NEXT-WAM: ANY-TO-ANY WORLD ACTION MODEL FOR ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models and World-Action Models (WAMs) have advanced robot manipulation by leveraging multimodal understanding and predictive modeling, respectively. Nevertheless, both paradigms struggle to jointly reason over task intent, forecast visual outcomes, and output precise actions. In this work, we introduce **Next-WAM**, an any-to-any World-Action Model that unifies these capabilities within a single multimodal framework. Built upon a Mixture-of-Transformers (MoT) architecture, our model consists of understanding, generation, and action experts with disjoint parameters and structured joint attention. It fuses modality-specific encodings while enabling cross-modal information exchange. Powered by this design, Next-WAM consumes visual history, produces subtask text as control guidance, predicts future observations, and outputs continuous action chunks. Notably, by masking future-image latents from the action attention stream, Next-WAM leverages visual prediction to supervise control representations, without needing to generate future images at action inference time. Trained without extra embodied pretraining, Next-WAM attains 98.4% success on LIBERO and 70.5% on LIBERO-Plus, outperforming Fast-WAM by 0.8% and 19.0%, respectively. Real-robot experiments further validate Next-WAM's capacity to reason from task instructions for manipulation. Code will be made public.

1 INTRODUCTION

The ambition behind general-purpose agents is to enable machines to exhibit human-level perception, reasoning, and operation within real-world physical environments. Recent progress in embodied models has largely been driven by *Vision-Language-Action (VLA)* policies (Brohan et al., 2022; 2023; Kim et al., 2024; Black et al., 2024; 2025; Bjorck et al., 2025; Wu et al., 2026). These methods extend pretrained Vision-Language Models (VLMs) to map visual observations and language command directly to robot actions. Subsequent approaches, such as ECoT (Zawalski et al., 2025), OneTwoVLA (Lin et al., 2025), and CoT-VLA (Zhao et al., 2025), incorporate explicit language reasoning into VLA models to guide action prediction, as shown in Fig. 1 (a). However, direct action prediction provides limited supervision about how actions change the environment. It may therefore struggle to capture physical dynamics and long-horizon task structure (Du et al., 2023; 2024).

World-Action Models (WAMs) (Hu et al., 2024; Pai et al., 2025; Bi et al., 2026; Kim et al., 2026; Ye et al., 2026; Yuan et al., 2026b) extend action learning to include the prediction of future visual observations, as shown in Fig. 1 (b). These methods usually build on video generation models and can be broadly grouped into two categories. Early works follow an imagine-then-execute paradigm, explicitly generating future visual trajectories and using an inverse dynamics model to infer actions from the imagined transitions (Du et al., 2023; Zhou et al., 2024). Recent WAMs use future visual prediction as a training objective to learn physical priors and representations of interaction dynamics for action generation without requiring future generation at inference time (Pai et al., 2025; Ye et al., 2026). Fast-WAM (Yuan et al., 2026b) further demonstrates efficient action generation through video co-training without requiring future imagination at inference time.

Despite substantial progress in WAMs, two challenges remain. First, language often serves as a task condition rather than an explicit reasoning process in WAMs. While future visual prediction helps capture environmental dynamics, it does not directly express instruction constraints, subtask decisions, or the semantic rationale behind an action. Recent methods, including WoG (Su et al.,

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

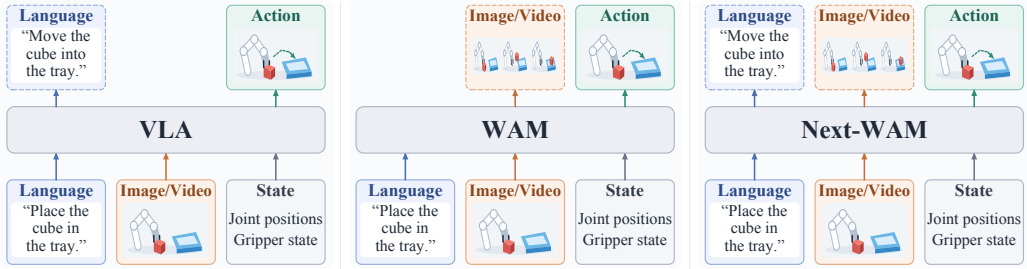


Figure 1: **Comparison of VLA, WAM, and Next-WAM.** (a) VLAs map visual observations, language instructions, and robot states to actions, with some also generating language (Lin et al., 2025; Zhao et al., 2025; Huang et al., 2026). (b) WAMs jointly model actions and future images or videos, typically without explicit language generation (Yuan et al., 2026b; Zhang et al., 2026). (c) Our Next-WAM unifies visual, language, and action generation with flexible multimodal conditioning.

2026) and BagelVLA (Hu et al., 2026), incorporate world modeling into VLAs, bringing semantic understanding and future visual prediction into a shared framework. These advances motivate a more explicit role for language reasoning in WAMs: connecting anticipated visual outcomes with task-level decisions that guide action generation. Second, language reasoning, visual prediction, and action generation have different representational requirements. Language reasoning relies on semantic abstraction, whereas visual prediction requires detailed spatial and temporal information, and action generation requires precise continuous control (Wu et al., 2025; Driess et al., 2025). Optimizing these tasks with fully shared parameters can introduce interference, making it difficult to improve one capability without compromising another.

To address these limitations, we introduce **Next-WAM**, an *any-to-any World-Action Model* supporting text, images, videos, and actions as inputs and outputs (Fig. 1(c)). Next-WAM integrates language reasoning with visual prediction and continuous control through three specialized experts. Separate experts reduce interference between language reasoning, visual prediction, and action learning, while joint attention allows the action expert to draw on both semantic and visual context. Motivated by Fast-WAM (Ma et al., 2026), our model learns from future-image supervision without requiring future-image generation during action inference. Table 2 compares the input and output modalities of Next-WAM and existing methods. This framework enables Next-WAM to support four key features:

Table 1: **Comparison of different models on inputs and outputs.** T: Text; I: Image; V: Video.

	Input			Output		
Method	T	I	V	T	I	V
OpenVLA (Kim et al., 2024)	✓	✓	✗	✗	✗	✗
$\pi_{0.5}$ (Black et al., 2025)	✓	✓	✗	✗	✗	✗
OneTwoVLA (Lin et al., 2025)	✓	✓	✗	✓	✗	✗
MEM (Torne et al., 2026)	✓	✓	✓	✗	✗	✗
BagelVLA (Hu et al., 2026)	✓	✓	✗	✓	✓	✗
Fast-WAM (Yuan et al., 2026b)	✓	✓	✗	✗	✗	✓
Image-WAM (Zhang et al., 2026)	✓	✓	✗	✗	✓	✗
Next-WAM (Ours)	✓	✓	✓	✓	✓	✓

(1) Visual History. By incorporating past observations alongside the current image, Next-WAM grounds control in the temporal evolution of the scene. This provides access to motion cues and previously observed information that a single image may not capture.

