LiMA: Bridging Long-term Imagination to Real-time Dexterous Manipulation via Asynchronous Diffusion

Ning Chen[1,2]*, Yankai Fu[1,2]*, Junkai Zhao[2]†, Qianpu Sun[1], Guocai Yao[2], Pengwei Wang[2], Zhongyuan Wang[2], Shanghang Zhang[1,2]✉

[1]State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University; [2]Beijing Academy of Artificial Intelligence

*Equal contribution, †Project leader, ✉Corresponding author

Project Webpage: LiMA Project

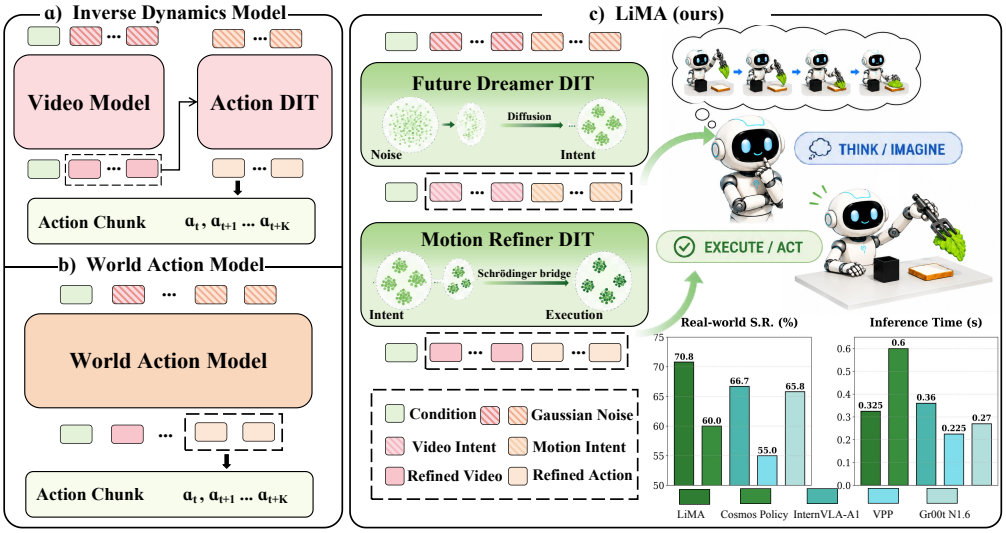


Figure 1: Unlike (a) Inverse Dynamics which sequentially generate observations before predicting actions, or (b) World Action Models that jointly denoise images and actions, we propose (c) LiMA, an asynchronous hierarchical framework. By decoupling long-term imagination from high-frequency execution, LiMA bridges latent foresight with multi-scale refinement, ensuring both strategic planning and real-time responsiveness across diverse horizons.

Abstract: Dexterous manipulation demands long-term foresight and rapid reactive control. Vision-Language-Action (VLA) models, while proficient in high-level reasoning, often lack a fine-grained understanding of physical dynamics and spatial perception. Conversely, World-Action Models (WAMs) typically suffer from high inference latency due to iterative generation. These deficiencies result in a critical temporal misalignment where the model's intent fails to adapt to rapid physical contact changes. To overcome this fundamental bottleneck, we propose **LiMA**, an asynchronous dual-system generative framework that systematically decouples intent planning from reactive execution. LiMA organizes computation into a multi-scale hierarchy: a slow system handles sparse long-horizon spatiotemporal intent generation, while a fast system focuses on dense high-frequency motion refinement. To align sparse intent predictions with dense action trajectories, we introduce a Latent Schrödinger Bridge Coupling mechanism that formulates refinement as an entropy-regularized probabilistic transport process. LiMA reduces inference latency by **45.8%** compared with Cosmos-Policy via asynchronous decoupling. Evaluated across six bimanual dexterous manipulation tasks spanning multiple horizons, LiMA achieves an overall success rate of **70.8%** and an average subtask success rate of **78.9%**, while maintaining performance in unseen scenarios.

Keywords: Dexterous Manipulation, World Action Model, VLA

10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

1 Introduction

Dexterous manipulation represents one of the most sophisticated challenges in robotics, requiring a seamless integration of long-term temporal foresight and rapid reactive precision [1, 2, 3]. Recent Vision-Language-Action (VLA) models have made substantial progress by scaling pretraining on large egocentric manipulation datasets [4, 5], enabling them to acquire broad visual, linguistic, and action priors for dexterous tasks [6, 7, 8, 9, 10, 11, 12]. Despite these advancements, existing VLA models still lack an explicit mechanism for anticipating and responding to the continuous evolution of the physical world, which is crucial for robust and generalizable dexterous manipulation.

World-Action Models (WAMs) have emerged as a promising paradigm, leveraging pretrained video generative models [13, 14] to internalize pixel-level physical world knowledge and exhibiting superior generalization and generative diversity. Current WAM frameworks generally follow two paths: (1) Inverse Dynamics Models (IDMs) [15, 16] employ a video model to "imagine" future observations as a conditional prior for action generation, improving policy generalization through future-aware visual priors. However, this sequential conditioning design loosely couples visual imagination with action prediction, limiting cross-modal feature interaction. (2) Integrated WAMs [17, 18, 19, 20] perform joint image-action generation within a unified denoising process, enabling richer spatiotemporal interactions. Yet, they are hindered by an excessively long latent feature space that leads to prohibitive inference latency, failing to meet the high-frequency reactive requirements of dexterous manipulation.

In this work, we introduce **LiMA**, an asynchronous multi-scale framework for jointly generating long-term foresight and robotic actions. LiMA employs a dual-system architecture that coordinates generation across distinct temporal scales, enabling rapid physical execution without sacrificing long-horizon planning. To maintain coherence between the two systems, we propose a Latent Schrödinger Bridge Coupling mechanism that models the transition from abstract intent to motor commands as a continuous probabilistic transport process, tightening the feature coupling between high-level imagination and low-level refinement. This integration is further enhanced by Spatiotemporal Adaptive Modulation, which grounds long-term foresight into real-time physical observations.

To evaluate LiMA, we conduct extensive experiments on diverse dexterous manipulation tasks across varying temporal horizons. Comparative analysis demonstrates that LiMA achieves a stronger balance between task success and execution efficiency. Compared with state-of-the-art VLA [21, 22] and WAM [16, 17] baselines, LiMA improves success rates especially in long-horizon and contact-intensive tasks, where stable task-level intent and fine-grained reactive correction are both required. Compared with integrated WAM baselines [17], LiMA substantially reduces inference latency through asynchronous coordination between long-range imagination and high-frequency action refinement, while maintaining their latent-level coupling. In summary, our contributions are as follows:

- We present **LiMA**, an asynchronous dual-system framework that decouples long-range imagination from reactive execution, overcoming generative inference bottlenecks.
- We introduce a Latent Schrödinger Bridge mechanism with adaptive modulation to ensure tight feature alignment between semantic intent and physical action.
- We provide a thorough evaluation demonstrating that LiMA achieves competitive results on real-world tasks while offering superior computational efficiency for real-time applications.

2 Related Work

2.1 Dexterous Manipulation

Dexterous manipulation poses a pioneering challenge due to high-dimensional action spaces and complex contact dynamics [23, 24, 25, 26, 27]. Early point-cloud-based methods [28, 1, 29] handled contact physics but lacked semantic depth. To address this, recent Vision-Language-Action (VLA) models [30, 31, 32, 33, 34] distilled large-scale reasoning priors into control, yet they often fail to ground abstract intent into fine-grained physical interactions. To better capture environment physics,

World-Action Models (WAMs) [35, 36, 37, 38, 39, 40] employ video synthesis as a generative prior, leveraging pixel-level dynamics to internalize robust physical world knowledge and improve out-of-distribution generalization. However, existing WAMs face a critical trade-off: sequential Inverse Dynamics Models (IDMs) suffer from sparse feature coupling, while unified generative models are bottlenecked by the computational cost of joint denoising. In contrast, LiMA introduces an asynchronous hierarchical framework that leverages video-based foresight while maintaining high-frequency reactive execution, reconciling physical grounding with real-time responsiveness.

2.2 Hierarchical Planning and Control in Robotics

To harmonize long-horizon reasoning with real-time sensorimotor control, recent literature has shifted toward dual-system hierarchies, loosely inspired by the cognitive division between deliberation and reaction [41, 42, 43]. Initial explorations focused on synchronous coordination, where a high-level cognitive model (System 2) generates latent embeddings to condition a lower-level policy head (System 1) [44, 45, 46, 47]. Despite their structural clarity, these methods are often bottlenecked by the lower sampling frequency of the deliberative module, failing to exploit the potential of System 1 for agile, high-rate execution. To break this temporal constraint, asynchronous architectures have emerged [48, 49, 50, 51], decoupling the refresh rates of task-level foresight and action-level response. However, current asynchronous designs treat the output of System 2 as a static extrinsic condition. This superficial coupling leads to fragile alignment and poor inheritance of reasoning priors. LiMA bridges this gap by interlinking both systems at the noise level via a distributional diffusion bridge. Instead of simple concatenation, our approach anchors the reactive controller to the generative manifold of System 2, ensuring multimodal consistency with real-time agility.

3 Method

3.1 Preliminaries

We formulate the sequential decision-making process of bimanual dexterous manipulation as a conditional generative problem. At each execution time step t, the system receives multi-modal sensory inputs comprising a language task instruction L, the robot's current proprioceptive joint state S_t, and multi-view visual observations consisting of three camera streams: a head-mounted ego-centric view and two wrist-mounted views, denoted as $\mathbf{O}_t = \{I_t^{\text{ego}}, I_t^{\text{left}}, I_t^{\text{right}}\}$.

