Z-WAM: WORLD-ACTION MODEL FROM IMAGE-EDITING PRIORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video-based world–action models (WAMs) connect robot action learning with dense future-video prediction, incurring the computational cost of synthesizing intermediate frames. We argue that this expensive visual prediction is unnecessary for learning effective actions: a policy can model the intended scene change and the actions that realize it without reconstructing every intermediate visual state. Motivated by this observation, we present Z-WAM, a vision–language–action framework that transfers pretrained image-editing priors to robot control. Conditioned on current observations and a language instruction, Z-WAM jointly predicts a single future multi-view observation and the corresponding action chunk. This formulation uses image editing to represent the outcome of a manipulation sequence while retaining the full sequence of executable actions. Sequential bidirectional attention couples the editing and action streams, allowing action hypotheses to refine visual features that subsequently guide action prediction. Progressive pretraining first adapts the editing model to robot observation transitions without action labels, then connects visual changes to controls through joint world–action learning. During task adaptation, grasp-aware objectives supervise cumulative arm motion and gripper transitions to encourage coordinated execution. Preliminary experiments on six real-world manipulation tasks support image-editing priors as a foundation for robot action learning without dense future-video prediction.

1 INTRODUCTION

Vision–language–action (VLA) models translate visual observations and language instructions into executable robot actions. By transferring representations from large-scale pretraining, these models connect semantic understanding with manipulation(Kim et al., 2024; Octo Model Team et al., 2024; Chi et al., 2025; Black et al., 2024a; Intelligence et al., 2025). World–action models (WAMs) extend this approach by incorporating future visual prediction into policy learning, connecting actions with their expected visual effects on the environment. Video-based approaches draw on generative visual priors to predict future trajectories, either as intermediate representations for control or jointly with robot actions (Hu et al., 2024; Zhu et al., 2025; Bi et al., 2026; Li et al., 2026).

However, dense future-video prediction introduces an expensive computation cost. The model must synthesize intermediate visual states throughout a motion sequence. This requires representing the appearance and temporal evolution of multiple future frames alongside the actions needed for execution. We argue that reconstructing every intermediate visual state is unnecessary for learning effective manipulation policies. A model can instead predict the visual outcome of an action chunk while learning the complete sequence of commands that produces it. This formulation retains temporal detail in the action sequence while concentrating visual prediction on a single future endpoint. It therefore motivates a different choice of generative prior, *i.e.*, directly models instruction-conditioned scene transformations.

Pretrained image-editing models provide such a prior. Given an image and a language instruction, they represent how the scene should change, making them a natural starting point for modeling the visual consequences of manipulation. Prior work has established the relevance of editing models to robot control: SuSIE generates visual subgoals for downstream policies (Black et al., 2024b), while ImageWAM conditions an action expert on intermediate editing features (Zhang et al., 2026b). These approaches motivate learning from image transformations without requiring dense future-

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.

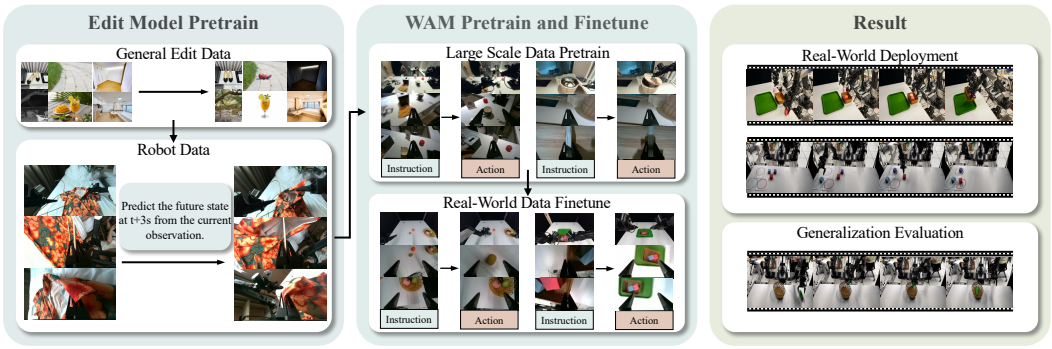


Figure 1: Overview of Z-WAM's learning process and real-world evaluation. An image editing prior is adapted to robot observations through future-state prediction, followed by joint world–action pretraining on large-scale robot data and fine-tuning on six real-world manipulation tasks. The example execution sequences illustrate real-world deployment and generalization evaluation.

video targets. However, an editing representation alone does not specify the executable motion needed to realize a transformation. A central challenge is therefore to connect the predicted scene change with the evolving action sequence, allowing both representations to inform one another.

