DECOUPLING PLANNING AND EXECUTION: A CEREBRUM-CEREBELLUM FRAMEWORK FOR ROBUST VISUAL-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual-Language-Action (VLA) models have demonstrated remarkable capabilities in end-to-end robotic manipulation by integrating multimodal information. However, these models frequently degenerate into static vision-action mappings, resulting in a critical failure to adhere to language instructions under complex constraints or target changes. We identify that this limitation stems from an inherent modality imbalance where physical actions correlate more strongly with visual states than with abstract linguistic descriptions. Consequently, models spontaneously develop a "vision-dominated bias" that marginalizes linguistic guidance during training. To address this, we propose a bio-inspired "Cerebrum-Cerebellum" architecture to structurally decouple task planning from execution. The language-dominant Cerebrum Module functions as a high-level planner to parse instructions into structured sub-tasks and tokens with visual information, while the vision-dominant Cerebellum Module executes these plans with real-time feedback. Across LIBERO-Plus and RoboTwin 2.0, our method improves instruction-following robustness, outperforming $\pi_{0.5}$ by 5.28 and 9.20 percentage points, respectively.

1 INTRODUCTION

Visual-Language-Action (VLA) models represent a paradigm shift in robotic autonomy, aiming to achieve end-to-end collaboration by integrating language understanding, visual parsing, and action generation Kim et al. (2024); Zitkovich et al. (2023); Zawalski et al. (2024); Qu et al. (2025); Li et al. (2024b;a; 2025). Driven by large-scale pre-training and cross-modal attention mechanisms, existing VLA models have demonstrated remarkable proficiency in simple visual-action collaboration tasks O'Neill et al. (2024); Khazatsky et al. (2024), demonstrating visual scene adaptation capabilities and action generation efficiency.

However, despite their success in static environments, these models exhibit a critical vulnerability: the failure to follow the language instructions when facing complex language constraints or target changes. Recent research suggests that current VLA models frequently degenerate into pure "Vision-Action" mappings, marginalizing linguistic input Fei et al. (2025b). Experiments on the LIBERO-Plus dataset Fei et al. (2025b) reveal three problems: (1) **Language Insensitivity**: Performance remains unaffected by random language perturbations; (2) **Modality Bias**: In blank instruction tests, models maintain high success rates on short-horizon tasks even without any language input, implying a reliance on visual overfitting; (3) **Adaptation Failure**: In target replacement scenarios, success rates drop to near zero as models stubbornly execute the original visual-motor trajectory, failing to adjust based on the updated language command. Building on these LIBERO-Plus findings, we refer to this tendency of existing VLA models to rely excessively on visual-action correlations as a "vision-dominated bias" Fei et al. (2025b). We assume that the cause of the above limitations is the inherent modality imbalance between vision and language. Since physical actions are naturally more correlated with visual states than with abstract linguistic descriptions, models spontaneously prioritize visual features during "Vision-Action-Language" triplet training. In practice, this means that the policy can often recover a plausible action sequence from vision alone, while language is treated as weak context and becomes easy to ignore when visual cues are salient. Consequently, the visual feature weight w_v significantly outweighs the language feature weight w_l

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

($w_v \gg w_l$), creating a "vision-dominated bias" where task constraints embedded in language are overshadowed by visual cues. Existing solutions primarily focus on enhancing feature alignment through data scaling or auxiliary reasoning tasks Zhou et al. (2025); Yang et al. (2025a); Zawalski et al. (2024); Sun et al. (2025). While these methods improve generalizability, they fail to address the architectural root of the "vision-dominated" bias. Mere feature-level optimization is insufficient to prevent visual information from suppressing language constraints in complex scenarios.

To bridge the gap between the visual feature and the language feature, this paper draws inspiration from the biological functional division of the central nervous system and proposes a "Cerebrum-Cerebellum Collaboration" architecture. This framework decouples task planning from action execution. The **Cerebrum Module** (language-dominant) acts as the planner. It parses abstract language instructions into structured sub-tasks with explicit visual constraints (e.g., target coordinates, interaction modes). The **Cerebellum Module** (vision-dominant) acts as the executor. It generates precise motor actions based on the Cerebrum's directives and real-time visual feedback. Furthermore, we establish a closed-loop mechanism where execution anomalies are fed back to the Cerebrum for re-planning. Experiments on LIBERO-Plus and RoboTwin 2.0 further demonstrate the robustness of this decoupled approach. On LIBERO-Plus, it improves the success rate by 5.28 percentage points under various dimensional perturbations, while on RoboTwin 2.0 it achieves 83.60% on Clean, 51.50% on Randomized, and 67.55% overall.

2 Related Work

2.1 Development of Visual-Language-Action (VLA) Models

Benefiting from the breakthroughs of pre-trained Visual-Language Models (VLMs) Beyer et al. (2024); Lu et al. (2024), Visual-Language-Action (VLA) models have become a central paradigm for autonomous embodied agents, generating executable actions under action supervision Brohan et al. (2022); Ding et al. (2024). RoboFlamingo Li et al. (2023) models temporal history with an explicit policy head to improve action coherence; OpenVLA Kim et al. (2024), the first open-source VLA model pre-trained on large-scale robot datasets O'Neill et al. (2024), provides a generalist backbone for diverse manipulation tasks. VLAS Zhao et al. (2025) extends the modality set with audio for richer multimodal perception, while UniVLA Bu et al. (2025) learns latent action representations from web videos to enhance downstream adaptability. ReconVLA Song et al. (2025) further introduces an implicit grounding paradigm by reconstructing gaze regions via a diffusion transformer, sharpening fine-grained visual representations. However, most of these methods still rely on monolithic, action-supervised architectures that easily overfit visual correlations and remain insensitive to linguistic variations or target changes. Our work instead explicitly re-balances the roles of language and vision, aiming to improve instruction sensitivity and robustness under changing goals and environments.