The global objective of the generative policy is to model the conditional joint probability distribution to predict a synchronized action chunk $\mathcal{A}$ over a future temporal horizon, which encapsulates the coordinated trajectories of both the robotic arms and all dexterous fingers:

$$\mathcal{A}_{t:t+K} \sim p\left(\mathcal{A} \mid \mathbf{O}_t, S_t, L\right) \tag{1}$$

To parameterize this inter-system transition, we adopt the Schrödinger Bridge framework [52], which constructs an entropy-regularized optimal transport path between two distributions. Unlike standard diffusion generation, which typically starts from an unstructured Gaussian prior, the Schrödinger Bridge provides a structured boundary-to-boundary formulation. In our setting, this allows the reactive controller to start from the Dreamer's strategic intent prior X_1 and progressively refine it toward the Refiner's clean execution latent X_0. Specifically, the intermediate state X_t is sampled from a conditional Gaussian posterior $q(X_t \mid X_0, X_1)$ along a deterministic interpolation path. This formulation enables a direct bridge-matching objective to learn the transition from the Dreamer's intent prior to the Refiner's execution latent, rather than relying on an unconditional noise prior. The detailed derivation and variance scheduling are provided in the Appendix.

3.2 Model Architecture

LiMA adopts a dual-DiT [53] architecture to integrate high-level generative imagination with low-level reactive control. As illustrated in Figure 2, the framework hierarchically channels the preliminary

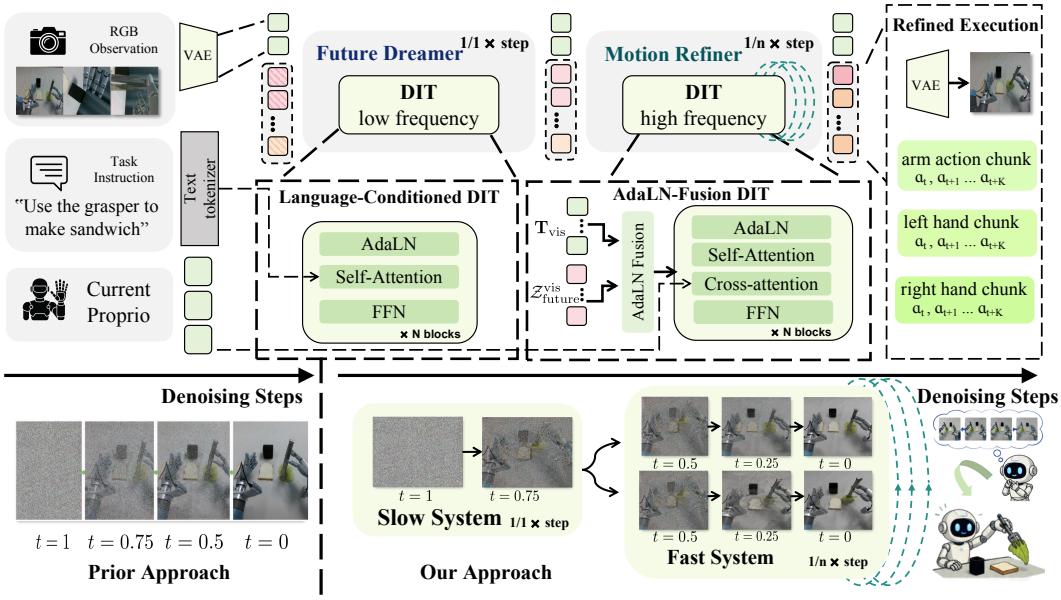


Figure 2: **Overview of LiMA.** LiMA asynchronously coordinates long-horizon foresight and real-time reactive control through a dual-system architecture. The Future Dreamer produces structured intent, while the Motion Refiner performs high-frequency latent refinement. A Latent Schrödinger Bridge Coupling mechanism maintains latent-level coherence between the two systems for manipulation.

denoising output of a Future Dreamer—which synthesizes a long-term spatiotemporal task imagination—as an informed latent prior into a Motion Refiner. This enables LiMA to effectively couple global strategic foresight with real-time action execution within a unified latent feature space.

Unified Multimodal Representation. LiMA maps heterogeneous sensory inputs into a unified latent sequence within a shared embedding space $\mathbb{R}^D$. Specifically, we employ a pre-trained Wan2.1 spatiotemporal VAE tokenizer [54] to compress multi-view visual frames $\mathbf{O}_t$ into visual tokens $\mathbf{T}_{\text{vis}}$, and a frozen T5-XXL encoder [55] to extract language embeddings $\mathbf{L}$. Crucially, the robot's proprioceptive state $\mathbf{S}_t$ is directly unified into the visual feature dimension via a structural repetitive padding mechanism to form state tokens $\mathbf{T}_{\text{state}}$, circumventing parametric projection layers.

Future Dreamer Flow (Slow System). The Future Dreamer ($\mathcal{G}_{\text{dream}}$) models long-horizon spatiotemporal task evolution as the slow branch of LiMA. It is initialized from a pre-trained Cosmos-Predict2-2B [56] to inherit physical world priors, and is further fine-tuned on task-specific demonstrations. The Dreamer adopts an asymmetric conditioning design: visual tokens $\mathbf{T}_{\text{vis}}$ form the main denoising stream, while language embeddings $\mathbf{L}$ modulate Transformer activations through Adaptive Layer Normalization (AdaLN) [57]. After preliminary denoising, it outputs a long-horizon intent representation $\mathcal{Z}_{\text{int}} = [\mathcal{Z}^{\text{vis}}_{\text{future}}, \mathcal{Z}^{\text{act}}_{\text{future}}]$, which contains imagined multi-view visual latents and a coarse macro-action prior. This intent representation is organized to match the Refiner's latent interface, allowing it to serve as the structured prior boundary for bridge refinement.

Motion Refiner Flow (Fast System). The Motion Refiner ($\mathcal{G}_{\text{refine}}$) acts as the fast branch for high-frequency action refinement. Unlike the pre-trained Dreamer, $\mathcal{G}_{\text{refine}}$ is a smaller 1B-parameter Transformer trained from scratch, enabling it to specialize in fine-grained motor coordination. Given the intent prior $\mathcal{Z}_{\text{int}}$, the Refiner uses a decoupled conditioning scheme that separately handles imagined visual latents, coarse action priors, and real-time proprioceptive states. It predicts a joint execution latent $\mathcal{Z}_{\text{out}} = [\mathcal{Z}^{\text{act}}_{\text{out}}, \mathcal{Z}^{\text{vis}}_{\text{out}}]$, which contains the near-term action chunk and the short-horizon ego-centric visual latent. The coarse action prior $\mathcal{Z}^{\text{act}}_{\text{future}}$ is replicated along the action horizon to initialize the action refinement trajectory, while the imagined visual prior $\mathcal{Z}^{\text{vis}}_{\text{future}}$ is fused with online visual tokens through a localized adaptive modulation layer, as shown in Figure 2.

Specifically, we exploit the spatial correspondence between imagined future views and streaming observations. For each camera slot i, the corresponding online visual tokens $\mathbf{T}^{(i)}_{\text{vis}}$ and imagined visual

prior $\mathcal{Z}_{\text{future}}^{\text{vis},(i)}$ are integrated through adaptive modulation:

$$\mathcal{T}_{\text{mod}}^{(i)} = \gamma(\mathcal{Z}_{\text{future}}^{\text{vis},(i)}) \cdot \text{LayerNorm}(\mathbf{T}_{\text{vis}}^{(i)}) + \beta(\mathcal{Z}_{\text{future}}^{\text{vis},(i)}), \tag{2}$$

where γ and β are affine scale and shift parameters predicted from the imagined visual prior. This modulation allows the Refiner to inject long-horizon visual intent into real-time observations while preserving sensitivity to local scene changes. In parallel, the instantaneous state tokens $\mathbf{T}_{\text{state}}$ are projected into continuous embedding streams and used as proprioceptive conditioning signals throughout the Refiner, keeping motion refinement grounded in the latest robot state.

3.3 Latent Schrödinger Bridge Coupling

Building on the I2SB formulation detailed in the Appendix, we construct a latent bridge between the Dreamer and the Refiner within a shared latent space. Instead of starting action refinement from an unconditional Gaussian prior, we use the Future Dreamer's denoised intent representation $\mathcal{Z}_{\text{int}}$ as the structured prior boundary, corresponding to X_1, and the ground-truth execution latent $\mathcal{Z}_{\text{out}}^{(0)}$ as the clean data boundary, corresponding to X_0. Since $\mathcal{Z}_{\text{int}}$ is organized to match the Refiner's latent interface, the bridge can be constructed directly without an additional alignment projection.

Given this boundary pair, the intermediate reactive bridge latent $\mathcal{Z}_{\text{out}}^{(t)}$ is sampled from the conditional posterior:

$$\mathcal{Z}_{\text{out}}^{(t)} \sim q\left(\mathcal{Z}_{\text{out}}^{(t)} \mid \mathcal{Z}_{\text{out}}^{(0)}, \mathcal{Z}_{\text{int}}\right) = \mathcal{N}\left(\mathcal{Z}_{\text{out}}^{(t)};\, \mu_t\left(\mathcal{Z}_{\text{out}}^{(0)}, \mathcal{Z}_{\text{int}}\right),\, \Sigma_t \mathbf{I}\right), \tag{3}$$

where the mean path μ_t defines the interpolation between the clean execution latent and the intent-conditioned prior, while Σ_t controls the stochastic perturbation along the bridge.

