatial relationships, such as the relative pose between peg and hole, fabric crease directions, or timing between arms. To mitigate this limitation, we conduct supervised fine-tuning (SFT) to incorporate essential 3D spatial priors into the pre-trained model while preserving its generalization capabilities across diverse tasks.

We fine-tune both the VLM and action expert modules jointly with a small learning rate. Dropout and action noise perturbation are applied during training to increase robustness under real-world conditions. Results show substantial gains in success rates on difficult tasks like insertion, folding, and bimanual door opening, while performance on previously learned simple tasks remains stable. This confirms that new capabilities can be added without degrading existing ones.

This two-stage approach, comprising large-scale pre-training followed by targeted supervised fine-tuning (SFT), effectively reconciles generality with task-specific precision. The pre-training phase endows the model with a broad understanding of reasonable action spaces, while the subsequent SFT stage specializes the policy for accurate execution under task constraints. Critically, this methodology requires no architectural modifications, thereby enabling flexible integration of new skills in a plug-and-play manner. Moreover, the framework supports continual expansion. Emerging sensor modalities such as tactile feedback and force-torque sensing, along with novel task formulations, can be seamlessly incorporated through analogous fine-tuning protocols.

4 Pre-Training

A fundamental challenge in advancing general-purpose embodied intelligence centers on distilling invariant representations from large-scale, heterogeneous manipulation datasets that span morphologically diverse robot embodiments and task distributions. Multi-source embodied datasets exhibit significant variation in

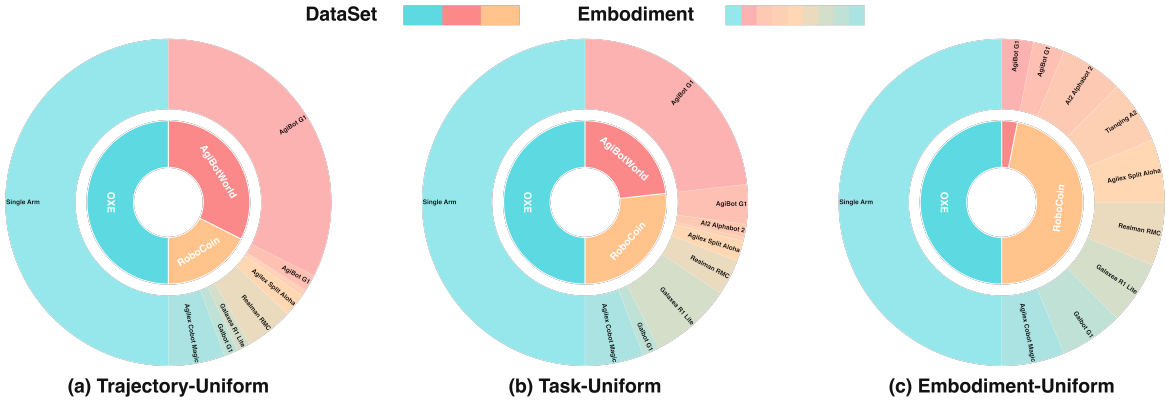


Figure 5 Embodiment distribution under different sampling strategies. Data are drawn from OXE [32], AgiBot-Beta [13], and RoboCoin [47]. The mixture receipe of single-arm data from OXE follows OpenVLA [20] and is fixed among these three strategies. We compare (a) Trajectory-Uniform, (b) Task-Uniform, and (c) Embodiment-Uniform sampling strategies and show the data distribution among different dataset and embodiments.

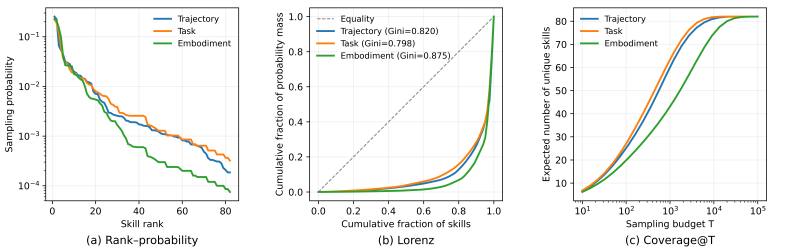


Figure 6 Skill Sampling Characteristics. Skill sampling behavior under different bimanual sampling strategies: (a) rank–probability distribution, (b) Lorenz curves of skill sampling probabilities, and (c) Coverage@T, measuring the number of unique skills covered as sampling progresses.

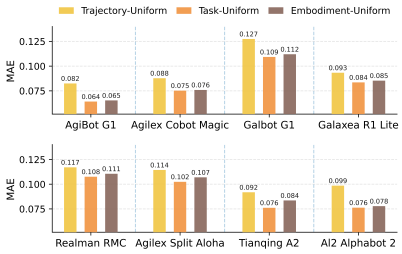


Figure 7 Validation Performance Across Bimanual Embodiments on RoboCoin.

trajectory length, task granularity, and embodiment diversity. Under naive trajectory-level uniform sampling, the training distribution becomes dominated by data volume, resulting in pathological biases. On one hand, long-tail but valuable domains, such as uncommon skills or rare robot morphologies, receive insufficient exposure, hindering effective convergence. On the other hand, limited training budgets are consumed by redundant sampling of frequent skills, which limits coverage of rare behaviors and ultimately undermines cross-task and cross-embodiment generalization. To address this issue, this section presents a systematic study of sampling strategies for bimanual learning. We propose multi-level stratified sampling to balance the embodiment coverage and skill acquisition efficiency, thereby establishing a data-aware trade-off that facilitates robust cross-embodiment and cross-task generalization. Section 4.1 conducts an in-depth analysis of different sampling strategies and propose a multi-level stratified sampling to balance the embodiment coverage and skill acquisition efficiency. Section 4.2 empirically validates how distributional disparities induced by these strategies impact generalization performance. The superiority of Task-Uniform is consistently validated across three dimensions: cross-embodiment generalization, cross-dataset transfer, and downstream task adaptation.

