ElasticWAM: Adaptive Spatiotemporal Imagination for Efficient Robot Control

Anonymous authors
Paper under double-blind review

Abstract

World Action Models (WAMs) benefit from imagining the future, but must they refine every region and complete future-video denoising to generate effective actions? We argue that future refinement should be guided by its contribution to action generation rather than visual completeness. We introduce ElasticWAM, an action-guided framework for adaptive spatiotemporal imagination. Built on a frozen pretrained WAM, ElasticWAM learns a unified refinement-demand field from action-prediction supervision to determine which future-video tokens to refresh, how far to advance denoising, and when to stop world refinement. It selectively updates future-video representations while retaining cached context from unrefreshed regions through the pretrained action-conditioning interface. Experiments across four manipulation benchmarks demonstrate the effectiveness of this approach. On RoboCasa-GR1, ElasticWAM improves success by 2.4 percentage points while reducing inference latency by 32.2% relative to Joint-WAM. Real-world dual-arm experiments further demonstrate long-horizon task execution under nominal conditions, lighting perturbations, and object replacement. These results support maintaining useful future context for control through selective refinement and cache reuse, without refreshing every region at every denoising step.

1 Introduction

World Action Models (WAMs) use imagined visual futures to support robot control (Ye et al., 2026; Yuan et al., 2026). In dense joint refinement, future-video and action representations are repeatedly updated together, introducing substantial inference latency. This process refines every future-video region throughout denoising, raising a question: **Does effective control require refreshing every region and denoising the imagined future to completion?**

Existing WAMs use different amounts of future refinement at inference time. Fast-WAM skips iterative future prediction, whereas Joint-WAM repeatedly updates future-video and action representations together (Yuan et al., 2026). Successful control with Fast-WAM shows that extensive future refinement is not always necessary. At the same time, additional denoising does not by itself guarantee better actions. These observations motivate a policy that learns whether to continue refining the future from the expected benefit to action generation.

Our central insight is that **future refinement should be guided by its expected contribution to action generation**. Existing future representations may already provide useful context for control, and leaving them unrefreshed does not make that information unavailable. This suggests preserving useful context while directing further refinement to regions where additional updates are expected to help. The resulting future representation need not be uniformly refined to support effective actions.

We introduce **ElasticWAM**, an action-guided framework for adaptive spatiotemporal imagination. Built on a frozen pretrained WAM, it learns a refinement-demand field from downstream action-prediction supervision to determine which future-video tokens to refresh, how far to advance denoising, and when to stop. After an initial dense pass, ElasticWAM adaptively decides whether to continue world refinement and how far to advance denoising. Each continued refinement step selectively refreshes tokens under a fixed per-step budget while reusing cached features elsewhere. Selected future-video tokens retain access to cached representations of unrefreshed regions, allowing selective updates to incorporate information from across the scene. The resulting representation supports the remaining action-denoising steps through the pretrained action-conditioning interface.

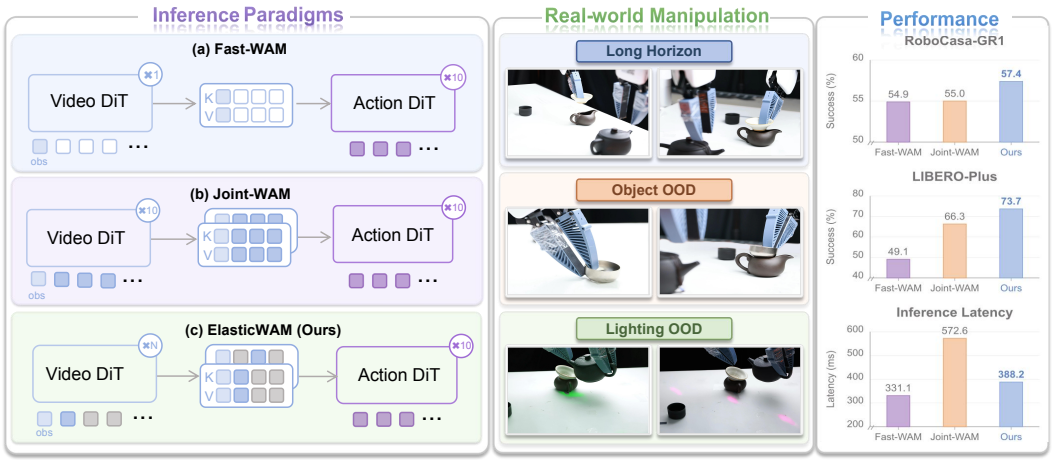


Figure 1: Overview of ElasticWAM. Left: Fast-WAM omits iterative future refinement, Joint-WAM densely refines future-video representations at every denoising step, and ElasticWAM adaptively determines when to stop, how far to advance denoising, and which regions to refresh while reusing cached context. Middle: Real-world long-horizon dual-arm tea preparation under nominal conditions, object replacement, and lighting perturbations. Right: Success rates on RoboCasa-GR1 and LIBERO-Plus, and inference latency on RoboCasa-GR1.

We evaluate ElasticWAM on LIBERO, LIBERO-Plus, LIBERO-Pro, and RoboCasa-GR1, together with a four-stage dual-arm tea-preparation task. On LIBERO-Plus, ElasticWAM achieves 73.7% episode-weighted overall success. On RoboCasa-GR1, it achieves 57.4% success while reducing inference latency by 32.2%. These results demonstrate that selective imagination can improve control performance while reducing the computational cost of dense joint refinement.

Our contributions are threefold:

- We identify and study whether selective spatiotemporal refinement can improve both action generation and inference efficiency in WAMs, examining how imagined futures can be refined to better serve downstream control.
- We propose ElasticWAM, an action-guided framework that unifies spatial token selection, denoising progression, and stopping through a learned refinement-demand field. Trained on a frozen pretrained WAM using action-prediction supervision, it selectively refreshes future representations while preserving access to cached context.
- We validate ElasticWAM on four manipulation benchmarks and a long-horizon real-world dual-arm tea-preparation task, demonstrating the effectiveness of selective future refinement for robot control. The results show that ElasticWAM can improve task success while reducing inference latency relative to Joint-WAM.

