

Robust Bimanual Vision-Language-Action Models via Embarrassingly Simple Modality Masking

Dongzhou Cheng^{1,2}, Ziang Li^{3,2}, Yixiao Zhou^{4,2}, Haojuan Li^{6,2}
Jinghao Zhang^{5,2}, Lei Lei^{5,2}, Minjing Dong⁷, Jie Gui¹, Jiaqi Wang²

¹Southeast University ²Shanghai Innovation Institute ³Wuhan University
⁴Zhejiang University ⁵University of Science and Technology of China
⁶Shanghai Jiao Tong University ⁷City University of Hong Kong

[m3vla.github.io](https://github.com/m3vla)

Abstract: Query-based Vision-Language-Action (VLA) models offer low-latency inference that is attractive for bimanual robotic manipulation, but we observe that they can still exhibit discontinuous actions and execution failures in complex dual-arm tasks. We hypothesize that unstable multi-view and language fusion is one contributing factor in these failures, often coinciding with attention spreading to distracting regions. To improve robustness, we introduce the Modality Masking Mechanism (M3), an *embarrassingly simple*, training-only strategy that requires no architectural changes or large-scale robot pretraining. M3 stochastically masks subsets of modality channels during training, exposing the policy to controlled partial observations and encouraging it to rely less on distracting cues and more on evidence that remains reliable. We evaluate M3 on ten bimanual tasks from RoboTwin 2.0 and on three long-horizon real-world tasks. Compared with the Adapter baseline, M3 improves average success by 21.7% in the Clean setting and 11.4% in Clean2Rand, where policies are trained on clean demonstrations and evaluated on randomized scenes, while also improving averaged real-world full-task success by over 30%. These results suggest that structured training-time masking is a practical way to improve the robustness of query-based VLA policies for bimanual manipulation.

Keywords: vision-language-action models, bimanual manipulation, multimodal learning, robotic control

1 Introduction

Bimanual robotic manipulation is widely viewed as a key stepping stone toward general-purpose embodied agents, since it demands coordinated dual-arm control [1, 2, 3], contact-rich interaction, and precise spatiotemporal reasoning. As Vision-Language-Action (VLA) models [4, 5, 6, 7, 8, 9, 10, 11, 12] progress from passive multimodal understanding of vision-language models (VLMs) [13, 14, 15] to acting in the physical world, efficient real-time inference becomes essential for bimanual control. Among existing paradigms, query-based VLA [6, 16] architectures decode actions with parallel learnable queries in a single forward pass, offering natural parallelism and low-latency inference.

However, we observe that query-based VLAs can still exhibit discontinuous actions in bimanual tasks, leading to poor execution (see Fig. 1). In many failure cases, the model’s attention spreads over irrelevant image regions rather than remaining concentrated on the intended object or contact region: multi-view inputs introduce salient yet task-irrelevant activations, which we find are often associated with unstable action rollout and reduced success rates (Fig. 8, Sec. 5).

Why does attention misalignment occur, and how can masking help? We observe a recurring failure pattern: when all camera views are always visible during training, the policy learns spurious

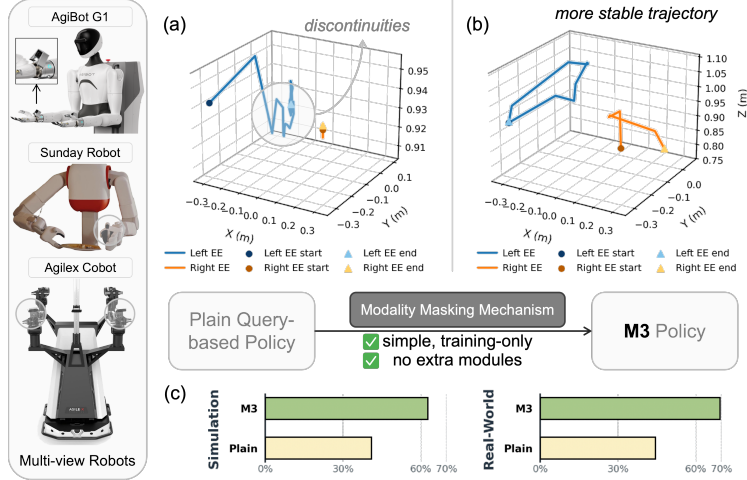


Figure 1: Left: representative multi-view, dual-arm platforms that reflect a prevalent form factor in current embodied manipulation research. (a) A plain query-based VLA exhibits trajectory discontinuities near the contact region. (b) The M3-trained policy produces more stable end-effector paths and more consistent contact alignment. (c) As shown in the bottom bar charts, M3 yields consistent improvements over the plain baseline across both simulation and real-world evaluations under clean conditions.

cross-view correlations rather than reasoning about which view provides reliable evidence for the current action. For instance, if a distractor in the idle arm’s view has high visual saliency, the model may incorrectly attend to it even when planning actions for the active arm. This manifests as diffuse attention spreading (Fig. 8) and severe degradation under distribution shift, where every method we evaluate drops sharply (Table 2): the Adapter baseline falls from 41.0% average success in Clean (Table 1) to 3.7%, and the strongest pretrained policy reaches only 12.9%. Our aim is to reduce this gap rather than to close it.

Why does this happen? Standard training exposes the model to all views simultaneously, allowing it to exploit any correlation—including unstable ones such as spurious visual feature matching across views or reliance on idle-arm saliency cues that happen to correlate with success in clean demonstrations. These correlations work when the training distribution is narrow but break when distractors, occlusions, or lighting changes alter one view’s appearance. The model never learns which cues remain informative under perturbation and which are coincidental to the clean training scenes.

Drawing on decision-making under partial observability [17, 18], we propose training under *structured view dropout*: masking the wrist-view group while preserving the egocentric view as a stable spatial reference forces the policy to solve tasks using evidence that remains reliable across different masking patterns. When the wrist views are jointly masked, the model cannot rely on wrist-to-ego saliency matching; it must instead extract task geometry from the egocentric frame, and learn to use wrist evidence only when those views provide non-redundant local information. We formalize this via the **Modality Masking Mechanism (M3)**, guided by three bimanual-specific design principles: (1) preserve the egocentric view as a consistent spatial anchor, (2) mask dual-wrist views *jointly* to prevent idle-arm interference, and (3) apply query-subset masking to encourage complementary action representations. Fig. 2 illustrates the resulting difference: the top panel contrasts the post-hoc attention structure of the Adapter baseline and M3, while the bottom panel shows their execution in the same *Handover Block* stage.

M3 is an *embarrassingly simple*, training-only intervention: it requires no architectural changes or large-scale pretraining, making it a low-overhead strategy for improving robustness. We evaluate on the challenging **RoboTwin 2.0** [3] benchmark that spans diverse bimanual tasks and provides both Clean and Clean2Rand evaluation settings, as well as on three long-horizon real-world tasks.

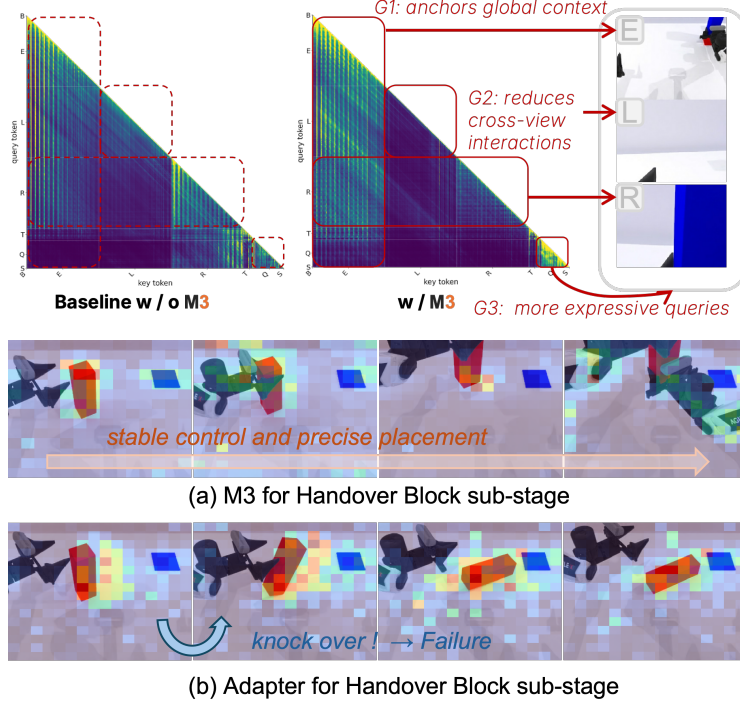


Figure 2: **Attention structure and contact-rich rollout behavior.** **Top:** Post-hoc attention-score maps for the Adapter baseline without M3 (left) and M3 (right). Rows are query tokens and columns are key tokens; E, L, R, T, and Q denote the egocentric, left-wrist, right-wrist, text, and action-query token groups, respectively. The M3 map exhibits patterns consistent with anchoring global context in the egocentric stream, reducing cross-view shortcuts, and encouraging more expressive query interactions. **Bottom:** At the handover-to-placement stage of the long-horizon *Handover Block* task, M3 maintains stable dual-arm coordination and completes placement in the shown rollout, whereas the Adapter baseline knocks the block over during contact and does not recover. Together, the panels juxtapose the intended change in attention structure with the qualitative execution difference observed in a representative rollout.

Clean2Rand trains policies on clean demonstrations and evaluates them on randomized scenes. With the same backbone and tuning budget as the Adapter baseline [16], M3 improves average success by **21.7%** in Clean and by **11.4%** in Clean2Rand. On the real robot, M3 raises the averaged full-task success rate by **25.0%** under clean conditions and by **48.6%** under OOD clutter across three long-horizon tasks. Overall, our **Contribution** lies in identifying a structured visibility design tailored to the bimanual multi-view setting, which can serve as a practical training-time strategy for improving the robustness of query-based VLA policies for bimanual robotic manipulation.

2 Preliminaries

We consider *query-based* VLA policies [6, 16] that turn multi-view images and language into an H -step action chunk in a single forward pass. For simplicity, we treat the proprioceptive inputs available during training as the default inputs to the VLA and omit further detailed description. At time t , let $V_t = [\phi_v(X_t^{(1)}), \dots, \phi_v(X_t^{(K)})] \in \mathbb{R}^{N_v \times d}$ denote visual tokens from K cameras, $T = \phi_\ell(P) \in \mathbb{R}^{N_p \times d}$ the language tokens for instruction P , and $Q \in \mathbb{R}^{N_q \times d}$ the decoder’s N_q action queries. A generic model g_θ predicts

$$\mathbf{A}_t = g_\theta(V_t, T, Q), \mathbf{A}_t[h] = a_{t+h}, h = 0, \dots, H-1. \quad (1)$$

where a_{t+h} denotes the h -step action and is trained with ℓ_1 objective:

$$\mathcal{L}(\theta) = \|\mathbf{A}_t - \hat{\mathbf{A}}_t\|_1, \quad (2)$$

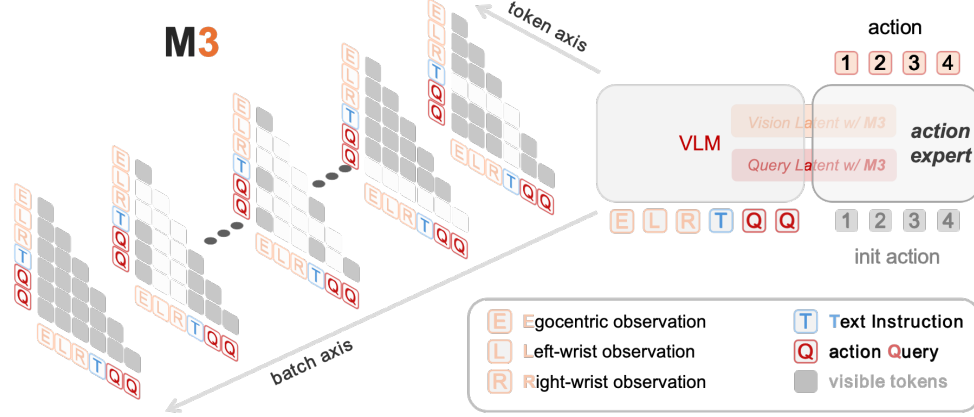


Figure 3: Overview of the Modality Masking Mechanism (M3). M3 is a training-time strategy that injects partial observability by stochastically masking modalities and action queries. As illustrated along the batch and token axes, different token subsets (comprising Egocentric (E), Left-wrist (L), Right-wrist (R) observations, Text Instructions (T), and Action Queries (Q)) are made visible (grey) at each training step. These partially observable inputs are processed by the VLM to produce M3-conditioned latents, which are then used by the action expert to decode actions. This process is intended to encourage more robust evidence selection and complementary representations without over-relying on any single input channel.

yielding efficient one-shot continuous control under an ℓ_1 objective. OpenVLA-OFT [6] treats the zero embedding as a special query, using $Q = \mathbf{0}$ so that the VLM extracts information directly from (V_t, T) , and then decodes the action chunks directly from the last-layer query embedding with a simple action head. VLA-Adapter [16], in contrast, replaces this zero query with learnable queries $Q = \text{Learnable}(N_q, d)$ and encodes vision and text conditions into the query latent by injecting Q through per-layer *bridge attention* [16], which modulates the τ -th layer action latent z^τ via Self-Attention and Cross-Attention:

$$\text{Action Expert} \begin{cases} z^0 = \mathbf{0}, \\ C^\tau = [\text{SA}(z^\tau), \text{CA}_{z^\tau}^{(t)}(\bar{V}^\tau), \text{CA}_{z^\tau}(\bar{Q}^\tau)], \\ z^{\tau+1} = \text{FFN}(C^\tau), \quad \tau = 0, \dots, L-1, \\ \mathbf{A}_t = \text{MLP}(z^L) \end{cases} \quad (3)$$

where τ indexes the layers of the VLM, $\text{CA}_{z^\tau}^{(t)}(\bar{V}^\tau)$ denotes cross-attention with z^τ as the query and the visual latents $\bar{V}^\tau$ from the VLM as keys/values, and the superscript (t) indicates the $\tanh(g)$ gate applied to bound the output [19].

3 Modality Masking Mechanism (M3)

Design guidelines of M3. The core idea of M3 is to train the policy under controlled partial observability. Instead of always providing all multimodal tokens at every update, we stochastically hide subsets of modalities and action queries during training (Fig. 3). The training objective remains the same regression objective, but the model is required to approximate the ground-truth even when some signals are missing. In our bimanual setting, we observe that naively dropping modalities at random can underperform a more structured approach, because careless masks may remove crucial global context, encourage diffuse attention patterns, or deprive the decoder of useful local signals. We therefore design M3 around three simple guidelines.