2.2 Hierarchical Decision-Making Driven Robot Policies

Multi-task decision-making and planning in complex environments remains a core challenge for VLA deployment. Researchers have leveraged the reasoning capabilities of pre-trained VLMs and LLMs to propose various hierarchical schemes: SayCan Ahn et al. (2022) uses a frozen LLM for subtask recognition; RT-H Belkhale et al. (2024) constructs language-guidable models to connect high-level decision-making with low-level actions; and RT-Trajectory Gu et al. (2023) links planning and action generation through intermediate spatial representations. To bridge the interface divergence between reasoning and execution, InstructVLA Yang et al. (2025b) proposes VLA Instruction Tuning (VLA-IT). It employs a Mixture-of-Experts (MoE) framework to unify autoregressive textual reasoning with action generation, effectively leveraging VLM knowledge to steer robotic control without catastrophic forgetting. Integrating high-level language planning with high-frequency action control remains challenging because frequent planning adds overhead while monolithic policies may underuse language. Our framework couples a language-dominant Cerebrum planner with a vision-dominant Cerebellum executor through language-grounded visual tokens. The Cerebrum is invoked initially and when execution feedback triggers replanning, while the Cerebellum handles high-frequency control, limiting planning overhead while preserving language guidance.

2.3 Application of Latent Space Reasoning in Cross-Modal Models

Recent studies indicate that reasoning in the latent space can significantly improve the performance of large models in complex tasks. Co-conut Hao et al. (2024) suggests that continuous latent embeddings are superior to explicit text-based Chain of Thought (CoT); CCoT Cheng & Van Durme (2024) compresses CoT into continuous tokens to increase reasoning density. This concept has been extended to VLMs to bridge the gap between semantic reasoning and perceptual grounding. Chain-of-Visual-Thought (CoVT) Qin et al. (2025) introduces continuous visual tokens that encode dense perceptual cues like depth and segmentation, allowing VLMs to think in a continuous visual space rather than just a discrete linguistic one. Similarly, Aurora Bigverdi et al. (2025) and Mirage Yang et al. (2025c) utilize latent representations for depth estimation and visual imagination. While these methods optimize reasoning efficiency or visual perception, they are not tailored for the unique language-action collaboration bottleneck in VLA models. The continuous visual token generation strategy in this paper converts language instructions into vision-aligned representations. By combining this with the dynamic reasoning mechanism of the cerebrum and cerebellum, we strengthen the language-vision-action association, ultimately improving the model's adaptability to complex language variations and fine-grained manipulation requirements.

3 Method

3.1 Method Overview

In this work, we try to address the vision-dominated bias in end-to-end VLA models by explicitly separating *language-driven task planning* from *vision-driven motor execution*. As illustrated in Figure 1, our framework consists of two modules. The Cerebrum Module is a low-frequency planner that parses natural language instructions into structured plans augmented with visual tokens from both segmentation and depth maps. The Cerebellum Module is a high-frequency executor that generates precise actions via conditional flow matching while monitoring execution progress.

In the following sections, we first detail the Cerebrum's mechanism for generating vision-grounded plans (Sec 3.2). We then describe the Cerebellum's flow-matching policy and its alignment objectives (Sec 3.3). Finally, we present the closed-loop inference pipeline enabled by a value-driven replanning mechanism (Sec 3.4).

3.2 The Cerebrum Module

Existing VLA models often fail to adhere to complex constraints because the linguistic constraints are drowned out by visual features during fusion. The role of Cerebrum Module is to act as a *high-level reasoner* that enforces linguistic authority. However, pure textual planning lacks spatial precision for robotic manipulation. Therefore, we design the module to output a **structured plan** that interleaves textual sub-tasks with **visual tokens**, explicitly grounding semantic concepts into the physical observation space.

Planning interface. The Cerebrum Module is instantiated by Qwen-VL 7B. Given a third-person RGB image I^{3rd} and an instruction x^{lang}, it generates a structured plan $y = f_{\text{Cer}}(x^{\text{lang}}, I^{\text{3rd}})$ as a sequence of parameterized manipulation primitives,

$$y = \Big[\texttt{robot}\langle\texttt{robot_seg}\rangle\,;\,\big[(a_i, \mathcal{A}_i, \mathcal{V}_i)\big]_{i=1}^{N}\,;\,\texttt{depth}\langle\texttt{depth}\rangle\Big], \tag{1}$$

where a_i denotes the action type, $\mathcal{A}_i$ contains its symbolic or numerical arguments, and $\mathcal{V}_i$ contains the action-specific visual tokens. Thus, every plan begins with the robot token, followed by one or more parameterized manipulation primitives, and ends with a global depth token. This interface supports pick-and-place, knob rotation, drawer opening/closing, and other fine-grained interactive motions. We first illustrate a representative template for classic pick-and-place manipulation: $y =$ `robot` ⟨`robot_seg`⟩;`pick obj_a` ⟨`obj_a_seg`⟩;`place in/on obj_b` ⟨`obj_b_seg`⟩; `depth` ⟨`depth`⟩.

where the tokens <robot_seg>, <obj_a_seg>, <obj_b_seg>, <depth> are special learnable embeddings representing the robot mask, pick-target mask, place-target mask, and a depth map of