(2) Text Reasoning. Next-WAM makes language an explicit intermediate output for control. Conditioned on instructions and visual observations, the understanding expert generates subtask text that provides semantic guidance for subsequent action prediction.

(3) Future Imagination. Next-WAM learns to anticipate visual changes through future-image prediction, bringing supervision of environmental dynamics into policy learning. This predictive objective shapes visual representations without requiring future-image generation during action inference.

(4) Action Prediction. A dedicated action expert combines semantic and visual context with proprioceptive information to predict continuous action chunks. By executing a prefix and replanning from updated observations, Next-WAM connects multimodal understanding with closed-loop control.

We extensively evaluate Next-WAM in both simulation and real-world environments. On LIBERO, Next-WAM outperforms or matches baseline models, while transferring the LIBERO-trained pol-

icy to LIBERO-Plus further demonstrates its robustness. In real-world experiments, Next-WAM achieves superior performance on tasks requiring text-based reasoning, where baseline methods often struggle. In summary, our contributions are threefold:

- We introduce **Next-WAM**, an **any-to-any World-Action Model** that unifies language reasoning, visual prediction, and continuous control, supporting flexible combinations of text, images, videos, and actions as inputs and outputs.
- We develop a **three-expert architecture** with joint attention to reduce cross-task interference, enabling action generation guided by predictive visual supervision without requiring future-image generation at inference time.
- Experiments on LIBERO, LIBERO-Plus, and real-world tasks demonstrate Next-WAM's **strong performance, transfer robustness**, and ability to handle complicated tasks requiring text-based reasoning.

2 Related Works

2.1 Vision-Language-Action models (VLA)

Vision-Language-Action (VLA) models learn to map visual observations and language instructions directly to executable robot actions. Pioneering works of this paradigm, including RT-1, RT-2 and OpenVLA, adopt Transformer architectures (Vaswani et al., 2017) and tokenize continuous actions into discrete tokens (Brohan et al., 2022; 2023; Kim et al., 2024). Subsequent models, including Diffusion Policy, π series, GR00T, and LingBot-VLA, adopt diffusion or flow-matching decoders as their action heads, yielding continuous control trajectories via an iterative denoising process (Chi et al., 2023; Black et al., 2024; 2025; Bjorck et al., 2025; Wu et al., 2026). Some follow-up studies further explore language reasoning by harnessing the reasoning capabilities of large language models, enabling robots to decompose long-horizon instructions, resolve ambiguous commands, and perform step-by-step planning before executing low-level actions (Zhao et al., 2025; Huang et al., 2026; Ye et al., 2025; Lin et al., 2025; Liu et al., 2026a;b).

More recently, several works have explored integrating VLA models with world modeling to improve action generation through future prediction. These approaches can be broadly categorized into two types: those predicting explicit future visual observations, and those predicting latent representations of future observations. The former includes GR-1 (Wu et al., 2024), GR-2 (Cheang et al., 2024), UP-VLA (Zhang et al., 2025), WorldVLA (Cen et al., 2025b), BagelVLA (Hu et al., 2026), and RynnVLA-002 (Cen et al., 2025a), which incorporate future image generation into VLA learning to capture environmental dynamics. The latter includes FLARE (Zheng et al., 2025), VLA-JEPA (Sun et al., 2026), and WoG (Su et al., 2026), which predict future latent representations to guide action generation. *Overall, VLA models primarily focus on action generation, with world modeling serving an auxiliary role, motivating more integrated World-Action Models (WAMs).*

2.2 World-Action Models (WAM)

World-Action Models (WAMs) integrate world prediction and action generation, leveraging learned environmental dynamics for robot control. Early works focus on video-based planning: UniPi (Du et al., 2023) and RoboDreamer (Zhou et al., 2024) generate language-conditioned visual plans and recover executable actions through inverse dynamics. Methods such as VidMan (Wen et al., 2024), VPP (Hu et al., 2024), mimic-video (Pai et al., 2025), and GE-Act (Liao et al., 2026) leverage dynamics information encoded within video models to generate actions without explicitly decoding future video frames. Related approaches, including UVA (Li et al., 2025), UWM (Zhu et al., 2025), and Motus (Bi et al., 2026), emphasize joint generative modeling of video and actions, supporting multiple prediction tasks within a unified framework. Building on large pretrained video models, DreamZero (Ye et al., 2026), Cosmos Policy (Kim et al., 2026), and LingBot-VA (Li et al., 2026a) leverage learned spatiotemporal priors to improve generalization in robot control. Notably, Fast-WAM (Yuan et al., 2026b) retains video supervision during training while bypassing future video generation during action inference. Follow-up methods (Yu et al., 2026; Cai et al., 2026a; Ma et al., 2026; Li et al., 2026b; Yuan et al., 2026a) further improve the robustness and efficiency of WAMs. Complementary approaches, such as $\pi_{0.7}$ (Ai et al., 2026) and τ_0-VLA (Cai et al., 2026b), augment