Guideline 1. Preserve a stable spatial reference frame. We always keep the egocentric view visible to serve as a consistent spatial anchor across timesteps and masking realizations. Unlike wrist views—which capture task-relevant detail but are prone to occlusion, lighting variation, and distractor

interference—the egocentric view provides a stable global frame for localizing targets and goals. Crucially, this is not a shortcut: masking the ego view instead is markedly worse (37.0% vs. 64.0%, Table 3a), since the wrist views alone often cannot localize the goal (e.g., when the placement target lies outside both wrist fields of view). Instead, by keeping ego visible while randomly masking wrist views, we force the model to learn *when and how* wrist views add value beyond what ego provides, rather than always fusing them indiscriminately. This encourages the policy to extract stable task geometry from the global view and only rely on local wrist views when they offer non-redundant, actionable evidence.

Guideline 2. Mask dual-wrist views jointly, not independently. We always mask both wrist cameras together rather than masking them independently. The rationale is that the left and right wrist views are spatially correlated: both typically capture overlapping regions of the workspace, especially near the target object. If we mask them independently, the model can still perform spurious cross-view feature matching between the visible wrist view and the egocentric view (e.g., matching high-contrast edges or object saliency across views). By masking both wrist views jointly, we create a clean separation between global evidence (egocentric) and local evidence (wrist), forcing the policy to extract task geometry from the stable global frame rather than relying on which wrist view happens to have better lighting or fewer occlusions. This design reduces idle-arm interference: when wrist views are masked, the model cannot attend to distractors in the non-executing arm’s camera, preventing the failure mode observed in Fig. 8 (top).

Guideline 3. Apply query-subset masking to encourage specialization. For action queries, we never mask all queries simultaneously (which would make the task unsolvable), but instead stochastically drop a random subset at each training step. This follows the classical dropout principle [20]: when different subsets of queries must solve the same task under different masking patterns, they are discouraged from learning redundant representations (co-adaptation). Instead, queries specialize: some may focus on coarse waypoint planning, others on fine-grained contact alignment, and still others on temporal consistency across the action horizon. By forcing queries to remain useful even when their “collaborators” are absent, we encourage them to extract complementary, non-redundant information from the multimodal context. Empirically, the Q-group attention structure in Fig. 2 (top) is more expressive under M3 than in the baseline, and we observe improved robustness when the policy encounters novel view configurations or partial occlusions at test time.

Empirically, our analysis (see Sec. 5 and the **Appendix**) is consistent with these guidelines: (1) preserving the ego stream helps sustain global planning, (2) (L+R) arm-view masking reduces spurious cross-view interactions, and (3) query-subset masking helps the model use diverse semantic cues under dynamic masking.

M3 Strategy. We introduce a *modality-level visibility*, which is integrated into the scaled dot-product attention through additive masks. This approach preserves all embeddings unchanged. Specifically, let $V_t = [V_t^{\text{ego}}, V_t^{\text{L}}, V_t^{\text{R}}]$ represent the visual token streams from the egocentric and two arm-mounted cameras. We propose to sample the mask from a Bernoulli distribution as $u_v \sim \text{Bernoulli}(1 - p_v)$ that masks both arm views **jointly**. Another mask is sampled from a Bernoulli distribution as $u_\ell \sim \text{Bernoulli}(1 - p_\ell)$ to mask the **entire** language modality, while ensuring the ego stream is always retained. For the action queries, we never fully drop any queries. Instead, we sample an element-wise vector $u_q \in \{0, 1\}^{N_q}$ using *i.i.d.* samples $u_q(i) \sim \text{Bernoulli}(1 - p_q)$ and enforce the constraint $\|u_q\|_1 \geq 1$, ensuring at least one active query at all times.

We define the token index sets for all modalities, which partition the full sequence: $\mathcal{S}_{\text{ego}}$ (egocentric), $\mathcal{S}_{\text{wrist}}$ (wrist), $\mathcal{S}_{\text{lang}}$ (language), and $\mathcal{S}_q$ (action queries). We then construct a single *token visibility indicator* vector m , where $m_i = 1$ if token i is visible and $m_i = 0$ otherwise. This vector is populated based on our sampled variables (letting i' be the relative index of i within $\mathcal{S}_q$):

$$m_i = \begin{cases} 1, & i \in \mathcal{S}_{\text{ego}}, \\ u_v, & i \in \mathcal{S}_{\text{wrist}}, \\ u_\ell, & i \in \mathcal{S}_{\text{lang}}, \\ u_q(i'), & i \in \mathcal{S}_q. \end{cases} \quad (4)$$

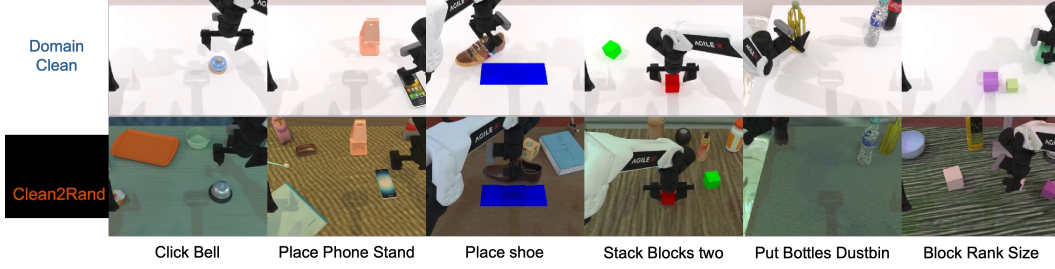


Figure 4: Example tasks from the RoboTwin 2.0 [3] benchmark, illustrating the visual distinction between the Clean (top row) and Clean2Rand (bottom row) evaluation settings. The six tasks shown are representative examples, with two selected from each of the short-, medium-, and long-horizon categories. Clean2Rand trains on clean demonstrations and evaluates on scenes with changed backgrounds, lighting, and distractor objects.

For self-attention, a masked token should neither “see” (row mask) nor “be seen” (column mask). We implement this with a single square additive mask $\tilde{M}$, which permits attention only between two tokens if **both** are visible:

$$\tilde{M}_{xy} = \begin{cases} 0, & m_x = 1 \text{ and } m_y = 1, \\ -\infty, & \text{otherwise.} \end{cases} \quad (5)$$

Finally, the VLM using scaled dot-product attention with this unified modality mask is:

$$\text{Attn} = \text{softmax} \left(\frac{\mathcal{Q}\mathcal{K}^\top}{\sqrt{d}} + M_c + \tilde{M} \right) \mathcal{V}, \quad (6)$$

where M_c represents the causal mask and $\tilde{M}$ is the unified modality-based mask. These masks are combined element-wise, enabling the model to train under controlled partial observability, with the goal of improving robustness under the input conditions.

Query Rescaling for Diverse Encoding. In addition to the attention mask defined in $\tilde{M}$, we introduce a *query rescaling* mechanism to promote diversity and robustness. This mechanism functions as *dropout regularization along the query dimension*. Specifically, while the unified mask $\tilde{M}$ excludes masked queries from the attention computation, we further rescale the embeddings of the *remaining* (visible) queries to maintain their expected magnitude, which can be formulated as

$$\tilde{\mathbf{q}}_i = \frac{1}{1 - p_q} \cdot \mathbf{q}_i, \quad \text{if } u_q(i') = 1 \text{ (visible)}. \quad (7)$$

This rescaling preserves the overall feature energy of the query embeddings. Consequently, the remaining queries may encode richer contextual information. Rather than relying on a fixed subset of queries, the model is encouraged to distribute task-relevant information across different query slots. Intuitively, each query may attend to distinct aspects of the visual-language context, since it cannot depend on a fixed partner query being consistently available. In our experiments, this dynamic adaptation is associated with improved robustness in downstream action prediction.

M3 Objective. While adhering to the standard $\mathcal{L}_1$ loss used in baselines, our M3 training objective is applied to predictions conditioned on M3-masked inputs. Formally, let V_t^{M3} , T_t^{M3} , and Q_t^{M3} denote the visible visual tokens, language tokens, and action queries after applying M3. The training objective remains the same $\mathcal{L}_1$ regression loss:

$$\mathcal{L}_{M3}(\theta) = \|\mathbf{g}_\theta(V_t^{M3}, T_t^{M3}, Q_t^{M3}) - \hat{\mathbf{A}}_t\|_1, \quad (8)$$

where $\mathbf{g}_\theta$ denotes the policy. As illustrated in Fig. 3, we follow the Adapter-style action expert, which decodes actions from visual latents and query latents that already carry the language signal. Training $\mathbf{g}_\theta$ to approximate $\hat{\mathbf{A}}_t$ under deliberate partial observability may encourage the query-based VLA model to learn more robust representations. In practice, this setup may help the model extract diverse, complementary information from these “dynamically observable latents,” potentially improving generalization.

Table 1: Domain-clean performance across multi-horizon tasks on the **RoboTwin 2.0** simulation platform. The best performance in each row is bolded, and the second-best is underlined. Δ denotes the relative improvement of **M3** over the Adapter baseline. Models marked with * are pretrained on large-scale robot data, while our method achieves superior performance through direct fine-tuning.

Category	Task Name	RDT*	π_0^*	ACT	DP	Adapter	M3	Δ
Short Horizon	Click Bell	80	44	58	54	84	97	+13
	Grab Roller	74	<u>96</u>	94	98	88	<u>96</u>	+8
	Place Phone Stand	15	35	2	13	10	55	+45
Medium Horizon	Place Bread Basket	10	17	6	14	11	22	+11
	Place A2B Right	1	27	0	13	4	28	+24
	Place Shoe	35	28	5	23	34	63	+29
	Stack Blocks Two	21	42	25	7	78	83	+5
Long Horizon	Handover Block	45	45	42	10	27	74	+47
	Put Bottles Dustbin	21	54	27	22	60	81	+21
	Block Rank Size	0	7	0	1	14	28	+14
Overall Avg		30.2	39.5	25.9	25.5	41.0	62.7	+21.7

4 Experiments

4.1 Setup

Benchmark. We adopt **RoboTwin 2.0** [3] as our primary simulation benchmark. The platform offers 50 dual-arm manipulation tasks with domain randomization over distractors, backgrounds, lighting, table heights, and language instructions. Following [21], we select 10 tasks grouped into *short-, medium-, and long-horizon* categories by mean step count (see Fig. 4). Each task is trained on 50 clean demonstrations and evaluated on 100 held-out scenarios in two settings: Clean uses held-out clean scenes, while Clean2Rand evaluates the same clean-trained policy on held-out randomized scenes. For simulation, we follow the official RoboTwin benchmark evaluation protocol and report the corresponding single-run results for direct comparison with prior work. **Real-world protocol.** We further construct three long-horizon bimanual tasks on the *Agilex Cobot* platform: bottle disposal with inter-arm handover, bowl stacking and shelving, and vegetable plating followed by plate centering. Each task exceeds 800 control steps and involves multi-stage grasping, handover, and coordinated transport. We collect 50 demonstrations per task and evaluate under two settings: *clean*, which uses the nominal layout with only task-relevant objects, and *OOD*, which introduces novel distractor objects near the targets following recent generalization protocols [22, 4]. For each task and each model, we repeat the real-world evaluation for three rounds: each round contains 16 clean trials and 8 OOD trials. For fair repeated measurement, the corresponding trial in each round uses the same scene layout, including the distractor arrangement in the OOD setting. We report averaged full-task success rates across the three rounds, corresponding to 48 clean and 24 OOD trials per task in total. Full details appear in Fig. 7 and **Appendix**.

Implementation details. Our VLM backbone is the Prismatic VLM [14], built upon the Qwen2.5-0.5B [23] language model. A key component is its hybrid vision encoder, which combines DI-NOv2 [24] and SigLIP [25], an architectural choice consistent with OpenVLA [5]. Furthermore, our *action expert* module is designed in alignment with the Adapter [16] approach, where, consistent with the baseline, proprioceptive inputs are utilized as default keys and values. Unless otherwise specified, all models use the *AdamW* optimizer, an initial learning rate of $2e - 4$, and are trained for 10 k steps with a MultiStep decay of $\times 0.1$ at 5 k steps. VLA-Adapter follows the official *pro* configuration as a strong baseline. Our method, M3, preserves the backbone architecture and introduces only training-time modality masking, and the LoRA rank is fixed to 64. We use identical hyperparameters across short/medium/long horizons in all experiments. Additional implementation details are provided in the **Appendix**.

Table 2: **Clean2Rand performance** across multi-horizon tasks on the **RoboTwin 2.0** simulation platform. All task-specific policies are trained on 50 clean demonstrations and evaluated on 100 held-out randomized scenes. Relative to the fine-tuned baseline, **M3** improves average success by **11.4%**. Unlike VLA policies such as π_0 [7] and RDT-1B [8], which rely on massive cross-embodiment, internet-scale or multi-robot pretraining to reduce the domain gap, **M3** uses only clean, task-specific training data and achieves comparable or better generalization.

Category	Task Name	RDT*	π_0^*	ACT	DP	Adapter	M3	Δ
Short Horizon	Click Bell	9	3	3	0	7	16	+9
	Grab Roller	43	80	25	0	28	54	+26
	Place Phone Stand	6	7	0	0	0	12	+12
Medium Horizon	Place Bread Basket	2	4	0	0	0	6	+6
	Place A2B Right	1	6	0	0	0	2	+2
	Place Shoe	7	6	0	0	0	11	+11
	Stack Blocks Two	2	1	0	0	0	4	+4
Long Horizon	Handover Block	14	8	0	0	0	6	+6
	Put Bottles Dustbin	4	13	1	0	2	40	+38
	Block Rank Size	0	1	0	0	0	0	0
Overall Avg		8.8	12.9	2.9	0.0	3.7	15.1	+11.4

4.2 Main Results

Domain-clean results. On RoboTwin 2.0 [3] (Table 1), M3 improves the overall success rate from 41.0% to 62.7%, outperforming the Adapter baseline as well as the compared pretrained policies in this evaluation. The gains are especially pronounced on long-horizon tasks, where the average rises from 33.7% to 61.0%; on *Handover Block* alone, success improves from 27% to 74%. As shown in the bottom panel of Fig. 2, M3 also exhibits more stable execution on this representative long-horizon task. These results indicate that structured training-time masking can substantially improve robustness for this query-based bimanual VLA setting.

Clean2Rand results. In Clean2Rand evaluation (Table 2), M3 achieves 15.1% overall success, a +11.4% gain over the Adapter baseline and the highest among all compared methods. The largest per-task gains appear on *Put Bottles Dustbin* (+38%) and *Grab Roller* (+26%). The sole exception is *Block Rank Size*, which stays near zero for every method. This task demands fine-grained relational size reasoning [12] and already exhibits residual diffuse attention in the Clean setting (Fig. 8), and the distractors in Clean2Rand further amplify this cross-view interference. Overall, these results indicate that *training-only* masking can meaningfully improve domain-shift robustness without large-scale robot data.