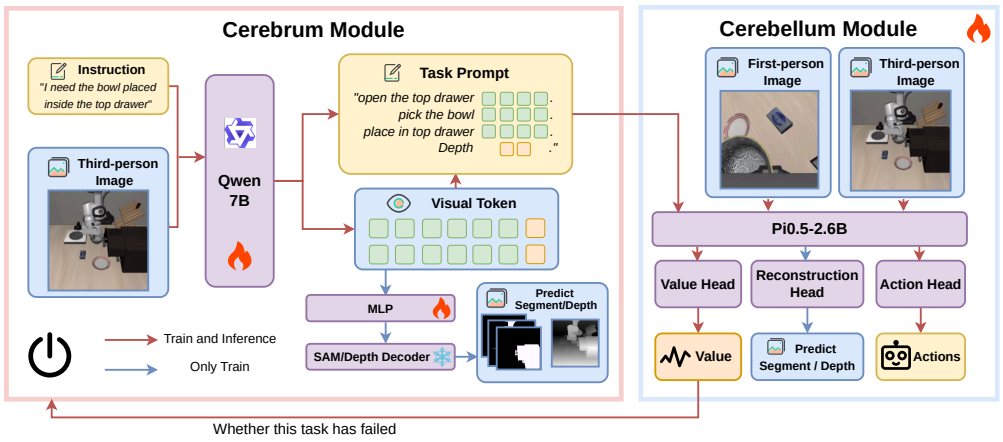


Figure 1: Overview of the Cerebrum-Cerebellum framework. The Cerebrum generates language-guided plans with visual tokens, while the Cerebellum executes them with visual feedback. Red arrows denote paths used during training and inference, whereas blue arrows denote training-only paths; replanning is triggered at initialization or when the value signal drops.

the scene, respectively. We denote the final-layer embedding of them as $h_{\text{robot}}, h_{\text{obj_a}}, h_{\text{obj_b}}, h_{\text{depth}} \in \mathbb{R}^{d_{\text{LLM}}}$.

Other primitives use action-specific arguments and visual tokens. For example, drawer interaction is represented as

$$\texttt{robot}\ \langle\texttt{robot_seg}\rangle; \texttt{open/close}\ \langle\texttt{drawer_seg}\rangle\ \langle\texttt{handle_seg}\rangle; \texttt{depth}\ \langle\texttt{depth}\rangle.$$

Thus, the action type and its arguments specify the interaction semantics, whereas the visual tokens ground the relevant object, handle, robot, and spatial geometry in the current observation.

Grounding visual tokens via frozen decoders. To ensure above tokens carry valid geometric information, we map them to the latent space of pre-trained vision experts. We employ the frozen decoder of both the Segment Anything Model (SAM)Kirillov et al. (2023) and the Depth Anything ModelYang et al. (2024).

For the segmentation tokens, we allocate one token for each segmented entity in the scene (the robot and every object instance), so that the model can reason about them individually. We then use a trainable linear projection P_{seg} to map each token embedding h to a SAM-compatible prompt embedding, i.e., $z_{\text{seg}} = P_{\text{seg}}(h)$. The frozen SAM decoder then generates a binary mask $\hat{M} \in \{0,1\}^{H\times W}$ conditioned on the image features.

In contrast, for depth we maintain a single global depth token h_{depth} that corresponds to the entire image rather than per-object depth tokens. We similarly use a linear projection P_{dep} to map h_{depth} to a latent code z_{dep}, which is then decoded into a dense depth map $\hat{D} \in \mathbb{R}^{H\times W}$. These frozen-decoder reconstruction paths serve as an explicit embedding-space alignment, forcing the learned visual-token embeddings to lie on the manifolds expected by the pre-trained vision experts.

Training objective. The Cerebrum is trained to minimize a joint loss of autoregressive language modeling and visual reconstruction:

$$\mathcal{L}_{\text{Cer}} = \mathcal{L}_{\text{lm}} + \lambda_{\text{seg}} \sum_{k\in\mathcal{K}} \ell_{\text{seg}}(\hat{M}_k, M^{\text{gt}}_{\pi(k)}) + \lambda_{\text{dep}} \left\| \text{norm}(\hat{D}) - \text{norm}(D^{\text{gt}}) \right\|_2^2, \tag{2}$$

where $\mathcal{L}_{\text{lm}}$ is the standard autoregressive language modeling loss over the generated plan sequence y, and $\mathcal{K}$ indexes the robot, pick-target, and place-target masks. The permutation π is obtained by Hungarian matching between predicted and ground-truth instances, and ℓ_{seg} combines pixel-wise cross-entropy and soft IoU. The depth term uses the squared L_2 error after per-image depth normalization.

This multi-task objective has two advantages: (i) the language modeling term keeps the Cerebrum faithful to the instruction and the fixed plan template; (ii) the reconstruction terms force the plan's visual tokens to carry precise, human-interpretable geometric cues (masks/depth), which reduces spurious vision-action shortcuts and makes the planner's constraints verifiable.

3.3 THE CEREBELLUM MODULE

While the Cerebrum ensures semantic correctness, open-loop execution is brittle to dynamic disturbances. The Cerebellum is motivated by the need for a robust, high-frequency controller that can adjust to real-time visual feedback while strictly adhering to the Cerebrum's plan y. A critical challenge here is preventing the policy from ignoring the plan y. Following the $\pi_{0.5}$-2.6B architecture, the Cerebellum uses a conditional flow-matching Action Head that takes the upstream VLM key-value cache H_t^{KV} as input and generates continuous actions.

Reconstruction Head for Alignment. To solve the instruction-ignoring problem, we introduce a novel auxiliary task: **Visual Token Reconstruction**. During training, we add a reconstruction head that takes the Cerebellum's fused features h_t and attempts to recover the original masks and depth maps encoded in the plan y. Specifically, conditioned on h_t, this head predicts latent codes $\tilde{z}_{\text{robot}}, \tilde{z}_{\text{obj_a}}, \tilde{z}_{\text{obj_b}}, \tilde{z}_{\text{dep}}$. These latent codes are then fed into the same frozen SAM and depth decoders used by the Cerebrum, yielding reconstructed masks and depth. And the objective $\mathcal{L}_{rec}$ forces the Cerebellum's internal representations to contain the geometric information specified by the Cerebrum, thereby mathematically enforcing that the execution policy is conditioned on the plan.