4.1 Sampling Ratio for Multi-Embodiment Learning

This section systematically analyzes three uniform sampling strategies for bimanual data, focusing on their effects on embodiment distribution and skill sampling behavior in multi-embodiment learning. We fix single-arm data at 50% of the sampling budget and perform controlled ablation of sampling strategies solely within the bimanual regime.

Embodiment-Level Distribution Analysis. At the embodiment level, the three sampling strategies induce

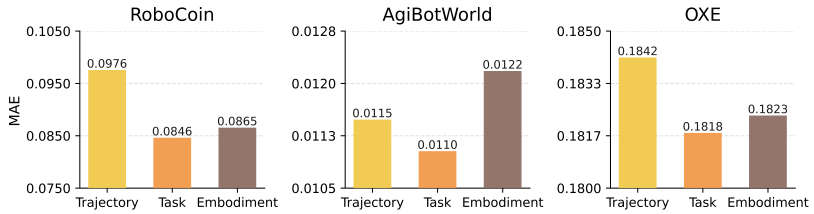


Figure 8 Dataset-Wise Validation Performance Under Different Sampling Strategies.

Method	Total
Trajectory-Uniform	71.3
Embodiment-Uniform	71.6
Task-Uniform	**72.4**

Table 2 Downstream Performance on Libero Plus.

markedly different effective training distributions, as shown in Figure 5. Trajectory-uniform sampling largely preserves the original dataset scale imbalance, causing training to be dominated by AgiBot-G1 and resulting in a highly concentrated embodiment distribution, which increases the risk of homogenization. In contrast, task-uniform sampling substantially alleviates this issue by increasing the sampling visibility of multi-task but single-embodiment data from RoboCoin. This mechanism allows long-tail embodiments to receive more exposure during training, while still retaining AgiBot-Beta as the primary data source, leading to a more stable balance between embodiment coverage and data scale.

Skill-Level Sampling Characteristics. Apart from embodiment balance, we examine the skill-level sampling behavior of these strategies, shown in Figure 6. Although all strategies exhibit pronounced long-tailed skill distributions, task-uniform sampling produces a noticeably less concentrated probability mass. This is reflected by a Lorenz curve closer to the equality line and a lower corresponding Gini coefficient. Moreover, under the same sampling budget, task-uniform sampling achieves faster growth in the number of covered unique skills, indicating higher efficiency in reducing redundant sampling and improving skill diversity. In contrast, embodiment-uniform sampling, while enforcing stronger balance across embodiments, further concentrates sampling probability on a small set of high-frequency skills, resulting in substantially slower coverage growth.

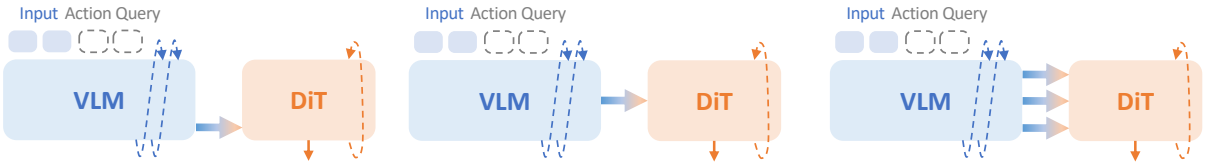
Overall, task-uniform sampling provides a more favorable trade-off between embodiment diversity and skill coverage, whereas trajectory-uniform sampling suffers from severe scale dominance and embodiment concentration, and embodiment-uniform sampling, despite improving embodiment balance, introduces stronger bias toward high-frequency skills and degrades skill coverage efficiency.

4.2 Generalization Evaluation

In Section 4.1, we have analyzed how different bimanual sampling strategies introduce structural biases during training from the perspectives of embodiment distribution and skill sampling statistics. To further examine whether these differences translate into measurable generalization behavior and downstream performance, this section conducts a systematic evaluation across datasets and embodiments. Notably, pretraining is conducted exclusively on OXE, AgiBot-Beta, and RoboCoin, while Libero is used only for downstream supervised fine-tuning.

Validation Set Construction. To avoid validation results being dominated by dataset scale or partition granularity, the validation set is constructed by aligning the sampling unit with the most representative diversity dimension of each dataset. Specifically, for OXE, since the single-arm sampling ratio is fixed, we randomly sample 1,000 trajectories as the validation set. For AgiBot-Beta, we stratify by task and randomly sample one trajectory for each task (183 in total) to ensure task coverage. For RoboCoin, we stratify by embodiment and randomly sample 30 trajectories per embodiment (240 in total) to ensure embodiment coverage. The evaluation metric is the mean absolute error (MAE) between the predicted and ground-truth actions. All models are trained under identical settings for 50k steps before evaluation.

Cross-Embodiment Generalization. We assess cross-embodiment generalization using the RoboCoin validation set stratified by bimanual embodiment. Results are shown in Figure 7. Task-Uniform and Embodiment-Uniform achieve similar overall MAE, while Task-Uniform is consistently lower on most embodiments. In contrast, Trajectory-Uniform performs substantially worse, exhibiting markedly higher MAE across



(a) Last Layer (b) Intermediate Layer (c) Multiple Layers

Figure 9 VLM Feature Interaction. Following pre-training on robotics data, we feed an action expert with either the VLM's raw hidden features or an additional action query, extracted from its final, intermediate, or multiple layers.

nearly all embodiments. These results suggest that task-level organization provides more transferable interaction supervision, enabling strong cross-embodiment performance without enforcing strict embodiment-level uniformity.

Cross-Dataset Generalization. Results in Figure 8 indicate that Task-Uniform achieves the lowest MAE on OXE, AgiBot-Beta, and RoboCoin, demonstrating a more stable and consistent advantage. In contrast, Trajectory-Uniform yields the highest errors on RoboCoin and OXE, while Embodiment-Uniform performs worst on AgiBot-Beta, reflecting the biases induced by different uniformity objectives under cross-dataset evaluation. This suggests that Task-Uniform better balances learning signal and coverage needs across data sources, leading to a more favorable overall trade-off in cross-dataset generalization.