As shown in Figure 1, we present **Z-WAM**, an image-editing-based world–action model that transfers generative editing priors to robot manipulation through joint visual–action modeling.Conditioned on current multi-view observations and a language instruction, Z-WAM jointly predicts a relative-action chunk and the multi-view observation at the end of that chunk. The visual target describes the outcome of the predicted commands, which may constitute only part of a longer task. An editing stream and an action stream interact through sequential bidirectional attention: the current action hypothesis first refines the editing features, and the updated visual representation then guides action refinement. Building on bidirectional visual–action interaction explored in video-based WAMs (Bi et al., 2026; Li et al., 2026), we apply this coupling to an image-editing backbone with a single future endpoint. The resulting policy jointly refines visual and action representations without generating the intervening video.

Transferring this prior to robot control also requires bridging the gap between general image transformations and robot demonstrations. We adopt progressive pretraining that first adapts the editing stream to pairs of current and future robot observations without action labels. Joint world–action pretraining then introduces action supervision, associating visual transitions with the commands that produce them. During task adaptation, we additionally address the coordination of arm motion and gripper transitions. Small errors in relative joint commands can accumulate before a grasp, leaving the arm short of its intended configuration when the gripper closes. We supervise cumulative joint displacement near reference grasp events using predicted closure weights, with an additional penalty for directional underreach. Transition-weighted gripper supervision complements this objective by emphasizing opening and closing events.

We report real-robot results across 6 manipulation tasks involving object placement, transport, and coordinated bimanual interaction. These results support the feasibility of learning manipulation policies from image-editing priors without dense future-video prediction. Our contributions are:

- **An image-editing-based world–action framework** that jointly predicts an action chunk and its endpoint observation, with sequential bidirectional interaction between editing and action representations.
- **A progressive pretraining strategy** that connects general editing priors to robot control through visual adaptation followed by joint world–action learning.
- **Grasp-aware action supervision** that combines PredU with transition-weighted gripper classification. PredU supervises cumulative joint displacement around reference closure events using prediction-dependent weights and a directional underreach penalty, while the gripper objective emphasizes opening and closing transitions.

2 Related Work

World Action Models for Robot Manipulation. Beyond VLA-based observation-to-action prediction, World Action Models (Wu et al., 2023; Du et al., 2023; Bi et al., 2026) incorporate future visual dynamics into robot policy learning, building upon advances in video generation models (Wan et al., 2025; HaCohen et al., 2026). Existing approaches generally follow two paradigms. The first adopts an *imagine-then-act* pipeline, where a video model generates or represents future visual trajectories, from which actions are recovered through inverse dynamics or an action decoder (Zhou et al., 2024; Hu et al., 2024). The second jointly models future video and robot actions within a unified generative process (Li et al., 2025; Cheang et al., 2024; Bi et al., 2026; Li et al., 2026; Zhang et al., 2026a), allowing action prediction to directly interact with future visual representations rather than relying on a separated video-to-action pipeline. Apart from video-based WAMs, ImageWAM (Zhang et al., 2026b) explores an image-editing-based formulation, using intermediate editing representations to condition action generation. Our Z-WAM follows this paradigm with a strong foundation model, enabling bidirectional interaction between visual and action representations without dense future-video prediction.

Image Editing Models. Text-guided image editing modifies a source image according to language instructions while preserving irrelevant content. Early diffusion-based methods enable flexible editing through stochastic generation and attention control (Meng et al., 2022; Hertz et al., 2023), while InstructPix2Pix (Brooks et al., 2023) learns instruction-conditioned transformations directly. Subsequent methods improve editing precision and generality through multimodal reasoning and large-scale generative pretraining (Fu et al., 2024; Sheynin et al., 2024; Yu et al., 2025). Recent foundation models, such as FLUX (Labs, 2025) and Z-Image (Cai et al., 2025), further scale this paradigm with powerful diffusion-transformer backbones. Image editing has also been explored for modeling state transitions in robot manipulation. SuSIE (Black et al., 2024b) generates visual subgoals for downstream control, while ImageWAM (Zhang et al., 2026b) repurposes an image-editing model as a world-action backbone to condition action generation. Building upon Z-Image (Cai et al., 2025), Z-WAM jointly predicts a future observation and the corresponding action chunk, enabling bidirectional visual–action interaction within a shared generative process.