Value Head: progress monitoring. The value head predicts task progress from the Cerebellum's fused representation h_t, which incorporates proprioception, visual observations, and the latest Cerebrum plan y. We use normalized progress along successful demonstrations as a dense surrogate for task value:

$$\mathcal{L}_{\text{val}} = \mathbb{E}_{\tau \sim \mathcal{D}_{\text{succ}}} \mathbb{E}_{t \sim \mathcal{U}\{1,\ldots,T_\tau\}} \left[V_\theta(h_t) - \frac{t}{T_\tau} \right]^2, \tag{3}$$

where $\mathcal{D}_{\text{succ}}$ is the set of successful demonstrations, τ is a demonstration of length T_τ, and t is a uniformly sampled timestep. V_θ denotes the value head with parameters θ, and t/T_τ is its normalized progress target. During execution, a decrease in predicted progress exceeding 0.3 over a 50-frame temporal window triggers replanning.

3.4 CEREBRUM–CEREBELLUM CLOSED LOOP

Our framework is trained in two successive stages rather than by joint end-to-end optimization. In the first stage, we train the Cerebrum Module independently using the ground-truth structured plan. The language-modeling and visual reconstruction objectives supervise both the symbolic manipulation primitives and their associated segmentation/depth tokens. After this stage, the Cerebrum parameters are frozen. In the second stage, we train the Cerebellum Module with plans generated by the frozen Cerebrum, rather than with ground-truth plans. The generated plan is passed to the Cerebellum as continuous visual-token embeddings together with the textual and action-related plan representations; it is not serialized and re-encoded as text. All parameters of the $\text{Pi}_{0.5}$ backbone used by the Cerebellum are updated during this stage, along with the action, reconstruction, and value heads. Since the Cerebrum is frozen, no gradient is propagated from the Cerebellum back into the planner.

At inference time, the two modules form a sparse closed loop. Given the language instruction and the initial third-person observation, the Cerebrum produces an initial plan $\hat{y}_0$ and its continuous visual-token embeddings. At each control step t, the Cerebellum receives the current observation o_t, proprioceptive inputs, and the latest plan representation, and predicts an action a_t together with a progress value $V_\theta(h_t)$. The Cerebellum therefore returns an execution-status signal to the planner through the value estimate, while the current observation remains available for replanning. If the predicted progress decreases by more than 0.3 within a 50-frame temporal window, we regard the execution as anomalous and send the latest third-person observation and the anomaly signal back to the Cerebrum. The Cerebrum then generates an updated plan and continuous visual-token embeddings, which replace the previous plan in the Cerebellum. Otherwise, the Cerebellum continues high-frequency execution without invoking the Cerebrum. This feedback loop allows the Cerebrum

to remain a low-frequency language-driven planner while the Cerebellum handles reactive motor control.

4 EXPERIMENTS

4.1 DATASETS AND METRICS

We conduct experiments on three main benchmarks: LIBERO-Plus Fei et al. (2025b), the original LIBERO benchmark Liu et al. (2023), and RoboTwin 2.0 RoboTwin Team (2026). We adopt task success rate (%) as the primary evaluation metric, measuring the proportion of correctly completed tasks. An additional Target Replacement Evaluation is provided in the appendix to assess sensitivity to changed language targets.

LIBERO-Plus. LIBERO-Plus is a comprehensive extension of LIBERO with 10,030 diverse tasks, designed to evaluate VLA models' robustness under controlled distribution shifts. It introduces perturbations along seven dimensions, each targeting a different source of failure for embodied agents: Camera (viewpoint changes), Robot (initial state variations), Language (instruction paraphrases and semantic variations), Light/Background/Noise (visual appearance shifts), and Layout (object placement and distractors). For clarity and consistency, we follow the reporting order in Table 1: Camera, Robot, Language, Light, Background, Noise, and Layout. Unless otherwise stated, we follow the official LIBERO-Plus protocol for training and evaluation.

LIBERO. On the original LIBERO benchmark, tasks are organized into four categories according to task type: spatial, object, goal, and long-horizon. For each category, we select 10 representative tasks, resulting in 40 tasks in total, and perform 50 evaluation rollouts per task. This setting provides a complementary assessment of our model's generalization and robustness on the canonical LIBERO suites.

RoboTwin 2.0. We further evaluate the proposed method on RoboTwin 2.0, a simulated bimanual manipulation benchmark with 50 tasks. We report the clean and randomized settings using the benchmark's clean2clean and clean2random protocols, respectively. Our evaluation contains 1,000 trials in each setting, giving 2,000 trials in total. For comparison, we use the publicly reported results for $\pi_{0.5}$, π_0, starVLA, and ACT from the RoboTwin leaderboard RoboTwin Team (2026). The overall score is the arithmetic mean of the two setting-level success rates.

4.2 COMPARISONS WITH THE SOTA

We compare our method with representative VLA baselines on LIBERO-Plus, including OpenVLA-OFT, MergeVLA, SRPO, and $\pi_{0.5}$. Table 1 shows that our approach achieves the best overall robustness, reaching a total success rate of 89.72% and improving upon the strongest baseline $\pi_{0.5}$ (84.44%) by 5.28 percentage points.

Importantly, the improvements are not confined to a single perturbation type as shown in Table 1. Instead, our method performs strongly across language, physical-state, and appearance shifts, which is consistent with our goal of counteracting the "vision-dominated" shortcut at the architectural level rather than relying on narrow data augmentation.

Language robustness. The largest gain appears in the language dimension (96.31%), substantially outperforming $\pi_{0.5}$ (82.23%) and OpenVLA-OFT (86.47%). This dimension directly tests whether the policy actually conditions on instructions (e.g., paraphrases or richer semantics) instead of collapsing into a pure vision-action mapping. We attribute the improvement to two coupled design choices: (i) *planning–execution decoupling*, which prevents high-frequency execution features from overwhelming language; and (ii) the Cerebrum's *visual token prompts*, which translate the instruction into explicit, verifiable geometric constraints (masks/depth). During training, the executor is additionally regularized to preserve these constraints (via visual-token reconstruction), making it harder for the policy to ignore the plan.