Real-world results. Fig. 5 compares representative rollouts under both the *clean* and *OOD* settings. In the clean setting, the Adapter baseline fails to discard the bottle into the bin, whereas M3 completes the task sequence successfully in this example rollout. Under the OOD setting, where novel distractor objects are placed near the targets, the baseline exhibits gripper misalignment during the inter-arm handover, while M3 appears more stable throughout the episode. More cases are shown in the **Appendix**. Quantitatively (Fig. 7), across three repeated evaluation rounds totalling 144 clean and 72 OOD trials, M3 raises the averaged full-task success rate from 44.4% to 69.4% under clean conditions and from 12.5% to 61.1% under OOD conditions. These numbers suggest that novel distractors unseen during training substantially reduce the baseline’s full-task success, whereas the M3-trained policy appears less affected under the same OOD protocol.

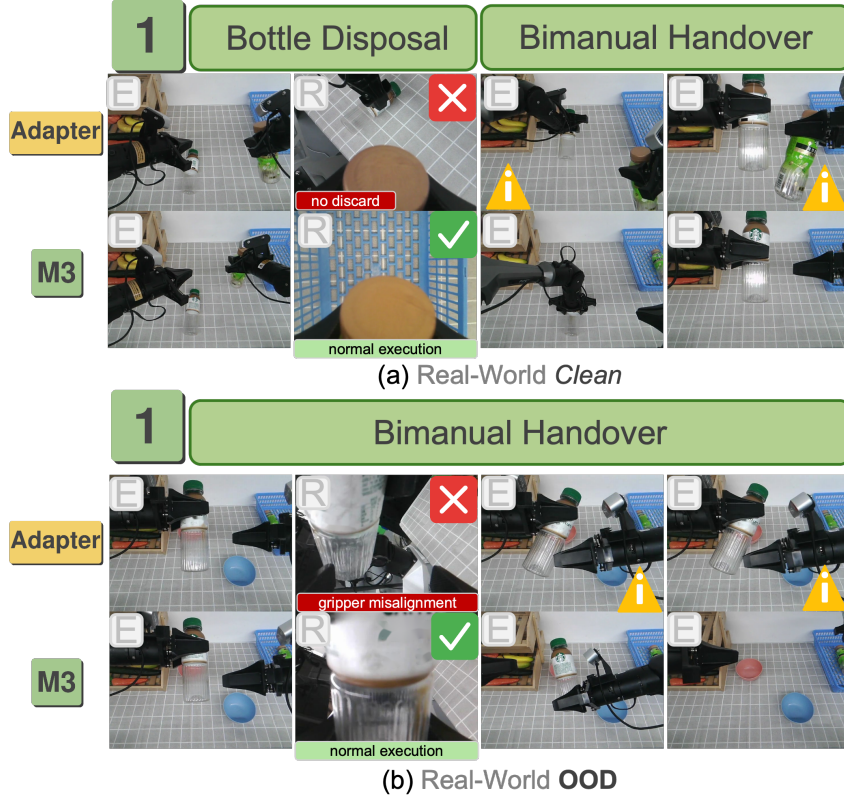


Figure 5: **Qualitative comparison of real-world execution under clean and OOD conditions.** The second column provides a close-up from the right-arm wrist camera. (a) In the clean setting, the Adapter baseline fails to release the bottle into the bin, whereas M3 completes the discard smoothly. (b) Under OOD clutter, the baseline suffers from gripper misalignment during handover, while M3 maintains stable alignment throughout the sequence.

5 Analysis

Cross-backbone transfer. To probe whether M3 may transfer beyond the Adapter backbone, we apply it to OpenVLA-OFT [6], a structurally distinct query-based VLA with parallel decoding and bidirectional attention. As shown in Fig. 6, M3 improves the overall average success rate of OpenVLA-OFT from 32.2% to 53.5% under the domain-clean RoboTwin protocol, with gains across short-, medium-, and long-horizon tasks. Taken together, these results suggest that the benefits of the proposed structured partial-observability training strategy may extend beyond the primary backbone studied in the main paper. Further per-task breakdowns are provided in the **Appendix**.

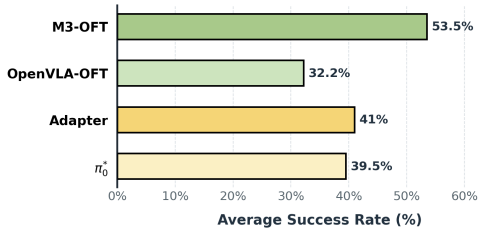


Figure 6: **Cross-backbone transfer.** Under the domain-clean RoboTwin protocol, M3-OFT raises average success from 32.2% to 53.5% and exceeds the evaluated baselines.

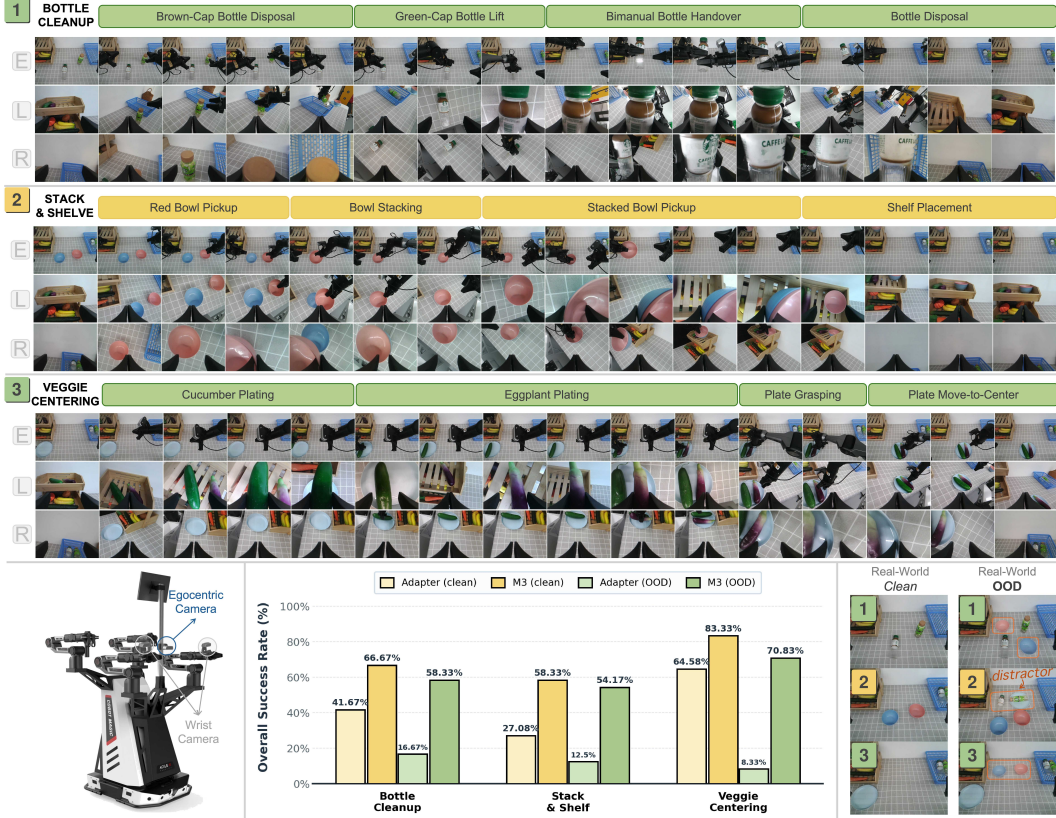


Figure 7: Overview of real-world evaluation. **Top**: three representative tasks successfully executed by M3: (1) Bottle Cleanup, (2) Stack Shelf, and (3) Veggie Centering. **Bottom-left**: our bimanual robot platform. **Bottom-center**: full-task success rates under clean and OOD settings, where M3 shows consistent improvement over the Adapter baseline. **Bottom-right**: illustration of the two evaluation settings (clean vs. OOD with novel distractor objects). The baseline degrades more sharply under OOD clutter, suggesting that M3 appears to be more robust to unseen distractors. Detailed task definitions, protocols, additional demonstrations, and tabulated results are given in Sec. 4.1 and the **Appendix**.

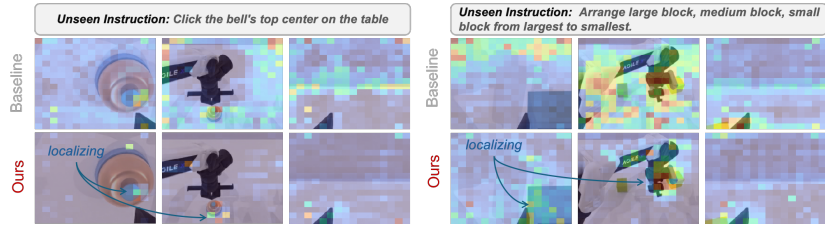


Figure 8: **Spatial attention under viewpoint variation**. On *Click Bell* and *Block Rank Size*, the Adapter baseline spreads attention over distractors, the gripper, and background regions, whereas M3 more consistently concentrates on the target and contact region. Attention remains partly diffuse on the harder *Block Rank Size*.

Visualization of M3. Fig. 8 compares spatial attention maps of the baseline and M3 on RoboTwain 2.0 tasks. We consistently observe that the baseline allocates more attention to distractors, the robot gripper, and background regions, whereas M3 tends to produce more compact attention around the object and contact region. The token-level attention-score analysis in the top panel of Fig. 2 complements these spatial maps by visualizing interactions among the egocentric, wrist, language, and action-query groups. Together with complementary statistics in the **Appendix**, these patterns are consistent with our hypothesis that training under structured partial observability

encourages more task-focused evidence selection; concurrent work likewise uses attention analysis to study VLA failures [26, 27].

Training efficiency. We define training efficiency as achieving a target success rate with fewer optimization steps while preserving or improving the final plateau. Fig. 9 compares learning curves on a short-horizon task (Place Phone Stand) and a long-horizon task (Handover Block) under identical settings. Across both settings, M3 rises more steeply and saturates at a higher level than the Adapter baseline. It reaches strong performance earlier and sustains a stable margin thereafter, indicating better sample efficiency and more reliable convergence. This trend suggests that M3 may provide a more favorable training signal in these settings.

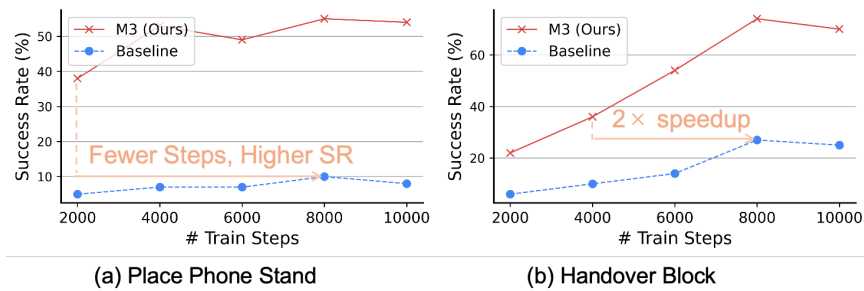


Figure 9: Success rate vs. training steps on *Place Phone Stand* (short-horizon) and *Handover Block* (long-horizon). Under the same compute budget, M3 tends to reach higher success rates earlier and maintains a consistent margin over the Adapter baseline throughout training.

Ablation of M3. At the view level, Table 3(a) shows that preserving the egocentric (ego) view while jointly masking both wrist views outperforms masking the ego view or only one wrist view. This supports the role of the ego view as a global anchor and suggests that paired wrist-view masking is preferable to independent single-wrist masking. Table 3(b) further shows that a light query-mask ratio works best, whereas overly aggressive query masking reduces success. Table 4(a) then decomposes M3 into vision, language, and query masking components. Single-component masking remains limited (29.0%–34.0% Avg. SR), and adding language to vision gives only 36.7%; in contrast, combining vision and query masking raises Avg. SR to 60.3%, while the full V+L+Q configuration reaches 64.0%. Table 4(b) compares M3 with generic regularization baselines: token dropout (31.8%), modality dropout (24.1%), visual augmentation (22.3%), and region augmentation (23.0%) all stay far below full M3. Overall, Table 4 indicates that the main gain comes from coupling structured vision masking with query masking, and that generic dropout or augmentation is not sufficient for this bimanual multi-view setting.

Table 3: **Targeted masking ablations.** (a) Vision view-level masking while non-view masking components follow the full-M3 setting. Rows compare masking the egocentric view, one wrist view (L or R), and both wrist views. (b) Query-mask ratio, including the unmasked Adapter baseline (0.0). Masking both wrist views and using a light query-mask ratio (0.1) achieve the highest average success rates in their respective comparisons.

(a) Vision view-level masking.					(b) Query-mask ratio.				
M3 variant	Place Phone Stand	Place Shoe	Handover Block	Avg	Query Mask Ratio	Place Phone Stand	Place Shoe	Handover Block	Avg
Mask ego	20	49	42	37.0	0.0 (Adapter)	10	34	27	23.7
Mask one wrist	22	39	37	32.7	0.1	55	63	74	64.0
Mask both wrists	55	63	74	64.0	0.3	50	59	59	56.0
					0.5	49	58	63	56.7
					0.7	52	55	55	54.0
					0.9	41	40	50	43.7

Table 4: **Ablation studies.** (a) Component ablation: V, L, and Q denote vision, language, and query masking. (b) Comparison with dropout and augmentation baselines. Avg. SR is the mean success rate over *Place Phone Stand*, *Place Shoe*, and *Handover Block*.

(a) Component ablation.			(b) Dropout and augmentation baselines.	
Method	Mask	Avg. SR	Method	Avg. SR
Adapter	none	23.7	Adapter	23.7
Language-only	L	29.0	Token dropout	31.8
Vision-only	V	34.0	Modality dropout	24.1
Query-only	Q	33.3	Visual aug.	22.3
Vision+language	V+L	36.7	Region aug.	23.0
Vision+query	V+Q	60.3	Mask one wrist	32.7
<i>M3 (full)</i>	V+L+Q	64.0	<i>M3 (full)</i>	64.0

6 Related Works

Vision-Language-Action Models. VLAs [4, 5, 6, 7, 8, 9, 10, 11, 28] have greatly advanced general-purpose robotic policies by leveraging large-scale pretrained vision-language models (VLMs) [13, 14, 15] to connect perception with action. Current VLAs generally fall into three modeling categories. (1) Autoregressive models such as RT-2 [4] and OpenVLA [5] discretize continuous actions into language tokens and predict them sequentially. CoT-VLA [29] further introduces Chain-of-Thought reasoning to imagine intermediate visual states before generating actions, separating visual reasoning from execution. However, such discretization can lead to quantization errors that reduce action precision in fine-grained manipulation. (2) Diffusion-based models operate directly in continuous action space. Diffusion Policy [30] learns high-quality grasp trajectories through iterative denoising, while π_0 [7] enhances sampling efficiency via a Mixture-of-Transformers [31] and flow matching [32]. Although effective, these approaches [5, 7] often depend on extensive pretraining. (3) Query-based models decode actions in a single forward pass and regression objective. ACT [1] adopts a DETR-style architecture with temporal ensembling for stable predictions, and OpenVLA-OFT [6] extends it with parallel decoding for efficient adaptation. VLA-Adapter [16] further scales this design to a compact 0.5B-parameter model, introducing action queries and bridge-attention experts, achieving competitive single-arm performance without large-scale retraining. Despite these advances, query-based VLAs remain underexplored in more complex bimanual settings [3]. To address this gap, we introduce a lightweight modality-masking mechanism that requires no architectural changes and incurs zero additional compute or parameters, and empirically improves policy generalization across diverse bimanual manipulation tasks.