Vision-Language-Action Policies. Vision-Language-Action policies have emerged as a prominent paradigm for general-purpose robot manipulation, directly mapping visual observations and language instructions to robot actions. From early efforts such as RT-2 (Zitkovich et al., 2023) to generalist policies including Octo (Octo Model Team et al., 2024) and OpenVLA (Kim et al., 2024), large-scale vision-language pretraining and diverse robot data have substantially improved transferable robot control. Recent advances increasingly focus on expressive action generation and scalable architectures. Diffusion and flow-matching policies, exemplified by RDT-1B (Liu et al., 2025b), π_0 (Black et al., 2024a), and $\pi_{0.5}$ (Intelligence et al., 2025), provide flexible continuous action modeling, while architectures such as CogAct (Li et al., 2024), DexVLA (Wen et al., 2025), and HybridVLA (Liu et al., 2025a) further integrate multimodal reasoning with dedicated action generation. Meanwhile, FAST (Pertsch et al., 2025), SmolVLA (Shukor et al., 2025), GR00T (Bjorck et al., 2025), and Gemini Robotics (Team et al., 2025) extend VLAs toward more efficient and general-purpose embodied control. Despite these advances, VLA policies largely retain a direct observation-to-action formulation, without explicitly modeling future visual dynamics.

3 Method

Z-WAM jointly predicts a relative-action chunk and its future endpoint observation by extending an instruction-conditioned image editor with an action stream. Its central design is to exchange information between the two streams during denoising: the current action hypothesis updates the editing representation, which then conditions action refinement. Progressive robot-domain pretraining establishes this observation–action correspondence before adaptation to the target tasks. During adaptation, PredU supervises accumulated arm motion near gripper closure, while transition-weighted gripper classification supervises opening and closing behavior. Figure 2 summarizes the architecture and objectives.

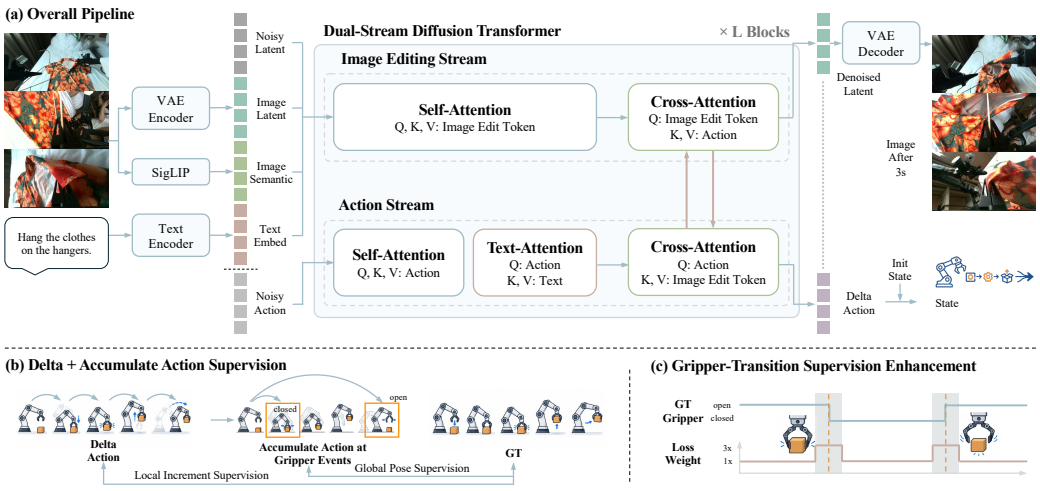


Figure 2: Z-WAM overview. (a) Editing and action streams jointly predict a future observation and a 48-step action chunk through sequential action-to-visual and visual-to-action attention. (b) PredU supervises cumulative joint displacement near closure and penalizes directional underreach. (c) Transition-weighted gripper classification emphasizes opening and closing.