Downstream Transfer on Libero Plus. Beyond MAE evaluation, we further transfer the pretrained models to the Libero Plus benchmark for supervised fine-tuning to assess downstream transferability. As reported in Table 2, we observe noticeable performance differences across sampling strategies, and the overall trend is consistent with the MAE evaluation. These results suggest that the choice of sampling strategy during pretraining can influence downstream learning behavior and final performance on manipulation tasks.

5 Perception to Action

A fundamental challenge in advancing general-purpose embodied intelligence lies in endowing models with dual capabilities: comprehending high-level task intent while simultaneously grounding actions in precise environmental structure. Vision–language models (VLMs) demonstrate strong capabilities in parsing natural language instructions, recognizing object categories, and inferring contextual semantics. Nevertheless, their spatial perception typically remains qualitative, capturing only semantic relations such as "the cup is to the left of the box" without metric precision regarding distance, reachability, or egocentric pose. In contrast, robotic action execution demands millimeter-level spatial reasoning and dynamic coordination. Therefore, relying solely on VLMs is insufficient for fine-grained manipulation. To address this, we discuss the choice of VLM feature for action expert in Section 5.1 and compare different mechanisms to inject 3D information in Section 5.2. VLM-derived visual features and geometry-aware 3D representations undergo hierarchical cross-attention within a shared latent space, producing a fused multimodal embedding that conditions the action expert to generate spatially grounded and temporally coherent action sequences.

5.1 VLM Feature Interaction

Our vision-language model (VLM) is pre-trained on large-scale vision-language-action (VLA) data, endowing it with strong generalization capabilities in embodied scenarios and the ability to directly predict discrete actions in textual form. However, architectural, training, and output-format differences persist between the pre-trained VLM and the downstream action expert policy model, such as the discrete versus continuous action representation. In contrast to prior work like VLA-Adapter [44] that focused on feature selection for VLMs without robotics pretraining, we explore which features from a VLM pretrained with robotic data are most critical for training an action expert.

We conduct ablation studies with Qwen3-VL-4B as the VLM backbone and a 16-layer Diffusion Transformer (DiT) as the action expert. We explore two key aspects of feature utilization: (a) Which layer(s) of the pre-

trained VLM provide the most effective features for policy learning? We evaluate features from three different layers, including features of final-layer(deepest representation), features of intermediate-layer (approximately the 70th percentile layer in depth), and concatenated features from all hidden layers. (b) Does incorporating action queries yield better performance than using raw VLM features alone? We also experimented with action queries drawn from the intermediate layer, the final layer, and the last 16 layers, respectively. Furthermore, we investigated using the concatenated features of the query and the hidden states from the last 16 layers of VLM as a condition. The schematic illustration is provided in Figure 9.

Experimental results shown in Table 8 (detailed in ablation study) lead to the following conclusions: (i) Using raw VLM features directly outperforms feature concatenation with action queries. (ii) Deep-layer VLM features consistently yield better policy performance than shallow or intermediate features. (iii) Aggregating features from multiple layers provides no significant improvement over using the final layer alone.

These findings suggest that large-scale VLA pre-training enables the VLM to internalize action-space semantics, where intermediate layers capture rich multi-modal scene representations and deeper layers progressively encode action-relevant semantic structures. Consequently, the model no longer requires auxiliary mechanisms such as action queries to adapt the VLM to action prediction tasks. The superior performance of the final-layer features surpasses both shallow-layer and multi-layer alternatives and demonstrates that a single deep representation effectively encapsulates the most discriminative action-space information. Moreover, the under-performance of action-query augmentation further confirms that the pre-trained VLM has already achieved sufficient alignment with the action domain, rendering additional adaptation modules redundant.

This study not only validates the effectiveness of our VLA pre-training paradigm but also provides a practical and efficient feature selection strategy for transferring pre-trained VLMs to downstream action generation modules—favoring simplicity (single deep-layer features) over complexity without sacrificing performance.

5.2 3D Information Injection

While Section 5.1 establishes that deep-layer VLM features alone already encode rich action-relevant semantics, they remain inherently limited by their 2D visual input. VLMs excel at semantic understanding but lack precise geometric awareness. It knows "where" but can not tell "how far" or "whether reachable". This gap becomes critical in precise manipulation, where spatial reasoning is required.

To complement the VLM's semantic feature with explicit 3D spatial structure, we introduce a plug-and-play 3D information injection module that operates alongside the VLM within our dual-stream architecture. Rather than replacing the VLM, this module enriches its representation with geometric priors, enabling the unified system to jointly reason about what to do and where to act.

Specifically, we integrate two sources of 3D awareness. (i) Feedforward single-image 3D features: We use a model pretrained on large-scale 3D data, such as VGGT [43], to extract 3D-aware features from a single RGB image. By jointly modeling appearance and geometry, VGGT infers the structure of the 3D environment even from monocular input. (ii) Implicit multi-view features: To further enhance robustness under occlusion and viewpoint variation, we employ Qwen-Image-Edit [45] to synthesize additional viewpoints from the original image, implicitly capturing 3D scene layout through view consistency.

Crucially, these 3D features are fused with the final-layer VLM features (identified in Section 5.1) as the most effective semantic representation. As illustrated in Figure 10, fusion occurs just before the action expert. We compare three different fusion strategies: concatenation, cross-attention (VLM features as queries, 3D features as keys/values), and Q-Former (learnable queries attending to multi-layer VLM and 3D features). Ablation studies in Section 6 show that single-layer cross-attention achieves optimal performance, confirming that attention-based interaction is key to harmonizing semantic and geometric streams.

For multi-view features, we lightly fine-tune Qwen-Image-Edit on Bridge [42] and collected LIBERO [28] data, using only 50 paired samples per dataset. Experiments reveal that two synthesized views significantly outperform one, particularly on viewpoint-sensitive tasks (e.g., +14% points on LIBERO-Plus's camera perturbation subset). Our final system thus adopts two synthetic views with two-step inference, with their representations fused into the action expert via cross-attention to enrich spatial awareness.