Learning-based Bimanual Manipulation. Leveraging structured reinforcement learning and coordination priors, recent bimanual manipulation works have achieved robust sim-to-real transfer across various dual-arm skills [33, 34]. Concurrently, multimodal perception, including visuo-tactile fusion and wrist-force sensing, has improved contact state estimation, robustness, and data efficiency [35, 36, 37]. Moreover, directly learning implicit action signals from human videos bridges the embodiment gap and lowers demonstration costs [38, 39, 40]. Within the most widely studied imitation-learning track, ACT and diffusion policies on low-cost and mobile whole-body ALOHA systems demonstrate strong real-world coordination under modest supervision [1, 30, 2, 41]; foundation-scale diffusion (RDT-1B) [8] and efficient 3D flow-matching policies [42] unify action spaces and set new state of the art across robots and tasks. However, diffusion models can incur nontrivial training and inference cost; our work instead explores a query-based foundation policy that directly decodes actions in a single step, aiming to retain efficiency while matching or surpassing pretrained models.

Dropout and training-time regularization. Dropout [20] randomly deactivates neurons during training to prevent co-adaptation. This principle has been extended to layer dropout [43], attention

dropout [44], and modality dropout for multimodal fusion [45, 46]. Our ablations (Table 4) show that generic token dropout and modality dropout reach only 31.8% and 24.1% average success, respectively, whereas the structured M3 configuration reaches 64.0%.

7 Limitations

While M3 shows consistent gains across RoboTwin 2.0 and three real-world tasks, broader evaluation on additional robot platforms, task categories, wider VLA architectures, and richer input modalities such as 3D geometric information, together with a deeper analysis of how structured masking influences multimodal fusion and diffusion-based action decoding, remains future work.

8 Conclusion

We study robustness issues in query-based VLA models for bimanual manipulation and introduce M3, a simple training-only modality masking strategy. Across RoboTwin 2.0 and three long-horizon real-world tasks, M3 consistently improves over the Adapter baseline while requiring no inference-time architectural changes. These results suggest that structured training-time masking is a practical way to strengthen query-based bimanual VLA policies.

References

- [1] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.
- [2] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.
- [3] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Z. Li, Q. Liang, X. Lin, Y. Ge, Z. 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.
- [4] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. 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.
- [5] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.
- [6] M. J. Kim, C. Finn, and P. Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.
- [7] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
- [8] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.
- [9] Y. Zhong, F. Bai, S. Cai, X. Huang, Z. Chen, X. Zhang, Y. Wang, S. Guo, T. Guan, K. N. Lui, et al. A survey on vision-language-action models: An action tokenization perspective. *arXiv preprint arXiv:2507.01925*, 2025.
- [10] R. Shao, W. Li, L. Zhang, R. Zhang, Z. Liu, R. Chen, and L. Nie. Large vlm-based vision-language-action models for robotic manipulation: A survey. *arXiv preprint arXiv:2508.13073*, 2025.

- [11] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.
- [12] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. pi0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.
- [13] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.
- [14] S. Karamcheti, S. Nair, A. Balakrishna, P. Liang, T. Kollar, and D. Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*, 2024.
- [15] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [16] Y. Wang, P. Ding, L. Li, C. Cui, Z. Ge, X. Tong, W. Song, H. Zhao, W. Zhao, P. Hou, et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. *arXiv preprint arXiv:2509.09372*, 2025.
- [17] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.
- [18] M. Lauri, D. Hsu, and J. Pajarinen. Partially observable markov decision processes in robotics: A survey. *IEEE Transactions on Robotics*, 39(1):21–40, 2022.
- [19] R. Zhang, J. Han, C. Liu, P. Gao, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li, and Y. Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.
- [20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [21] H. Li, Y. Zuo, J. Yu, Y. Zhang, Z. Yang, K. Zhang, X. Zhu, Y. Zhang, T. Chen, G. Cui, et al. Simplevla-rl: Scaling vla training via reinforcement learning. *arXiv preprint arXiv:2509.09674*, 2025.
- [22] Q. Li, Y. Liang, Z. Wang, L. Luo, X. Chen, M. Liao, F. Wei, Y. Deng, S. Xu, Y. Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.
- [23] Q. A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y.-C. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, Z. Qiu, S. Quan, and Z. Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024.
- [24] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P. Huang, S. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jégou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024.
- [25] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

- [26] H. Huang, F. Liu, L. Fu, T. Wu, M. Mukadam, J. Malik, K. Goldberg, and P. Abbeel. Otter: A vision-language-action model with text-aware visual feature extraction. *arXiv preprint arXiv:2503.03734*, 2025.
- [27] W. Song, Z. Zhou, H. Zhao, J. Chen, P. Ding, H. Yan, Y. Huang, F. Tang, D. Wang, and H. Li. Reconvla: Reconstructive vision-language-action model as effective robot perceiver. *arXiv preprint arXiv:2508.10333*, 2025.
- [28] Z. Li, D. Cheng, Y. Wang, S. Wang, X. Xu, L. Weng, J. Wang, and J. Wang. Light-wam: Efficient world action models with state-fusion action decoding. *arXiv preprint arXiv:2606.08242*, 2026.
- [29] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713, 2025.
- [30] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.
- [31] W. Liang, L. Yu, L. Luo, S. Iyer, N. Dong, C. Zhou, G. Ghosh, M. Lewis, W.-t. Yih, L. Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.
- [32] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- [33] Y. Fan, X. Li, K. Zhang, C. Qian, F. Zhou, T. Li, and Z. Huang. Learning robust skills for tightly coordinated arms in contact-rich tasks. *IEEE Robotics and Automation Letters*, 9(3):2973–2980, 2024.
- [34] Y. Cui, Z. Xu, L. Zhong, P. Xu, Y. Shen, and Q. Tang. A task-adaptive deep reinforcement learning framework for dual-arm robot manipulation. *IEEE Transactions on Automation Science and Engineering*, 22:466–479, 2024.
- [35] B. Huang, Y. Wang, X. Yang, Y. Luo, and Y. Li. 3d-vitac: Learning fine-grained manipulation with visuo-tactile sensing. *arXiv preprint arXiv:2410.24091*, 2024.
- [36] Y. Lin, A. Church, M. Yang, H. Li, J. Lloyd, D. Zhang, and N. F. Lepora. Bi-touch: Bimanual tactile manipulation with sim-to-real deep reinforcement learning. *IEEE Robotics and Automation Letters*, 8(9):5472–5479, 2023.
- [37] S. Stepputtis, M. Bandari, S. Schaal, and H. B. Amor. A system for imitation learning of contact-rich bimanual manipulation policies. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11810–11817. IEEE, 2022.
- [38] A. Bahety, P. Mandikal, B. Abbatematteo, and R. Martín-Martín. Screwmimic: Bimanual imitation from human videos with screw space projection. *arXiv preprint arXiv:2405.03666*, 2024.
- [39] H. Zhou, R. Wang, Y. Tai, Y. Deng, G. Liu, and K. Jia. You only teach once: Learn one-shot bimanual robotic manipulation from video demonstrations. *arXiv preprint arXiv:2501.14208*, 2025.
- [40] H. Zhou and K. Jia. Binomap: Learning category-level bimanual non-prehensile manipulation primitives. *arXiv preprint arXiv:2509.21256*, 2025.
- [41] T. Z. Zhao, J. Thompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid. Aloha unleashed: A simple recipe for robot dexterity. *arXiv preprint arXiv:2410.13126*, 2024.

- [42] N. Gkanatsios, J. Xu, M. Bronars, A. Mousavian, T.-W. Ke, and K. Fragkiadaki. 3d flowmatch actor: Unified 3d policy for single-and dual-arm manipulation. *arXiv preprint arXiv:2508.11002*, 2025.
- [43] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, pages 646–661. Springer, 2016.
- [44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- [45] N. Neverova, C. Wolf, G. Taylor, and F. Nebout. Moddrop: Adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1692–1706, 2016.
- [46] Y. Liu, F. Pardo, and T. B. Schön. Sensor dropout: Learning efficient sensing policies. In *Workshop on Bayesian Deep Learning, NeurIPS*, 2017.
- [47] C. Li, J. Liu, B. Li, B. Gao, Y. Yuan, Y. He, Y. Li, and J. Tang. DTP: A simple yet effective distracting token pruning framework for vision-language action models. *arXiv preprint arXiv:2601.16065*, 2026. doi:10.48550/arXiv.2601.16065.

Table 5: VRAM footprint during simulation on RTX 4090 (24 GB). OpenVLA-OFT approaches the 24 GB ceiling and frequently triggers OOM failures, whereas VLA-Adapter (0.5B) operates with a substantially smaller footprint, enabling stable iteration [16].

Model	VRAM (GB)	Status
OpenVLA-OFT (7B)	$\approx 23\text{-}24$	OOM-prone
VLA-Adapter (0.5B)	≈ 10	Stable

A Experimental Settings

A.1 Backbone Rationale and Motivation

We adopt VLA-Adapter [16] as our primary backbone, motivated by its parameter efficiency and alignment with our research objectives.

Parameter and Inference Efficiency. VLA-Adapter leverages a 0.5B-scale vision-language backbone with a lightweight bridging attention strategy, enabling parameter-efficient fine-tuning without large-scale robotic pretraining. This aligns well with our goal of enhancing bimanual manipulation under limited training resources. It also offers low-latency inference via one-shot decoding of learnable action queries.

Practical and Computational Constraints. In contrast, the 7B-parameter OpenVLA-OFT [6] demands more parameters and query embeddings for bimanual tasks. This imposes significant overhead in multi-view and long-horizon scenarios, causing high VRAM sensitivity. This burden is exacerbated during Clean2Rand evaluation, where diverse environment assets must be rendered concurrently. In our engineering setup, we simulate with an RTX 4090 GPU since the H100 lacks Vulkan support for SAPIEN rendering. Consequently, the combined VRAM footprint of OpenVLA-OFT and simulation rendering frequently approaches the 24GB ceiling (see Table 5). This leads to frequent **Out-Of-Memory (OOM)** failures and impedes experimental throughput. Therefore, to support faster iteration on commodity-grade GPUs, we prioritize VLA-Adapter.

Empirical Suitability in Simulation. This preference is not solely resource-driven. In the original VLA-Adapter paper, the authors report that VLA-Adapter achieves performance broadly comparable to OpenVLA-OFT across multiple simulated results, and in some cases slightly better, despite using a substantially smaller backbone [16]. We observe a similar tendency in our own domain-clean evaluation: as summarized later in Table 8, the Adapter baseline remains competitive and attains a higher average success rate than OpenVLA-OFT. While these comparisons are drawn from different benchmarks and implementations, they suggest that selecting VLA-Adapter is a practically reasonable backbone choice rather than merely a compromise for memory efficiency.

Methodological Motivation. Methodologically, we observe that query-based VLA models (e.g., VLA-Adapter) can exhibit unstable multi-view and language fusion in bimanual control, often coinciding with attention spreading to distracting regions. This observation motivates our proposed improvements. To this end, we introduce M3: a modality masking strategy effective exclusively during the training phase. Based on a heuristic design, M3 operates by stochastically masking partial modality channels and action query tokens. Without altering the baseline architecture, this approach provides structured partial observability during training and encourages more robust evidence use.

A.2 Baselines.

Action Chunking with Transformers (ACT) [1]. ACT, based on a conditional variational autoencoder (CVAE) imitation learning formulation, uses action chunking to predict future action sequences and applies temporal ensembling techniques to ensure stable and smooth execution. Inspired by DETR, the model adopts an encoder-decoder architecture where action queries feed into the transformer decoder to predict actions. It trains through direct behavior cloning with a regression loss

applied to complete action chunks, and empirically demonstrates the ability to solve fine-grained bimanual manipulation tasks on low-cost hardware.

Diffusion Policy (DP) [30]. Diffusion Policy treats visuomotor control as a conditional denoising diffusion process over action trajectories. It learns to iteratively refine noisy action sequences into expert-like control signals conditioned on current observations. The forward process adds Gaussian noise to ground-truth action horizons, while a neural network learns the reverse denoising dynamics given images and robot state. At inference, the model samples a full future action horizon via iterative denoising and executes only the initial segment before re-planning. This enables robust receding-horizon control and naturally captures multimodal action distributions. Empirically, it outperforms standard behavioral cloning across many manipulation tasks and serves as the canonical “diffusion over actions” formulation that many VLA architectures adopt as their continuous-action head or expert.

Robotics Diffusion Transformer (RDT) [8]. Addressing the high complexity and data scarcity of bimanual manipulation, RDT-1B presents itself as a 1.2B parameter diffusion-based foundation model. The work introduces a “Physically Interpretable Unified Action Space” to unify action formats, which facilitates the pre-training of its Robotics Diffusion Transformer (RDT) on a massive 1M+ multi-robot trajectory dataset. The model subsequently undergoes fine-tuning on 6K+ bimanual-specific episodes. By leveraging a scalable Transformer and diffusion modeling to represent multi-modal action distributions, RDT-1B achieves strong zero-shot generalization to new objects and scenes, follows language instructions, and learns new skills from just 1-5 demonstrations.

π_0 [7]. This model represents a large Vision-Language-Action (VLA) foundation policy built on a pretrained Vision-Language Model (VLM) [13], augmented with a continuous-action flow-matching [32] expert to enable generalist robot control. The VLM encodes images and natural-language instructions, inheriting internet-scale semantic knowledge, while the flow model learns a conditional vector field that transforms noise into continuous robot actions given visual, linguistic, and proprioceptive context. π_0 trains on a variety of datasets for single-arm, bimanual, and mobile manipulators, follows complex language instructions, and allows for fine-tuning to acquire new skills. As a VLM-based prototype VLA foundation model, it influences a large body of subsequent diffusion-based VLA work.