2 Related Work

2.1 Future-Aware Robot Policies and World-Action Models

Vision-Language-Action (VLA) models typically predict robot actions directly from visual observations and language instructions (Zitkovich et al., 2023; Kim et al., 2025b; Black et al., 2024). Recent work instead introduces explicit future reasoning into robot policies. CoT-VLA predicts future images as visual subgoals before action generation (Zhao et al., 2025), while WorldVLA jointly models future images and actions within a unified autoregressive framework (Cen et al., 2025). DreamZero further develops this direction with video-based World-Action Models (WAMs), using predicted future dynamics to improve physical generalization (Ye et al., 2026). Such future-aware policies make world evolution an explicit part of policy inference, but also introduce additional generative computation. Recent work has therefore explored more efficient forms of future conditioning, for example by sparsifying video–action interactions while retaining future representations (Zhao et al., 2026).

ElasticWAM is complementary to these approaches: rather than removing or restructuring future conditioning, we adaptively allocate computation *within* the recurrent world-refinement process.

2.2 Adaptive Denoising and Computation Allocation

Diffusion models incur substantial inference cost from iterative denoising. Classical acceleration methods reduce the number of sampling steps through non-Markovian sampling, efficient ODE solvers, or consistency distillation (Song et al., 2021; Lu et al., 2022; Luo et al., 2024), but typically apply a common inference schedule across inputs. More recent methods make this computation input dependent: AdaDiff learns sample-specific denoising step budgets (Zhang et al., 2025a), while AdaptiveDiffusion dynamically skips redundant noise-prediction computation based on the evolution of intermediate latents (Ye et al., 2024). For WAMs, SANTS directly learns a state-adaptive video denoising trajectory, jointly deciding when to stop world refinement and how far to progress along the noise axis using downstream action quality (Sun et al., 2026). These methods primarily adapt computation along the temporal denoising trajectory. ElasticWAM instead couples temporal stopping and continuous progression with state-dependent spatial refinement, representing all three decisions through a unified spatiotemporal refinement-demand field.

2.3 Sparse and Dynamic Computation for Visual Generation

A complementary line of work reduces the per-step cost of video diffusion by exploiting spatial redundancy. VSA learns hardware-efficient sparse attention by identifying critical token regions (Zhang et al., 2025b), while VORTA dynamically routes inputs among specialized sparse-attention patterns (Sun et al., 2025). Jenga further combines dynamic attention carving with progressive-resolution generation to exploit redundancy across both denoising stages and video tokens (Zhang et al., 2025c). These methods primarily allocate computation to preserve visual generation quality and typically modify attention connectivity or latent resolution. ElasticWAM instead sparsifies the *refresh* of future-video queries while preserving their full logical attention context. Moreover, its spatial allocation is learned from the effect of sparse world refinement on downstream action prediction, coordinating temporal and spatial computation within a shared refinement field.

3 Method

As shown in Fig. 2, our ElasticWAM adapts world refinement to the state-dependent temporal and spatial redundancy of world-action models, determining **WHEN** to stop refinement, **HOW FAR** to progress, and **WHERE** to allocate computation. We model these decisions through a unified spatiotemporal refinement field, in which geometry-aware local curves represent the residual refinement demand over space and denoising trajectory (Sec. 3.1). To obtain an executable refinement policy, we instantiate this field with a shared temporal basis and an absolute–relative amplitude decomposition, while enforcing an external exact-token resource budget (Sec. 3.2). We learn the resulting temporal, relative spatial, and absolute demand components progressively using same-state terminal counterfactuals, adjacent-state flow supervision, and whole-prefix supervision with terminal-counterfactual consolidation (Sec. 3.3). At inference, ElasticWAM starts from a dense joint update and recurrently performs cache-preserving sparse refinement until the accumulated demand triggers World Commit, after which the current world state is directly reused for Action Finish (Sec. 3.4).

3.1 Unified Spatiotemporal Refinement Field

We represent the remaining world refinement demand over both denoising progression and future video space. Given a recurrent state s, we partition the $N_F = 196$ future video tokens into $G = 64$ geometry-aware groups that preserve temporal and camera-view boundaries. Each group g contains n_g valid tokens, where $n_g \in \{1, 2, 4\}$ and $\sum_g n_g = N_F$. Each group has a local refinement curve:

$$q_g(z \mid s) = a_g(s)\ \mathrm{Beta}(z; m_g(s), c_g(s))\,, \qquad z \in [0, 1], \tag{1}$$

where z denotes the normalized progression coordinate over the admissible denoising interval. The amplitude a_g represents the residual refinement demand of group g, while m_g and c_g characterize where this demand is concentrated along the progression axis.