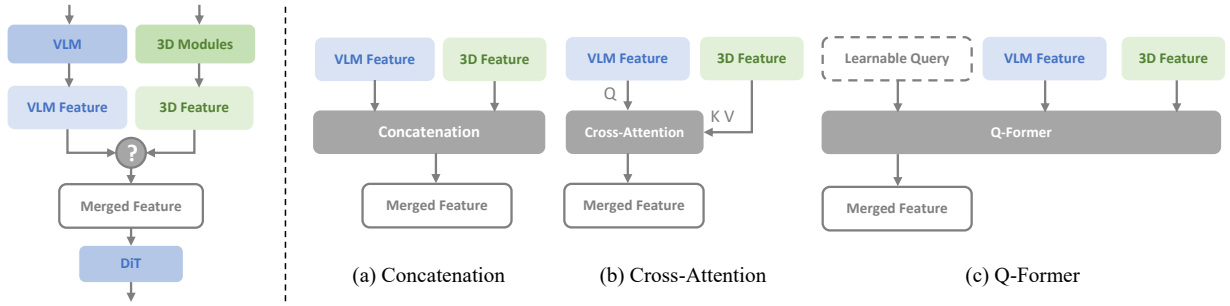


Figure 10 3D Information Injection. Left: pipeline for fusing VLM and 3D features. Right: three fusion strategies.

Note the 3D module is fully modular. It can be toggled on/off or combined (single-view + multi-view) based on task demands, offering flexible deployment without retraining the core VLM. The semantic understanding from the VLM and geometric grounding from 3D models together enables robust and precise action generation in complex 3D environments.

6 Evaluation

6.1 Experiment Settings

To comprehensively evaluate ABot-M0, we conducted extensive experiments on multiple simulation benchmarks, including LIBERO [28], LIBERO-Plus [15], RoboCasa GR1 Tabletop Tasks [31] and Robotwin2.0 [10], to assess the model's generalization, robustness, and adaptability to both single and dual-arm robots. Our training pipeline is built upon the StarVLA [38] framework. ABot-M0 employs Qwen3-VL [1] 4B as its VLM backbone, the 0.16B DiT with Action Manifold Learning (AML) as the action expert, and the VGGT [43] as an optional 3D spatial model. By default, we use 4 denoising steps and an action chunk size of 16. For visual inputs, images are resized to 224 × 224. For pre-training, we use a learning rate of 1e-5, a total batch size of 1024, and train the model for 100K steps.

6.2 Main Results

6.2.1 LIBERO

Table 3 Evaluation results on the LIBERO benchmark. We train one ABot-M0 jointly on all suites and report the success rate on each suite.

Method	L-Spatial	L-Object	L-Goal	L-Long	Average
Diffusion Policy [12]	78.5	87.5	73.5	64.8	76.1
OpenVLA [20]	84.7	88.4	79.2	53.7	76.5
SpatialVLA [35]	88.2	89.9	78.6	55.5	78.1
CoT-VLA [49]	87.5	91.6	87.6	69.0	83.9
π_0-Fast [34]	96.4	96.8	88.6	60.2	85.5
GR00T-N1 [31]	94.4	97.6	93.0	90.6	93.9
π_0 [3]	98.0	96.8	94.4	88.4	94.4
F1 [30]	98.2	97.8	95.4	91.3	95.7
InternVLA-M1 [14]	98.0	99.0	93.8	92.6	95.9
Discrete Diffusion VLA [27]	97.2	98.6	97.4	92.0	96.3
$\pi_{0.5}$ [17]	**98.8**	98.2	98.0	92.4	96.9
GR00T-N1.6 [31]	97.7	98.5	97.5	94.4	97.0
OpenVLA-OFT [21]	97.6	98.4	97.9	94.5	97.1
X-VLA [51]	98.2	98.6	97.8	**97.6**	98.1
ABot-M0 (Ours)	**98.8**	**99.8**	**99.0**	96.6	**98.6**

The main experimental results on the LIBERO [28] benchmark are presented in Table 3. ABot-M0 demonstrates outstanding performance across the LIBERO test suites, achieving an average success rate of 98.6%. Notably, it attained success rates of 98.8% and 96.6% on the spatial long-horizon trajectory tasks, respectively. These results indicate that ABot-M0 possesses strong spatial understanding capabilities and can effectively execute complex multi-step manipulation tasks.

6.2.2 LIBERO Plus

Table 4 Zero-shot performance on LIBERO-Plus. All methods are trained only on the standard LIBERO dataset without fine-tuning on LIBERO-Plus dataset.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Total
OpenVLA [20]	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
OpenVLA-OFT [21]	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
OpenVLA-OFT_w [21]	10.4	38.7	70.5	76.8	**93.6**	49.9	69.9	55.8
Openvla-OFT_m [21]	55.6	21.7	81.0	92.7	91.0	78.6	68.7	67.9
NORA [16]	2.2	37.0	65.1	45.7	58.6	12.8	62.1	39.0
WorldVLA [7]	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
UniVLA [4]	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
π_0 [3]	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
π_0-Fast [34]	**65.1**	21.6	61.0	73.2	73.2	74.4	68.8	61.6
RIPT-VLA [39]	55.2	31.2	77.6	88.4	91.6	73.5	74.2	68.4
ABot-M0 (Ours)	60.4	**67.9**	**86.4**	**96.2**	91.6	**86.4**	**82.6**	**80.5**

To further assess the generalization ability and robustness of ABot-M0 in unseen scenarios, we also evaluated it on LIBERO-Plus [15]. LIBERO-Plus is a variant of LIBERO that introduces controllable perturbations across seven dimensions, including vision and language, for systematic vulnerability analysis. As shown in Table 4, we conducted zero-shot evaluation on LIBERO-Plus using our ABot-M0 model trained solely on LIBERO. ABot-M0 achieves a success rate of 80.5%, substantially outperforming prior methods like OpenVLA and OFT by 12.6-64.9%. This result demonstrates the model's strong inherent robustness and generalization against visual and language perturbations.