VLA-Adapter (Adapter) [16]. VLA-Adapter builds a VLA by freezing a 0.5B vision-language backbone and learning lightweight adapter and policy modules that bridge multimodal features to actions. Rather than pre-training the VLM on robot data, it identifies the most control-relevant vision-language conditions and injects them into the action space through a policy module with Bridge Attention. With modest amounts of robotic data, VLA-Adapter reaches competitive performance on simulated and real benchmarks while retaining fast inference and much lower training cost than billion-parameter VLAs. It is therefore a representative small-scale VLA baseline centered on efficient VL-to-action bridging instead of backbone scaling or heavy diffusion or flow heads.

A.3 Additional Configuration

All models are trained on a high-performance compute cluster equipped with $4 \times$ NVIDIA H100 GPUs to ensure optimization efficiency, while all evaluations and inference benchmarks are conducted on a single NVIDIA RTX 4090 GPU. We maintain consistent training hyperparameters between the VLA-Adapter baseline and our M3 method, employing a global batch size of 48 (12 per GPU). For the OpenVLA-OFT experiments, we adjust the global batch size to 16 (4 per GPU) to accommodate the higher memory footprint of the 7B backbone, while keeping all other training configurations identical to those of the VLA-Adapter. Furthermore, adhering to the standard dual-arm Aloha protocols [1] established in RoboTwin 2.0 and OpenVLA-OFT, we strictly adopt an action chunk size of 25 steps across all training and evaluation sessions.

Table 6: Task statistics and horizon categorization on the RoboTwin 2.0 benchmark.

Task Name	Steps	Horizon	Horizon Group
Short Horizon Tasks			
Click Bell	62	Short	Average: 95 steps Count: 3 tasks
Grab Roller	95	Short	
Place Phone Stand	128	Short	
Medium Horizon Tasks			
Place Bread Basket	222	Medium	Average: 197 steps Count: 4 tasks
Place A2B Right	140	Medium	
Place Shoe	149	Medium	
Stack Blocks Two	277	Medium	
Long Horizon Tasks			
Handover Block	287	Long	Average: 461 steps Count: 3 tasks
Put Bottles Dustbin	592	Long	
Block Rank Size	505	Long	
Overall Statistics	Total: 10 tasks, Average: 245 steps.		

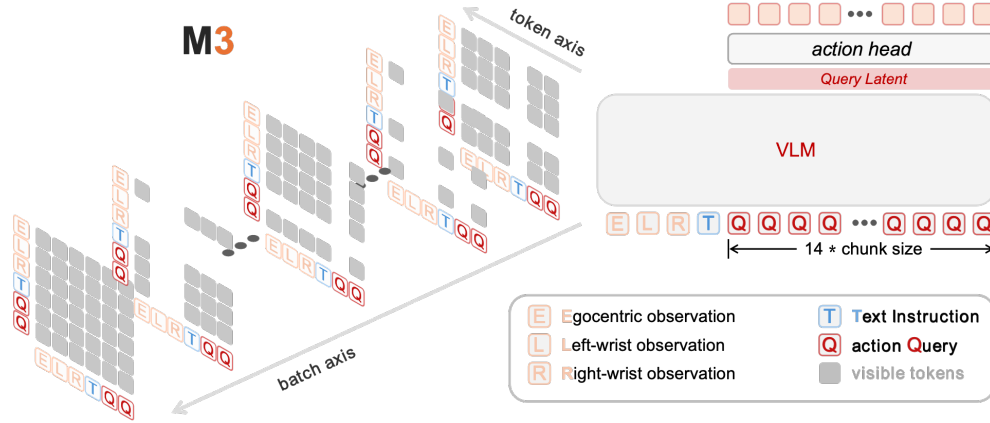


Figure 10: Overview of the Modality Masking Mechanism (M3) for OpenVLA-OFT [6], which is built upon the OpenVLA-7B [5] backbone. The legend follows the same color and token conventions as the main-paper overview figure. The three main differences from VLA-Adapter [16] include: (1) the adoption of parallel decoding with bidirectional attention, (2) the input of zero-query embeddings with shape (action axes) $\times$ (chunk size), and (3) unlike VLA-Adapter, which interacts vision and query latents via an action expert at every layer, here the final-layer query latents are mapped pointwise to the minimal unit of the action signal. For example, if we infer $T = 25$ action steps and each bimanual timestep has $A = 14$ action axes, we provide $A \times T = 14 \times 25 = 350$ query embeddings.

A.4 Tasks.

To evaluate our proposed M3 strategy, we conduct experiments on the RoboTwin 2.0 benchmark [3]. Following standard protocols, we select a set of 10 bimanual manipulation tasks that involve varying degrees of coordination, contact-rich interaction, and spatiotemporal reasoning.

Task Horizons and Complexity. As detailed in Table 6, we categorize these tasks into three groups based on their planning horizon and average step count to analyze performance across different temporal complexities:

- **Short Horizon:** This category includes tasks such as *Click Bell*, *Grab Roller*, and *Place Phone Stand*. These tasks typically involve fundamental reaching or grasping primitives with an average duration of approximately 95 steps.

Table 7: Representative seen (training) and unseen (evaluation) language instructions. During evaluation, instructions are stochastically sampled from the complete unseen set.

Category	Task Name	Seen Instructions	Unseen Instructions
Short Horizon	Click Bell	Touch the bell at its top center.	Find the palm-sized bell and click its top center.
	Grab Roller	Grab the roller on the table.	Use arms to grab the light brown cylindrical roller.
	Place Phone Stand	Move the palm-sized phone onto the compact plastic phone-stand.	Set the handheld phone with silver frame on the adjustable phone holder.
Medium Horizon	Place Bread Basket	Grab both the small bread and the golden bread loaf, drop into the beige plastic breadbasket.	Use one arm to grab the palm-sized braided golden bread, drop in the round beige breadbasket.
	Place A2B Right	Place the manual blue stapler precisely to the right of the woodenblock with back grooves.	Put the hard black box for card deck to the right of the bright blue soap bar.
	Place Shoe	Move the white shoe with rounded toe from the table to the mat in one fluid motion.	Grab the synthetic shoe upper from the table and set it on the mat.
	Stack Blocks Two	Shift red block to the center and place green block on top.	Place red block in the middle, then stack green block on it.
Long Horizon	Handover Block	Hold the red block with the left arm and place it on the pad.	Grab the red block with the left arm, switch to the right, and place it on the blue pad.
	Put Bottles Dustbin	Take the bottle made of smooth plastic, drop it in the dustbin with lid and bag, then move the blue-accented cylindrical bottle and the black and red bottle to the dustbin with lid and bag.	Grab the yellow bottle with elongated cylinder body and place it in the curved rectangular trash can, then do the same for the plastic bottle and the bottle with red screw cap.
	Block Rank Size	Bring large block, medium block, and small block to the center, sorting largest to smallest.	Sort the three blocks by size at the center, from the largest block to the smallest.

- **Medium Horizon:** Tasks such as *Place Bread Basket*, *Place A2B Right*, *Place Shoe*, and *Stack Blocks Two* fall into this category. They require sequential actions—such as pick-and-place operations or dual-arm coordination—averaging 197 steps in length.
- **Long Horizon:** The category comprises *Handover Block*, *Put Bottles Dustbin*, and *Block Rank Size*. These tasks generally demand extended reasoning and multi-stage execution. For instance, *Block Rank Size* involves sorting multiple objects, extending the horizon beyond 500 steps (avg. 461 steps).

This multi-horizon stratification allows us to examine M3 across varying temporal complexities, including longer sequences where we observe that the baseline query-based VLA can exhibit drift or instability.

Instructions. To assess the model’s language grounding capability and robustness to linguistic perturbations, we follow the official RoboTwin 2.0 benchmark settings to evaluate performance under two instruction settings, as enumerated in Table 7:

Table 8: Domain-clean performance across multi-horizon tasks on the **RoboTwin 2.0** simulation platform. The columns labeled **+M3** denote the integration of our M3 training strategy into the *preceding* baseline (Adapter and OpenVLA-OFT, respectively). The best performance in each row is bolded, and the second-best is underlined. Δ^{oft} denotes the relative improvement of **M3** over the OpenVLA-OFT baseline. Overall, these results provide additional supporting evidence that M3 may transfer beyond the primary VLA-Adapter backbone in the evaluated domain-clean setting.

Category	Task Name	RDT*	π_0^*	Adapter	+M3	OpenVLA-OFT	+M3	Δ^{oft}
Short Horizon	Click Bell	80	44	84	<u>97</u>	86	100	+14
	Grab Roller	74	96	88	<u>96</u>	94	97	+3
	Place Phone Stand	15	35	10	<u>55</u>	24	56	+32
Medium Horizon	Place Bread Basket	10	<u>17</u>	11	22	3	13	+10
	Place A2B Right	1	<u>27</u>	4	28	8	10	+2
	Place Shoe	35	28	34	63	17	<u>50</u>	+33
	Stack Blocks Two	21	42	<u>78</u>	83	22	69	+47
Long Horizon	Handover Block	45	45	27	74	28	39	+11
	Put Bottles Dustbin	21	54	60	81	40	<u>75</u>	+35
	Block Rank Size	0	7	14	28	0	<u>26</u>	+26
Overall Avg		30.2	39.5	41.0	62.7	32.2	<u>53.5</u>	+21.3

1. **Seen Instructions:** The set of instructions that are explicitly visible to the model during the training phase (e.g., “*Touch the bell at its top center*”).
2. **Unseen Instructions:** Instructions that share the same semantic intent as the training commands but employ more varied and complex phrasing. These are used exclusively during evaluation to assess the model’s robustness to linguistic diversity (e.g., “*Find the palm-sized bell...*”).

B Additional Experiments

To further examine transfer beyond the primary backbone studied in the main paper, we extend M3 to the OpenVLA-OFT [6] framework, which features a distinct query-based architecture with parallel decoding and bidirectional attention, as illustrated in Fig. 10. As detailed in Table 8, this integration improves the average success rate by 21.3% over the standard OpenVLA-OFT baseline in the domain-clean setting. The consistency of these gains across short-, medium-, and long-horizon tasks provides additional supporting evidence that M3 may transfer beyond VLA-Adapter in the evaluated setting.

B.1 Quantitative Attention-Misalignment Analysis

To complement the qualitative heatmaps shown in the main paper, we further analyze whether action discontinuities are statistically associated with a larger amount of attention allocated away from prompt-relevant visual patches. Following the analysis style of recent empirical studies on visuomotor attention quality [47, 26, 27], we revisit the three representative tasks used in our ablation study, namely *Place Phone Stand*, *Place Shoe*, and *Handover Block*. For each task, we recollect both temporally smooth rollouts and discontinuous rollouts from the evaluation logs and compute a DTP-style *unimportant attention* statistic from the model’s internal self-attention tensors [47].

Concretely, we first form an important patch set G on the concatenated three-view visual tokens using prompt-to-visual relevance, and then sum the action-to-visual attention mass outside G . In the configuration used here, prompt-to-visual relevance is averaged over the last 12 transformer layers and G retains the top 50% visual patches by relevance. This post-hoc statistic does not affect action prediction.

Because these trajectories have different horizons, we normalize each rollout to the interval $[0, 1]$ and aggregate the per-step attention statistic along normalized time. The resulting comparison is shown in Fig. 11. Across all three tasks, discontinuous trajectories tend to exhibit higher unimportant-attention values than their smooth counterparts. This gap is more visible in the middle-to-late phases of

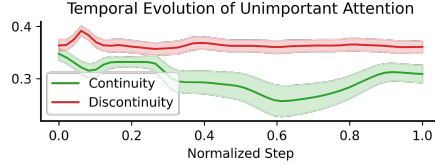


Figure 11: Quantitative attention-misalignment analysis on three representative tasks. Discontinuous trajectories tend to exhibit higher unimportant attention than smooth trajectories over normalized rollout time.

execution, where contact formation, object transfer, and final placement require accurate localization and temporally stable evidence selection. The trajectories produced by M3 tend to maintain a lower level of attention outside the prompt-relevant patch set throughout the rollout, suggesting that the policy may be less prone to drift toward distractors once the manipulation enters contact-rich stages.

This result provides a quantitative complement to the visualizations in the main paper: rollouts with larger action attention outside the prompt-relevant patch set tend to be more jittery or discontinuous. Together with the main-paper heatmaps, this trend is consistent with the hypothesis that training-time masking may reduce reliance on salient but task-irrelevant cues during cross-view fusion.

Bimanual-specific interpretation. The above trend is particularly relevant to dual-arm manipulation. In single-arm settings, the wrist camera typically remains aligned with the acting hand and the egocentric view, so visual evidence from different viewpoints is less likely to compete. In our bimanual setting, however, the idle arm can introduce a distractor wrist view that remains visually salient but action-irrelevant for the current sub-step. This creates a characteristic form of cross-view interference that is weaker or absent in single-arm manipulation. Our masking rules, especially preserving the egocentric stream while jointly masking the wrist views, were designed to address this hypothesized failure mode and may help explain the generalization gains observed at inference. The trend in Fig. 11 is consistent with this interpretation.

C Visualization Results

Real-World stage-level trends and qualitative cases. Table 9 provides a stage-level breakdown of the same three-task real-world evaluation summarized in the main paper. The performance differences are distributed across multiple phases rather than concentrated in a single substep. In **Bottle Cleanup**, the largest clean-setting gaps appear at the bimanual handover and final disposal stages, where coordination demands are highest. In **Stack & Shelf**, the two methods achieve comparable rates at the initial bowl pickup, while the gap is most pronounced at the bowl-stacking phase. In **Veggie Centering**, the gaps grow progressively from the early placement phases to the later plate-transport and centering stages. Under OOD clutter, baseline rates decline across all phases in each task, while M3 degrades comparatively less in this evaluation.

Figs. 14–22 provide qualitative context for these aggregate rates. For **Bottle Cleanup**, Fig. 14 shows a clean case where the baseline reaches the bin with the brown-cap bottle but fails to release it, whereas M3 completes both disposal branches. Fig. 15 presents a second clean case in which both methods progress through the early pickups, yet only M3 enters the later transfer-and-disposal stage. Fig. 16 depicts an OOD cluttered scenario where the baseline stalls at an early phase. For **Stack & Shelf**, Fig. 17 shows a clean case where the baseline narrowly completes the full task despite an imprecise bowl stack, whereas M3 maintains smoother execution across multiple stages. Fig. 18 shows a harder clean case where the baseline fails to complete the later pickup-and-placement stages. Fig. 19 presents an OOD example in which the baseline misses the initial bowl grasp, whereas M3 completes the full stack-and-shelf sequence. For **Veggie Centering**, Fig. 20 shows a baseline failure at the final centering stage, with visible downward drift that perturbs the cloth. Fig. 21 presents a second clean case where the baseline reaches the transport stage but leaves the transfer incomplete.