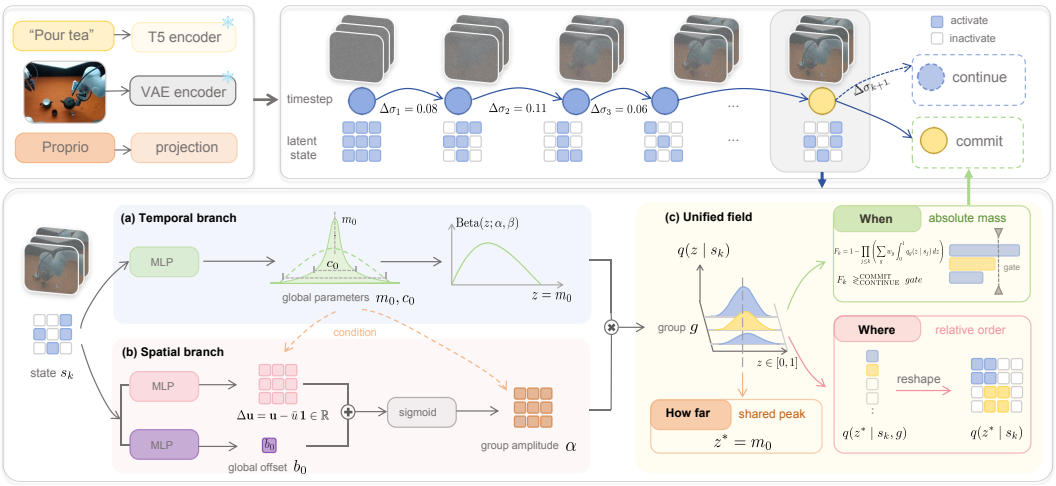


Figure 2: Adaptive refinement controller in ElasticWAM. Language, visual observations, and proprioception condition the world state. Temporal and spatial branches produce a refinement-demand field that determines when to commit, how far to advance, and which latent regions to update.

Since the Beta density is normalized, the amplitude directly corresponds to the mass of the local curve. We aggregate the local curves according to their actual token support:

$$\int_0^1 q_g(z \mid s)\, dz = a_g(s), \qquad Q(z \mid s) = \sum_{g=1}^{G} \frac{n_g}{N_F}\, q_g(z \mid s). \tag{2}$$

Thus, $Q(z \mid s)$ forms a unified spatiotemporal refinement field: its total mass describes how much useful world refinement remains, its distribution along z describes where further progression is preferred, and its local components indicate where refinement demand is spatially concentrated. This formulation provides a common representation for temporal stopping, denoising progression, and spatial allocation. We next introduce a conditional instantiation that factorizes the field according to the execution structure of the underlying WAM.

3.2 Conditional Instantiation for Adaptive Inference

The general field in Sec. 3.1 allows each group to have an independent temporal profile. Since all future video tokens share a single noise progression, ElasticWAM adopts a shared temporal basis:

$$q_g(z \mid s) = a_g(s) B_0(z \mid s), \qquad B_0(z \mid s) = \mathrm{Beta}(z; m_0(s), c_0(s))\,. \tag{3}$$

The global field therefore factorizes as

$$Q(z \mid s) = D(s) B_0(z \mid s), \qquad D(s) = \sum_{g=1}^{G} \frac{n_g}{N_F} a_g(s), \tag{4}$$

where B_0 captures the shared temporal progression and $D(s)$ measures the remaining refinement demand. To separate absolute demand from its spatial distribution, we area-center the relative logits:

$$\Delta u_g(s) = u_g^{\mathrm{rel}}(s) - \sum_{j=1}^{G} \frac{n_j}{N_F} u_j^{\mathrm{rel}}(s), \qquad \sum_{g=1}^{G} \frac{n_g}{N_F} \Delta u_g(s) = 0. \tag{5}$$

The local amplitudes are then defined by

$$a_g(s) = \mathrm{sigmoid}\left(b_0(s) + \mathrm{stopgrad}[\Delta u_g(s)]\right), \tag{6}$$

where Δu_g preserves spatial ordering and the common mode b_0 controls absolute demand.

WHEN. The continuation masses $D(s_k)$ determine the cumulative commit probability:

$$\begin{aligned} C_k &= D(s_k), \qquad \Delta H_k = -\log(\mathrm{clamp}(C_k, \varepsilon, 1)), \\ F_k &= 1 - \exp\left(-\sum_{j \le k} \Delta H_j\right) = 1 - \prod_{j \le k} C_j. \end{aligned} \tag{7}$$

ElasticWAM performs World Commit when $F_k \ge \eta$, with $\eta = 0.5$.

HOW FAR. Given continuation, the temporal distribution is the normalized refinement field:

$$\begin{aligned} p(z \mid s, \mathrm{CONTINUE}) = \frac{Q(z \mid s)}{D(s)} = B_0(z \mid s), \\ \Delta\sigma_k = \delta_{\min} + z_k(\delta_{\max} - \delta_{\min}), \qquad z_k = m_0(s_k), \end{aligned} \tag{8}$$

where $\delta_{\min} = 0.05$ and $\delta_{\max} = \min(0.20, \sigma_k - \sigma_{\mathrm{floor}})$.

WHERE. Because the temporal basis and common mode are shared across groups, they do not alter the spatial ordering. Under an external exact-token budget K_0, we select

$$M_k^* = \arg\max_{M \in \{0,1\}^G} \sum_{g=1}^{G} n_g M_g \Delta u_g(s_k), \qquad \text{s.t.} \quad \sum_{g=1}^{G} n_g M_g = K_0. \tag{9}$$

We use $K_0 = 96$ by default in ElasticWAM, treating it as a fixed per-update computation budget.

3.3 Supervised Policy Learning

We learn the three components of ElasticWAM progressively while keeping the pretrained WAM frozen. Each stage targets a distinct quantity: the global temporal basis, the relative spatial demand, and the absolute demand used for cumulative World Commit.