The Motion Refiner is trained to directly predict the clean execution latent from the intent-conditioned bridge state:

$$\mathcal{L}_{\text{refine}} = \mathbb{E}_{t, \mathcal{Z}_{\text{out}}^{(0)}, \mathcal{Z}_{\text{int}}}\left[\left\|\hat{\mathcal{Z}}_{\text{out}}^{(0)} - \mathcal{Z}_{\text{out}}^{(0)}\right\|_2^2\right], \quad \hat{\mathcal{Z}}_{\text{out}}^{(0)} = \mathcal{G}_\theta\left(\mathcal{Z}_{\text{out}}^{(t)}, t, \mathcal{Z}_{\text{int}}\right). \tag{4}$$

This objective encourages the Refiner to recover the clean joint execution latent from an intent-conditioned intermediate state, rather than denoising from an unconditional Gaussian prior.

During real-time inference, the clean boundary $\mathcal{Z}_{\text{out}}^{(0)}$ is unavailable. Therefore, we initialize the refinement process from the intent-conditioned prior:

$$\mathcal{Z}_{\text{out}}^{(1)} \sim \mathcal{N}\left(\mathcal{Z}_{\text{int}}, \sigma_1^2 \mathbf{I}\right). \tag{5}$$

Starting from this structured prior reduces the sampling burden of the Refiner and keeps the generated action trajectory aligned with the Dreamer's long-horizon intent. As a result, the fast system can focus on local motion correction while preserving consistency with the high-level foresight.

3.4 Asynchronous Dual-System Coordination

To deploy compute-intensive generative loops for real-time control, LiMA adopts an asynchronous temporal decoupling mechanism to resolve the inherent trade-off between cognitive capacity and inference latency. Specifically, while the Future Dreamer performs long-horizon spatiotemporal imagination to provide global strategic foresight, its heavy computational cost limits execution frequency. To bridge this mismatch, we introduce a periodic execution interval $\Delta T \in \mathbb{Z}^+$ to decouple the inference frequencies of the dual systems.

Formally, at update steps where $t \bmod \Delta T = 0$, the compute-heavy Future Dreamer is triggered to process the current observation $\mathbf{O}_t$ and language instruction L, generating a refreshed strategic intent prior $\mathcal{Z}_{\text{int},t}$ in the unified latent space. The Motion Refiner simultaneously consumes this updated prior together with $\mathbf{O}_t$ to produce the fine-grained execution chunk $\mathcal{Z}_{\text{out},t}$.

At intermediate non-update steps where $t \bmod \Delta T \neq 0$, the Future Dreamer bypasses inference and freezes its latent output. Meanwhile, the lightweight Motion Refiner remains active at full

Table 1: **Comparison of LiMA and baselines on Real-World Tasks.** Each experiment is evaluated with 20 trials. Inference speed is evaluated on a NVIDIA H100 GPU.

Method	Stack cup		Roll T-shirt		Cook Rice		Make Sandwich		Make Coffee		Assemble package		Time (ms)
	SR	PSR	SR	PSR	SR	PSR	SR	PSR	SR	PSR	SR	PSR	-
GR00T N1.6	85.0%	87.5%	70.0%	71.7%	70.0%	72.0%	65.0%	72.5%	55.0%	**71.7%**	**50.0%**	61.7%	270
VPP	80.0%	80.0%	55.0%	56.7%	55.0%	60.0%	50.0%	61.3%	50.0%	56.7%	40.0%	48.3%	**225**
Internvla-a1	**90.0%**	90.0%	**75.0%**	**88.3%**	70.0%	78.0%	65.0%	77.5%	50.0%	66.7%	**50.0%**	58.3%	360
cosmos-policy	80.0%	82.5%	60.0%	70.0%	60.0%	67.0%	60.0%	71.3%	55.0%	63.3%	45.0%	51.7%	600
LiMA (Ours)	**90.0%**	**95.0%**	**75.0%**	83.3%	**80.0%**	**82.0%**	**70.0%**	**80.0%**	**60.0%**	70.0%	**50.0%**	**63.3%**	325

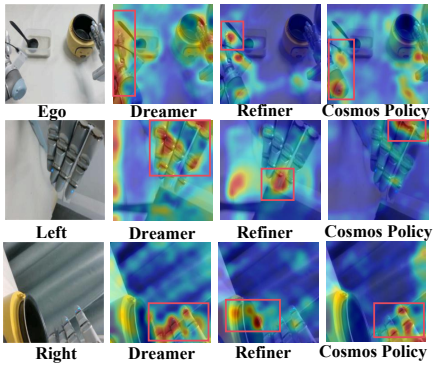


Figure 3: **Attention Map Comparison.**

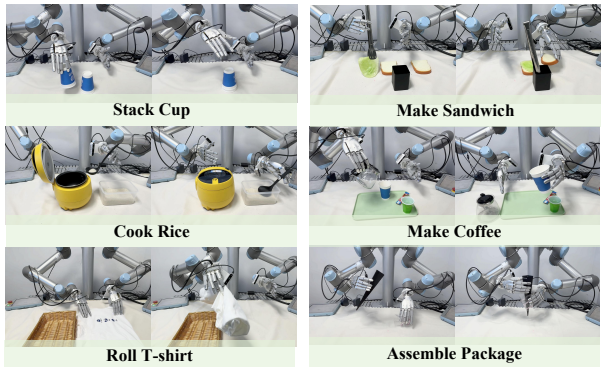


Figure 4: **Visualization of Dexterous Tasks.**

operational frequency, continuously integrating the latest observation $\mathbf{O}_t$ with the cached strategic prior $\mathcal{Z}_{\text{int},\lfloor t/\Delta T \rfloor \Delta T}$ to synthesize real-time actions. This sparse planning coupled with dense, high-frequency motion refinement enables robust reactivity under unforeseen environmental perturbations.

3.5 Training Objective

We train LiMA with a joint optimization objective over the Future Dreamer and the Motion Refiner. For the Future Dreamer, the learning objective $\mathcal{L}_{\text{dream}}$ is a joint-space score-matching loss that supervises the generation of long-horizon action chunks and three-view future visual latents. To improve the robustness of the Dreamer–Refiner interface during real-time inference, we introduce a Boundary Noise Injection regularizer on the conditioning path of the fast system. Specifically, during training, the Refiner receives a lightly perturbed copy of the intent latent with a small probability p_{bni}:

$$\tilde{\mathcal{Z}}_{\text{int}} = \begin{cases} \mathcal{Z}_{\text{int}} + \gamma \epsilon_{\text{noise}}, & \text{with probability } p_{\text{bni}}, \\ \mathcal{Z}_{\text{int}}, & \text{otherwise}, \end{cases} \qquad \epsilon_{\text{noise}} \sim \mathcal{N}(0, \mathbf{I}), \quad \gamma \ll 1. \tag{6}$$

Conditioned on $\tilde{\mathcal{Z}}_{\text{int}}$, the Motion Refiner is optimized with $\mathcal{L}_{\text{refine}}$ to directly recover the clean joint execution latent $\mathcal{Z}_{\text{out}}^{(0)}$ from an intent-conditioned bridge state. This target contains fine-grained dual-arm actions, left/right anthropomorphic hand motions, and near-term ego-centric visual latents. The total training objective is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{dream}} + \lambda \mathcal{L}_{\text{refine}}. \tag{7}$$

4 Experiment

4.1 Experiment Setup

Robot Setup. Our real-world evaluation platform consists of a bimanual setup featuring two 6-DoF UR5 robotic arms equipped with two 22-DoF SharpaWave five-fingered dexterous hands. The multimodal perception suite comprises three Intel RealSense D435 cameras: one head-mounted camera capturing the global ego-centric view and two wrist-mounted cameras capturing localized coordination views. Following [9], we use VIVE Trackers to capture the relative spatial trajectories of the human wrists for real-time end-effector control of the UR5 arms. Concurrently, human finger

configurations are recorded via MetaGlove Pro gloves and mapped onto the 22-DoF SharpaWave hands through a joint-space kinematic retargeting pipeline.

Tasks Setup. We evaluate LiMA across two short-horizon tasks—*Stack Cup*, *Roll T-shirt*—and four challenging long-horizon bimanual dexterous tasks, as shown in Figure 4: *Cook Rice*, *Make Sandwich*, *Make Coffee*, and *Assemble Package*. For each task, we collect 100 high-quality expert teleoperation demonstrations. During physical deployment, the policy is evaluated in 20 trials per task, and the success rate is reported as the main metric.

Baselines and Evaluation Protocol. We evaluate LiMA against four state-of-the-art baselines, including two Vision-Language-Action (VLA) models — Gr00T N1.6 [21] and InternVLA-a1 [22] — and two World-Action Models (WAM) —VPP [16] and Cosmos-Policy [17]. To rigorously quantify performance across diverse dexterous scenarios, we employ Success Rate (SR) to measure overall task completion and Progress Success Rate (PSR) as a fine-grained metric for subtask advancement.

4.2 Results and Analysis

Real-World Quantitative Analysis. LiMA achieves strong overall performance across the six real-world manipulation tasks, as shown in Table 1. On short-horizon tasks such as Stack Cup and Roll T-shirt, LiMA performs comparably to strong VLA baselines, indicating that the hierarchical generative design does not compromise basic manipulation accuracy. More importantly, on long-horizon and contact-intensive tasks such as Cook Rice and Make Sandwich, LiMA shows clearer advantages in task progress and execution stability, as reflected by its improved PSR and competitive SR. This suggests that the Dreamer's long-horizon intent helps the Refiner maintain task progress and reduce intermediate failures during complex bimanual interactions. Compared with unified WAM baselines, LiMA further improves the efficiency–performance trade-off, reducing inference latency from 600ms for Cosmos-Policy to 325ms, corresponding to a 45.8% latency reduction. These results show that LiMA maintains strong task performance while enabling more efficient real-time execution for high-frequency closed-loop dexterous manipulation.