Table 9: Per-phase and full-task success rates (%) for three real-world bimanual tasks, pooled over three evaluation rounds (clean: $3 \times 16 = 48$ trials; OOD: $3 \times 8 = 24$ trials per task). Full-task rows report the pooled rate together with the sample standard deviation across the three round-wise full-task success rates.

Task	Phase	Clean		OOD	
		Adapter	M3	Adapter	M3
Bottle Cleanup	Brown-bottle disposal	68.8	85.4	20.8	70.8
	Green-bottle lift	85.4	91.7	37.5	75.0
	Bimanual handover	54.2	79.2	16.7	62.5
	Final disposal	54.2	79.2	16.7	62.5
	Full task	41.7 $\pm$ 9.6	66.7 $\pm$ 3.6	16.7 $\pm$ 7.2	58.3 $\pm$ 19.1
Stack & Shelf	Bowl pickup	87.5	87.5	20.8	79.2
	Bowl stacking	45.8	75.0	16.7	70.8
	Stacked-bowl pickup	39.6	58.3	12.5	54.2
	Shelf placement	39.6	58.3	12.5	54.2
	Full task	27.1 $\pm$ 9.6	58.3 $\pm$ 3.6	12.5 $\pm$ 12.5	54.2 $\pm$ 7.2
Veggie Centering	Cucumber placement	91.7	97.9	25.0	87.5
	Eggplant placement	87.5	95.8	16.7	83.3
	Plate-rim grasp	75.0	89.6	12.5	70.8
	Plate centering	64.6	83.3	8.3	70.8
	Full task	64.6 $\pm$ 3.6	83.3 $\pm$ 9.6	8.3 $\pm$ 14.4	70.8 $\pm$ 7.2

Fig. 22 depicts an OOD example in which the baseline fails at an early grasp while M3 proceeds to completion. Across these cases, the qualitative differences align with the phase-level trends in Table 9.

Qualitative Attention Patterns. A core motivation of M3 is to reduce overfitting to spurious visual correlations, such as background textures or the robot’s own gripper. These distractions are prevalent when fusing high-dimensional multi-view inputs. As illustrated in Fig. 12, the baseline policy often exhibits attention misalignment. In the *Handover Block* and *Click Bell* tasks, the baseline’s attention maps are scattered across the table surface and irrelevant regions. This lack of focus is often accompanied by failures to localize the target object or the correct transfer point. In contrast, the M3-trained policy shows more concentrated attention in the shown examples. The heatmaps suggest that the model attends more consistently to task-relevant semantic regions, including the object to be grasped, the receiving hand, contact points, or the final placement pad. These qualitative patterns are consistent with the hypothesis that stochastic masking of arm views and queries during training may encourage the model to rely less on transient, salient pixel features. We also note that M3 does not completely remove attention misalignment in Clean2Rand. In several failure cases, distractor backgrounds, lighting variation, or visually salient arm regions still attract attention away from the true contact area. We attribute part of this gap to training policies on clean demonstrations while Clean2Rand evaluates them on scenes with amplified nuisance factors.

Qualitative Execution Stability. Temporal execution differences are qualitatively consistent with the attention patterns discussed above. Fig. 12 provides a timeline comparison between the baseline and M3. In the *Click Bell* task (Fig. 12, bottom), the baseline policy struggles to ground the bell instruction, resulting in a prolonged search phase ($T = 40$ s) that ultimately ends in failure. In the shown rollout, the M3 policy appears to locate the target earlier and executes the click action in $T = 6$ s. Similarly, in the *Handover Block* task (Fig. 12, top), the baseline policy inadvertently knocks over the target object during the interaction. This collision drives the scene into an out-of-distribution state where the policy becomes disoriented and unable to recover, causing the execution to stall at

$T = 80$ s. By contrast, the M3 policy completes the successful handover and placement in 24s in the shown example. Across these examples, more focused attention coincides with more stable execution.

Qualitative Long-Horizon Rollout Comparisons. Bimanual manipulation often requires sequential reasoning, where early minor errors can compound into catastrophic failure. Fig. 13 illustrates comparative rollouts for complex, long-horizon tasks. In the *Block Rank Size* task (Fig. 13, top), which requires sorting objects by size, the baseline correctly grasps the first block but fails to maintain the logical ordering during placement, leading to a ranking error at $T = 120$ s. In the shown example, M3 successfully arranges the blocks from largest to smallest by $T = 44$ s. Furthermore, in the *Put Bottles Dustbin* and *Handover Block* tasks (Fig. 13, middle and bottom), the baseline often fails during contact-rich phases. In the *Put Bottles Dustbin* task, the policy makes unstable contact during the manipulation process. These errors accumulate over time and result in the target bottle being knocked over, as observed in the final failure state at $T = 169$ s. In contrast, M3 completes these intricate sequences without disturbing the environment state in the shown rollouts. Overall, these examples are qualitatively consistent with improved long-horizon stability under M3.

Discussion. Collectively, these visualizations are consistent with the possibility that the dynamic visibility constraints introduced by M3 *may function as a regularization mechanism*. By encouraging the model to extract complementary evidence from vision and language under partial observability, the strategy may reduce reliance on spurious correlations. We hypothesize that such stochastic occlusion may penalize reliance on salient but task-irrelevant features and encourage the model to anchor its decision-making on more durable semantic and geometric cues. These qualitative patterns may help explain the more focused attention maps, lower execution instability, and improved long-horizon performance observed in our experiments.

NOTE: See Fig. 12 for attention maps, Fig. 13 for long-horizon rollouts, and Figs. 14–22 for real-world cases on the following pages.

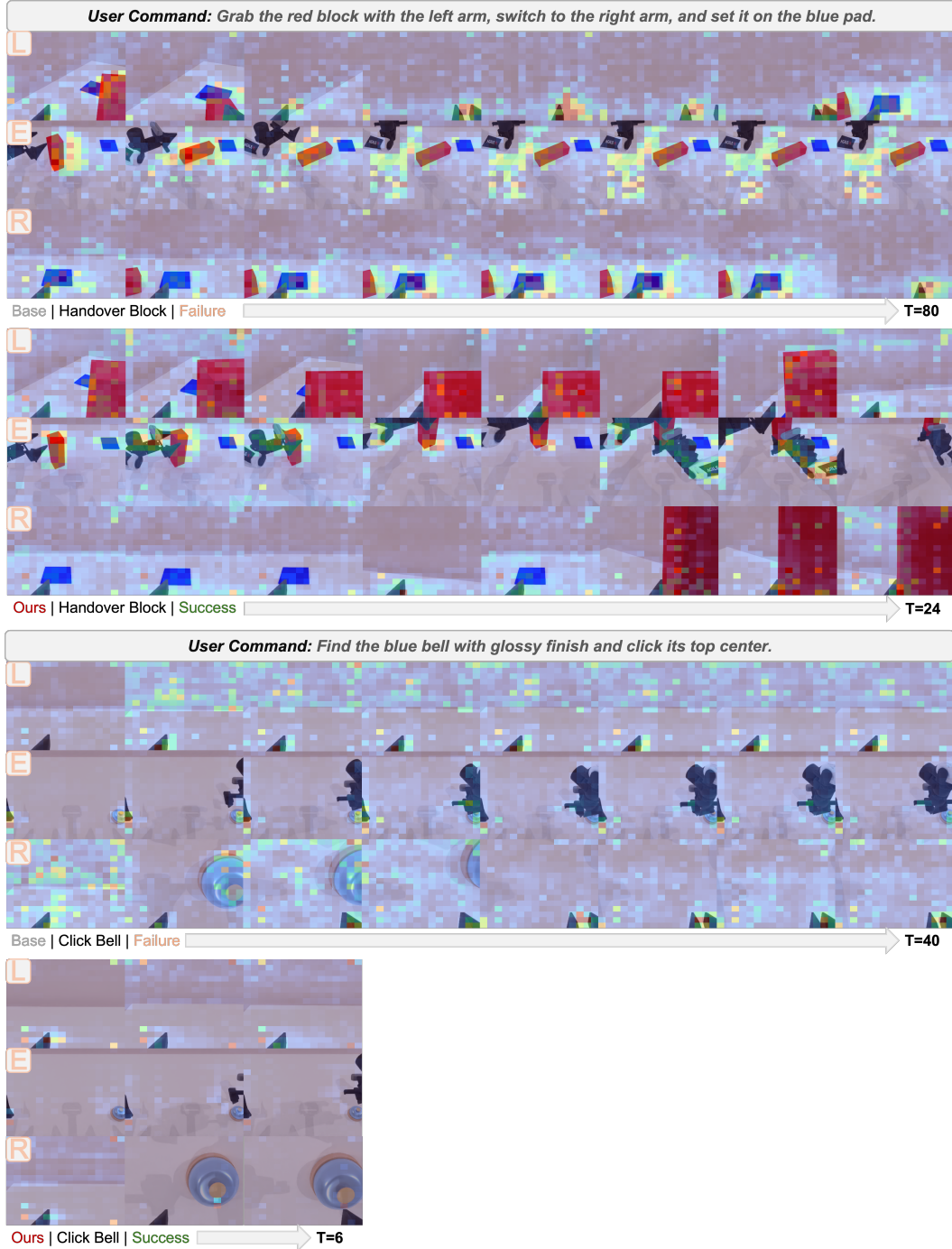


Figure 12: Qualitative attention maps and rollout timelines comparing the baseline with M3 across representative bimanual tasks.

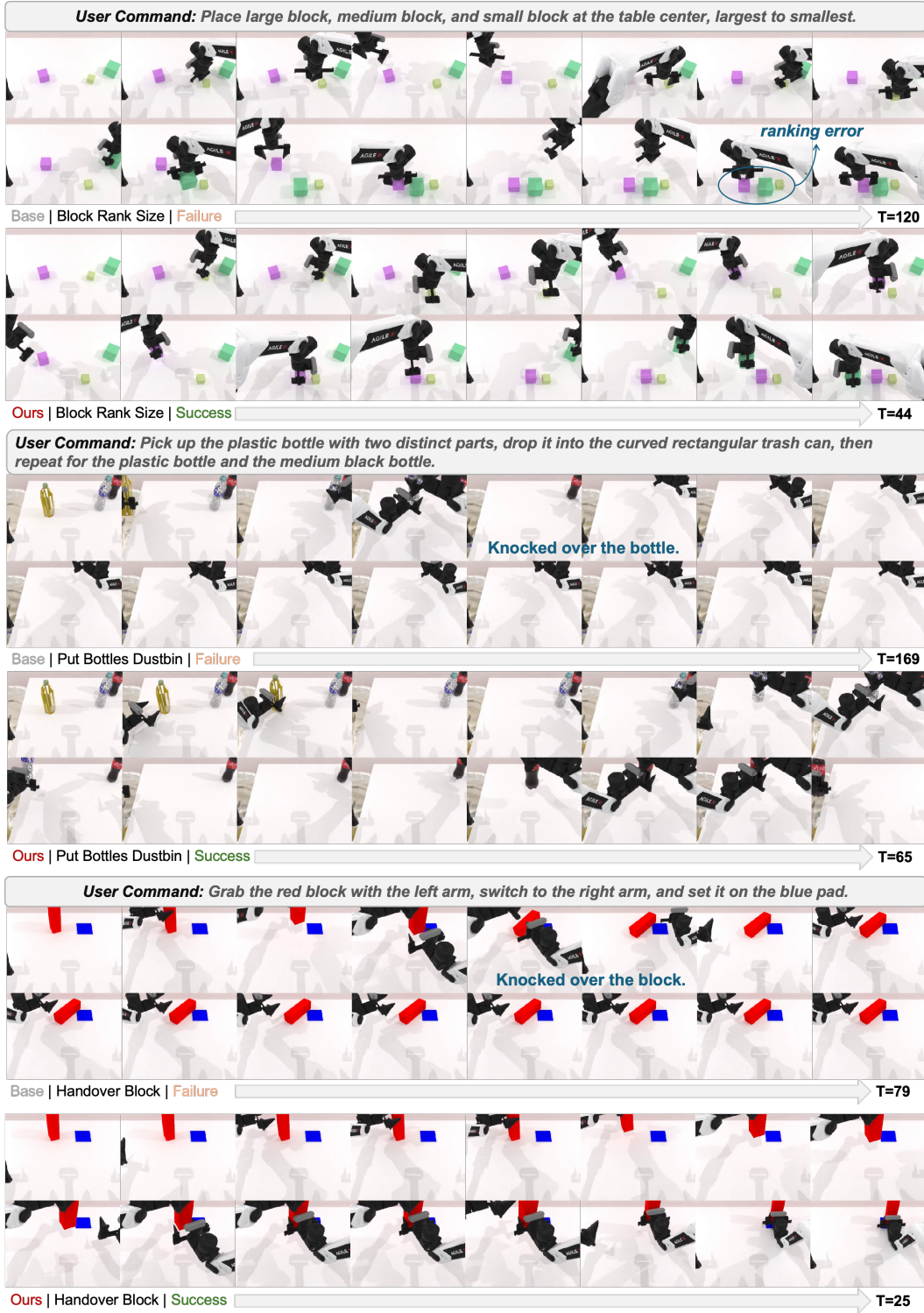


Figure 13: Qualitative rollout comparisons on representative long-horizon bimanual manipulation tasks.