Global Temporal Basis. For each recurrent state s, we construct same-state counterfactuals by comparing immediate World Commit with multiple valid continuation distances. Each continuation performs one additional Joint update, followed by forced Commit and a complete Action Finish. We evaluate each candidate using the terminal action error

$$E(s, y) = 0.6E_{\mathrm{seq}}(s, y) + 0.4E_{\Delta}(s, y), \tag{10}$$

and convert the candidate errors into a soft target for the Beta progression. The stopping teacher compares immediate Commit with the expected error under the learned continuation policy:

$$\begin{aligned} E^{\pi}_{\mathrm{continue}}(s) &= \sum_{\delta} P_{\pi}(\delta \mid s) E_{\delta}(s), \\ A_{\mathrm{stop}}(s) &= E^{\pi}_{\mathrm{continue}}(s) - E_{\mathrm{commit}}(s), \qquad p^*_{\mathrm{stop}}(s) = \mathrm{sigmoid}\left(\frac{A_{\mathrm{stop}}(s)}{T(s)}\right). \end{aligned} \tag{11}$$

Relative Spatial Demand. We learn Δu_g with adjacent-state flow supervision. A dense previous state at σ_0 is used to obtain the recurrent context and the frozen temporal policy selects σ_1. At the same teacher-forced state x_{cur} at σ_1, we compare a no-gradient dense teacher with a differentiable sparse student. The hard exact-budget mask is optimized through a straight-through relaxation:

$$\begin{aligned} M^{\mathrm{ST}} &= M^{\mathrm{hard}} + M^{\mathrm{soft}} - \mathrm{stopgrad}(M^{\mathrm{soft}}), \\ \mathcal{L}_{\mathrm{where}} &= \mathcal{L}_{\text{action-GT}} + \mathcal{L}_{\text{action-Dense}} + 0.1\mathcal{L}_{\text{video-GT}} + 0.01\mathcal{L}_{\mathrm{budget}}. \end{aligned} \tag{12}$$

After training, the relative spatial ordering is frozen.

Absolute Demand. With Δu_g fixed, we transfer the temporal teacher into the curve amplitudes. The teacher hazard defines a continuation target,

$$C_k^* = \exp\left(-\Delta H_k^{\mathrm{teacher}}\right), \qquad \sum_{g=1}^{G} \frac{n_g}{N_F} \mathrm{sigmoid}\left(b_0^*(s_k) + \Delta u_g(s_k)\right) = C_k^*, \tag{13}$$

which uniquely determines the target common mode b_0^*. Since World Commit depends on accumulated demand, we preserve complete recurrent prefixes and optimize

$$\begin{aligned}\mathcal{L}_{\text{seq}} &= \mathcal{L}_{b_0} + \mathcal{L}_{\text{conditional}} + \mathcal{L}_{\text{cumulative}} + 0.5\mathcal{L}_{\text{commit-time}},\\ \mathcal{L}_{\text{final}} &= \mathcal{L}_{\text{seq}} + 0.1\mathcal{L}_{\text{terminal-CF}}.\end{aligned} \tag{14}$$

The terminal-counterfactual term provides a low-weight task-utility correction on fresh policy-visited states, while the temporal basis and relative spatial branch remain frozen.

3.4 Recurrent Sparse Inference

At each replanning cycle, ElasticWAM starts with a mandatory dense Joint update over all future video tokens, establishing a complete recurrent state and layer-wise cache. The refinement field is then evaluated to obtain the cumulative World-Commit signal, the continuous denoising progression, and the exact-K spatial allocation described in Sec. 3.2.

Recurrent Sparse Refinement. If refinement continues, all video and action latents are advanced to the next noise level, while the selected exact-K mask determines which future video queries are refreshed in the next Joint update. Observation and action queries remain dense, and active future video queries retain access to the full logical key–value context. Inactive future-video tokens preserve their own hidden states, keys, values, and last-refresh velocities, which are reused until they are selected again. This reduces future-query computation while preserving the recurrent world state.

World Commit and Action Finish. The process repeats until the cumulative demand reaches the commit threshold or a safety condition is triggered. ElasticWAM then commits the current world representation and directly reuses its latent state, full logical hidden state, and layer-wise cache for Action Finish. No additional video prefill is performed after World Commit.

4 Experiments

4.1 Experimental Setup

We evaluate ElasticWAM on LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2025), LIBERO-Pro (Zhou et al., 2025), and RoboCasa-GR1 (NVIDIA et al., 2025), covering standard manipulation and robustness to distribution shifts. We report task success rates and additionally measure inference latency on RoboCasa-GR1. The pretrained WAM remains frozen while the refinement controller is trained using action-derived supervision (Section 3.3). On LIBERO-Plus, we evaluate the LIBERO-trained policy and controller without additional adaptation.

We further evaluate ElasticWAM, Fast-WAM, and Joint-WAM on a real-world dual-arm tea-preparation task under nominal conditions, lighting OOD, and object OOD. The task comprises four sequential stages: (1) placing the white cup on the uncovered teapot; (2) pouring tea from the covered teapot into the white cup; (3) returning the white cup to its original position; and (4) pouring tea from the uncovered teapot into the black cup and returning the uncovered teapot to its original position. For lighting OOD, we introduce neon lighting to alter the scene illumination. For object OOD, the white cup is replaced throughout the manipulation sequence by a silver-colored cup with a slightly different shape. Figure 3 illustrates both perturbations. We collect 200 demonstration trajectories for this task. We report full-task success rate (SR) and progress score (PS), defined as the average fraction of the four stages completed. Each completed stage receives one point, and full-task success requires completing all four stages. Additional experimental details are provided in Appendix A.

4.2 Main Results

Tables 1–4 summarize the simulation results. On LIBERO, ElasticWAM achieves an average success rate of 98.8%, with performance close to Joint-WAM on Spatial, Object, and Goal and higher success on Long. On LIBERO-Plus, the LIBERO-trained policy and controller achieve 73.7% episode-weighted overall success without additional adaptation. Relative to the reported Joint-WAM results, improvements appear across all seven perturbation categories and are more pronounced under camera viewpoint changes, robot initialization changes, and sensor noise. ElasticWAM also improves over Joint-WAM across all four LIBERO-Pro categories, although Position and Task remain challenging.

RoboCasa-GR1 further reveals the performance–efficiency trade-off. ElasticWAM improves success by 2.4 percentage points while reducing inference latency by 32.2% relative to Joint-WAM. Compared with Fast-WAM, it achieves higher success but incurs additional latency. These results indicate that selective iterative refinement can improve control performance while avoiding the full computational cost of dense joint updates.