Attention-Map Analysis. As shown in Figure 3, we visualize and compare the attention maps of action tokens across three camera views for Cosmos-Policy, which adopts a single DiT architecture, and LiMA, which consists of the Dreamer and the Refiner. In the Cook Rice task, the robot is required to grasp a ladle with the left hand while keeping the right hand relatively stationary, making both local manipulation accuracy and bimanual coordination important. We observe that Cosmos-Policy produces relatively dispersed attention, with noticeable weights assigned to regions that are less relevant to the current manipulation. In contrast, LiMA exhibits a clearer hierarchical division of attention. The Dreamer attends more to global spatial relationships and task-relevant context, supporting long-horizon intent generation, while the Refiner focuses more on localized manipulation regions for fine-grained grasping and real-time motion adjustment. These visualizations provide qualitative evidence that LiMA can separate long-term planning from local execution, leading to more stable interaction in complex bimanual manipulation tasks.

Latency Analysis. We measure end-to-end inference latency from receiving the multi-view observations to decoding an executable action chunk. Under the asynchronous execution schedule, the Dreamer is refreshed once every four Refiner updates ($\Delta T = 4$). We therefore report the amortized latency per Refiner update, computed as $T_{\text{avg}} = (T_{\text{Dreamer}} + 4T_{\text{Refiner}})/4$. LiMA requires 325 ms on average to produce a 32-step action chunk. Therefore, the reported value represents chunk-generation latency rather than latency per individual control action. For a fair comparison, all methods are evaluated on the same NVIDIA H100 GPU using the same batch size and numerical precision, without model-specific operator fusion, model compilation, quantization, or edge-device deployment optimization.

4.3 Ablation Study

Bridge Design Ablation. To isolate the contribution of the proposed I^2SB coupling, we compare LiMA with three controlled variants in Table 2: (i) removing future visual intent, (ii) initializing

Table 2: **Bridge design ablation.** Success rates (%) are reported as mean ± standard deviation over three seeds, with 20 real-world trials per seed.

Variant	Cook Rice	Stack Cup
w/o Future Visual Intent	61.7 ± 7.6	76.7 ± 2.9
Gaussian Init. + Cross-Attn	70.0 ± 5.0	78.3 ± 2.9
Flow Matching	73.3 ± 2.9	83.3 ± 5.8
LiMA	$\mathbf{78.3 \pm 2.9}$	$\mathbf{93.3 \pm 5.8}$

Table 3: **Ablation Studies.** We evaluate LiMA across three dimensions: architecture components, denoising steps, and system ratios ($N_{slow} : N_{fast}$). S.R.: Success Rate, Time: Inference Latency.

(a) Component Ablation			**(b) Denoising Steps**			**(c) System Ratio** ($N_s : N_f$)		
Component	S.R.	Time	Steps	S.R.	Time	Ratio	S.R.	Time
Full (LiMA)	**80%**	**0.325s**	5	65%	0.23s	1:9	50%	**0.28s**
w/o Adapt.	60%	-	10	80%	**0.325s**	2:8	65%	0.31s
w/o Tokens	55%	-	15	**85%**	0.43s	3:7	**80%**	0.325s
w/o Fast-Slow	80%	0.45s	20	80%	0.51s	4:6	70%	0.34s

the Refiner from Gaussian noise while injecting the Dreamer latent through cross-attention, and (iii) replacing I^2SB with Flow Matching under the same architecture and conditioning setup. Removing future visual intent leads to the largest performance degradation on both tasks, confirming that the imagined visual latent provides essential task-level guidance. The Gaussian-initialized cross-attention variant also underperforms LiMA. In this variant, the Dreamer latent is supplied only as an auxiliary condition, while the Refiner must construct the execution trajectory from an unstructured Gaussian source. Such indirect conditioning may be insufficient when the few-step Dreamer prediction remains coarse or imperfect. In contrast, I^2SB treats the Dreamer intent and target execution latent as paired boundary states and constructs a stochastic diffusion bridge with analytically tractable intermediate marginals. This formulation directly anchors the entire refinement trajectory to the structured intent prior, rather than relying on the network to inject intent indirectly through cross-attention. Although Flow Matching also transports probability mass between distributions, our standard Flow Matching variant learns a deterministic velocity field along a prescribed probability path. The stronger performance of I^2SB suggests that its stochastic, boundary-anchored formulation better accommodates the uncertainty and residual mismatch between coarse Dreamer intents and fine-grained Refiner executions.

Component Ablation. To validate the effectiveness of each design choice, we first ablate LiMA's major components, as shown in Table 3. Removing spatiotemporal adaptive modulation decreases SR from 80% to 60%, indicating that simple conditioning is insufficient to align the Dreamer's long-horizon intent with online visual observations. Removing multi-action tokens further reduces SR to 55%, and the policy tends to produce jittery and unstable motions during fine-grained grasping. This suggests that explicit action-token decomposition is important for dexterous bimanual coordination. In contrast, removing the fast–slow asynchronous mechanism does not improve task success, but increases inference latency from 0.325s to 0.45s. This shows that the asynchronous dual-system design mainly improves execution efficiency while preserving manipulation accuracy.

Bridge Denoising Steps. We further analyze the influence of bridge denoising steps. When only 5 denoising steps are used, the generated intent prior remains relatively coarse, leading to a clear drop in success rate. Increasing the number of steps to 15 slightly improves the success rate, but also introduces substantially higher inference latency. Therefore, 10 denoising steps provide a more practical performance–efficiency trade-off for real-time control.

Slow-Fast Denoising Ratio. Finally, we evaluate the computation allocation between the Dreamer and the Refiner. The 3:7 slow–fast configuration achieves the best overall result, suggesting that LiMA requires sufficient computation for the Dreamer to generate informative long-term intent, while allocating more capacity to the Refiner for high-frequency and fine-grained motion correction.

Overall, these ablation results demonstrate that LiMA's robust dexterous manipulation performance depends not only on the dual-system architecture itself, but also on an appropriate computation allocation between long-term imagination and real-time control.

4.4 Generalization

Beyond standard task evaluation, we further assess LiMA's out-of-distribution (OOD) generalization on the Cook Rice task, which requires coordinated bimanual execution across multiple task stages. We introduce four common visual perturbations: unseen backgrounds, novel object instances, unseen illumination conditions, and cluttered scenes. These settings evaluate different aspects of visual robustness. For each setting, we conduct 20 real-world trials, and Gr00T N1.6 and Cosmos-Policy are evaluated under the same protocol. As shown in Figure 5, LiMA consistently achieves higher success rates across all OOD settings. Its performance under unseen backgrounds and illumination conditions indicates that the policy does not merely memorize environment-specific textures or appearance statistics. LiMA also remains robust in cluttered scenes, suggesting that the model can preserve task-relevant intent despite additional visual distractors. Most notably, LiMA maintains a 70% success rate when manipulating novel object instances, demonstrating that it learns transferable interaction patterns rather than relying exclusively on object-specific visual cues observed during training. We attribute this robustness to the complementary roles of the two systems. The Future Dreamer captures long-horizon task structure and global object relationships, providing a relatively stable intent prior under visual changes. Meanwhile, the Motion Refiner continuously integrates the latest observations to correct local execution and accommodate deviations between imagined futures and the current physical state. This separation between task-level intent generation and observation-conditioned motion refinement reduces sensitivity to individual sources of visual variation.

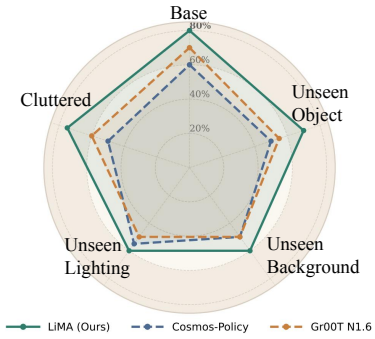


Figure 5: **Generalizations.**

5 Limitations

While LiMA excels in diverse dexterous tasks, its performance degrades under severe visual occlusion and low visual variance, where vision-based planning is inherently constrained. The current reliance on camera inputs limits the Dreamer's ability to maintain accurate trajectories when key features are obscured. We believe integrating tactile feedback is a crucial next step; incorporating haptic sensing would compensate for visual uncertainty and enable the Refiner to manage contact-rich interactions more robustly when visual information is compromised or temporarily unreliable. In addition, the reported inference latency is measured using a general implementation without deployment-specific acceleration, such as operator fusion, model compilation, quantization, or multi-GPU parallelism. Although this setting enables a consistent architecture-level comparison across different methods, it may not reflect the maximum achievable efficiency of LiMA. Future work will investigate fused computational kernels, reduced-precision inference, parallel execution of the Dreamer and Refiner, and hardware-aware multi-GPU deployment to further reduce latency and improve real-time performance.

6 Conclusion

In this paper, we presented **LiMA**, an asynchronous diffusion-based framework for bridging long-term imagination and high-frequency reactive execution. LiMA coordinates the Future Dreamer and the Motion Refiner through a dual-system architecture, enabling long-horizon foresight while preserving real-time responsiveness. A Latent Schrödinger Bridge Coupling mechanism further aligns strategic intent with clean execution latents. Extensive real-world experiments show that LiMA improves complex long-horizon manipulation while maintaining robustness and efficient closed-loop control.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62476011), the Beijing Natural Science Foundation (L252060). We are grateful to Xiansheng Chen for his support with the hardware setup, and to Yuqi Cao, Liangwei Hu, and Liwen Tong for their valuable insights and in-depth discussions on the mathematical derivations.