Physical perturbations. Our model is also more stable under robot perturbations (63.20%) and layout shifts (89.94%). These settings require the policy to adapt when the initial state or object arrangement deviates from the nominal demonstrations. Qualitatively, decoupling helps because the Cerebellum can focus on short-horizon control under the current observation, while the Cerebrum

Table 1: **Results of different models on LIBERO-Plus benchmarks.** Values represent success rates (%). **Bold** indicates the best result and underline indicates the second best. "Total" is computed as a task-count-weighted aggregate over all perturbation dimensions, rather than their arithmetic mean.

VLA Method	Camera	Robot	Language	Light	Background	Noise	Layout	Total
OpenVLA-OFTKim et al. (2025)	92.97	30.43	86.47	94.75	94.17	89.42	78.70	79.75
MergeVLAFu et al. (2025)	71.0	51.3	78.5	**99.0**	95.5	85.1	79.5	80.0
SRPOFei et al. (2025a)	83.4	62.0	73.6	97.2	**97.7**	85.7	75.2	82.1
π_0Black et al. (2024)	87.75	42.80	84.95	94.90	94.65	92.30	83.20	82.94
$\pi_{0.5}$Intelligence et al. (2025)	92.59	52.06	82.23	95.18	95.47	94.22	86.08	84.44
ours	**93.78**	**63.20**	**96.31**	97.69	96.11	**95.61**	**89.94**	**89.72**

Table 2: Results of different models on the original LIBERO benchmark.Values represent success rates (%). **Bold** indicates the best result and underline indicates the second best.

Method	Libero Spatial	Libero Object	Libero Goal	Libero Long	Average
Diffusion PolicyChi et al. (2025)	78.3	92.5	68.3	50.5	72.4
OctoTeam et al. (2024)	78.9	85.7	84.6	51.1	75.1
TraceVLA Zheng et al. (2024)	84.6	85.2	75.1	54.1	74.8
OpenVLA Kim et al. (2024)	84.7	88.4	79.2	53.7	76.5
SpatialVLA Qu et al. (2025)	88.2	89.9	78.6	55.5	78.1
π_0-FASTPertsch et al. (2025)	96.4	96.8	88.6	60.2	85.5
π_0Black et al. (2024)	96.8	**98.8**	95.8	85.2	94.2
UniVLABu et al. (2025)	96.5	96.8	95.6	92.0	95.2
$\pi_{0.5}$Intelligence et al. (2025)	**98.8**	98.2	98.0	92.4	96.9
OpenVLA-OFTKim et al. (2025)	97.6	98.4	97.9	94.5	97.1
Ours	**98.8**	98.7	**98.2**	**95.5**	**97.8**

provides a consistent task specification. Moreover, when the execution starts drifting (e.g., due to compounding errors), the value-driven replanning mechanism offers an explicit correction pathway by refreshing the plan from the current third-person observation.

Visual perturbations. On **Light/Background/Noise**, our performance remains comparable to or better than strong baselines. This suggests that strengthening language grounding does not require sacrificing visual generalization. In other words, our method improves instruction sensitivity *without* making the model brittle to appearance changes.

To confirm that our method's advantages are not limited to perturbed scenarios, we additionally evaluate all baselines on the **original LIBERO** benchmark. Notably, our model exhibits exceptional competitiveness in LIBERO long-horizon tasks, achieving a state-of-the-art success rate of 95.5% — this outperforms the second-best method OpenVLA-OFT. As shown in Table 2, our model also attains the best-leading average success rate of 97.8% across all task categories, while obtaining optimal or suboptimal results on individual spatial, object and goal tasks. These results collectively verify that our decoupled architecture delivers superior performance in standard, perturbed, and long-horizon task settings alike, confirming its strong generalizability rather than overfitting to distribution shifts. We note that the differences among top-performing methods on unperturbed LIBERO are small, and Table 2 contains marginal gaps at the average score. Therefore, the main takeaway of our method is its prominent robustness improvements on LIBERO-Plus, while maintaining the top-tier and competitive nominal performance on original standard tasks.

On **RoboTwin 2.0**, Our method achieves 83.60% on Clean and 51.50% on Randomized, corresponding to 836/1,000 and 515/1,000 successful trials, respectively. The combined success rate is 1,351/2,000 = 67.55%. This outperforms $\pi_{0.5}$ by 9.20 percentage points overall and retains a 5.50-point advantage under Randomized evaluation, indicating that the proposed planning–execution decoupling improves robustness to visual and state variation.

The **Target Replacement Evaluation** is summarized in Appendix A. Our method achieves 74.0% success, showing improved sensitivity to changed language targets.

4.3 Ablation Study

To evaluate the contribution of each core component in our framework, we perform a comprehensive ablation study on the LIBERO-Plus benchmarks, including four task suites (Object, Goal, Spatial, Long) and the average performance. As presented in Table 4, we systematically ablate four key

Table 3: RoboTwin 2.0 results. Clean and Randomized denote clean2clean and clean2random evaluation, respectively. Values are success rates (%). Baseline results are taken from the public RoboTwin 2.0 leaderboard RoboTwin Team (2026).

Method	Clean	Randomized	Overall
ACTZhao et al. (2023)	29.74	1.74	15.74
starVLAStarVLA Community (2026)	46.52	3.16	24.84
π_0Black et al. (2024)	46.42	16.34	31.38
$\pi_{0.5}$Intelligence et al. (2025)	70.70	46.00	58.35
Ours	**83.60**	**51.50**	**67.55**

Table 4: **Ablation study** results validating the effectiveness of each component (Decomposition, Visual Token, Reconstruction Head, and Value Head) on LIBERO-Plus benchmarks.