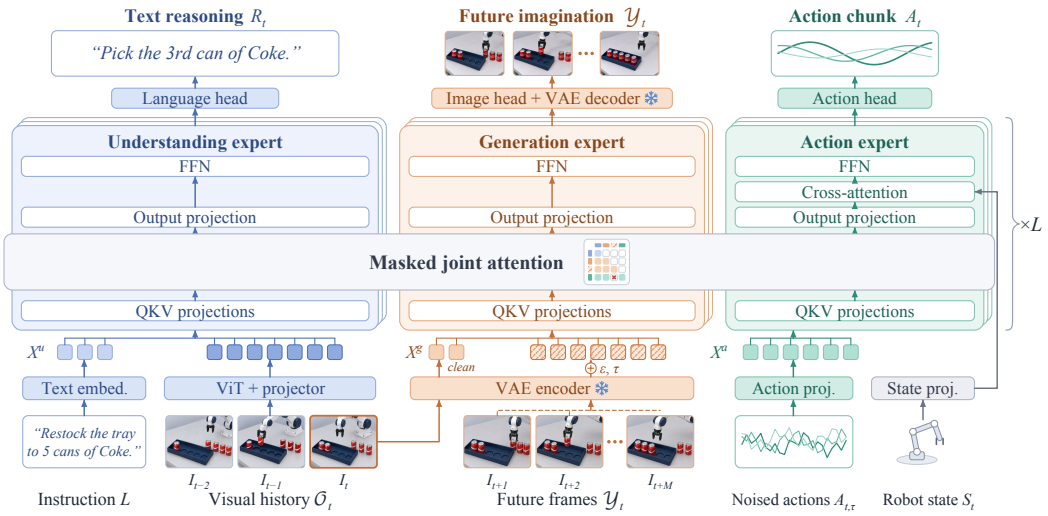


Figure 2: **Overview of Next-WAM.** Three experts, *i.e.*, understanding expert, generation expert, and action expert, unify language and visual understanding, future-image prediction, and continuous action generation within a Mixture-of-Transformers (MoT) architecture. Each expert uses separate query, key, and value projections and feed-forward networks, while exchanging information through structured joint attention. The attention mask is illustrated in Fig. 3.

VLA policies with separate generative world models to provide visual subgoals. *Despite their strong performance, many of these approaches focus on visual dynamics and action generation, leaving the language reasoning capabilities of LLMs underexplored.*

3 METHOD

Next-WAM unifies language-visual understanding, future visual prediction, and continuous robot control via three interoperable experts, as visualized in Fig. 2. We begin by detailing each expert module. Next, we introduce token routing and structured attention mechanisms that mediate cross-expert information exchange. Lastly, we describe the joint training objective and action generation.

3.1 PROBLEM FORMULATION

At decision step t, the robot observes a natural language instruction L, a sequence of K images $\mathcal{O}_t = (I_{t-K+1}, \ldots, I_t)$ sampled at fixed intervals, and a proprioceptive state S_t. The robot policy requires predicting an action chunk $A_t = (a_t, \ldots, a_{t+H-1})$:

$$A_t \sim \pi_\theta(\cdot \mid L, \mathcal{O}_t, S_t). \tag{1}$$

Beyond action prediction, our model supports future-image prediction and optional text reasoning. Let $\mathcal{Y}_t = (I_{t+1}, \ldots, I_{t+M})$ denote the future image sequence and R_t the textual reasoning sequence. In the language-supervised setting, the joint prediction is formulated as:

$$(R_t, \mathcal{Y}_t, A_t) \sim \pi_\theta(\cdot \mid L, \mathcal{O}_t, S_t). \tag{2}$$

Next, we present a detailed description of the model architecture.

3.2 MODEL ARCHITECTURE

To support language reasoning, image generation, and action prediction within a unified framework, Next-WAM adopts a Mixture-of-Transformers (MoT) architecture (Liang et al., 2024). Each expert processes a dedicated token group with its own parameters, while masked joint attention enables information exchange across groups.

Understanding expert. The understanding expert extracts semantic context from language and visual observations. It processes the instruction and observed image history as a sequence of tokens:

$$X_t^{\mathrm{u}} = [E_{\mathrm{text}}(L); E_{\mathrm{vis}}(I_{t-K+1}); \ldots; E_{\mathrm{vis}}(I_t)], \tag{3}$$

where $[\cdot;\cdot]$ denotes token concatenation. E_{text} is the language model's token embedding layer, and E_{vis} consists of the Qwen2.5-VL ViT (Bai et al., 2025) and an MLP projector that transforms visual features. These semantic visual tokens connect scene content with the task instruction, while the observation history provides temporal context for interpreting scene changes. The resulting representations serve as semantic conditioning for image and action generation through joint attention. When subtask annotations are available, the expert predicts R_t autoregressively through a language head, allowing intermediate task decisions to be expressed as text.

Generation expert. The generation expert models future visual observations using continuous latent representations that preserve appearance and spatial structure complementary to the understanding expert's semantic features. A frozen Wan2.2 VAE (Team Wan et al., 2025) encodes the current and future images:

$$X_t^{\text{g}} = [E_{\text{vae}}(I_t); E_{\text{vae}}(\mathcal{Y}_t)]. \tag{4}$$

The current-image latents remain clean and provide a visual reference for future prediction, while future-image latents are corrupted at flow-matching time τ. The latent features are linearly projected into the expert's hidden space and augmented with position and noise-level embeddings. For notational simplicity, Eq. 4 omits noise corruption and the latent-to-token projection. Conditioned on the semantic context and current-image latents, the expert predicts future-image latent velocities through a linear output head. This prediction objective trains the model to capture visual changes associated with robot interactions.

Action expert. The action expert predicts continuous actions, using a linear projection to encode the noised action chunk into tokens:

$$X_t^{\text{a}} = E_{\text{a}}(A_{t,\tau}), \tag{5}$$

where $A_{t,\tau}$ denotes the action chunk corrupted at flow-matching time τ. The proprioceptive state S_t is linearly projected and supplied through cross-attention. With noise-conditioned Transformer blocks, this expert predicts action velocities through an action output head.

Mixture-of-Transformers. The three experts maintain separate Transformer parameters and exchange information through joint attention. At each layer, let X^e denote the token representations of expert $e \in \{\text{u}, \text{g}, \text{a}\}$. Each expert computes its own queries, keys, and values:

$$Q^e = X^e W_Q^e, K^e = X^e W_K^e, V^e = X^e W_V^e. \tag{6}$$

These representations are concatenated across experts:

$$Q = [Q^{\text{u}}; Q^{\text{g}}; Q^{\text{a}}], K = [K^{\text{u}}; K^{\text{g}}; K^{\text{a}}], V = [V^{\text{u}}; V^{\text{g}}; V^{\text{a}}]. \tag{7}$$

Joint attention is then computed as:

$$[U^{\text{u}}; U^{\text{g}}; U^{\text{a}}] = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + \mathcal{M}\right) V, \tag{8}$$

where d_k is the attention head dimension and $\mathcal{M}$ controls token visibility. Each expert processes its attention outputs with a separate output projection and feed-forward network, using residual connections. This preserves expert-specific parameters while enabling masked information exchange.

Token visibility. Motivated by Fast-WAM (Yuan et al., 2026b), we retain future-image supervision while masking future-image latents from action attention. As shown in Fig. 3, action tokens X^{a} attend to the understanding tokens X^{u}, the current-image latents in X^{g}, and all action tokens, but not to future-image latents. Each future-image segment attends to X^{u}, the current-image latents, and itself. Within the observed context, segments attend to preceding segments, with causal attention within text segments and bidirectional attention within visual segments. Neither X^{u} nor X^{g} attends to X^{a}. This token visibility design allows future-image supervision to improve visual representations while avoiding future-image generation during action inference.