References

[1] Y. Fu, Q. Feng, N. Chen, Z. Zhou, M. Liu, M. Wu, T. Chen, S. Rong, J. Liu, H. Dong, et al. Cordvip: Correspondence-based visuomotor policy for dexterous manipulation in real-world. *arXiv preprint arXiv:2502.08449*, 2025.

[2] J. He, D. Li, X. Yu, Z. Qi, W. Zhang, J. Chen, Z. Zhang, Z. Zhang, L. Yi, and H. Wang. Dexvlg: Dexterous vision-language-grasp model at scale. *arXiv preprint arXiv:2507.02747*, 2025.

[3] S. Bai, W. Song, J. Chen, Y. Ji, Z. Zhong, J. Yang, H. Zhao, W. Zhou, W. Zhao, Z. Li, et al. Towards a unified understanding of robot manipulation: A comprehensive survey. *arXiv preprint arXiv:2510.10903*, 2025.

[4] R. Punamiya, S. Kareer, Z. Liu, J. Citron, R.-Z. Qiu, X. Cai, A. Gavryushin, J. Chen, D. Liconti, L. Y. Zhu, et al. Egoverse: An egocentric human dataset for robot learning from around the world. *arXiv preprint arXiv:2604.07607*, 2026.

[5] D. Chen, T. Kasarla, Y. Bang, M. Shukor, W. Chung, J. Yu, A. Bolourchi, T. Moutakanni, and P. Fung. Action100m: A large-scale video action dataset. *arXiv preprint arXiv:2601.10592*, 2026.

[6] R. Yang, Q. Yu, Y. Wu, R. Yan, B. Li, A.-C. Cheng, X. Zou, Y. Fang, X. Cheng, R.-Z. Qiu, et al. Egovla: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025.

[7] H. Luo, Y. Wang, W. Zhang, S. Zheng, Z. Xi, C. Xu, H. Xu, H. Yuan, C. Zhang, Y. Wang, et al. Being-h0. 5: Scaling human-centric robot learning for cross-embodiment generalization. *arXiv preprint arXiv:2601.12993*, 2026.

[8] R. Hoque, P. Huang, D. J. Yoon, M. Sivapurapu, and J. Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025.

[9] Y. Fu, N. Chen, J. Zhao, S. Shan, G. Yao, P. Wang, Z. Wang, and S. Zhang. Metis: Multi-source egocentric training for integrated dexterous vision-language-action model. *arXiv preprint arXiv:2511.17366*, 2025.

[10] X. Cai, R.-Z. Qiu, G. Chen, L. Wei, I. Liu, T. Huang, X. Cheng, and X. Wang. In-n-on: Scaling egocentric manipulation with in-the-wild and on-task data. *arXiv preprint arXiv:2511.15704*, 2025.

[11] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[12] 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.

[13] X. Chi, P. Jia, C.-K. Fan, X. Ju, W. Mi, Z. Qin, K. Zhang, W. Tian, K. Ge, H. Li, et al. Wow: Towards a world omniscient world model through embodied interaction. *arXiv preprint arXiv:2509.22642*, 2025.

[14] R. G. Goswami, A. Bar, D. Fan, T.-Y. Yang, G. Zhou, P. Krishnamurthy, M. Rabbat, F. Khorrami, and Y. LeCun. World models for learning dexterous hand-object interactions from human videos, 2026. URL https://arxiv.org/abs/2512.13644.

[15] Y. Jia, J. Liu, S. Liu, R. Zhou, W. Yu, Y. Yan, X. Chi, Y. Guo, B. Shi, and S. Zhang. Video2act: A dual-system video diffusion policy with robotic spatio-motional modeling, 2025. URL https://arxiv.org/abs/2512.03044.

[16] Y. Hu, Y. Guo, P. Wang, X. Chen, Y.-J. Wang, J. Zhang, K. Sreenath, C. Lu, and J. Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

[17] M. J. Kim, Y. Gao, T.-Y. Lin, Y.-C. Lin, Y. Ge, G. Lam, P. Liang, S. Song, M.-Y. Liu, C. Finn, and J. Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

[18] L. Li, Q. Zhang, Y. Luo, S. Yang, R. Wang, F. Han, M. Yu, Z. Gao, N. Xue, X. Zhu, Y. Shen, and Y. Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

[19] S. Ye, Y. Ge, K. Zheng, S. Gao, S. Yu, G. Kurian, S. Indupuru, Y. L. Tan, C. Zhu, J. Xiang, A. Malik, K. Lee, W. Liang, N. Ranawaka, J. Gu, Y. Xu, G. Wang, F. Hu, A. Narayan, J. Bjorck, J. Wang, G. Kim, D. Niu, R. Zheng, Y. Xie, J. Wu, Q. Wang, R. Julian, D. Xu, Y. Du, Y. Chebotar, S. Reed, J. Kautz, Y. Zhu, L. J. Fan, and J. Jang. World action models are zero-shot policies, 2026. URL https://arxiv.org/abs/2602.15922.

[20] J. Cen, S. Huang, Y. Yuan, K. Li, H. Yuan, C. Yu, Y. Jiang, J. Guo, X. Li, H. Luo, F. Wang, F. Wang, and D. Zhao. Rynnvla-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025.

[21] NVIDIA, J. Bjorck, N. C. Fernando Castañeda, X. Da, R. Ding, L. J. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, J. Jang, Z. Jiang, J. Kautz, K. Kundalia, L. Lao, Z. Li, Z. Lin, K. Lin, G. Liu, E. Llontop, L. Magne, A. Mandlekar, A. Narayan, S. Nasiriany, S. Reed, Y. L. Tan, G. Wang, Z. Wang, J. Wang, Q. Wang, J. Xiang, Y. Xie, Y. Xu, Z. Xu, S. Ye, Z. Yu, A. Zhang, H. Zhang, Y. Zhao, R. Zheng, and Y. Zhu. GR00T N1: An open foundation model for generalist humanoid robots. In *ArXiv Preprint*, March 2025.

[22] J. Cai, Z. Cai, J. Cao, Y. Chen, Z. He, L. Jiang, H. Li, H. Li, Y. Li, Y. Liu, et al. Internvla-a1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026.

[23] Z. Zhang, W. Wang, H. Sun, Q. Ding, X. Chu, G. Fang, and K. Au. Fingervip: Learning real-world dexterous manipulation with fingertip visual perception. *arXiv preprint arXiv:2604.21331*, 2026.

[24] X. Li, Y. Sun, L. Zhang, B. Huang, Y. Peng, Y. Meng, H. Jiang, S. Xie, G. Yao, A. Knoll, et al. Deco: Decoupled multimodal diffusion transformer for bimanual dexterous manipulation with a plugin tactile adapter. *arXiv preprint arXiv:2602.05513*, 2026.

[25] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning*, pages 2549–2564. PMLR, 2023.

[26] S. Funabashi, T. Isobe, F. Hongyi, A. Hiramoto, A. Schmitz, S. Sugano, and T. Ogata. Multi-fingered in-hand manipulation with various object properties using graph convolutional networks and distributed tactile sensors. *IEEE Robotics and Automation Letters*, 7(2):2102–2109, 2022.

[27] Y. Ye, Y. Fu, Y. Lv, B. Hou, J. Cen, L. Kong, D. Zheng, T. Chen, J. Liu, Z. Cao, Y. Lou, W. Chow, X. Sun, Y. Wang, K. Ge, X. Chi, X. Zhang, Z. Pang, Y. Zhong, S. Han, Z. Lu, W. Yuan, Q. Chen, M. Y. Wang, Y. Mu, Z. Liu, J. Yang, P. Luo, and S. Zhang. Data Pyramid for Embodied Manipulation, 2026. URL https://arxiv.org/abs/2607.24744.

[28] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

[29] Y. Zhong, Q. Jiang, J. Yu, and Y. Ma. Dexgrasp anything: Towards universal robotic dexterous grasping with physics awareness. *arXiv preprint arXiv:2503.08257*, 2025.

[30] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

[31] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[32] 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.

[33] F. Lin, R. Nai, Y. Hu, J. You, J. Zhao, and Y. Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning, 2025. URL https://arxiv.org/abs/2505.11917.

[34] Y. Ye, J. Ma, J. Cen, and Z. Lu. Token expand-merge: Training-free token compression for vision-language-action models. *arXiv preprint arXiv:2512.09927*, 2025.

[35] J. Pai, L. Achenbach, V. Montesinos, B. Forrai, O. Mees, and E. Nava. mimic-video: Video-action models for generalizable robot control beyond vlas. *arXiv preprint 2512.15692*, 2025.

[36] H. Bi, H. Tan, S. Xie, Z. Wang, S. Huang, H. Liu, R. Zhao, Y. Feng, C. Xiang, Y. Rong, et al. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025.

[37] R. Li, H. Zhang, J. Jin, Q. Zeng, Z. Zhuang, Y. Tang, S. Lyu, and D. Wang. World-value-action model: Implicit planning for vision-language-action systems. *arXiv preprint arXiv:2604.14732*, 2026.

[38] T. Yuan, Z. Dong, Y. Liu, and H. Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

[39] Q. Feng, J. Yu, J. Liu, Y. Jia, Z. Wu, H. Chen, Z. Qian, S. Gu, P. Jia, S. Ma, and S. Zhang. Harmowam: Harmonizing generalizable and precise manipulation via adaptive world action models, 2026.

[40] Y. Lou, Y. Ye, Y. Fu, J. Cen, X. Chi, Y. Lyu, P. Jia, S. Han, Z. Lu, and S. Zhang. Dreamtac: A unified tactile world action model for contact-rich robot manipulation. *arXiv preprint arXiv:2606.08737*, 2026.