6.2.3 RoboCasa GR1 Tabletop Tasks

We further evaluated our method on the Robocasa GR1 Tabletop Tasks [31]. This benchmark comprises 24 distinct manipulation tasks involving complex interactions with articulated objects of various geometries, designed to assess general-purpose robot policies in a wide range of household scenarios. In this evaluation, we predict a 29 action space encompassing both dual arms, hands, and the waist, with an action chunk size of 16, to test the robustness of the new AML paradigm for predicting 464 high-dimensional actions under limited information capacity. As shown in Table 5, ABot-M0 achieves a 58.3% success rate, comprehensively outperforming prior noise prediction-based experts like GR00T-N1.6 and action regression methods such as OFT, thereby setting a new SOTA. These results demonstrate that, under the Action Manifold Hypothesis, directly predicting actions is a more effective paradigm than predicting noise, validating the superiority of AML.

6.2.4 RoboTwin 2.0

To evaluate the generalization capability of our method, we conduct multi-task training on RoboTwin2.0 [10]: ABot-M0 and all baselines are trained on 2500 demonstrations collected in clean scenes (50 per task) plus 25000 demonstrations gathered in heavily randomized scenes (500 per task). The randomizations include random backgrounds, table clutter, table height perturbations, and random lighting. The results for $\pi_{0.5}$ and X-VLA are taken from Motus [2]. As shown in Table 6, ABot-M0 outperforms strong VLA models like $\pi_{0.5}$

Table 5 Evaluation Results on RoboCasa GR1 Tabletop Tasks. We train one model jointly on all 24 tasks, and report mean results over 50 rollouts per task.

Task	GR00T-N1.6	Qwen3GR00T	Qwen3PI	Qwen3OFT	Qwen3FAST	**ABot-M0 (Ours)**
PnPBottleToCabinetClose	51.5	46.0	26.0	30.0	38.0	**86.0**
PnPCanToDrawerClose	13.0	**80.0**	62.0	76.0	44.0	74.0
PnPCupToDrawerClose	8.5	54.0	42.0	44.0	**56.0**	48.0
PnPMilkToMicrowaveClose	14.0	48.0	**50.0**	44.0	44.0	46.0
PnPPotatoToMicrowaveClose	41.5	28.0	42.0	32.0	14.0	**50.0**
PnPWineToCabinetClose	16.5	46.0	32.0	36.0	14.0	**66.0**
PnPNovelFromCuttingboardToBasket	58.0	48.0	40.0	50.0	54.0	**70.0**
PnPNovelFromCuttingboardToCardboardbox	46.5	40.0	46.0	40.0	42.0	**58.0**
PnPNovelFromCuttingboardToPan	68.5	68.0	60.0	70.0	58.0	**76.0**
PnPNovelFromCuttingboardToPot	65.0	52.0	40.0	54.0	58.0	**66.0**
PnPNovelFromCuttingboardToTieredbasket	46.5	**56.0**	44.0	38.0	40.0	38.0
PnPNovelFromPlacematToBasket	**58.5**	42.0	44.0	32.0	36.0	52.0
PnPNovelFromPlacematToBowl	57.5	44.0	52.0	58.0	38.0	**66.0**
PnPNovelFromPlacematToPlate	**63.0**	48.0	50.0	52.0	42.0	60.0
PnPNovelFromPlacematToTieredshelf	**28.5**	18.0	28.0	24.0	18.0	26.0
PnPNovelFromPlateToBowl	57.0	**60.0**	52.0	**60.0**	52.0	54.0
PnPNovelFromPlateToCardboardbox	43.5	**50.0**	40.0	**50.0**	30.0	48.0
PnPNovelFromPlateToPan	51.0	54.0	36.0	**66.0**	48.0	**66.0**
PnPNovelFromPlateToPlate	**78.7**	70.0	48.0	68.0	50.0	64.0
PnPNovelFromTrayToCardboardbox	51.5	38.0	34.0	44.0	28.0	**54.0**
PnPNovelFromTrayToPlate	**71.0**	56.0	64.0	56.0	34.0	68.0
PnPNovelFromTrayToPot	**64.5**	50.0	44.0	62.0	46.0	64.0
PnPNovelFromTrayToTieredbasket	57.0	36.0	50.0	54.0	36.0	**60.0**
PnPNovelFromTrayToTieredshelf	31.5	16.0	28.0	30.0	16.0	**38.0**
Average	47.6	47.8	43.9	48.8	39.0	**58.3**

and X-VLA in both the clean and randomized multi-task settings on RoboTwin2.0, achieving a success rate of over 80%.

6.3 Ablation Study

6.3.1 Action Manifold Learning

To validate the superiority of AML, we present a comprehensive comparison in Table 7 between its action prediction paradigm and the noise prediction paradigm of GR00T. For a fair comparison, both GR00T and ABot-M0 are initialized from the same Qwen3-VL-FAST pretrained model, with their action experts both sized at 0.16B. Furthermore, ABot-M0 does not employ an additional 3D spatial module in this setup, ensuring that the only distinction between the two models is their prediction paradigm.

Under the default configuration of 4 denoising steps and an action chunk size of 8, ABot-M0 outperforms GR00T by 1.7%, indicating that under typical settings, directly predicting the action is more efficient than predicting noise. When varying only the number of denoising steps, ABot-M0 consistently outperforms GR00T, especially in the extreme scenario of just 2 denoising steps. However, when the number of denoising steps is increased to 10, neither model shows performance gains, suggesting that both ABot-M0 and GR00T can already achieve high-quality action generation efficiently.