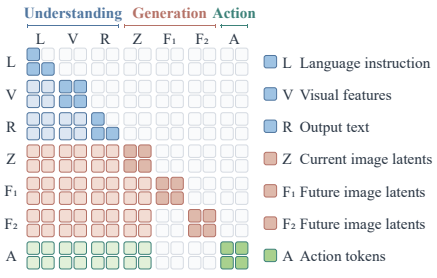


Figure 3: **Token visibility.** Colored cells indicate allowed attention. Empty cells are masked.

3.3 Training Objective and Action Generation

The three experts are trained jointly. The generation and action experts use conditional flow matching. For a clean target x and noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$,

$$x_\tau = (1-\tau)x + \tau\epsilon, u = \epsilon - x, \tau \in [0,1], \tag{9}$$

with image and action times sampled independently. Each expert regresses its own velocity: the generation expert on future-image latents, and the action expert on the action chunk. Observed-image latents stay clean and receive no loss. When an output text $R_t = (r_1, \ldots, r_J)$ is available, the understanding expert predicts it autoregressively from the instruction and the segment-start image,

$$\mathcal{L}_{\text{text}} = \frac{1}{J}\sum_{j=1}^{J} \log p_\theta(r_j \mid L, I^{\text{s}}, r_{<j}). \tag{10}$$

Samples without output text omit this term. The joint objective is:

$$\mathcal{L} = \lambda_{\text{img}}\mathcal{L}_{\text{img}} + \lambda_{\text{act}}\mathcal{L}_{\text{act}} + \lambda_{\text{text}}\mathcal{L}_{\text{text}}, \tag{11}$$

where $\mathcal{L}_{\text{img}}$ and $\mathcal{L}_{\text{act}}$ are the mean squared velocity errors of the two flow-matching experts.

Reason & Act. To switch between subtask reasoning and action generation, we designate two special control tokens, $\langle$*Reason*$\rangle$ and $\langle$*Act*$\rangle$. During training, the understanding expert learns to predict the control token from the instruction, current observation, proprioceptive state, and previous subtask. At inference, the model first selects a control token. Predicting $\langle$*Reason*$\rangle$ generates a new subtask until the end-of-sequence token, while predicting $\langle$*Act*$\rangle$ generates an action chunk conditioned on the retained subtask. The subtask persists across policy calls until the model chooses to update it, enabling adaptive reasoning as execution progresses.

Action generation. During inference, action chunks are sampled by integrating the action expert's predicted velocity from $\tau = 1$ to $\tau = 0$, starting from Gaussian noise. The robot executes a prefix of the sampled action chunk and replans upon receiving new sensory observations. Crucially, action inference requires no future-image generation or decoding.

4 Experiments

4.1 Implementation Details

Simulation benchmarks. We assess language-conditioned manipulation on LIBERO (Liu et al., 2023) using Spatial, Object, Goal, and Long, with 10 tasks in each suite. Next-WAM is trained on the original LIBERO demonstrations, *without additional embodied pretraining*. Evaluation uses 50 rollouts per task. To test whether performance carries over to changed conditions, we evaluate the LIBERO-trained checkpoint directly on LIBERO-Plus (Fei et al., 2025).

Real-robot tasks. We also evaluate two task families on a dual-arm mobile robot equipped with one egocentric and two wrist-mounted cameras: *Conditional Object Grasping* , and *Coke Restocking*. *Conditional Object Grasping* requires picking up the eggplant if an orange or a pear is present on the table, and picking up the mushroom if neither is present. The orange and pear serve as decision cues rather than grasp targets. *Coke Restocking* specifies a target inventory, like "Please restock 4 cokes". The robot must add missing cans and stop replenishing once the target is reached. Unlike an isolated pick-and-place task, this setting tests the interpretation of a numerical goal and the tracking of completed transfers across successive action cycles. Our preliminary evaluation considers $0\rightarrow1$, $1\rightarrow3$, and $3\rightarrow5$. In summary, beyond low-level manipulation, these tasks probe text reasoning, like language-conditioned logical selection, and numerical goal tracking.

Training. We initialize Next-WAM from the pretrained Lance-3B (Fu et al., 2026) and attach an action expert with about 382M parameters. The action expert contains 36 Transformer blocks with hidden and feed-forward dimensions of 512 and 2048, respectively. We train with AdamW for 10 epochs on LIBERO and 50k steps on real-robot tasks. We use an effective batch size of 128 across eight GPUs. The VAE is frozen, and all remaining model parameters are optimized. More training details can be found in Appendix A.

Table 2: **Performance comparison on LIBERO.** We report success rates (%) on the four suites and their average. Embodied P.T. indicates whether a method uses additional embodied pretraining.

Method	Embodied P.T.	Spatial	Object	Goal	Long	Average
OpenVLA (Kim et al., 2024)	✓	84.7	88.4	79.2	53.7	76.5
π_0 (Black et al., 2024)	✓	96.8	98.8	95.8	85.2	94.1
$\pi_{0.5}$ (Black et al., 2025)	✓	98.8	98.2	98.0	92.4	96.9
LingBot-VA (Li et al., 2026a)	✓	98.5	99.6	97.2	98.5	98.5
Motus (Bi et al., 2026)	✓	96.8	99.8	96.6	97.6	97.7
Fast-WAM (Yuan et al., 2026b)	✗	98.2	100.0	97.0	95.2	97.6
Next-WAM (Ours)	✗	98.6	99.8	98.2	97.0	98.4

Table 3: **Robustness comparison on LIBERO-Plus.** We report success rates (%) across seven perturbation categories and their average. Our Next-WAM is evaluated directly using the LIBERO-trained checkpoint, without further adaptation. P.T. denotes additional embodied pretraining.

Method	P.T.	Camera	Robot	Language	Light	Background	Noise	Layout	Avg
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., 2025)	✓	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., 2025b)	✓	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
FastWAM (Yuan et al., 2026b)	✗	16.4	44.5	68.9	78.2	53.7	37.7	60.7	51.5
Next-WAM (Ours)	✗	61.0	64.1	75.9	83.1	64.6	74.2	72.2	70.5

Inference. The default policy generates action chunks with 10 flow-matching integration steps. To accelerate inference, we cache the layer-wise keys and values of the visual and language context during the first step and update only the action tokens in subsequent steps. We further use CUDA Graphs for the initial joint forward pass and the remaining action integration loop to reduce kernel launch overhead. The context cache is refreshed at each replanning cycle, and action prediction does not require future-image generation. More details about inference acceleration are in Appendix D.