[41] H. Xue, J. Ren, W. Chen, G. Zhang, Y. Fang, G. Gu, H. Xu, and C. Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

[42] H. Xu, J. Ding, J. Xu, R. Wang, J. Chen, J. Mai, Y. Fu, B. Ghanem, F. Xu, and M. Elhoseiny. Diffusion-based imaginative coordination for bimanual manipulation, 2025. URL https://arxiv.org/abs/2507.11296.

[43] A. Authors. Hume: Introducing system-2 thinking in visual-language-action model. *arXiv preprint arXiv:2505.21432*, 2025.

[44] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn, A. Handa, M.-Y. Liu, D. Xiang, G. Wetzstein, and T.-Y. Lin. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[45] Q. Bu, Y. Yang, J. Cai, S. Gao, G. Ren, M. Yao, P. Luo, and H. Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

[46] W. Song, J. Chen, W. Li, X. He, H. Zhao, P. D. S. Su, F. Tang, X. Cheng, D. Wang, Z. Ge, et al. Rationalvla: A rational vision-language-action model with dual system. *arXiv preprint arXiv:2506.10826*, 2025.

[47] J. Wen, Y. Zhu, Z. Tang, J. Li, Y. Peng, C. Shen, and F. Feng. Dexvla: Vision-language model with plug-in diffusion expert for visuomotor policy learning. *arXiv preprint arXiv:2502.05855*, 2025.

[48] Z. Liu, J. Liu, H. Chen, J. Yu, Z. Guo, C. Hou, C. Gu, X. Mi, R. Zhang, K. Wu, et al. Last _{0}: Latent spatio-temporal chain-of-thought for robotic vision-language-action model. *arXiv preprint arXiv:2601.05248*, 2026.

[49] H. Chen, J. Liu, C. Gu, Z. Liu, R. Zhang, X. Li, X. He, Y. Guo, C.-W. Fu, S. Zhang, and P.-A. Heng. Fast-in-slow: A dual-system foundation model unifying fast manipulation within slow reasoning, 2025. URL https://arxiv.org/abs/2506.01953.

[50] G. Team. Galaxea g0: Open-world dataset and dual-system vla model. *arXiv preprint arXiv:2509.00576v1*, 2025.

[51] J. Zhang, Y. Guo, X. Chen, Y.-J. Wang, Y. Hu, C. Shi, and J. Chen. Hirt: Enhancing robotic control with hierarchical robot transformers. *arXiv preprint arXiv:2410.05273*, 2024.

[52] G.-H. Liu, A. Vahdat, D.-A. Huang, E. A. Theodorou, W. Nie, and A. Anandkumar. I^2sb: Image-to-image schrödinger bridge. *arXiv preprint arXiv:2302.05872*, 2023.

[53] W. Peebles and S. Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

[54] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, J. Wang, J. Zhang, J. Zhou, J. Wang, J. Chen, K. Zhu, K. Zhao, K. Yan, L. Huang, M. Feng, N. Zhang, P. Li, P. Wu, R. Chu, R. Feng, S. Zhang, S. Sun, T. Fang, T. Wang, T. Gui, T. Weng, T. Shen, W. Lin, W. Wang, W. Wang, W. Zhou, W. Wang, W. Shen, W. Yu, X. Shi, X. Huang, X. Xu, Y. Kou, Y. Lv, Y. Li, Y. Liu, Y. Wang, Y. Zhang, Y. Huang, Y. Li, Y. Wu, Y. Liu, Y. Pan, Y. Zheng, Y. Hong, Y. Shi, Y. Feng, Z. Jiang, Z. Han, Z.-F. Wu, and Z. Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[55] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[56] NVIDIA, N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, T. Cai, P. Chattopadhyay, Y. Chen, Y. Cui, Y. Ding, D. Dworakowski, J. Fan, M. Fenzi, F. Ferroni, S. Fidler, D. Fox, S. Ge, Y. Ge, J. Gu, S. Gururani, E. He, J. Huang, J. Huffman, P. Jannaty, J. Jin, S. W. Kim, G. Klár, G. Lam, S. Lan, L. Leal-Taixe, A. Li, Z. Li, C.-H. Lin, T.-Y. Lin, H. Ling, M.-Y. Liu, X. Liu, A. Luo, Q. Ma, H. Mao, K. Mo, A. Mousavian, S. Nah, S. Niverty, D. Page, D. Paschalidou, Z. Patel, L. Pavao, M. Ramezanali, F. Reda, X. Ren, V. R. N. Sabavat, E. Schmerling, S. Shi, B. Stefaniak, S. Tang, L. Tchapmi, P. Tredak, W.-C. Tseng, J. Varghese, H. Wang, H. Wang, H. Wang, T.-C. Wang, F. Wei, X. Wei, J. Z. Wu, J. Xu, W. Yang, L. Yen-Chen, X. Zeng, Y. Zeng, J. Zhang, Q. Zhang, Y. Zhang, Q. Zhao, and A. Zolkowski. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[57] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

Appendix

A Additional Method Details

A.1 Additional Preliminaries

Latent Schrödinger Bridge Coupling. Schrödinger Bridge formulates a stochastic transport problem between two prescribed boundary distributions. Unlike standard diffusion generation, which typically starts from an unstructured Gaussian prior during sampling, Schrödinger Bridge constructs a bridge process whose endpoints are constrained by two informative boundary distributions. I^2SB [52] provides a practical instantiation of this idea by defining an analytic posterior over intermediate bridge states between a structured source endpoint and a clean target endpoint.

Given a paired boundary sample (X_0, X_1), the intermediate state X_t is sampled from:

$$q(X_t \mid X_0, X_1) = \mathcal{N}\left(X_t; \mu_t(X_0, X_1), \Sigma_t \mathbf{I}\right), \tag{8}$$

where

$$\mu_t(X_0, X_1) = \frac{\bar{\sigma}_t^2}{\sigma_t^2 + \bar{\sigma}_t^2} X_0 + \frac{\sigma_t^2}{\sigma_t^2 + \bar{\sigma}_t^2} X_1, \qquad \Sigma_t = \frac{\sigma_t^2 \bar{\sigma}_t^2}{\sigma_t^2 + \bar{\sigma}_t^2}. \tag{9}$$

Here, $\sigma_t^2 = \int_0^t \beta_\tau d\tau$ and $\bar{\sigma}_t^2 = \int_t^1 \beta_\tau d\tau$ denote the accumulated variances along the bridge. The mean path $\mu_t(X_0, X_1)$ defines the deterministic interpolation between the two endpoints, while $\Sigma_t \mathbf{I}$ introduces controlled stochasticity around this path. This gives a boundary-conditioned posterior rather than an unconditional Gaussian prior.

The endpoint behavior of this posterior clarifies the bridge direction. At $t = 0$, we have $\sigma_t^2 = 0$, which makes $\mu_t(X_0, X_1) = X_0$ and $\Sigma_t = 0$. At $t = 1$, we have $\bar{\sigma}_t^2 = 0$, which makes $\mu_t(X_0, X_1) = X_1$ and $\Sigma_t = 0$. Equivalently, a sampled intermediate state can be written as $\mu_t(X_0, X_1) + \sqrt{\Sigma_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. Thus, the stochastic term vanishes at both endpoints, while intermediate timesteps retain controlled stochasticity between the two meaningful latent endpoints. This differs from standard diffusion generation, where the sampling trajectory is typically anchored by an uninformative Gaussian prior.

In LiMA, we use this boundary-conditioned posterior as the mathematical interface between the Future Dreamer and the Motion Refiner. The key adaptation lies in redefining the two bridge boundaries for asynchronous robotic control: the structured source boundary X_1 is instantiated as the Dreamer's strategic intent prior,

$$X_1 \equiv \mathcal{Z}_{\text{int}}, \tag{10}$$

while the clean target boundary X_0 is instantiated as the Refiner's clean joint execution latent,

$$X_0 \equiv \mathcal{Z}_{\text{out}}^{(0)}. \tag{11}$$

Substituting these two boundaries into the I^2SB posterior yields the latent bridge used in LiMA:

$$q\left(\mathcal{Z}_{\text{out}}^{(t)} \mid \mathcal{Z}_{\text{out}}^{(0)}, \mathcal{Z}_{\text{int}}\right) = \mathcal{N}\left(\mathcal{Z}_{\text{out}}^{(t)}; \mu_t\left(\mathcal{Z}_{\text{out}}^{(0)}, \mathcal{Z}_{\text{int}}\right), \Sigma_t \mathbf{I}\right). \tag{12}$$

This turns the original boundary-to-boundary bridge into an intent-to-execution latent coupling mechanism. As a result, LiMA initializes refinement from a structured long-horizon intent prior rather than an unconditional Gaussian latent, while retaining stochastic refinement capacity through the bridge posterior.