When the action chunk size is increased to 10, GR00T's performance shows no significant change, whereas ABot-M0's performance improves further to 72.4. This is because most tasks in LIBERO-Plus are simple, short-horizon tasks, for which a chunk size of 10 is a more optimal choice. Notably, as we increase the action chunk size further to 30, this oversized chunk becomes ill-suited for the majority of these simple, short-horizon tasks. Concurrently, a larger action chunk increases the dimensionality of the action sequence to be predicted, thereby increasing task difficulty. Under these conditions, GR00T's performance drops sharply by 23.6%,

Table 6 Evaluation results on RoboTwin 2.0 Benchmark. (Clean vs Randomized, 50+ tasks).

Simulation Task	$\pi_{0.5}$		X-VLA		ABot-M0 (Ours)	
	Clean	Rand.	Clean	Rand.	Clean	Rand.
Place Dual Shoes	12	7	79	**88**	75	79
Move Stapler Pad	16	18	**78**	73	49	72
Stack Blocks Two	48	56	92	87	**100**	98
Scan Object	42	38	14	36	**100**	80
Place Object Stand	74	65	86	88	**100**	94
Place Fan	25	36	80	75	84	**91**
Move Pillbottle Pad	33	29	73	71	93	**100**
Pick Dual Bottles	10	6	47	36	77	**100**
Blocks Ranking Rgb	43	35	83	83	68	**85**
.....(50 tasks)	-	-	-	-	-	-
Turn Switch	5	6	40	61	52	**81**
Pick Diverse Bottles	5	3	58	36	70	**93**
Place Bread Basket	48	56	81	71	**91**	87
Stack Blocks Three	15	16	6	10	**72**	49
Put Bottles Dustbin	12	9	74	**77**	74	70
Place Can Basket	19	25	49	52	45	**66**
Stamp Seal	36	23	76	**82**	76	76
Handover Block	18	19	73	37	**93**	69
Stack Bowls Three	33	35	76	86	**100**	88
Place Object Basket	43	36	44	39	**97**	95
Open Microwave	35	37	79	71	**93**	49
Average	42.98	43.84	72.80	72.84	80.42	**81.16**

while ABot-M0 maintains an impressive 62.8% success rate. This validates the "Action Manifold Hypothesis", which posits that predicting high-dimensional noise is a difficult and inefficient paradigm requiring vast information capacity, whereas directly predicting the action itself is a superior approach. Particularly for future challenges such as more difficult long-horizon tasks requiring longer action chunks, and dexterous manipulation or whole-body control demanding higher action dimensionality, the AML paradigm lays a solid foundation for the long-term advancement of embodied intelligence.

6.3.2 VLM Feature Interaction

In Table 8, we investigate the overall effect of different feature conditions on action generation after the VLM has been pre-trained on robotics data. Conditioning on the features from the final hidden layer yields the highest success rate of 71%, suggesting that this layer has become highly abstract and task-relevant, encapsulating the most critical information required for action generation. For the action queries, we employ 64 queries to strike a balance between efficiency and performance. Using queries alone results in performance slightly inferior to that of the original features. This indicates that robotics pre-training has already effectively aligned the VLM's internal feature space with the action space, rendering the benefit of an additional, from-scratch query interface marginal. In fact, it may even underperform the direct use of the already-optimized original features. Furthermore, concatenating features and queries as the conditioning input resulted in the lowest performance. We posit that this is because the pre-trained VLM has already formed a coherent internal representational structure across its layers. In such a case, injecting query features, which may introduce redundant or conflicting signals, can disrupt the learning process of the policy network. Future work could also explore assigning different weights to these two types of features or more effective ways to combine them. In summary, after robotics pre-training, the final layer features are well-adapted to the action domain and achieve strong performance, while the gains from incorporating intermediate layer features or additional action queries are limited.

6.3.3 3D Information Injection

To systematically validate the effectiveness of our proposed 3D feature injection mechanism, we conduct ablation studies on LIBERO [28] and its more challenging extension, LIBERO-Plus [15]. All experiments are initialized from the same pretrained vision-language model (Qwen3-VL 4B) and differ only in the

Table 7 Ablation studies of Action Manifold Learning on LIBERO-Plus benchmark.

Method	Hparams	Camera	Robot	Language	Light	Background	Noise	Layout	Total
Qwen3-VL-GR00T	Denoising Steps 4	41.0	61.2	86.5	91.3	91.6	53.8	73.4	69.3
ABot-M0	Action Chunk 8	39.9	63.7	85.9	90.8	93.1	60.6	76.8	71.0
Qwen3-VL-GR00T	Denoising Steps 2	35.4	59.7	86.7	90.2	90.1	49.8	73.3	67.2
ABot-M0	Denoising Steps 2	37.4	59.6	82.7	**93.1**	92.9	56.7	**80.5**	69.7
Qwen3-VL-GR00T	Denoising Steps 10	36.7	61.0	88.0	91.9	91.9	51.1	74.6	68.6
ABot-M0	Denoising Steps 10	**44.6**	56.9	81.4	90.2	90.1	62.9	78.3	70.2
Qwen3-VL-GR00T	Action Chunk 10	37.3	61.6	**88.7**	92.8	92.8	51.7	75.2	69.3
ABot-M0	Action Chunk 10	42.5	**64.1**	86.3	92.7	**93.3**	**63.2**	78.0	**72.4**
Qwen3-VL-GR00T	Action Chunk 30	7.9	31.7	64.9	83.2	73.6	24.0	55.2	45.7
$\Delta(Change)$	Action Chunk 30	-33.1	-29.5	-21.6	-8.1	-18.0	-29.8	-18.2	-23.6
ABot-M0	Action Chunk 30	23.5	53.9	74.6	92.8	91.4	53.1	68.7	62.8
$\Delta(Change)$	Action Chunk 30	-16.4	-9.8	-11.3	+2.0	-1.7	-7.5	-8.1	-8.2

Table 8 Ablation studies of VLM feature interaction manners on LIBERO-Plus benchmark.