Table 1: LIBERO success rates (%).

Method	Spat.	Obj.	Goal	Long	Avg.
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	98.8	98.2	98.0	92.4	96.9
LingBot-VA (Li et al., 2026)	98.5	99.6	97.2	98.5	98.5
Motus (Bi et al., 2025)	96.8	99.8	96.6	97.6	97.7
Fast-WAM (Yuan et al., 2026)	98.2	100.0	97.0	95.2	97.6
Joint-WAM (Yuan et al., 2026)	99.6	99.4	98.2	96.8	98.5
ElasticWAM (Ours)	99.4	99.6	98.0	98.2	**98.8**

Table 2: LIBERO-Pro success rates (%).

Method	Obj.	Pos.	Sem.	Task	Avg.
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	96.0	20.8	95.8	0.8	53.4
OpenVLA (Kim et al., 2025b)	93.0	0.0	97.3	0.0	47.6
X-VLA (Zheng et al., 2025)	79.0	0.8	96.8	6.8	45.9
Fast-WAM (Yuan et al., 2026)	78.0	1.0	83.2	15.8	44.5
Joint-WAM (Yuan et al., 2026)	84.5	6.3	96.2	24.2	52.8
ElasticWAM (Ours)	85.6	8.4	97.7	24.7	**54.1**

Table 3: LIBERO-Plus success rates (%).

Method	Cam.	Rob.	Lang.	Light	Backg.	Noise	Layout	SR.
UniVLA (Bu et al., 2025)	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
OpenVLA-OFT (Kim et al., 2025a)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
π_0 (Black et al., 2024)	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
π_0-Fast (Pertsch et al., 2025)	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
WorldVLA (Cen et al., 2025)	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
Fast-WAM (Yuan et al., 2026)	18.8	45.7	70.1	83.2	45.7	29.8	62.7	49.1
Joint-WAM (Yuan et al., 2026)	37.5	64.5	93.0	95.0	55.9	47.3	79.6	66.3
LingBot-VA (Li et al., 2026)	40.9	83.0	86.4	82.3	53.1	64.4	76.2	69.5
ElasticWAM (Ours)	52.1	76.3	95.5	95.1	60.0	61.1	81.4	**73.7**

Table 4: RoboCasa-GR1.

Method	Latency (ms) ↓	Avg.
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	80.52	37.0
Fast-WAM (Yuan et al., 2026)	331.1	54.9
Joint-WAM (Yuan et al., 2026)	572.6	55.0
ElasticWAM (Ours)	388.2	**57.4**

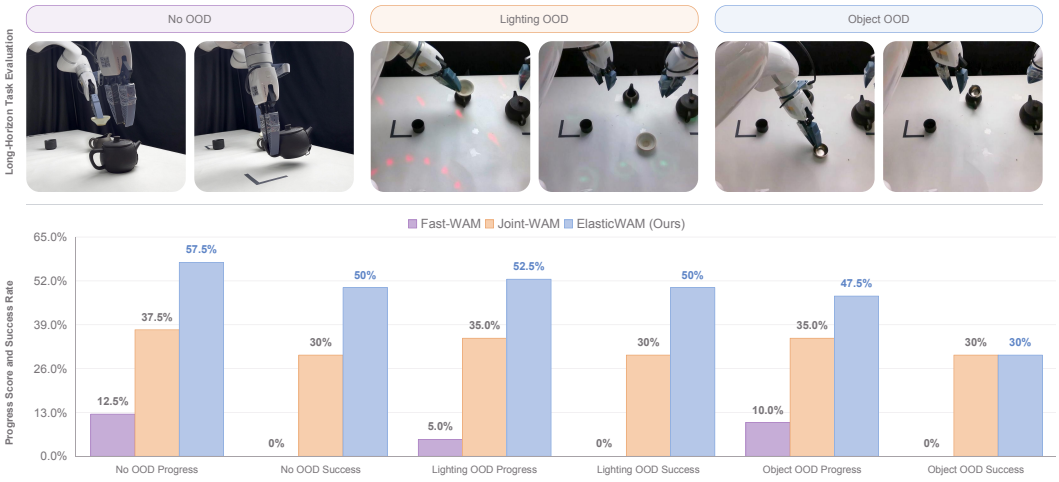


Figure 3: Real-world long-horizon tea-preparation results under nominal conditions, lighting perturbations, and cup replacement. Progress score (PS) measures the average fraction of the four stages completed; success rate (SR) measures full-task completion. Both metrics are reported as percentages, with higher values indicating better performance.

Figure 3 summarizes the real-world results under nominal conditions, lighting perturbations, and cup replacement. ElasticWAM achieves 50% full-task success under both nominal conditions and lighting perturbations, compared with 30% for Joint-WAM. Under cup replacement, both methods achieve 30% success, while ElasticWAM attains a higher progress score of 47.5% versus 35.0%.

ElasticWAM achieves the highest progress score across all three conditions, whereas Fast-WAM does not complete the full task under any condition. Under cup replacement, the higher progress score despite equal full-task success indicates that ElasticWAM completes more stages on average. Progress score therefore captures improvements in long-horizon execution that full-task success alone does not reveal.

4.3 Ablation Studies

We ablate spatial selection, temporal control, and token-refresh budget on RoboCasa-GR1 (1,200 episodes) and a shared LIBERO-Plus subset (1,506 episodes). Overall success aggregates successes across episodes; Adaptive and $K = 96$ in Table 6 use the same reference configuration.

Spatial selection and future refinement. On RoboCasa-GR1 (Table 5), learned selection at $K = 96$ achieves 57.42% success, exceeding random selection at the same budget by 3.59 percentage points and all-active refinement ($K = 128$) by 2.25 points. These results support selective regional updates: learned allocation improves control while refreshing 25% fewer future tokens per subsequent Joint update than all-active refinement. The full policy also exceeds FirstKV, which reuses the first-pass world cache within each replan, by 2.92 points, supporting continued adaptive refinement in this setting. The gains over FirstKV and all-active refinement support combining cache reuse with selective subsequent updates. At the same per-update token budget, learned selection also outperforms random selection, indicating that the allocation of refinement matters for control.