A.2 Additional Implementation Details

Latent token layout. Following the latent tensor convention of Cosmos-Policy [17], LiMA represents multimodal information in a five-dimensional latent tensor. Each latent tensor is organized as

$$\mathcal{Z} \in \mathbb{R}^{B \times C \times N_{\text{slot}} \times H \times W}, \tag{13}$$

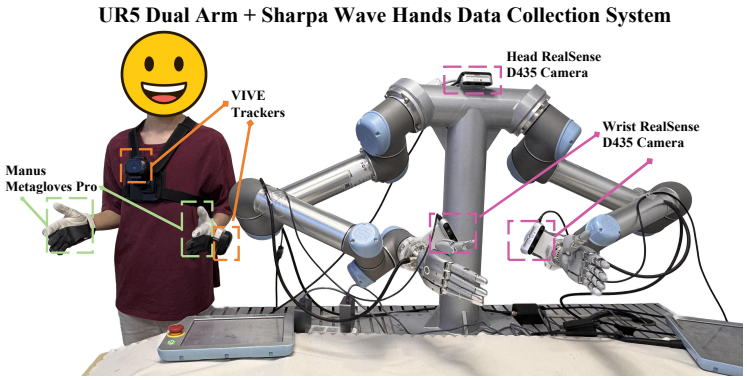


Figure 6: Overview of the real-world bimanual manipulation platform. The system consists of two UR5 robotic arms equipped with SharpaWave dexterous hands, a three-view RealSense D435 camera setup with one head-mounted camera and two wrist-mounted cameras, and a wearable teleoperation interface based on VIVE Trackers and MetaGlove Pro gloves for expert demonstration collection.

where B is the batch size, $C = 16$ is the latent channel dimension, $N_{\text{slot}} = 8$ denotes the number of organized latent slots, and $H = W = 28$ are the spatial latent resolutions. This slot-based organization provides an explicit interface between visual observations, action priors, and future imagination tokens.

The Future Dreamer and Motion Refiner use compatible slot layouts, each with $N_{\text{slot}} = 8$. Specifically, the Future Dreamer uses the slot set $\mathcal{S}_{\text{dream}}$ = [`blank`, `left_wrist`, `right_wrist`, `primary`, `action`, `left_future`, `right_future`, `primary_future`]. The Motion Refiner uses the slot set $\mathcal{S}_{\text{refine}}$ = [`blank`, `left_wrist`, `right_wrist`, `primary`, `arm_action`, `left_hand_action`, `right_hand_action`, `primary_future`]. This compatible slot organization allows the Dreamer's intent latent $\mathcal{Z}_{\text{int}}$ to be directly consumed by the Motion Refiner as a structured prior boundary. Therefore, the latent Schrödinger bridge can be constructed in the shared latent space without introducing an additional projection or alignment network.

Training hyperparameters. For training LiMA, the Future Dreamer is initialized from the pretrained Cosmos-Predict2-2B checkpoint, while the Motion Refiner is randomly initialized and trained from scratch. We use a batch size of 14 and optimize the model with a peak learning rate of 1.0×10^{-4}. The learning rate is warmed up for 2.5K steps, then decayed to 0.3 of the peak value within the first 40K steps. After that, the learning rate is kept constant at 0.06 of the peak learning rate.

B Additional Experimental Setup

B.1 Robot System Setup

We further provide implementation details of the real-world robot system, including the system operation pipeline, demonstration collection, and policy execution protocol. The experimental platform follows the bimanual dexterous manipulation setup described in the main paper, as illustrated in Figure 6. It consists of two UR5 robotic arms, two SharpaWave five-fingered dexterous hands, three Intel RealSense D435 cameras, VIVE Trackers, and MetaGlove Pro gloves.

Perception System. The three RealSense D435 cameras provide one global view and two local wrist views. The head-mounted camera captures the overall task scene, including target objects, the manipulation workspace, and task progress, while the two wrist-mounted cameras are installed near the end-effectors of the left and right robotic arms to provide local visual observations around the contact regions. All cameras are fixed before the experiments to ensure consistent multi-view observations within the same robot workspace.

Demonstration Collection. During human demonstration collection, the operator uses VIVE Trackers to record the relative spatial trajectories of both wrists, which are transformed and mapped to end-effector reference trajectories for the two UR5 arms. Meanwhile, MetaGlove Pro gloves record the finger joint configurations of both hands, which are mapped to the 22-DoF SharpaWave dexterous hands through a joint-space kinematic retargeting pipeline. This process produces synchronized dual-arm trajectories, dual-hand joint trajectories, multi-view visual observations, and language task instructions.

Policy Execution. During policy execution, each policy generates a short-horizon action chunk based on the current observations. The predicted action chunk contains motion commands for the two UR5 arms and joint commands for the left and right SharpaWave dexterous hands. Before each real-world trial, the experimental scene is reset to a predefined initial state, including the robot initial posture and the corresponding task instruction. All methods are evaluated under the same robot platform, camera configuration, task initialization protocol, and success criteria, ensuring that the comparison between LiMA and the baselines mainly reflects differences in policy design rather than hardware conditions or evaluation settings.

B.2 Baseline Settings

We compare LiMA with representative baselines covering integrated world-action modeling, vision-language-action modeling, and video-prediction-based policy learning. All baselines are evaluated under the same real-world robot platform, camera configuration, task initialization protocol, and success criteria as LiMA. For each baseline, we follow the training or fine-tuning protocol recommended by its original paper whenever applicable, and adapt its action outputs to the same bimanual dexterous robot execution interface.

Cosmos-Policy. Cosmos-Policy [17] is used as an integrated world-action modeling baseline. It maps robot actions into the visual latent space and jointly denoises action latents and image latents within a unified DiT-based denoising process. This makes Cosmos-Policy the closest integrated generative baseline to LiMA, as both methods model future visual information and robot actions within a shared generative framework.

GR00T N1.6. GR00T N1.6 [21] is used as a representative state-of-the-art vision-language-action baseline. It takes robot visual observations and language instructions as inputs to a vision-language model for semantic understanding, and conditions an action head with the robot state to generate continuous actions through a denoising process. In our setting, we found that the officially recommended 2,000 training steps did not fully expose its best performance on our bimanual dexterous manipulation tasks. Therefore, we report the results after training GR00T N1.6 for 50,000 steps on 1 NVIDIA H100 GPU.

VPP. VPP [16] is used as an inverse-dynamics-model baseline. It uses the future visual predictions produced by a video model as conditioning information and feeds them into an action DiT to generate robot actions. This baseline represents the paradigm of first predicting future visual states and then deriving actions from the predicted visual condition.

InternVLA-A1. InternVLA-A1 [22] adopts a Mixture-of-Transformers (MoT) architecture that unifies scene understanding, generative foresight, and action modeling within a single framework, enabling the model not only to acquire a rich representation of the environment but also to leverage predicted future visual observations to guide action generation. To accommodate the full joint configuration of our dual-arm, dual-hand system—comprising two 6-DoF robot arms and two 22-DoF dexterous hands (a total of 56-DoF)—we extend the action dimensionality accordingly and initialize the model from the pretrained InternVLA-A1-3B checkpoint.

To ensure a fair comparison, all baselines are trained or fine-tuned on the same real-world demonstration dataset whenever task-specific training is required. They use the same task instructions, multi-view robot observations, robot state inputs, and evaluation protocol as LiMA. Since different

Figure 7: **Visualization of task progress.**

baselines adopt different native action representations, we adapt their output interfaces to the same bimanual execution space, including arm motion commands and dexterous hand joint commands.

C Additional Evaluation Details

C.1 Task Details

We provide additional details on the real-world manipulation tasks used in our evaluation. We design six dexterous manipulation tasks with varying temporal horizons and interaction complexity, including Stack Cup, Roll T-shirt, Cook Rice, Make Sandwich, Make Coffee, and Assemble Package. These tasks cover both short-horizon single-arm manipulation and long-horizon bimanual coordination, requiring the robot to generate continuous actions from multi-view visual observations and language instructions. Visualizations of these real-world tasks are shown in Figure 7.

Stack Cup. Stack Cup is a short-horizon single-arm manipulation task. The robot is required to use its right hand to gently grasp a cup, align it above another cup, and place it down to complete the stacking process. This task mainly evaluates basic grasping, alignment, and precise placement ability.

Roll T-shirt. Roll T-shirt is a bimanual deformable-object manipulation task. The robot first uses both hands to grasp two corners of the T-shirt and folds it forward for several steps until the garment becomes easier to grasp with one hand. Finally, the robot uses its right hand to pick up the folded garment and place it into a basket. This task evaluates the model's ability to localize manipulation regions under diverse cloth deformations and to recognize different task stages during deformable-object manipulation.

Cook Rice. Cook Rice is a long-horizon multi-stage task. The robot first uses its left hand to gently grasp a spoon, scoop rice, and steadily transfer the rice into a rice cooker. After pouring the rice, the robot uses its right hand to close the rice cooker and then returns the spoon with its left hand. This task requires smooth motion over a longer horizon and consistent maintenance of task-level intent across multiple manipulation stages.

Make Sandwich. Make Sandwich is a long-horizon tool-use task. The robot first uses its right hand to pick up a pair of tongs, grasp a piece of lettuce, and place it onto a slice of bread. After accurately

Table 4: **Progress stage definitions for real-world tasks.** Each task is decomposed into several progress stages for computing Progress Success Rate (PSR).

Task	P1	P2	P3	P4	P5
Stack Cup	Grasp cup	Stack cup	–	–	–
Roll T-shirt	Grasp corners	Fold garment	Pick and place	–	–
Cook Rice	Grasp spoon	Scoop rice	Transfer rice	Close cooker	Return spoon
Make Sandwich	Pick up tongs	Grasp lettuce	Place lettuce and return tongs	Cover with bread	–
Make Coffee	Pick up coffee bag	Pick up kettle	Hand over cup	–	–
Assemble Package	Pick up black part	Pick up transparent part	Combine parts	–	–

returning the tongs, the robot uses its left hand to pick up another slice of bread and place it on top of the lettuce. This task mainly evaluates tool use, fine-grained object interaction, and precise multi-step placement.

Make Coffee. Make Coffee is a long-horizon multi-object interaction task. The robot first uses its left hand to pick up a coffee bag and pour coffee into a cup, then uses its right hand to pick up a kettle and pour water into the cup. Finally, the robot uses its left hand to grasp the cup and hand it to a person in front of the workspace. For safety and hardware protection, including water and dust resistance considerations, this task is performed with simulated pouring actions rather than real liquid transfer. The task mainly evaluates the model's ability to handle multi-object interactions over a long temporal horizon.