Ablation Module				Benchmark (LIBERO-Plus)				
Decomposition	Visual Token	Reconstruction Head	Value Head & Replanning	Object	Goal	Spatial	Long	Average
×	×	×	×	85.90	81.52	90.09	80.23	84.44
✓	×	×	×	87.26	84.49	91.53	81.50	86.20
×	×	✓	×	87.04	85.62	92.68	80.82	86.54
×	✓	✓	×	88.24	88.02	93.21	80.44	87.48
×	×	×	✓	86.20	81.66	91.62	83.30	85.70
✓	✓	✓	✓	89.99	90.43	94.42	84.04	89.72

modules: Decomposition, Visual Token, Reconstruction Head, and Value Head & Replanning, to verify their individual effectiveness and synergies.

(1) Decomposition Module. Starting with the full ablation baseline (all modules disabled, 84.44%), we first activate only the Decomposition module. This brings a clear performance improvement, with the average success rate reaching 86.20%. The decomposition module standardizes unstructured natural language instructions into structured, step-wise task representations, reducing linguistic ambiguity and bringing stable gains across all task scenarios, especially for spatial and object-centric tasks.

(2) Reconstruction Head. On the baseline model, we individually enable the Reconstruction Head, which boosts the average score to 86.54%. The reconstruction objective optimizes visual feature learning by enforcing the model to capture fine-grained visual details of target objects and scene layouts. This module effectively enhances the model's basic visual perception ability, contributing to consistent performance improvements on both Spatial and Goal task benchmarks.

(3) Visual Token + Reconstruction Head. We further combine Visual Token with Reconstruction Head while keeping other modules disabled, achieving an average success rate of 87.48%. This combination significantly strengthens the model's attention to task-related target objects and enhances semantic-visual grounding. As a result, the performance on the LIBERO-Goal benchmark is remarkably improved, demonstrating that visual token alignment and reconstruction constraints effectively anchor goal entities to visual cues, which is critical for goal-oriented manipulation tasks.

(4) Value Head & Replanning. We then individually introduce Value Head & Replanning into the baseline model, yielding an average performance of 85.70%. This module achieves the most significant improvement on the LIBERO-Long benchmark. The model monitors task progress and triggers replanning when predicted progress drops, providing a correction mechanism for execution drift in long-horizon tasks.

(5) Full model. When all four modules are integrated together, our full model achieves the best average performance of 89.72%, surpassing all ablated variants. The consistent performance gains across all benchmarks validate the strong synergies between semantic decomposition, visual grounding, and decision optimization. Each component plays an irreplaceable role in improving the robustness and success rate of robotic manipulation tasks.

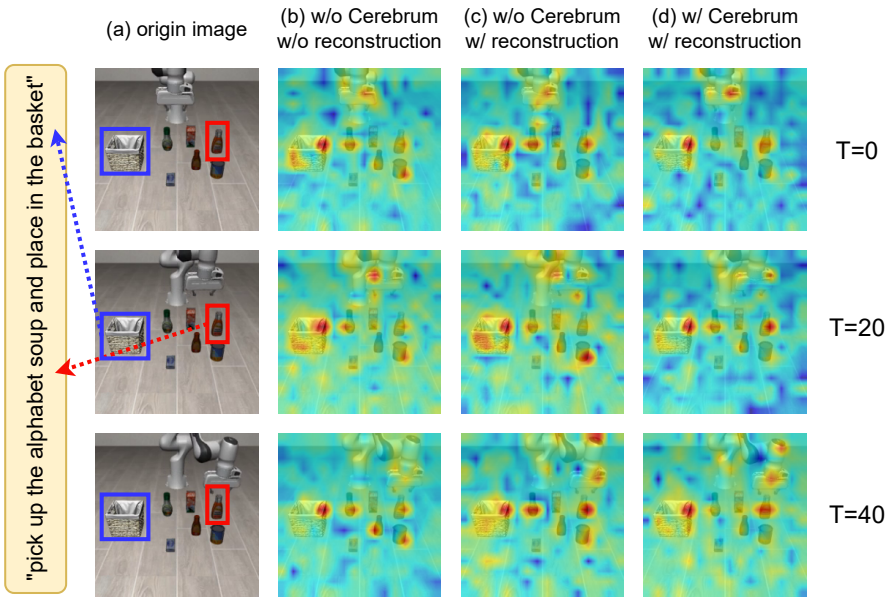


Figure 2: Attention visualization of different model variants on the task "pick up the alphabet soup and place in the basket" at time steps T=0, T=20, and T=40. (a) Original scene image; (b) Without Cerebrum and reconstruction; (c) Without Cerebrum but with reconstruction; (d) With both Cerebrum and reconstruction. The visualization clearly shows that the Cerebrum module provides critical semantic-visual constraints to guide attention toward the target and mechanical arm.

4.4 In-Depth Analysis

Attention Visualization. Figure 2 provides a qualitative comparison of attention maps for several ablated variants on the task "pick up the alphabet soup and place in the basket" at T=0/20/40. In the weakest variant (Fig. 2(b)), attention is diffuse over many objects, which is consistent with the failure mode discussed above that the policy can over-rely on visual shortcuts and under-utilize instruction-related signals.

When we gradually add components, the attention maps become increasingly concentrated on the target object and the relevant robot regions. Specifically, enabling the Cerebellum-side visual-token reconstruction objective (Fig. 2(c)) already reduces attention diffusion compared to the weakest variant, while adding the full Cerebrum plan with visual-token prompts (Fig. 2(d)) further sharpens attention on instruction-relevant entities. We stress that these visualizations are *interpretability evidence* rather than standalone proof; nevertheless, they are aligned with the quantitative improvements on the language perturbation axis in Table 1 and with the gains from adding visual-token alignment in Table 4. Taken together, the results suggest that grounding the plan into explicit segmentation/depth cues helps the execution policy focus on instruction-relevant entities instead of drifting to nearby distractors.