Table 5: Spatial selection and future refinement on RoboCasa-GR1 (success rate, %).

Variant	Cab.	Drawer	Micro.	CutBoard	Placemat	Plate	Tray	Overall
Learned WHERE, $K = 96$	65.00	65.00	50.00	63.60	48.50	60.00	53.20	**57.42**
Random WHERE, $K = 96$	61.00	63.00	50.00	55.60	47.50	59.00	48.00	53.83
All-active, $K = 128$	61.00	60.00	45.00	60.40	50.00	59.50	50.40	55.17
FirstKV	62.00	54.00	53.00	59.60	46.00	58.50	50.80	54.50

The first three rows use adaptive temporal control. K counts refreshed future tokens after the initial dense Joint update.

Adaptive temporal control. With learned WHERE fixed at $K = 96$, WHERE-only uses fixed progression and commit, achieving 71.71% success (Table 6(a)). Enabling adaptive commit raises success to 72.31%; additionally enabling adaptive progression yields 73.71%. The full controller improves over WHERE-only by 2.00 points, with gains in six of seven perturbation categories, largest under Robot and Camera perturbations. This supports adapting both temporal decisions along the evaluated sequence of configurations.

Table 6: Temporal control and token budget on LIBERO-Plus (success rate, %).

Variant	Camera	Robot	Lang.	Light	Backg.	Noise	Layout	Overall
(a) Temporal control: learned WHERE, $K = 96$								
WHERE-only	47.92	71.72	94.67	93.21	59.46	58.75	81.82	71.71
Fixed step	47.50	75.25	94.26	94.44	61.08	59.14	80.91	72.31
Adaptive	52.08	76.26	95.49	95.06	60.00	61.09	81.36	**73.71**
(b) Token-refresh budget: adaptive temporal control								
$K = 49$	48.75	73.23	94.26	96.30	62.16	59.14	82.73	72.84
$\boldsymbol{K = 96}$	52.08	76.26	95.49	95.06	60.00	61.09	81.36	**73.71**
$K = 147$	46.67	68.69	94.67	94.44	60.54	59.92	82.73	71.71
$K = 196$	46.25	69.19	94.26	95.06	59.46	57.98	80.45	70.92

Fixed step uses fixed progression with adaptive commit. K counts refreshed future tokens after the initial dense Joint update, out of 196 available tokens.

Token-refresh budget. Among $K \in \{49, 96, 147, 196\}$, $K = 96$ achieves the highest overall success at 73.71%, while $K = 49$ remains within 0.87 points (Table 6(b)). Larger budgets of $K = 147$ and 196 yield 71.71% and 70.92%, respectively. Thus, refreshing more regions does not necessarily improve control; selective refinement achieves stronger aggregate performance in this evaluation. The best-performing budget also varies across perturbation categories, suggesting that a single fixed refresh budget may not be optimal across evaluation conditions.

4.4 Qualitative Comparisons

Additional precision-sensitive placement comparisons with Fast-WAM and a long-horizon real-world rollout are provided in Appendix B.

Selective refinement under observation noise. Figure 4 compares Joint-WAM and ElasticWAM under noisy observations in LIBERO-Plus. Although their action-to-video attention maps show broadly similar spatial focus, the video representations supplied to the action branch differ. ElasticWAM selectively refreshes future-video representations and retains cached keys and values elsewhere, allowing action generation to draw on regional features with different refinement histories. Similar attention patterns therefore need not produce similar conditioning signals. Retaining earlier representations may help preserve useful action cues when further denoising produces less reliable predictions. In the illustrated example, ElasticWAM succeeds with one to three updates per future token, compared with ten for Joint-WAM. Selective refresh thus reduces video-side computation while preserving access to the full future context.

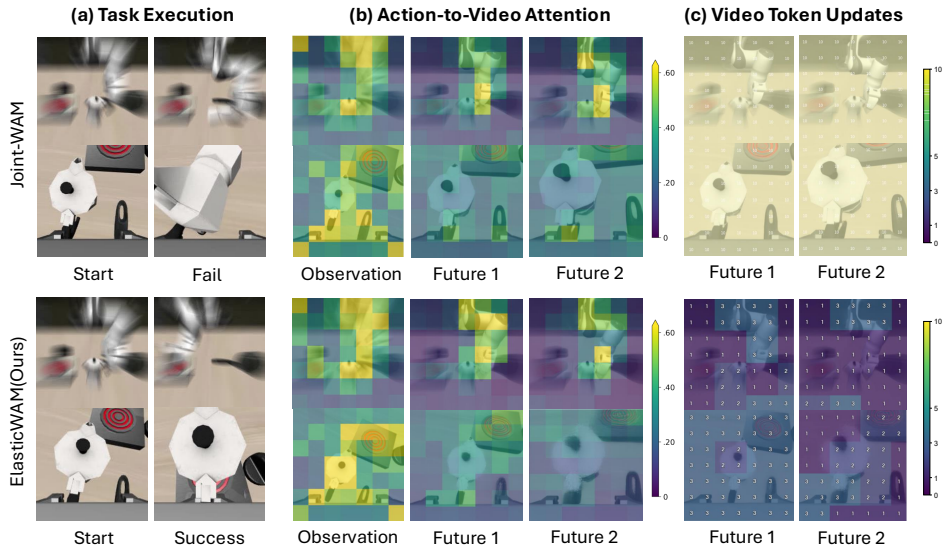


Figure 4: Joint-WAM (top) and ElasticWAM (bottom) under visual observation noise in LIBERO-Plus. (a) Task outcomes. (b) Action-to-video attention. (c) Per-token future-update counts.