3.1 Task Formulation and Action Representation

Given three camera views $o_t = \{o_t^{\text{head}}, o_t^{\text{left}}, o_t^{\text{right}}\}$ and a language instruction ℓ, we model

$$p_\theta(A_t^{\text{rel}}, o_{t+H} \mid o_t, \ell), \qquad A_t^{\text{rel}} = (a_t^{\text{rel}}, \ldots, a_{t+H-1}^{\text{rel}}). \tag{1}$$

The visual target contains three views at a single future time, after $H = 48$ control transitions, corresponding to 3.2 s at the recorded rate of 15 Hz. This endpoint describes the outcome associated with the action chunk, while the action sequence retains the intermediate commands needed for execution.

Arm channels encode joint increments, whereas gripper channels retain absolute opening commands. Given reference arm commands q^{cmd} and measured starting joints s_t^{arm}, we define and reconstruct arm commands as

$$\begin{aligned} \delta q_0 &= q_t^{\text{cmd}} - s_t^{\text{arm}}, \qquad \delta q_k = q_{t+k}^{\text{cmd}} - q_{t+k-1}^{\text{cmd}} \quad (1 \le k < H), \\ \hat{q}_{t+k}^{\text{cmd}} &= s_t^{\text{arm}} + \sum_{i=0}^{k} \widehat{\delta q}_i \qquad (0 \le k < H). \end{aligned} \tag{2}$$

The measured starting joints anchor command reconstruction but are not supplied as Transformer tokens. Reconstructed values are commanded targets, not measured outcomes after execution. Task adaptation supervises twelve arm and two gripper channels, masking inactive dimensions.

This representation makes accumulated motion an explicit function of the predicted increments. Small errors across several increments can therefore shift the commanded arm configuration at a grasp, even when individual increments are close to their targets. This motivates the cumulative, event-centered supervision introduced in Section 3.5.

3.2 Bidirectional Editing–Action Coupling

We extend the Z-Image editing backbone (Cai et al., 2025) with an action stream and reciprocal cross-attention. The editing sequence M contains visual latents and text/image conditioning representations; the action sequence U represents the noisy action chunk. Cross-stream interaction lets the action hypothesis influence the representation of the future observation, while that updated representation provides context for refining the action.

At each depth, the editing Transformer block first produces $M^\star$. Action self-attention and language cross-attention produce $U^\star$. We then apply action-to-visual attention followed by visual-to-action

attention:

$$\begin{aligned} \widetilde{M} &= M^\star + g_M \,\mathrm{CA}_M(M^\star, U^\star), \\ \widetilde{U} &= U^\star + g_U \,\mathrm{CA}_U(U^\star, \widetilde{M}), \\ U^+ &= \widetilde{U} + \mathrm{FFN}_U(\widetilde{U}), \qquad M^+ = \widetilde{M}. \end{aligned} \tag{3}$$

Here $\mathrm{CA}(X, Y)$ uses X as queries and Y as keys and values, and g_M, g_U are learned scalar residual gates.

The sequential order defines the information path $U^\star \to \widetilde{M} \to \widetilde{U}$. The action update reads $\widetilde{M}$, which already incorporates the current action hypothesis. Consequently, action features can affect both visual refinement and the subsequent action update within the same depth. The residual gates control the contribution of each exchange while retaining the stream-specific transformations.

Cross-attention covers all valid editing tokens, including conditioning tokens. Both streams participate in joint denoising, and their interaction occurs in feature space. The action stream does not require a decoded future RGB image; RGB decoding is needed only when exporting the visual prediction. The editing stream itself remains part of policy inference.

3.3 Progressive World–Action Pretraining

Progressive pretraining transfers a general editing model to joint robot observation and action prediction. We separate adaptation to robot visual changes from learning their correspondence with control sequences, then specialize the complete model to the target tasks.

Starting from a pretrained instruction-conditioned image editor, we first adapt the editing stream to current–future robot observation pairs conditioned on task descriptions. This stage uses visual supervision without action labels. It adapts the editor to manipulation-related observation changes before introducing control prediction, and the resulting parameters initialize the editing stream for joint pretraining.

We then introduce the action stream and reciprocal interaction modules. OpenGalaxea training windows pair current and future observations with cleaned task descriptions and the intervening controls. Joint flow matching supervises future visual latents and the available native action dimensions. The editing stream, action stream, and interaction modules are optimized together, so both prediction targets shape the representations exchanged through Eq. equation 3.