4.2 Main Results

LIBERO. Table 2 compares Next-WAM with representative VLAs, such as OpenVLA (Kim et al., 2024), π_0 (Black et al., 2024), $\pi_{0.5}$ (Black et al., 2025), LingBot-VA (Li et al., 2026a), and WAMs, such as Motus (Bi et al., 2026) and Fast-WAM (Yuan et al., 2026b). Next-WAM achieves an average success rate of 98.4%, with 98.6%, 99.8%, 98.2%, and 97.0% on Spatial, Object, Goal, and Long, respectively. **It obtains the highest average among the compared methods without embodied pretraining.** Its average success rate is only 0.1 percentage points below that of LingBot-VA. Compared with Fast-WAM (Yuan et al., 2026b), Next-WAM improves average success by 0.8 percentage points, demonstrating competitive manipulation performance.

Table 4: Temporal offsets of input and predictions.

Input/Predictions	Spatial	Object	Goal	Long	Avg
$[0, 32]$	98.6	99.8	98.2	97.0	98.4
$[-8, -4, 0, 32]$	96.2	99.8	97.2	96.6	97.4
$[-16, -8, 0, 32]$	98.0	99.8	95.6	96.2	97.4
$[0, 4, \ldots, 32]$	98.4	99.4	97.8	93.4	97.2

Visual history. We vary the input image history while keeping the predicted image at offset 32, as shown in Table 4. Using only the current image at offset 0 achieves an average success rate of 98.4%, whereas adding images at offsets $[-8, -4]$ or $[-16, -8]$ yields 97.4% in both cases. Additional input history therefore has only a modest effect on performance under the evaluated configurations. Further, we examine whether denser future supervision affects performance. Using the current image at offset 0 as input, we predict images at offsets $4, 8, \ldots, 32$ and compare this setting with predicting only the image at offset 32. This variant yields competitive performance.

Future imagination. We visualize the future imagination when validating on LIBERO. As seen in Appendix C, the model forecasts plausible scene evolution and viewpoint changes induced by robot movement. Notably, the action policy does not require these predicted images during inference.

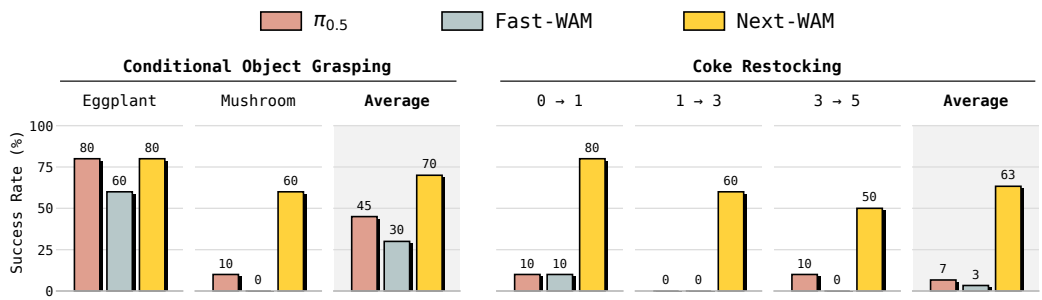


Figure 4: **Real-robot success rates.** Each setting is evaluated over 10 trials. *Conditional Object Grasping* covers the eggplant and mushroom branches of the same instruction, and *Coke Restocking* covers three count settings. Each Average column pools all trials of its task family.

Table 5: **Ablation studies** for (a) action expert parameter count and (b) IDM variant on LIBERO.

(a) Action expert parameter count.

AE	Spatial	Object	Goal	Long	Average
0.2B	96.6	98.8	91.0	82.2	92.1
0.4B	98.6	99.8	98.2	97.0	98.4
1.0B	98.0	99.6	97.8	93.2	97.2

(b) IDM variant.

Policy	Spatial	Object	Goal	Long	Average
Next-WAM	98.6	99.8	98.2	97.0	98.4
Next-WAM-IDM	97.4	99.6	97.2	95.0	97.3

LIBERO-Plus. Table 3 evaluates robustness across seven perturbation categories on LIBERO-Plus benchmark. Next-WAM achieves 70.5% aggregate success, the highest reported overall score among the compared methods. It outperforms Fast-WAM (Yuan et al., 2026b) by 19.0 percentage points. Next-WAM improves over Fast-WAM in every perturbation category. The largest gains occur under camera viewpoint changes and sensor noise, where success increases from 16.4% to 61.0% and from 37.7% to 74.2%, respectively. These results show that the gains observed on standard LIBERO are accompanied by broader robustness improvements.

Real-robot experiments We compare $\pi_{0.5}$ (Black et al., 2025), Fast-WAM (Yuan et al., 2026b), and Next-WAM on two manipulation task families: *Coke Restocking*, and *Conditional Object Grasping*. The policies run closed-loop on the same dual-arm mobile manipulator, with identical camera inputs and low-level controllers, and each setting is evaluated over 10 trials. As shown in Fig. 4, Next-WAM achieves an average success rate of 70% on Conditional Object Grasping, compared with 45% for $\pi_{0.5}$ and 30% for Fast-WAM. The baselines grasp the eggplant reliably but rarely select the mushroom when neither cue object is present (10% and 0%, versus 60% for Next-WAM), indicating that they do not follow the conditional logic of the instruction. On Coke Restocking, Next-WAM averages 63% across the three count settings, whereas $\pi_{0.5}$ and Fast-WAM reach only 7% and 3%. This is because the baselines typically fail to stop once the target count is reached.

Text reasoning. Fig. 5 shows example Next-WAM rollouts with subtask reasoning generated online during closed-loop execution. Triggered by the ⟨Reason⟩ and ⟨Act⟩ control tokens introduced in §3.3, the model produces natural-language subtasks to support logical object selection and numerical inventory counting. For conditional object grasping, the model reasons over the presence or absence of visual cue objects to select the correct target. For coke restocking, it keeps track of the current can inventory and stops replenishment once the target count is satisfied. The generated subtask acts as persistent task state across replanning cycles and conditions subsequent action generation, allowing the policy to maintain task progress without external memory.

4.3 Ablation Study

In this section, we perform a set of ablation experiments on the LIBERO benchmark. Unless noted otherwise, all experiments adopt our default configuration with the 0.4B action expert. Additional ablation results are provided in Appendix B.