Layers	Feature	Query	Camera	Robot	Language	Light	Background	Noise	Layout	Total
Last	✓	×	39.9	**63.7**	85.9	90.8	**93.1**	60.6	**76.8**	**71.0**
	×	✓	38.4	61.5	87.2	91.6	90.3	59.1	75.3	70.0
Intermediate	✓	×	38.2	49.9	88.7	92.7	89.5	**64.3**	73.6	69.0
	×	✓	**42.3**	54.4	**89.1**	87.5	88.7	53.9	75.1	68.3
Last 16 layers	✓	×	36.9	54.1	87.1	**93.9**	90.3	51.1	74.4	67.4
	×	✓	25.8	57.5	80.9	88.7	87.5	37.3	74.1	65.2
	✓	✓	25.0	50.4	88.5	90.1	88.5	50.0	70.3	63.8

components under evaluation, which are introduced via supervised fine-tuning (SFT). We strictly exclude other enhancements to ensure that any observed performance change can be unambiguously attributed to our 3D perception mechanism.

Results are reported in Table 9 and Table 10. First, we evaluate VGGT-based single-image 3D features under three fusion strategies: Concatenation, Cross-Attention, and Q-Former. In all cases, integrating VGGT features consistently improves performance on both benchmarks, confirming the general utility of single-view 3D priors for VLA tasks. Among the three, a single-layer Cross-Attention yields the best results and is adopted as the default fusion strategy in subsequent experiments.

We then assess implicit multi-view features generated by Qwen-Image-Edit [45]. Incorporating synthesized views significantly enhances generalization, yielding absolute success rate gains of 2.8% on LIBERO and 2-4% on LIBERO-Plus. Notably, using two synthesized views further strengthens 3D spatial awareness. This improvement is most pronounced on the "camera viewpoint perturbation" subset, where accuracy increases by up to 14%, demonstrating that multi-view information effectively mitigates reliance on fixed observation angles.

7 Future Work & Conclusion

7.1 Future Work

We will further take scaling data with human demonstrations and UMI-collected [11, 50] trajectories to approach data capacity limits. Finer control over data quality, especially task balance and embodiment coverage, requires deeper study. A self-evolving data engine could form a closed loop of execution, failure

Table 9 Ablation studies of 3D feature injection on LIBERO benchmark.

Method	L-Spatial	L-Object	L-Goal	L-Long	Average
Baseline	97.8	98.2	94.6	90.8	95.4
VGGT (cross attention)	99.4	99.2	96.2	95.6	97.6
VGGT (concat)	99.4	98.6	96.2	93	96.8
VGGT (Q-former)	99.2	99.2	97.2	94	97.4
Qwen-Image-Edit (1 view)	99.4	**99.4**	97.2	96.4	98.1
Qwen-Image-Edit (2 views)	**99.6**	99.0	**97.4**	**96.6**	**98.2**

Table 10 Ablation studies of 3D feature injection on LIBERO-plus benchmark.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Total
Baseline	32.9	50.8	86.3	85.7	86.3	62.0	73.6	66.4
VGGT (cross attention)	45.8	**53.8**	86.8	**97.2**	**93.9**	**65.5**	69.8	**71.1**
VGGT (concat)	41.2	51.2	87.0	93.7	88.4	63.9	70.6	68.9
VGGT (Q-former)	44.3	51.0	87.2	95.1	91.3	62.0	**71.0**	69.6
Qwen-Image-Edit (1 view)	38.5	52.6	87.5	94.9	90.3	62.5	64.6	68.0
Qwen-Image-Edit (2 views)	**46.7**	53.5	**87.6**	96.6	91.8	60.6	69.5	70.2

analysis, augmentation, and model update, using model predictions to guide data collection and labeling, reducing manual cleaning and improving data-policy co-adaptation. Unifying multi-modal sensing—force, touch, temperature—into a single action space enables end-to-end perception-decision-action for active interaction. 3D representation learning can shift from post-hoc injection to intrinsic modeling, learning geometric priors during pre-training via self-supervised depth and pose estimation. True "one-brain, many-forms" intelligence demands generalization to legged systems, drones, and humanoids through architectures that abstract hardware details and learn from universal physical principles.

7.2 Conclusions

We present ABot-M0, a unified framework that jointly optimizes data standardization and architecture to build hardware-agnostic embodied agents. Integrating six major open-source datasets, we build UniACT-dataset, a foundation of 6M+ trajectories, and apply delta actions in the end-effector frame, pad-to-dual-arm modeling, and dual-level reweighting to unify sources and support cross-embodiment transfer. We introduce the *Action Manifold Hypothesis* and *Action Manifold Learning (AML)*, shifting action prediction from denoising to direct generation, improving decoding efficiency and policy robustness. A plug-and-play dual-stream architecture fuses VLM semantics with enhanced 3D perception. Experiments show ABot-M0 achieves state-of-the-art performance on Libero, Libero-Plus, RoboCasa and Robotwin, surpassing pi0 and UniVLA. These results confirms that high-performance embodied intelligence is achievable without proprietary data or custom hardware when public resources are systematically engineered. We open-source all pipelines and code to advance community-driven progress. This effort advocates a new paradigm that embodied intelligence evolving through open collaboration, growing smarter via shared knowledge.

8 Contributions

Author contributions in the following areas are as follows:

- **Data Collection & Analysis:** Yandan Yang, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dekang Qi
- **Data Standardization:** Yandan Yang, Tong lin, Xinyuan Chang
- **Data Pipeline:** Tong Lin, Shuang Zeng, Dongjie Huo
- **Model Architecture:** Shuang Zeng, Junjin Xiao
- **Training:** Shuang Zeng, Tong Lin, Junjin Xiao
- **Evaluation:** Junjin Xiao, Shuang Zeng, Tong Lin
- **Writing:** Yandan Yang, Xinyuan Chang, Shuang Zeng, Tong Lin, Dekang Qi, Junjin Xiao
- **Project Lead:** Xinyuan Chang, Feng Xiong
- **Advisor:** Mu Xu[†], Zhiheng Ma, Xing Wei

[†]Corresponding author: xumu.xm@alibaba-inc.com

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xiong-Hui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Rongyao Fang, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Qidong Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Li Ying Meng, Xuancheng Ren, Xin yi Ren, Sibo Song, Yu-Chen Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yihe Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Botao Zheng, Humen Zhong, Jingren Zhou, Fanxi Zhou, Jingren Zhou, Yuanzhi Zhu, and Keming Zhu. Qwen3-vl technical report. arXiv preprint arXiv:2511.21631, 2025.