5 Conclusion

We presented ElasticWAM, an action-guided framework for adaptive spatiotemporal imagination in world–action models. ElasticWAM learns which future-video regions to refresh, how far to advance denoising, and when to stop world refinement. It selectively updates future-video representations while reusing cached representations of unrefreshed regions through the pretrained action-conditioning interface. Our results show that effective control does not require refreshing every future-video region at every denoising step. Across multiple manipulation benchmarks and long-horizon real-world dual-arm tasks, ElasticWAM improves task success while reducing inference latency relative to dense joint refinement. Future work will investigate state-dependent token-refresh budgets and more efficient action-denoising schedules, aiming to further reduce inference latency while maintaining control performance.

Reproducibility Statement

Our method and training procedure are described in Section 3, and evaluation settings and protocols are provided in Section 4 and Appendix A. We will release the code for refinement-controller training and inference, together with the trained controller checkpoints, upon publication.

AI Use Statement

In this work, we used ChatGPT to assist with experimental code implementation. Additionally, ChatGPT was used to improve the language and organization of the manuscript. The authors reviewed the AI-assisted text and checked and tested the AI-assisted code used in the experiments. We take responsibility for the final content of this work, including the manuscript, experimental results, and accompanying code.

References

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025. URL https://arxiv.org/abs/2512.13030.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *ArXiv Preprint ArXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025. URL https://arxiv.org/abs/2505.06111.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. WorldVLA: Towards autoregressive action world model. *ArXiv Preprint ArXiv:2506.21539*, 2025. URL https://arxiv.org/abs/2506.21539.

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

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025a. URL https://arxiv.org/abs/2502.19645.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning*, pp. 2679–2713. PMLR, 2025b.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *ArXiv Preprint ArXiv:2601.21998*, 2026. URL https://arxiv.org/abs/2601.21998.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023. URL https://arxiv.org/abs/2306.03310.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent Consistency Models: Synthesizing high-resolution images with few-step inference. In *International Conference on Learning Representations*, 2024.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. URL https://arxiv.org/abs/2503.14734.

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

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

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

Wenhao Sun, Rong-Cheng Tu, Yifu Ding, Zhao Jin, Jingyi Liao, Shunyu Liu, and Dacheng Tao. VORTA: Efficient video diffusion via routing sparse attention. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Yirui Sun, Guangyu Zhuge, Keliang Liu, Jie Gu, Shiqin Dai, Xinyu Bing, Zhongxue Gan, and Chunxu Tian. SANTS: A state-adaptive scheduler for world action models. *ArXiv Preprint ArXiv:2605.27947*, 2026. URL https://arxiv.org/abs/2605.27947.

Hancheng Ye, Jiakang Yuan, Renqiu Xia, Xiangchao Yan, Tao Chen, Junchi Yan, Botian Shi, and Bo Zhang. Training-free adaptive diffusion with bounded difference approximation strategy. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *ArXiv Preprint ArXiv:2602.15922*, 2026. URL https://arxiv.org/abs/2602.15922.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang 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.

Hui Zhang, Zuxuan Wu, Zhen Xing, Jie Shao, and Yu-Gang Jiang. AdaDiff: Adaptive step selection for fast diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9914–9922, 2025a. doi: 10.1609/aaai.v39i9.33075.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric Xing, and Hao Zhang. Faster video diffusion with trainable sparse attention. *Advances in Neural Information Processing Systems*, 38:152509–152534, 2025b.

Yuechen Zhang, Jinbo Xing, Bin Xia, Shaoteng Liu, Bohao Peng, Xin Tao, Pengfei Wan, Eric Lo, and Jiaya Jia. Training-free efficient video generation via dynamic token carving. In *Advances in Neural Information Processing Systems*, volume 38, 2025c.

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

Weiheng Zhao, Haoyi Jiang, Xin Shi, Liu Liu, Fan Huang, Zhizhong Su, Wei Sui, and Xinggang Wang. Faster-WAM: Efficient inference-time future conditioning for robust world action models. *arXiv preprint arXiv:2608.04404*, 2026. URL https://arxiv.org/abs/2608.04404.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025. URL https://arxiv.org/abs/2510.10274.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025. URL https://arxiv.org/abs/2510.03827.

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

A Experimental Details

This appendix expands the experimental setup in Section 4.1. Task success is the primary metric, and inference latency is additionally reported to assess the performance–efficiency trade-off.

Evaluation protocols and baseline sources. For LIBERO and LIBERO-Plus, we follow the evaluation configurations and protocols used in Faster-WAM (Zhao et al., 2026). Baseline results sourced from Faster-WAM are quoted as reported. For RoboCasa-GR1, we follow the evaluation setting provided by StarVLA (StarVLA Community, 2026).

LIBERO. We use the Spatial, Object, Goal, and Long suites of LIBERO (Liu et al., 2023) to evaluate language-conditioned manipulation across different spatial arrangements, manipulated objects, task goals, and extended action sequences. Each suite comprises 10 tasks with 50 demonstrations per task. Our evaluation uses 50 rollouts for each task, resulting in 2,000 trials across the four suites. We report success rates for individual suites and their aggregate performance.

LIBERO-Plus. We assess out-of-distribution robustness using LIBERO-Plus (Fei et al., 2025), which introduces seven perturbation categories: camera viewpoint, robot initialization, language instruction, illumination, background appearance, sensor noise, and object layout. The LIBERO-trained policy and refinement controller are evaluated without additional adaptation to these perturbations. Results are reported separately for each category to characterize robustness beyond the standard LIBERO setting. For ElasticWAM, overall success is the total number of successful episodes divided by the total number of evaluated episodes, weighting categories by their episode counts. On the 1,506-episode subset, 1,110 successes yield 73.71%, reported as 73.7% in Table 3.