Additional **Success-Speed analysis** is provided in Appendix B, and the standalone **real-world generalization experiment** is provided in Appendix C.

5 Conclusion

In conclusion, we propose a Cerebrum-Cerebellum architecture that decouples language-driven planning from vision-based action execution and uses progress-triggered replanning to improve instruction following under complex manipulation conditions. Experiments on LIBERO-Plus, original LIBERO, RoboTwin 2.0, and real-world tasks demonstrate improved robustness under the evaluated settings while maintaining competitive nominal performance.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Mahtab Bigverdi, Zelun Luo, Cheng-Yu Hsieh, Ethan Shen, Dongping Chen, Linda G Shapiro, and Ranjay Krishna. Perception tokens enhance visual reasoning in multimodal language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3836–3845, 2025.

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.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

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. *arXiv preprint arXiv:2505.06111*, 2025.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*, 2024.

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.

Pengxiang Ding, Han Zhao, Wenjie Zhang, Wenxuan Song, Min Zhang, Siteng Huang, Ningxi Yang, and Donglin Wang. Quar-vla: Vision-language-action model for quadruped robots. In *European Conference on Computer Vision*, pp. 352–367. Springer, 2024.

Senyu Fei, Siyin Wang, Li Ji, Ao Li, Shiduo Zhang, Liming Liu, Jinlong Hou, Jingjing Gong, Xianzhong Zhao, and Xipeng Qiu. Srpo: Self-referential policy optimization for vision-language-action models. *arXiv preprint arXiv:2511.15605*, 2025a.

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

Yuxia Fu, Zhizhen Zhang, Yuqi Zhang, Zijian Wang, Zi Huang, and Yadan Luo. Mergevla: Cross-skill model merging toward a generalist vision-language-action agent. *arXiv preprint arXiv:2511.18810*, 2025.

Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *arXiv preprint arXiv:2311.01977*, 2023.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

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.

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

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Hao Li, Shuai Yang, Yilun Chen, Yang Tian, Xiaoda Yang, Xinyi Chen, Hanqing Wang, Tai Wang, Feng Zhao, Dahua Lin, et al. Cronusvla: Transferring latent motion across time for multi-frame prediction in manipulation. *arXiv preprint arXiv:2506.19816*, 2025.

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*, 2024a.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023.

Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024b.

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

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.

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)*, pp. 6892–6903. IEEE, 2024.

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.

Yiming Qin, Bomin Wei, Jiaxin Ge, Konstantinos Kallidromitis, Stephanie Fu, Trevor Darrell, and XuDong Wang. Chain-of-visual-thought: Teaching vlms to see and think better with continuous visual tokens. *arXiv preprint arXiv:2511.19418*, 2025.

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. *arXiv preprint arXiv:2501.15830*, 2025.

RoboTwin Team. Robotwin 2.0 leaderboard. RoboTwin Platform, September 2026. URL https://robotwin-platform.github.io/leaderboard. Accessed 2026-09-20.

Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. Reconvla: Reconstructive vision-language-action model as effective robot perceiver. *arXiv preprint arXiv:2508.10333*, 2025.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing, 2026. URL https://arxiv.org/abs/2604.05014.

Qi Sun, Pengfei Hong, Tej Deep Pala, Vernon Toh, U-Xuan Tan, Deepanway Ghosal, and Soujanya Poria. Emma-x: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14199–14214, 2025.

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.

Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal ai agents. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14203–14214, 2025a.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10371–10381, 2024.

Shuai Yang, Hao Li, Yilun Chen, Bin Wang, Yang Tian, Tai Wang, Hanqing Wang, Feng Zhao, Yiyi Liao, and Jiangmiao Pang. Instructvla: Vision-language-action instruction tuning from understanding to manipulation. *arXiv preprint arXiv:2507.17520*, 2025b.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. *arXiv preprint arXiv:2506.17218*, 2025c.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

Wei Zhao, Pengxiang Ding, Min Zhang, Zhefei Gong, Shuanghao Bai, Han Zhao, and Donglin Wang. Vlas: Vision-language-action model with speech instructions for customized robot manipulation. *arXiv preprint arXiv:2502.13508*, 2025.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Yaxin Peng, Chaomin Shen, Feifei Feng, et al. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5377–5395, 2025.

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*, pp. 2165–2183. PMLR, 2023.

A TARGET REPLACEMENT EVALUATION

We construct this evaluation from LIBERO by changing the manipulated objects and placement targets specified in the instructions while keeping the initial scene layouts unchanged. The training data comprise 10 tasks with 50 expert trajectories per task. This evaluation tests whether policies follow the updated language targets rather than relying on fixed visual cues.

Table 5: Results of the Target Replacement Evaluation. Values represent success rates (%)

π_0Black et al. (2024)	OpenVLA Kim et al. (2024)	UniVLABu et al. (2025)	$\pi_{0.5}$Intelligence et al. (2025)	Ours
6.0	8.5	21.0	55.0	**74.0**

Table 5 presents the results of the Target Replacement Evaluation. Vanilla VLA models (π_0, OpenVLA, UniVLA) achieve 6.0%, 8.5%, and 21.0%, respectively, while $\pi_{0.5}$ reaches 55.0%. Our method achieves 74.0%, indicating improved adaptation to changed language targets.

B ADDITIONAL IN-DEPTH ANALYSIS

Figure 3 summarizes the trade-off between average success rate and inference frequency (actions/s, excluding first-frame latency) on the LIBERO benchmark, with bubble size indicating model scale. Our design decouples high-frequency execution from low-frequency planning: the Cerebellum runs in the main control loop, while the Cerebrum is invoked sparsely for initial planning and value-triggered replanning. This separation amortizes planning overhead over an episode and preserves the executor's fast inference path.