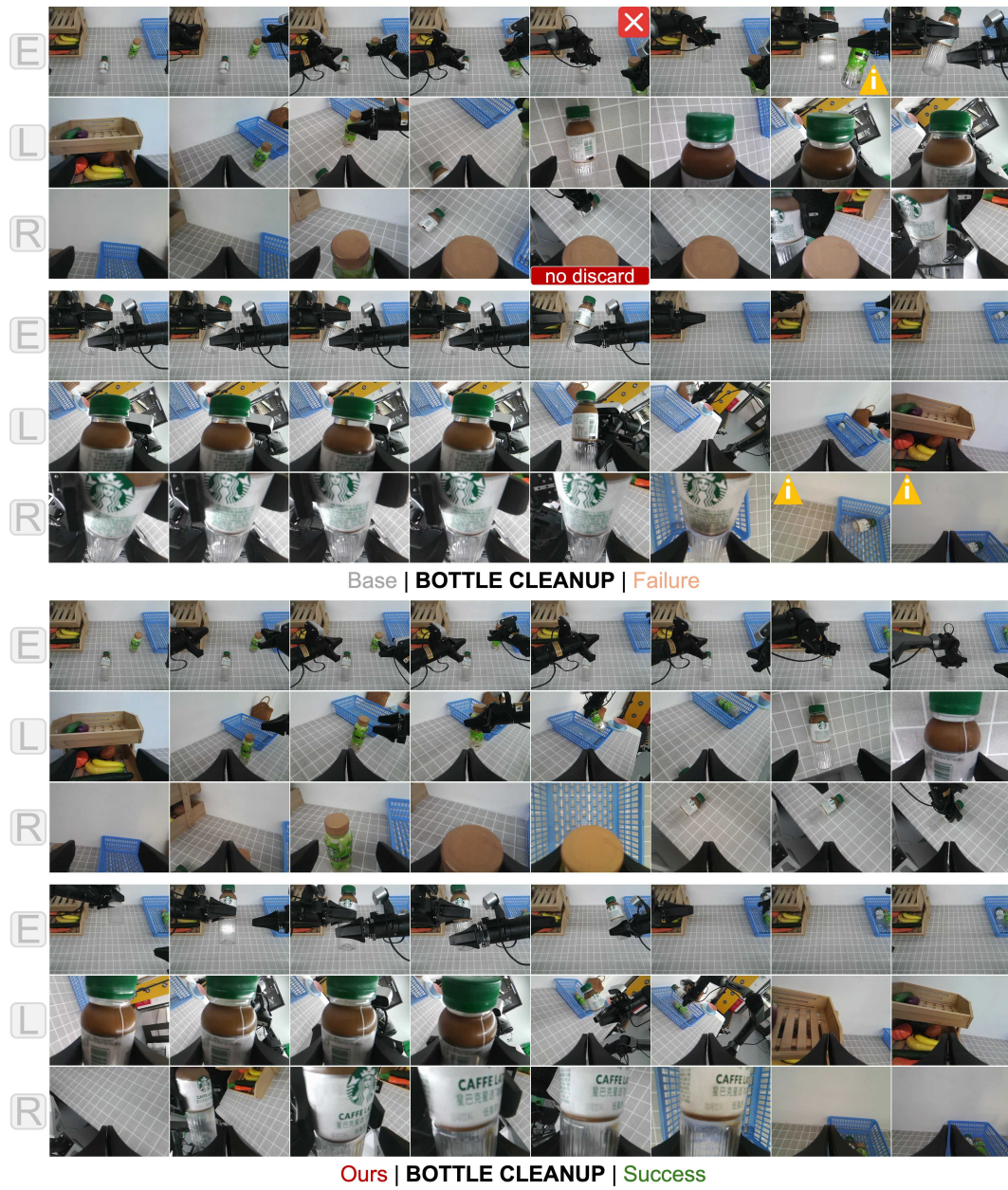


Figure 14: Bottle Cleanup: Case 1.

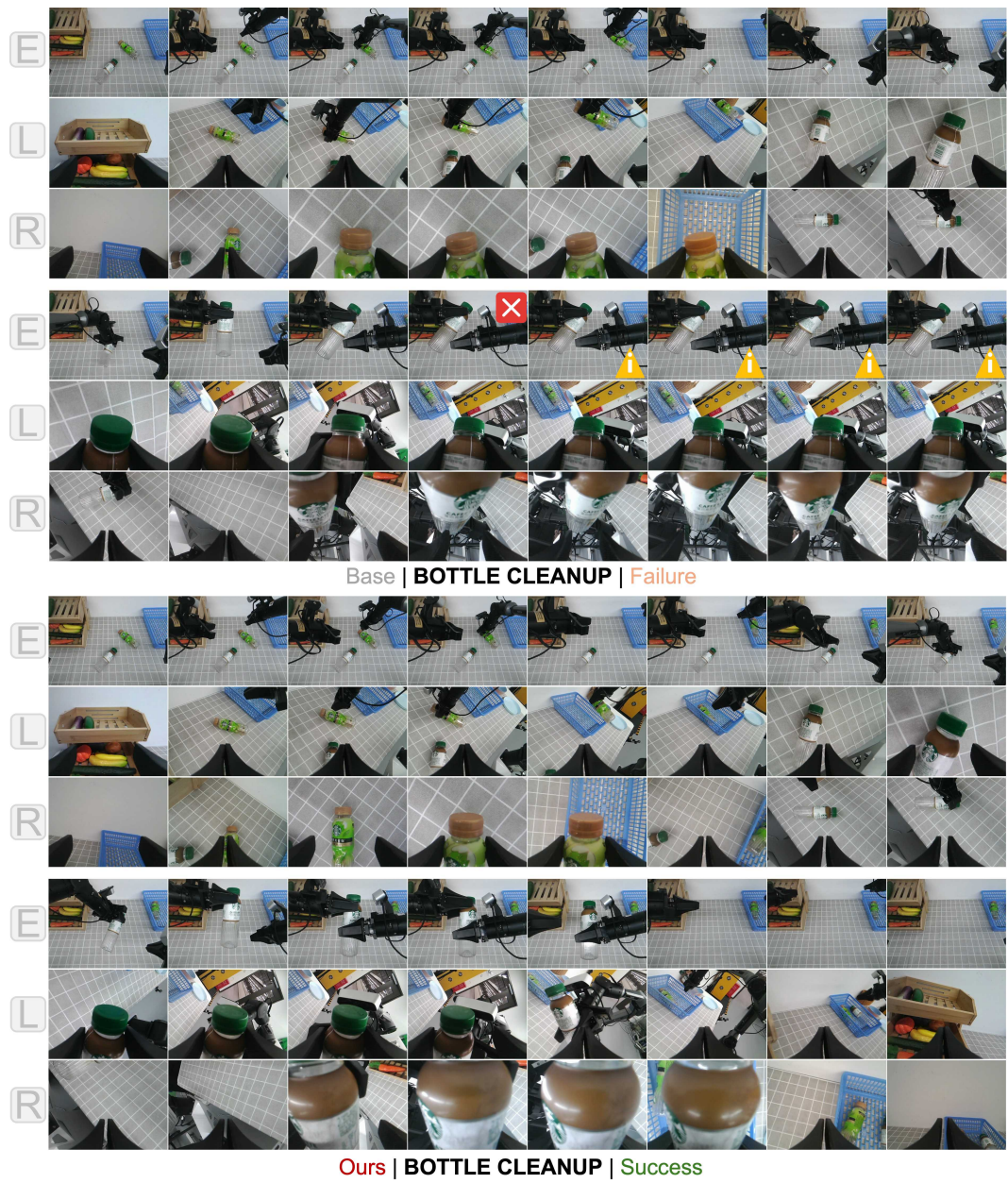


Figure 15: Bottle Cleanup: Case 2.

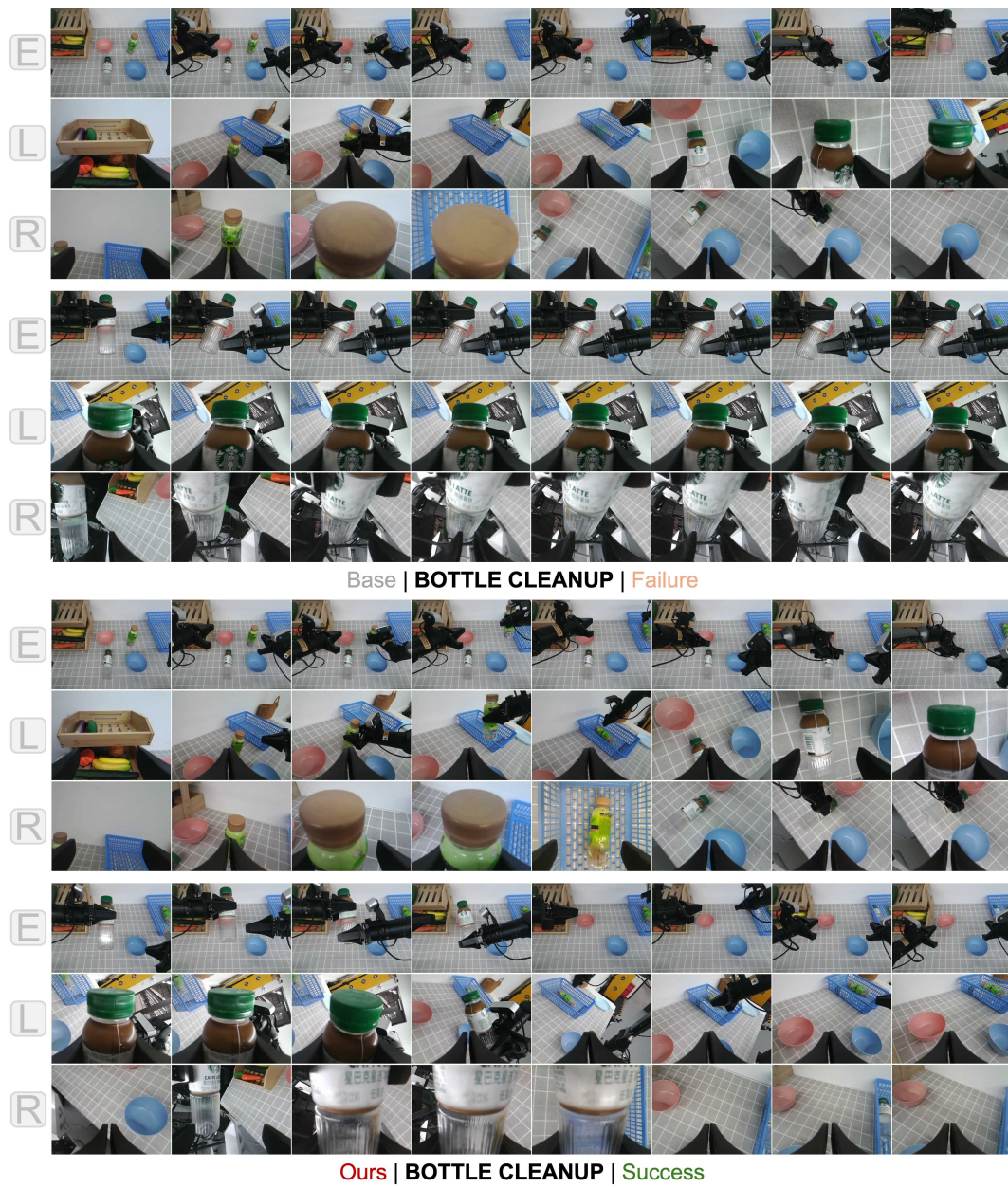


Figure 16: Bottle Cleanup: OOD Clutter.



Figure 17: Stack & Shelf: Case 1.

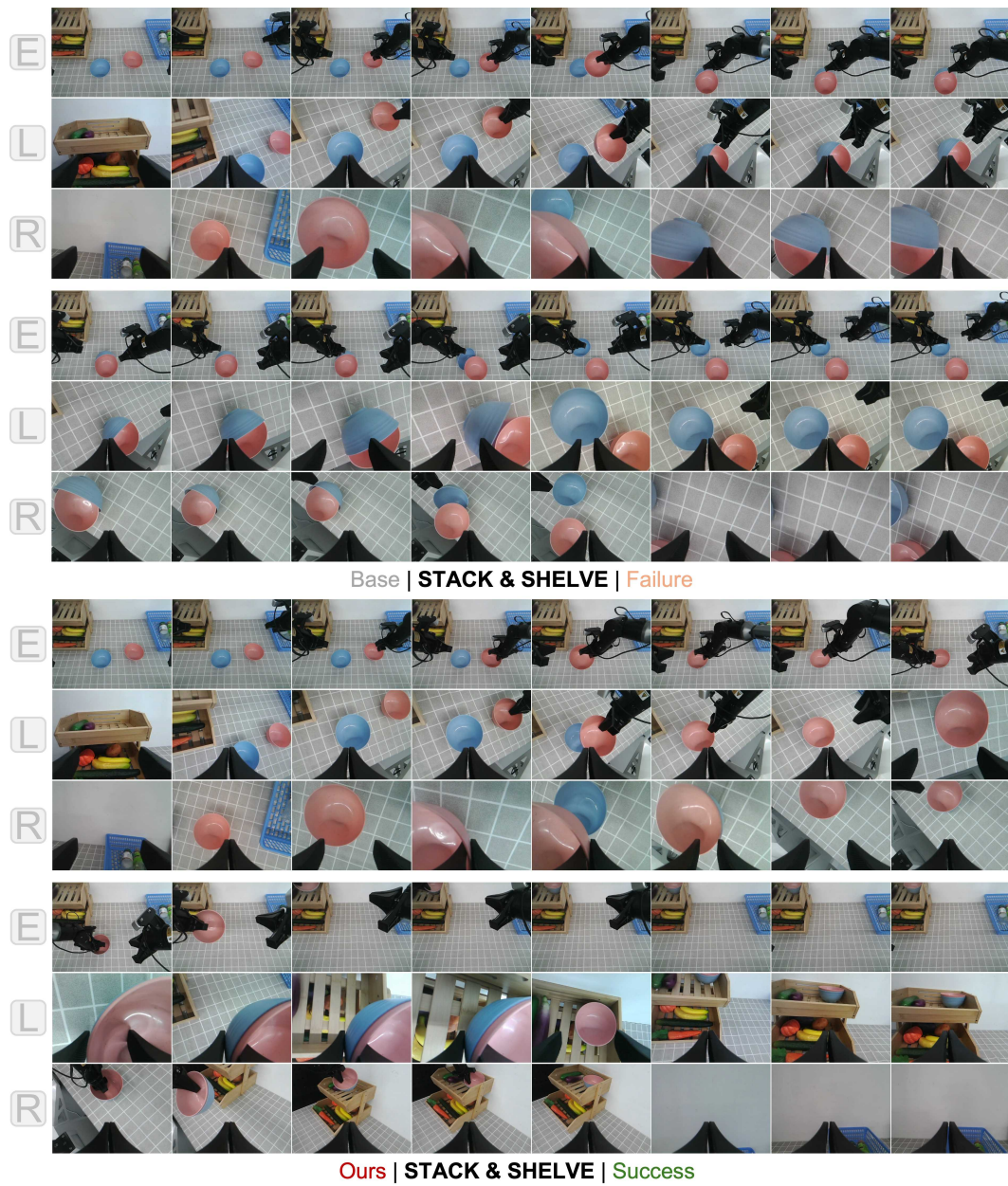


Figure 18: Stack & Shelf: Case 2.