Action-expert capacity. We evaluate action experts with approximately 0.2B, 0.4B, and 1.0B parameters, as shown in Table 5a. These variants use hidden dimensions of 288, 512, and 1024, with

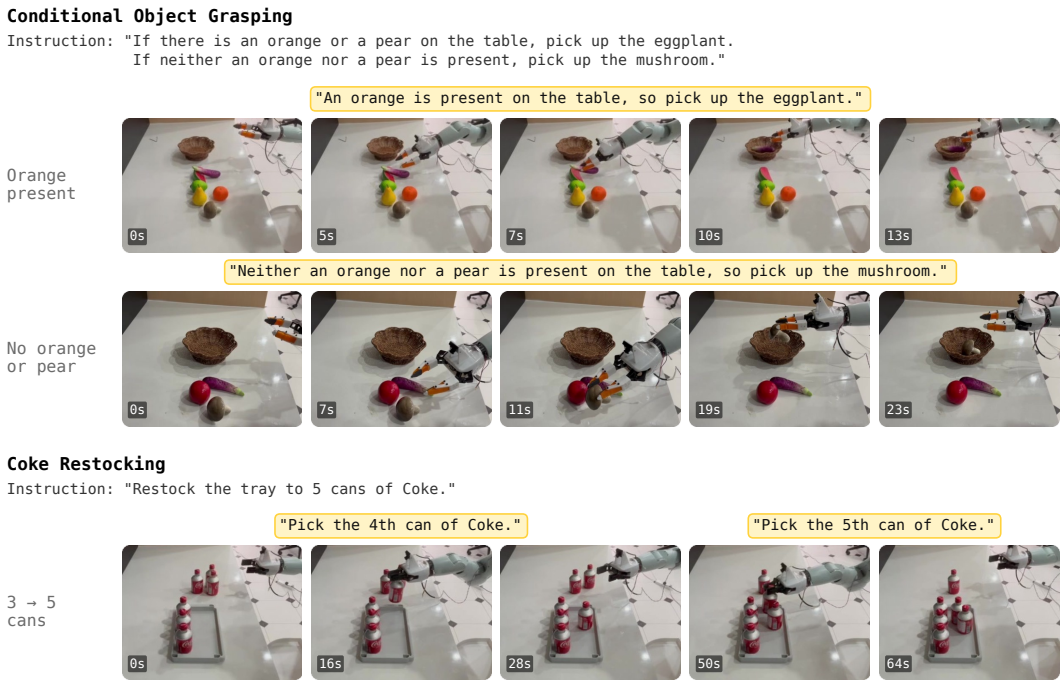


Figure 5: **Real-robot rollouts of Next-WAM.** Key frames from closed-loop policy executions. Timestamps are shown in the lower-left corner. Text in yellow boxes denotes online-generated subtask reasoning produced by Next-WAM. The model dynamically adapts its reasoning to scene states. For *Conditional Object Grasping*, it selects eggplant when an orange is present, and mushroom otherwise for conditional grasping. For *Coke Restocking*, the generated subtask tracks the number of cans on the tray to enforce the target inventory.

corresponding feed-forward dimensions of 1152, 2048, and 4096. All three maintain 36 Transformer layers and 16 attention heads with a head dimension of 128, preserving the same joint-attention interface with the understanding and generation experts. Their average success rates are 92.1%, 98.4%, and 97.2%, respectively. The 0.4B expert achieves the highest success rate with fewer parameters than the 1.0B variant, and we therefore adopt it for our main experiments.

Conditioning on generated futures. We compare direct action prediction (Next-WAM) with a variant that generates future-image latents and conditions action prediction on them (Next-WAM-IDM). Next-WAM-IDM achieves 97.3% average success, compared with 98.4% for Next-WAM (Table 5b). It performs slightly worse across all four LIBERO suites. Thus, conditioning on generated futures provides no benefit in this setting, and we use direct action prediction in our main experiments.

5 Conclusion

We introduced Next-WAM, an any-to-any World-Action Model that brings visual history, text reasoning, future imagination, and action prediction into a unified framework. Three specialized experts exchange information through structured joint attention, accommodating the distinct requirements of multimodal understanding, visual prediction, and continuous control. Future-image supervision contributes to representation learning, while action inference proceeds without generating future images. Without additional embodied pretraining, Next-WAM achieves 98.4% success on LIBERO and 70.5% on LIBERO-Plus, demonstrating competitive manipulation performance and robustness to environmental perturbations. Real-robot evaluations further validate its online text reasoning capabilities for logical object selection and numerical inventory tracking. Future work will explore richer compositional language reasoning, longer-horizon temporal contexts, and stronger generalization across diverse manipulation tasks and robot embodiments.

AI USE STATEMENT

We used generative AI tools to assist with code writing and to polish the language of the manuscript, including improving grammar, clarity, and readability. We did not use generative AI tools to generate research ideas, develop the research methodology, or interpret experimental results. We reviewed all AI-assisted code and manuscript edits to ensure consistency with our intended methodology and technical content. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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.

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

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

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

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.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, Ru Ying, Ran Zheng, and Yao Mu. AHA-WAM: Asynchronous horizon-adaptive world-action modeling with observation-guided context routing. *arXiv preprint arXiv:2606.09811*, 2026a.

Xiaowei Cai, Yunuo Cai, Bingao Chen, Jingxiao Chen, Zhi Chen, Siyuan Feng, Tengyu Hou, Jingshun Huang, Han Jiang, Runkun Ju, et al. τ_0-VLA: a hierarchical robot foundation model with world-model-guided test-time computation. *arXiv preprint arXiv:2608.16885*, 2026b.

Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Bohan Hou, Yuming Jiang, Jiayan Guo, Xin Li, et al. Rynnvla-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025a.

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*, 2025b.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *arXiv preprint arXiv:2505.23705*, 2025.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, Leslie Kaelbling, et al. Video language planning. In *International Conference on Learning Representations*, volume 2024, pp. 31138–31155, 2024.

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.

Fengyi Fu, Mengqi Huang, Shaojin Wu, Yunsheng Jiang, Yufei Huo, Hao Li, Yinghang Song, Fei Ding, Jianzhu Guo, Qian He, et al. Lance: Unified multimodal modeling by multi-task synergy. *arXiv preprint arXiv:2605.18678*, 2026.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

Yucheng Hu, Jianke Zhang, Yuanfei Luo, Yanjiang Guo, Xiaoyu Chen, Xinshu Sun, Kun Feng, Qingzhou Lu, Sheng Chen, Yangang Zhang, et al. Bagelvla: Enhancing long-horizon manipulation via interleaved vision-language-action generation. *arXiv preprint arXiv:2602.09849*, 2026.

Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *Advances in Neural Information Processing Systems*, 38:82782–82802, 2026.

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

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

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

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*, 2026a.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified Video Action Model. In *Proceedings of Robotics: Science and Systems*, 2025.

Ying Li, Xiaobao Wei, Jiajun Cao, Hao Wang, Xiaowei Chi, Chengyu Bai, Qianpu Sun, Jiajun Li, Xiaojie Zhang, Peidong Jia, Jian Tang, Sirui Han, and Shanghang Zhang. WAM4D: Fast 4D world action model via spatial register tokens. *arXiv preprint arXiv:2606.14048*, 2026b.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Si Liu, Jianlan Luo, Liliang Chen, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 88446–88463, 2026.

Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2505.11917*, 2025.

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.

Yicheng Liu, Zibin Dong, Baijun Ye, Tianyuan Yuan, Tao Jiang, Anqi Yang, Shicheng Cao, Haonan Liu, Yue Sun, Zihan Guo, et al. G0. 5: One autoregressive stream for robot reasoning and action. *arXiv preprint arXiv:2608.11739*, 2026a.

Zhuoyang Liu, Jiaming Liu, Hao Chen, Jiale Yu, Ziyu Guo, Chengkai Hou, Chenyang Gu, Xiangju Mi, Renrui Zhang, Kun Wu, et al. Last $_{\{0\}}$: Latent spatio-temporal chain-of-thought for robotic vision-language-action model. *arXiv preprint arXiv:2601.05248*, 2026b.

Liheng Ma, Rui Heng Yang, Zhanguang Zhang, Mateo Clemente, Ziwen Hu, Tongtong Cao, and Yingxue Zhang. Faster-WAM: Do world action models need deep action modules? *arXiv preprint arXiv:2608.02365*, 2026.

Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond vlas. *arXiv preprint arXiv:2512.15692*, 2025.

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.

Yue Su, Sijin Chen, Haixin Shi, Mingyu Liu, Zhengshen Zhang, Ningyuan Huang, Weiheng Zhong, Zhengbang Zhu, Yuxiao Liu, and Xihui Liu. World guidance: World modeling in condition space for action generation. *arXiv preprint arXiv:2602.22010*, 2026.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. In *European Conference on Computer Vision*, pp. 478–497. Springer, 2026.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Marcel Torne, Karl Pertsch, Homer Walke, Kyle Vedder, Suraj Nair, Brian Ichter, Allen Z. Ren, Haohuan Wang, Jiaming Tang, Kyle Stachowicz, Karan Dhabalia, Michael Equi, Quan Vuong, Jost Tobias Springenberg, Sergey Levine, Chelsea Finn, and Danny Driess. MEM: Multi-scale embodied memory for vision language action models, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Youpeng Wen, Junfan Lin, Yi Zhu, Jianhua Han, Hang Xu, Shen Zhao, and Xiaodan Liang. Vidman: Exploiting implicit dynamics from video diffusion model for effective robot manipulation. *Advances in Neural Information Processing Systems*, 37:41051–41075, 2024.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12966–12977, 2025.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026.

Angen Ye, Zeyu Zhang, Boyuan Wang, Xiaofeng Wang, Dapeng Zhang, and Zheng Zhu. Vla-r1: Enhancing reasoning in vision-language-action models. *arXiv preprint arXiv:2510.01623*, 2025.

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.

Hanyang Yu, Haitao Lin, Jingbo Zhang, Wenyao Zhang, Chenghao Gu, Heng Li, and Ping Tan. MaskWAM: Unifying mask prompting and prediction for world-action models. *arXiv preprint arXiv:2606.13515*, 2026.

Shanglin Yuan, Weiheng Zhao, Xin Shi, Haoyi Jiang, Xianda Guo, Liu Liu, Wenyu Liu, Wei Sui, and Xinggang Wang. DreamWAM: Beyond RGB future prediction for world action models. *arXiv preprint arXiv:2608.04996*, 2026a.

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*, 2026b.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3157–3181. PMLR, 2025.

Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent. *arXiv preprint arXiv:2501.18867*, 2025.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. ImageWAM: Do world action models really need video generation, or just image editing?, 2026.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1702–1713. IEEE, 2025.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified World Models: Coupling Video and Action Diffusion for Pretraining on Large Robotic Datasets. In *Proceedings of Robotics: Science and Systems*, 2025.

A Training Details

We initialize Next-WAM from Lance-3B (Fu et al., 2026) and add an action expert with approximately 382M parameters. The expert contains 36 Transformer blocks with hidden and feed-forward widths of 512 and 2048, respectively. The VAE remains frozen during training. We use eight GPUs with DeepSpeed ZeRO-1 and BF16 mixed precision. A per-GPU batch size of 8 and two gradient accumulation steps give an effective batch size of 128. We optimize with AdamW, linearly warm up the learning rate for 500 steps. Table 6 summarizes the default training configuration.

Table 6: Training hyperparameters.

Parameter	**Value**
GPUs	8 NVIDIA H100
Distributed strategy	DeepSpeed ZeRO-1
Mixed precision	BF16
Optimizer	AdamW
Optimizer betas	$(0.9, 0.95)$
Learning rate	10^{-4}
Minimum learning rate	10^{-6}
Warmup steps	500
Batch size per GPU	8
Gradient accumulation steps	2
Image / action loss weights	1.0 / 1.0
Action flow shift (training / inference)	5.0 / 5.0

LIBERO data. We jointly train on demonstrations from the Spatial, Object, Goal, and Long suites for 10 epochs, without additional embodied pretraining. Each camera view is resized to 224×224, and the external and wrist views are concatenated horizontally. The default configuration uses the current observation to predict 32 seven-dimensional actions and the image at offset $+32$. An eight-dimensional proprioceptive state provides additional conditioning. Actions and states use min–max normalization. We retain training windows with padded trajectory endpoints. The image and action experts are trained with flow matching, with their loss weights both set to 1.0.

Real-robot data. We combine an egocentric view with two wrist views. The egocentric image is resized to 256×320, while each wrist image is resized to 128×160. The wrist images are placed side by side below the egocentric image, producing a 384×320 input. We train for 50,000 steps using AdamW. Text prediction is supervised only for samples with output-text annotations. Table 7 summarizes the image inputs and training durations for the two settings.

Table 7: Dataset-specific input processing and training duration.