[2] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. arXiv preprint arXiv:2512.13030, 2025.

[3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164, 2024.

[4] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. RSS, 2025.

[5] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Jess Moss, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024.

[6] Gunnar Carlsson. Topology and data. Bulletin of the American Mathematical Society, 46(2):255–308, 2009.

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

[8] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-Supervised Learning. The MIT Press, 09 2006. ISBN 9780262033589. doi: 10.7551/mitpress/9780262033589.001.0001. URL https://doi.org/10.7551/mitpress/9780262033589.001.0001.

[9] Chilam Cheang, Sijin Chen, Zhongren Cui, Yingdong Hu, Liqun Huang, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Xiao Ma, et al. Gr-3 technical report. arXiv preprint arXiv:2507.15493, 2025.

[10] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. arXiv preprint arXiv:2506.18088, 2025.

[11] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. arXiv preprint arXiv:2402.10329, 2024.

[12] 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. The International Journal of Robotics Research, 44(10-11):1684–1704, 2025.

[13] AgiBot World Colosseum contributors. Agibot world colosseum. https://github.com/OpenDriveLab/AgiBot-World, 2024.

[14] InternVLA-M1 Contributors. Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy. arXiv preprint arXiv:2510.13778, 2025.

[15] 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. Libero-plus: In-depth robustness analysis of vision-language-action models. arXiv preprint arXiv:2510.13626, 2025.

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

[17] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. *pi_{0.5}*: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054, 2025.

[18] Guanhua Ji, Harsha Polavaram, Lawrence Yunliang Chen, Sandeep Bajamahal, Zehan Ma, Simeon Adebola, Chenfeng Xu, and Ken Goldberg. Oxe-auge: A large-scale robot augmentation of oxe for scaling cross-embodiment policy learning. arXiv preprint arXiv:2512.13100, 2025.

[19] Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, Xiao Liu, Shuiqi Cheng, Jiyang Gao, Huazhe Xu, and Hang Zhao. Galaxea open-world dataset and g0 dual-system vla model. arXiv preprint arXiv:2509.00576, 2025.

[20] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246, 2024.

[21] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. RSS, 2025.

[22] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. arXiv preprint arXiv:2508.07917, 2025.

[23] LejuRobotics. Let:full-size humanoid robot real-world dataset. https://huggingface.co/datasets/LejuRobotics/let_dataset, 2025.

[24] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. arXiv preprint arXiv:2411.19650, 2024.

[25] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. arXiv preprint arXiv:2511.13720, 2025.

[26] Litian Liang, Liuyu Bian, Caiwei Xiao, Jialin Zhang, Linghao Chen, Isabella Liu, Fanbo Xiang, Zhiao Huang, and Hao Su. Robo360: a 3d omnispective multi-material robotic manipulation dataset. arXiv preprint arXiv:2312.06686, 2023.

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

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

[29] Songming Liu, Bangguo Li, Kai Ma, Lingxuan Wu, Hengkai Tan, Xiao Ouyang, Hang Su, and Jun Zhu. Rdt2: Exploring the scaling limit of umi data towards zero-shot cross-embodiment generalization. arXiv preprint arXiv:2602.03310, 2026.

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

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

[32] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In 2024 IEEE International Conference on Robotics and Automation (ICRA), pages 6892–6903. IEEE, 2024.

[33] William Peebles and Saining Xie. Scalable diffusion models with transformers. arXiv preprint arXiv:2212.09748, 2022.

[34] 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. arXiv preprint arXiv:2501.09747, 2025.

[35] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. RSS, 2025.

[36] Sabela Ramos, Sertan Girgin, Léonard Hussenot, Damien Vincent, Hanna Yakubovich, Daniel Toyama, Anita Gergely, Piotr Stanczyk, Raphael Marinier, Jeremiah Harmsen, et al. Rlds: an ecosystem to generate, share and use datasets in reinforcement learning. arXiv preprint arXiv:2111.02767, 2021.

[37] Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies. arXiv preprint arXiv:2509.04996, 2025.

[38] starVLA Contributors. Starvla: A lego-like codebase for vision-language-action model developing. GitHub repository, 1 2025. URL https://github.com/starVLA/starVLA.

[39] Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. arXiv preprint arXiv:2505.17016, 2025.

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

[41] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research, 11(12), 2010.

[42] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In Conference on Robot Learning, pages 1723–1736, 2023.

[43] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2025.

[44] Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, Siteng Huang, Yifan Tang, Wenhui Wang, Ru Zhang, Jianyi Liu, and Donglin Wang. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. AAAI, 2026.

[45] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

[46] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. arXiv preprint arXiv:2412.13877, 2024.

[47] Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, Bowen Yang, Zhe Li, Kai Zhu, Hongyu Wu, Yiheng Liu, et al. Robocoin: An open-sourced bimanual robotic data collection for integrated manipulation. arXiv preprint arXiv:2511.17441, 2025.

[48] Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, Xing Wei, and Ning Guo. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. arXiv preprint arXiv:2505.17685, 2025.

[49] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In CVPR, 2025.

[50] Zhaxizhuom Zhaxizhuoma, Kehui Liu, Chuyue Guan, Zhongjie Jia, Ziniu Wu, Xin Liu, Tianyu Wang, Shuai Liang, Pengan Chen, Pingrui Zhang, et al. Fastumi: A scalable and hardware-independent universal manipulation interface with dataset. In Conference on Robot Learning, pages 3069–3093. PMLR, 2025.

[51] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. ICLR, 2025.

[52] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Conference on Robot Learning, pages 2165–2183. PMLR, 2023.