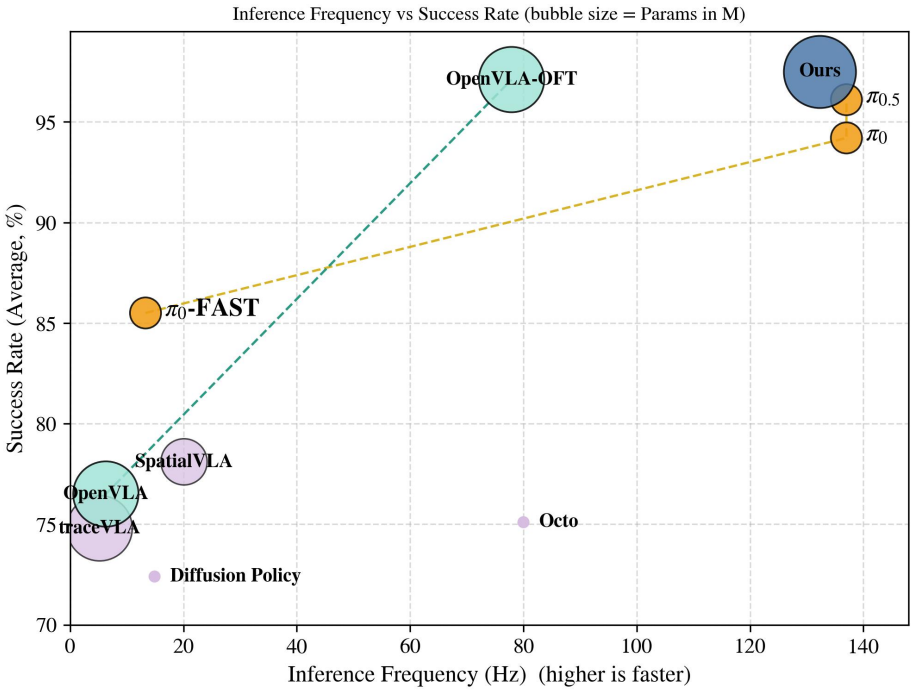


Figure 3: Inference Frequency vs. Average Success Rate on the Libero Benchmark, where bubble size denotes the number of model parameters (in billions). The inference frequency is measured as actions per second, excluding the initial frame's startup latency.

C Real-World Generalization

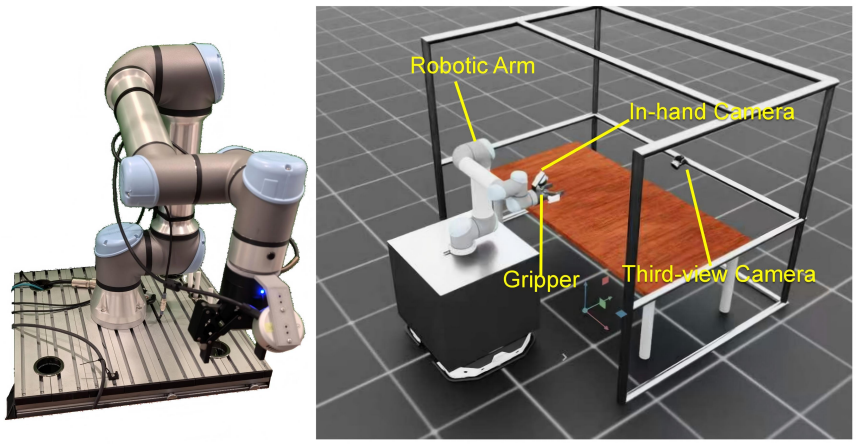


Figure 4: Real-world experimental setup. Left: a UR5E robotic arm equipped with a Robotiq-85 gripper and an in-hand Intel L515 camera. Right: the complete setup with both wrist-mounted and third-person cameras.

Setup. We evaluate the proposed method on a 6-DoF UR5E robotic arm equipped with a Robotiq-85 gripper. An Intel L515 camera is mounted on the robot wrist to provide an egocentric observation, while a second camera provides a third-person view. The experimental setup is shown in Figure 4.

Tasks and evaluation protocol. We consider six representative manipulation tasks: Push in the bottom drawer, Pick cube and place in the top drawer, Pick cube and place in the left caddy, Pick cup and place on the cabinet, Push in the top drawer, and Pick sponge and place on the plate. For training, we collect an average of 30 demonstration trajectories for each task. For evaluation, each task is tested for 10 trials under the same out-of-distribution (OOD) conditions for both $\pi_{0.5}$ and our method. These conditions jointly vary the task description, object placement, and visual background.

Results. Table 6 reports the success rates under the aforementioned OOD conditions. Our method achieves an average success rate of 80.0% across the six tasks, compared with 58.3% for $\pi_{0.5}$. The largest improvements are observed on the tasks "Pick cube and place in the top drawer," "Pick cup and place on the cabinet," and "Push in the top drawer." These results suggest that the proposed planning and execution decoupling can improve instruction following and robustness under the evaluated real-world distribution shifts. The improvement is not uniform across all tasks: on "Push in the bottom drawer," our method achieves 70.0%, compared with 80.0% for $\pi_{0.5}$. Since each task contains 10 evaluation trials, one successful or failed trial changes the reported success rate by 10 percentage points.

Table 6: Real-world evaluation results under OOD conditions. Each entry is based on 10 evaluation trials per task.

Task	$\pi_{0.5}$	Ours
Push in the bottom drawer	80	70
Pick cube and place in the top drawer	60	90
Pick cube and place in the left caddy	30	40
Pick cup and place on the cabinet	60	100
Push in the top drawer	50	100
Pick sponge and place on the plate	70	80
Average	58.3	80.0