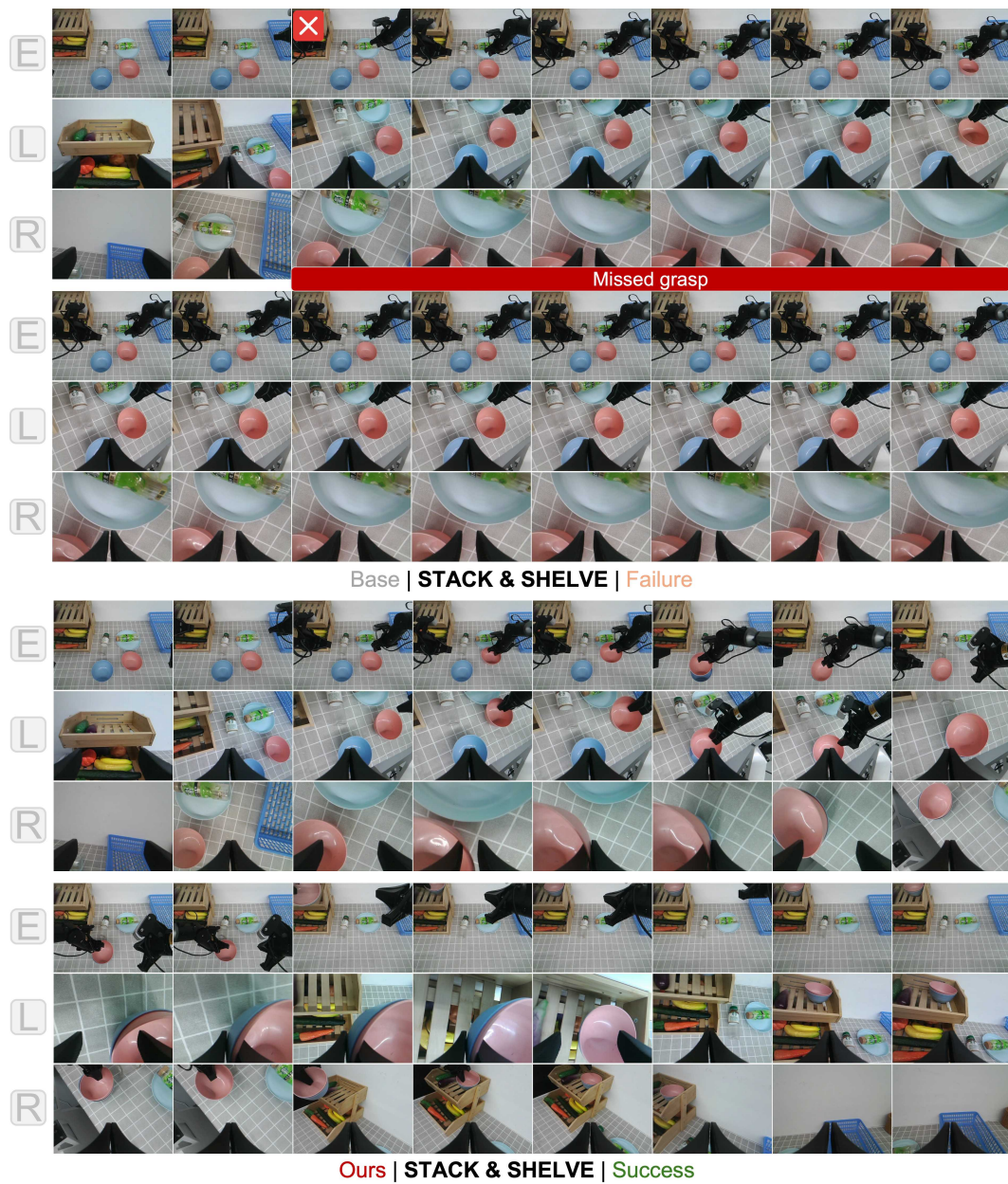


Figure 19: Stack & Shelf: OOD Clutter.

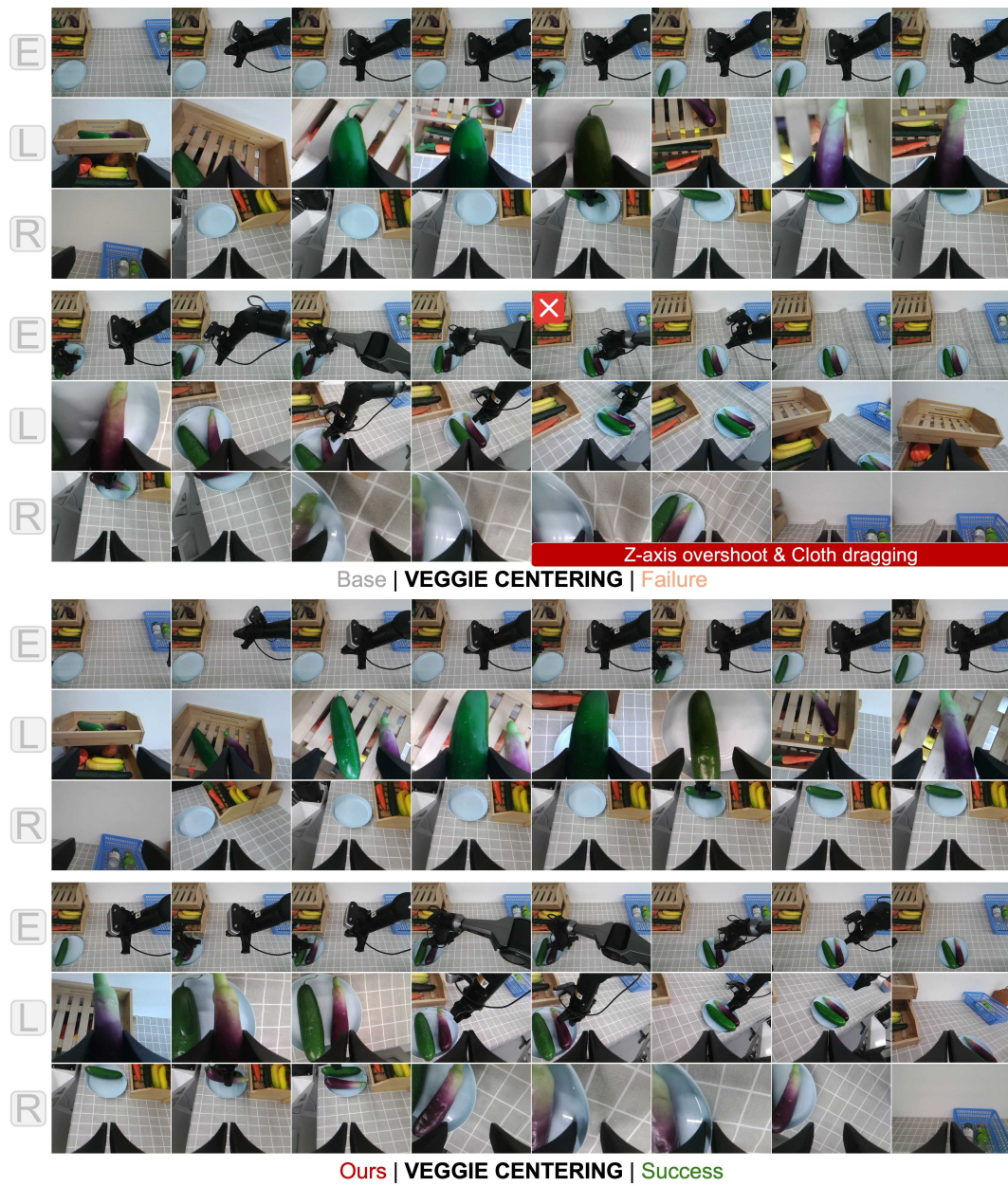


Figure 20: Veggie Centering: Case 1.

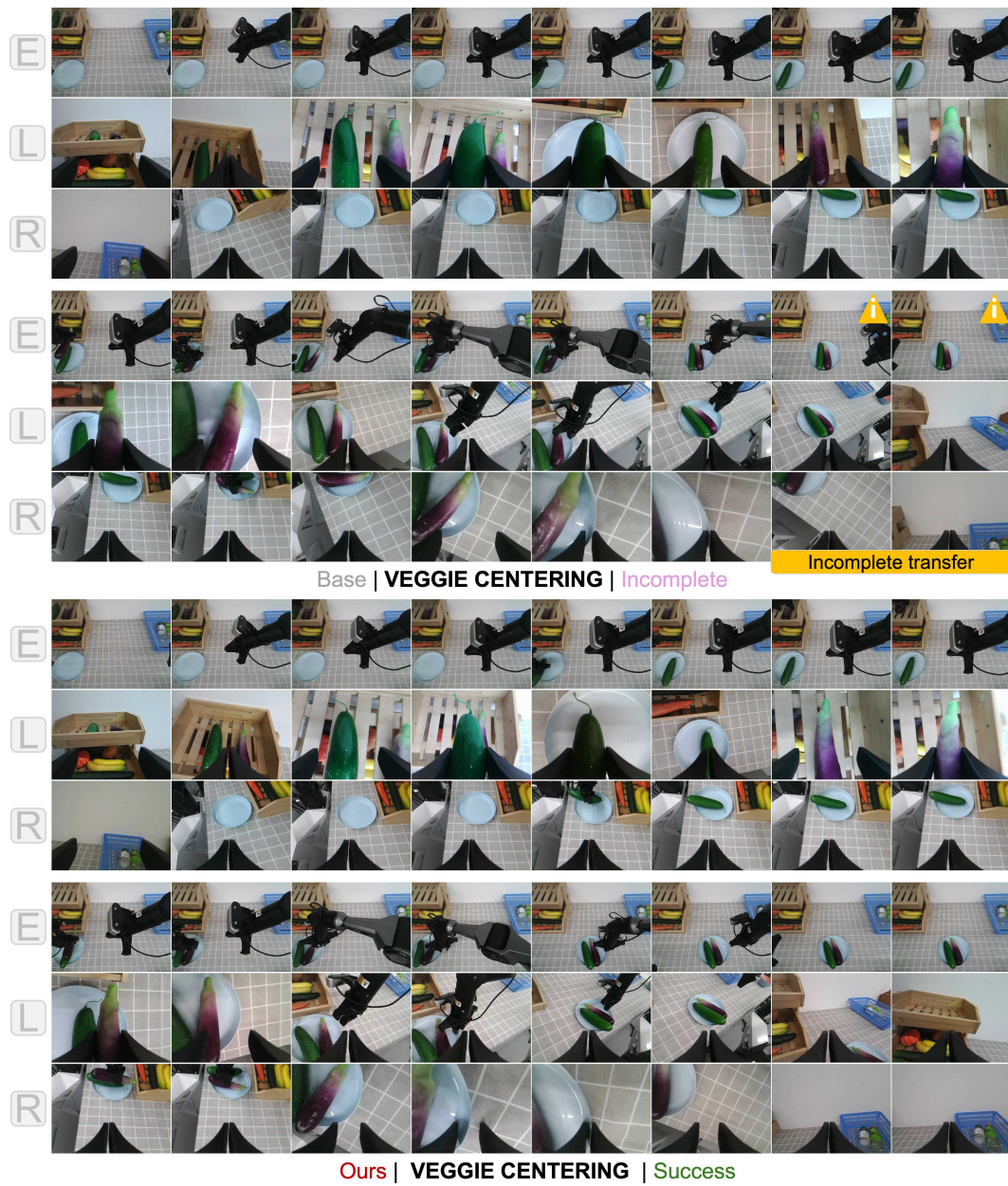


Figure 21: Veggie Centering: Case 2.

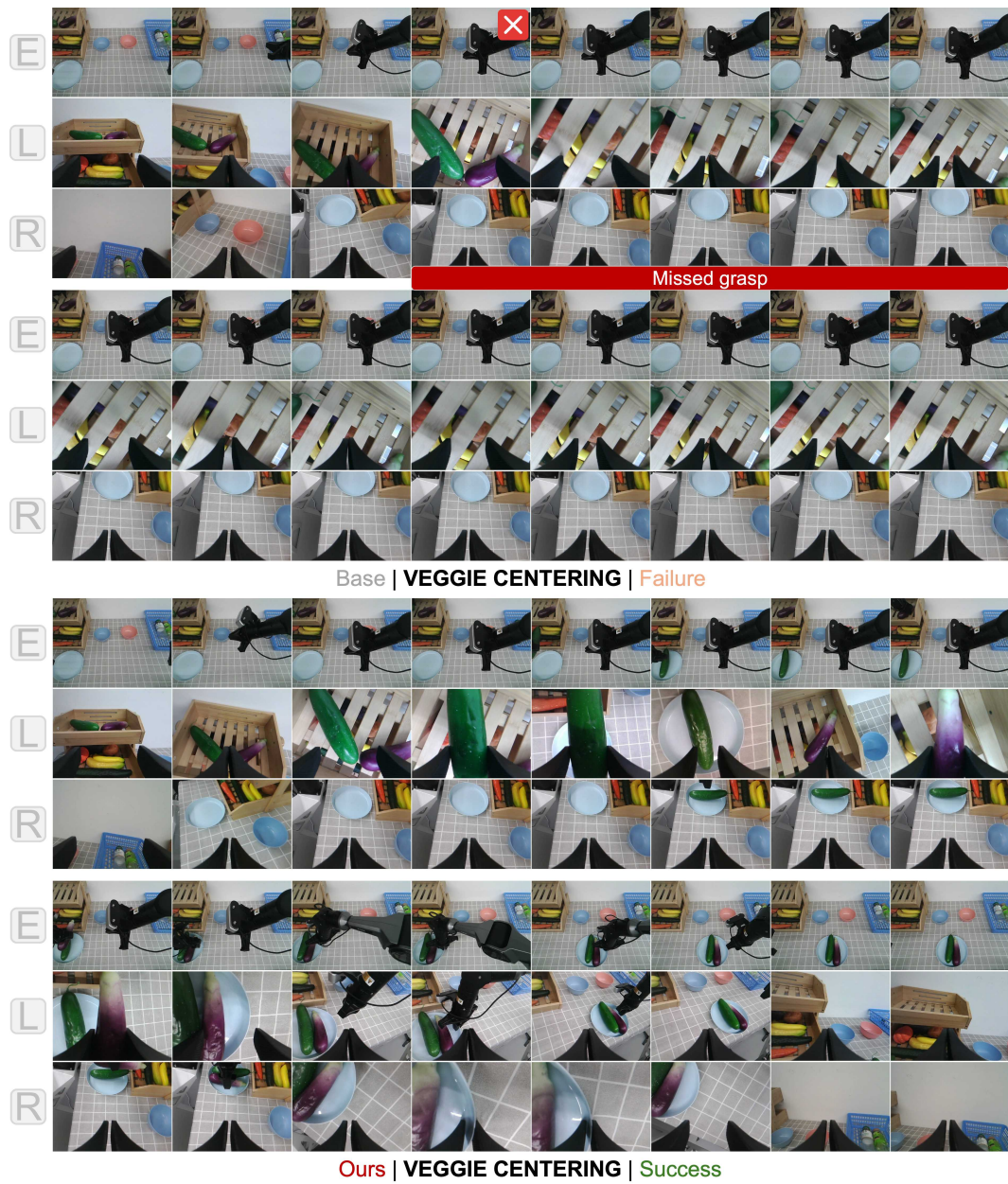


Figure 22: Veggie Centering: OOD Clutter.