Parameter	**LIBERO**	**Real-robot tasks**
Camera views	External + wrist	Egocentric + two wrists
View layout	Horizontal	Egocentric above wrist pair
Combined image resolution	224×448	384×320
Training duration	10 epochs	50,000 steps

B Additional Ablation Studies

Table 8 reports additional LIBERO ablations using the 0.4B action expert. The default policy predicts $H = 32$ actions and executes $E = 10$ actions before replanning.

Action horizon and replanning. At $H = 32$, reducing the executed prefix from 10 to 8 or 5 actions lowers average success from 98.40% to 97.85% and 96.20%, respectively. The largest decline occurs on Long, from 97.00% to 92.00% with $E = 5$. At $H = 16$, executing 10 actions also outperforms executing 8 (98.20% versus 97.50%).

Visual prediction supervision. Reducing the image loss weight from 1.0 to 0.5 lowers average success by 2.00%, from 98.40% to 96.40%. This comparison favors the default image loss weight of 1.0 under the evaluated training recipe.

Flow shift. Setting the action flow shift to 1 during training and inference reduces average success to 96.95%, a decrease of 1.45 percentage points. The largest losses occur on Goal and Long, favoring the default shifted schedule in this comparison.

Table 8: **Additional ablations on LIBERO.** H denotes the predicted action horizon and E the number of executed actions before replanning. Unless specified, each row uses the default settings.

Configuration	Spatial	Object	Goal	Long	Average
Default ($H = 32$, $E = 10$)	98.60	99.80	98.20	97.00	**98.40**
$H = 32$, $E = 8$	98.80	99.80	98.00	94.80	97.85
$H = 32$, $E = 5$	98.40	96.80	97.60	92.00	96.20
$H = 16$, $E = 10$	98.60	99.80	98.00	96.40	98.20
$H = 16$, $E = 8$	98.40	99.40	98.20	94.00	97.50
Image loss weight 0.5	97.40	98.60	97.20	92.40	96.40
Flow shift 1	98.20	99.60	96.40	93.60	96.95

C Qualitative Results of Future Imagination

Fig. 6 presents predicted future observations on LIBERO, with one example from each suite. Each image combines the external and wrist-camera views. The predictions retain much of the scene layout while depicting changes in the robot pose and wrist-camera viewpoint. These examples illustrate visual anticipation of robot motion. The default policy generates actions directly from observed context without using these predicted images.

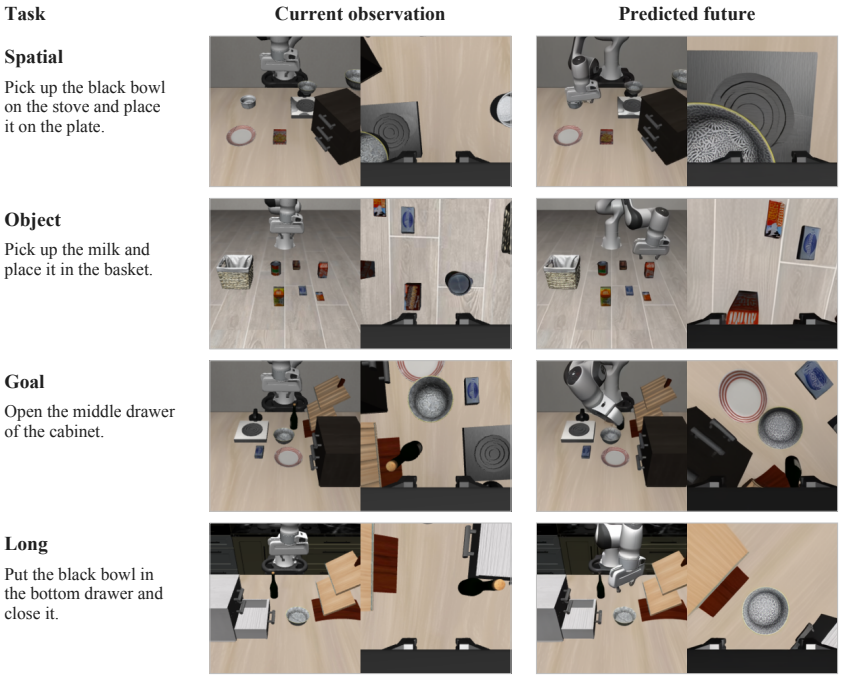


Figure 6: **Future imagination on LIBERO.** Current observations and predicted futures from four task suites. Each image combines external (left) and wrist (right) views.

D INFERENCE ACCELERATION

Experimental setup. All experiments use a single NVIDIA RTX 4090 with BF16 precision and batch size 1. The model receives two conditioning frames at 384×320 resolution and a 20-dimensional robot state, and predicts a 30×20 action sequence.

Condition caching and graph execution. The conditioning branch is independent of the evolving action tokens. We therefore cache its layer-wise keys and values during the first denoising step and reuse them throughout subsequent steps, avoiding repeated evaluation of the 36-layer Lance backbone. We separately capture the initial conditioning forward pass and the remaining action denoising loop with CUDA Graphs. The latter includes action encoding, Transformer computation, output projection, and scheduler updates. Input and KV buffers are refreshed before graph replay.

Attention optimization and compilation. We replace ViT attention with FlashAttention 2 while retaining SDPA for action attention. We further remove conditioning KV entries that are inaccessible to action queries. Since these queries share the same conditioning visibility and use full action self-attention, the compact representation eliminates the explicit attention mask. Finally, we compile the cached action layers using `torch.compile` with `mode="default"`, `fullgraph=True`, and `dynamic=False`, completing compilation warm-up before graph capture.

Results. Table 9 summarizes successive experiments under the same hardware and input specifications. The combined execution optimizations reduce 20-step inference latency from 1965.98 ms to 201.64 ms. Reducing the denoising schedule to 10 steps further lowers latency to 168.29 ms, yielding an overall $11.68\times$ speedup.

Table 9: Cumulative inference acceleration on a single NVIDIA RTX 4090. Latency is the median over 30 measurements. Speedup is relative to the original 20-step baseline.

Configuration	Inference time (ms)	Speedup
Original inference	1965.98	$1.00\times$
+ Condition KV caching	512.28	$3.84\times$
+ Prefill and action-loop CUDA Graphs	364.68	$5.39\times$
+ ViT FlashAttention 2	296.20	$6.64\times$
+ Compact KV	250.44	$7.85\times$
+ Cached action-layer compilation	201.64	$9.75\times$
+ Reduce denoising steps from 20 to 10	**168.29**	$\mathbf{11.68}\times$