Assemble Package. Assemble Package is a long-horizon bimanual coordination task. The robot first uses its right hand to pick up a black package component and then uses its left hand to pick up a transparent package component. The two hands then cooperate to align and combine the two components into a complete package. This task evaluates bimanual coordination, spatial alignment, and execution stability in a structured assembly scenario.

C.2 Evaluation Metrics

We provide additional details on the evaluation metrics used in our real-world robot experiments. We mainly evaluate different methods using Success Rate (SR), Progress Success Rate (PSR), and inference time. SR measures whether a policy can complete the entire task. For each real-world trial,

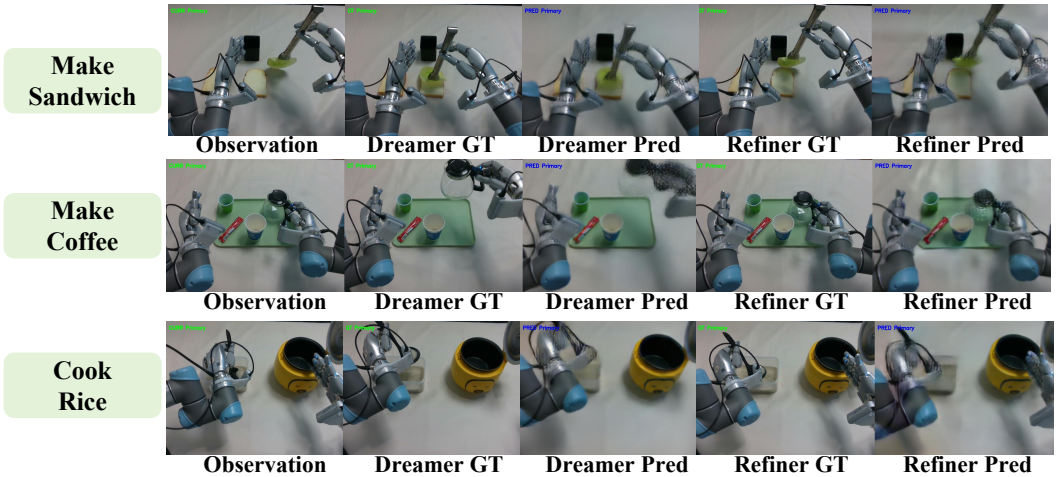


Figure 8: **Visualization of Dreamer & Refiner Generations.**

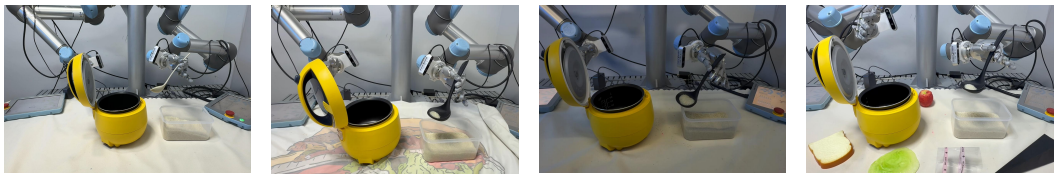
Table 5: **Per-stage success rates for progress evaluation.** We report the success rate of each progress stage to analyze where different methods fail during task execution.

Task	Method	P1(%)	P2(%)	P3(%)	P4(%)	P5(%)	PSR(%)
Stack Cup	LiMA	**100**	**90.0**	–	–	–	**95.0**
	Cosmos-Policy	85.0	80.0	–	–	–	82.5
	GR00T N1.6	90.0	85.0	–	–	–	87.5
	VPP	80.0	80.0	–	–	–	80.0
	InternVLA-A1	90.0	**90.0**	–	–	–	90.0
Roll T-shirt	LiMA	95.0	80.0	**75.0**	–	–	83.3
	Cosmos-Policy	80.0	70.0	60.0	–	–	70.0
	GR00T N1.6	75.0	70.0	70.0	–	–	71.7
	VPP	60.0	55.0	55.0	–	–	56.7
	InternVLA-A1	**100**	**90.0**	**75.0**	–	–	**88.3**
Cook Rice	LiMA	**85.0**	**85.0**	**80.0**	**80.0**	**80.0**	**82.0**
	Cosmos-Policy	75.0	70.0	70.0	60.0	60.0	67.0
	GR00T N1.6	80.0	70.0	70.0	70.0	70.0	72.0
	VPP	65.0	60.0	60.0	60.0	55.0	60.0
	InternVLA-A1	**85.0**	80.0	**80.0**	75.0	70.0	78.0
Make Sandwich	LiMA	90.0	**85.0**	**75.0**	**70.0**	–	**80.0**
	Cosmos-Policy	90.0	70.0	65.0	60.0	–	71.3
	GR00T N1.6	80.0	75.0	70.0	65.0	–	72.5
	VPP	75.0	60.0	60.0	50.0	–	61.3
	InternVLA-A1	**100**	75.0	70.0	65.0	–	77.5
Make Coffee	LiMA	**80.0**	70.0	**60.0**	–	–	70.0
	Cosmos-Policy	70.0	65.0	55.0	–	–	63.3
	GR00T N1.6	**80.0**	**80.0**	55.0	–	–	**71.7**
	VPP	60.0	60.0	50.0	–	–	56.7
	InternVLA-A1	**80.0**	70.0	50.0	–	–	66.7
Assemble Package	LiMA	**70.0**	**70.0**	**50.0**	–	–	**63.3**
	Cosmos-Policy	60.0	50.0	45.0	–	–	51.7
	GR00T N1.6	**70.0**	65.0	**50.0**	–	–	61.7
	VPP	60.0	45.0	40.0	–	–	48.3
	InternVLA-A1	65.0	60.0	**50.0**	–	–	58.3

the trial is considered successful only if the robot completes all predefined task goals in the correct order. Thus, SR reflects the final task-level success of each policy.

For long-horizon multi-stage tasks, binary SR alone cannot fully reflect partial task completion. Therefore, we also use the Progress Success Rate (PSR) to measure the progress of the task. The PSR is calculated on the basis of the proportion of progress stages successfully completed in each trial. The progress stages used for the PSR calculation are defined in Table 4, and the success rates per-stage are reported in Table 5. This enables a more fine-grained analysis of where different methods fail during task execution. In addition to these quantitative progress metrics, we visualize LiMA's predicted ego-centric future observations and the corresponding ground-truth observations for three long-horizon tasks in Figure 8.

We also report inference time to evaluate real-time deployability in closed-loop robot control. Inference time denotes the average time required for a policy to generate one executable action chunk from the current observation. For synchronous inference models, this time is measured by the forward-pass time of each complete policy inference. For LiMA, since the Future Dreamer is updated at a lower frequency while the Motion Refiner runs at the high-frequency control loop, we report the effective average inference time under asynchronous execution. Specifically, it is computed as the sum of one Future Dreamer inference time and all Motion Refiner inference times within one asynchronous update cycle, divided by the number of action chunks generated in that cycle. All inference-time measurements are collected under the same robot platform and execution protocol to ensure a fair comparison.



Unseen Object **Unseen Background** **Unseen Lighting** **Cluttered**

Figure 9: **Visualization of Generalizations.**

C.3 Generalization Evaluation

We provide additional qualitative analysis for the generalization evaluation, with task visualizations shown in Figure 9. The quantitative success rates are reported in the main paper. In the unseen-object setting, LiMA still maintains a success rate of 70%, indicating that the policy can adapt to changes in task-relevant object appearance while preserving the overall manipulation intent. In the cluttered-scene setting, LiMA achieves a success rate of 75%, suggesting that the proposed architecture can effectively focus on task-relevant regions and remain robust to irrelevant visual distractions. Compared with LiMA, Cosmos-Policy shows more frequent grasping failures under these perturbed scenes. This suggests that a unified DiT-based world-action model may be more sensitive to visual clutter and local object appearance shifts during contact-rich execution. GR00T N1.6 also exhibits failure modes that are not observed under the base setting, such as skipping the rice-scooping stage in the Cook Rice task.

For unseen-background and unseen-lighting settings, LiMA shows larger success-rate fluctuations than in the base setting, as these perturbations directly affect the visual appearance of the task region. Nevertheless, LiMA still outperforms the baselines under these conditions. These results suggest that the asynchronous Dreamer–Refiner design and latent intent-to-execution coupling help LiMA maintain stronger robustness when both task-relevant and task-irrelevant visual factors are perturbed.

D Failure Cases

We further analyze the typical failure cases of LiMA in real-world deployment. The failures mainly come from two aspects. First, LiMA is sensitive to the field of view of the installed cameras. The policy relies on multi-view visual observations to maintain long-horizon task intent and local motion correction, so the task scene needs to be sufficiently covered by the cameras. For example, in the Assemble Package task, some manipulation steps may partially move outside the camera view due to the limited visual coverage. In addition, transparent objects are harder to perceive reliably from RGB observations, which further limits LiMA's long-horizon planning and execution stability. This partially explains why the success rate of Assemble Package is lower than that of other tasks.

Second, LiMA is still limited in perceiving fine-grained contact information from vision alone. For example, in the Roll T-shirt task, when the manipulated garment has a color similar to the surrounding environment, the model may be less sensitive to the exact contact region. In some trials, the robot occasionally pinches empty space instead of the garment, leading to failed folding or grasping. These results suggest that although LiMA can maintain stable task-level intent in most cases, its performance is still constrained by visual observability and contact-state estimation. In future work, these limitations can be mitigated by using cameras with a wider field of view and incorporating tactile sensing to provide more reliable contact feedback.