LIBERO-Pro. We further evaluate robustness using the Object, Position, Semantic, and Task categories of LIBERO-Pro (Zhou et al., 2025). We report category-wise success rates together with their unweighted average, providing a complementary assessment of sensitivity to changes in task specification and scene configuration.

RoboCasa-GR1. We evaluate tabletop manipulation using the GR1 benchmark (NVIDIA et al., 2025), which contains 24 tasks involving object rearrangement and interaction with articulated objects. We report task success alongside inference latency to examine the performance–efficiency trade-off of ElasticWAM relative to Fast-WAM and Joint-WAM.

Real-world evaluation. We collect 200 demonstration trajectories for the four-stage dual-arm tea-preparation task. ElasticWAM, Fast-WAM, and Joint-WAM are evaluated under nominal conditions, lighting perturbations, and replacement of the white cup with a silver-colored cup, without human intervention. We report full-task success rate (SR) and progress score (PS). SR measures the percentage of evaluations in which all four stages are completed. PS is the average number of completed stages divided by four, expressed as a percentage; each stage receives one point only when all operations within that stage are completed. The task sequence is described in Section 4.1, quantitative results are reported in Section 4.2 and Figure 3, and a successful rollout is shown in Figure 6.

Task instruction. The instruction is: "Place the white cup on top of the uncovered teapot, pour tea from the covered teapot into the white cup, return the white cup to its original position, then pour tea from the uncovered teapot into the black cup and return the teapot to its original position."

B Additional Qualitative Comparisons

Precision-sensitive placement. Figure 5 shows two RoboCasa-GR1 examples requiring accurate object placement. Fast-WAM moves the objects near their destinations but fails to place them correctly, whereas ElasticWAM achieves successful placement. These examples illustrate differences in execution accuracy during the final placement stage.

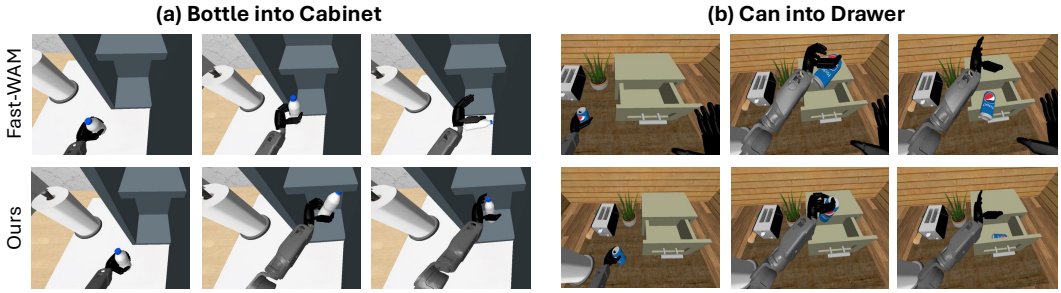


Figure 5: Precision-sensitive placement on RoboCasa-GR1: (a) bottle into cabinet and (b) can into drawer. Rows show Fast-WAM (top) and ElasticWAM (bottom); keyframes progress from left to right and highlight the placement stage.

Real-world long-horizon execution. Figure 6 presents keyframes from a successful rollout of the four-stage tea-preparation task described in Section 4.1. The sequence illustrates successive grasping, placement, pouring, and object-return operations on the dual-arm robot.

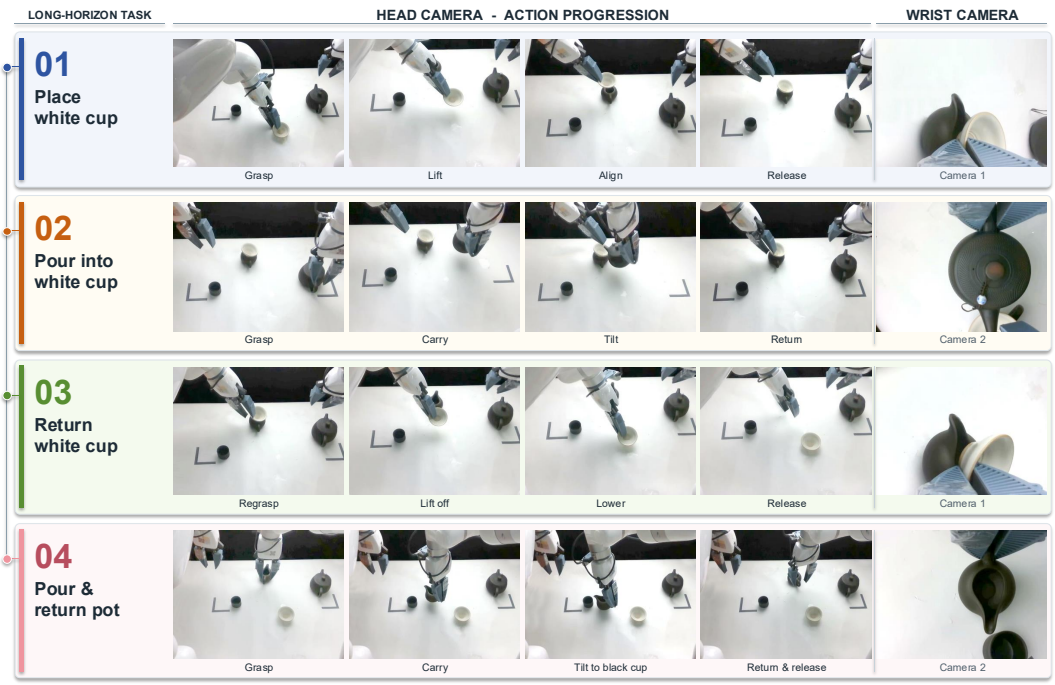


Figure 6: Real-world long-horizon tea preparation. Keyframes show a successful rollout through the four sequential stages described in Section 4.1, including cup placement, pouring, and object return.