Finally, all three components are initialized from the joint checkpoint and fine-tuned on the six target tasks. We retain joint flow matching and add PredU and transition-weighted gripper supervision to constrain cumulative arm motion and gripper state around interaction events. Section 4.1 and Appendix B provide the data protocol, initialization sources, and training details.

3.4 Joint Flow Matching

We train visual latents and standardized actions with flow matching (Lipman et al., 2022; Liu et al., 2022). For modality $r \in \{o, a\}$, clean target x_0^r, Gaussian noise $\epsilon^r \sim \mathcal{N}(0, I)$, and independently sampled modality noise levels σ_r, define

$$x_{\sigma_r}^r = (1 - \sigma_r)x_0^r + \sigma_r \epsilon^r, \qquad v^{r*} = x_0^r - \epsilon^r, \tag{4}$$

$$\hat{x}_0^r = x_{\sigma_r}^r + \sigma_r v_\theta^r. \tag{5}$$

Under this sign convention, $\mathrm{d}x_{\sigma_r}^r / \mathrm{d}\sigma_r = -v^{r*}$, and sampling integrates the corresponding learned field as σ_r decreases from 1 to 0.

The base objective is

$$\mathcal{L}_{\mathrm{base}} = \mathbb{E}\left[\frac{w(\sigma_o)}{D_o}\|v_\theta^o - v^{o*}\|_2^2\right] + \mathbb{E}\left[\frac{\sum_{k,d} m_{k,d}(v_{\theta,k,d}^a - v_{k,d}^{a*})^2}{\sum_{k,d} m_{k,d}}\right]. \tag{6}$$

Here D_o counts visual target elements, w is the visual noise weighting, and m assigns unit weights to available native action dimensions during pretraining and the fourteen active channels during task adaptation, with zeros elsewhere. Each modality is normalized by its supervised elements.

The auxiliary objectives below use the clean action estimate $\hat{x}_0^a$ from Eq. equation 5, restored to physical command units. This connects denoising to cumulative joint motion and gripper state, whose meanings are defined in command space.

3.5 POSE-AWARE ACTION OPTIMIZATION

The base action objective supervises individual denoising targets, whereas grasp execution depends on the arm configuration reached when the gripper closes. We therefore supervise cumulative joint displacement near reference closures. Here *pose* denotes commanded joint configuration, not Cartesian end-effector pose.

For one sample and six-joint arm, suppressing their indices, define

$$\hat{d}_k = \sum_{i=0}^{k} \widehat{\delta q}_i, \qquad d_k^* = \sum_{i=0}^{k} \delta q_i^*. \tag{7}$$

Because prediction and reference use the same starting configuration, $\hat{d}_k - d_k^*$ is also the error between their reconstructed joint commands. Supervising this displacement couples all preceding increments through their accumulated effect at the selected event.

Prediction-weighted closure alignment. A fixed reference timestamp does not account for a local shift in predicted gripper closure. We therefore use reference closures to anchor a small temporal neighborhood and predicted gripper commands to weight the candidate configurations within it.

Higher gripper values denote opening. Reference closures satisfy $g_{\tau-1}^* \geq 0.5$ and $g_\tau^* < 0.5$ for $1 \leq \tau < H$; no event is synthesized at the first row. Within

$$\mathcal{C}_\tau = \{\tau - 1, \tau, \tau + 1\} \cap \{1, \ldots, H - 1\},$$

predicted opening commands define

$$\begin{aligned} s_k &= \mathrm{sigmoid}(10(\hat{g}_{k-1} - 0.6))\,\mathrm{sigmoid}(10(0.6 - \hat{g}_k))\,, \\ \alpha_{\tau,k} &= \mathrm{stopgrad}\left(\frac{s_k + 0.05}{\sum_{j \in \mathcal{C}_\tau}(s_j + 0.05)}\right). \end{aligned} \tag{8}$$

The score favors an open-to-closed change in the predicted commands. The additive constant retains weight on the local candidates when the predicted transition is weak. Detaching the normalized weights blocks the direct gradient path from the pose objective to the gripper predictions through this selector.

Every candidate is compared with the same reference closure target d_τ^*:

$$\ell_{\mathrm{Pred}}(\tau) = \sum_{k \in \mathcal{C}_\tau} \alpha_{\tau,k} \frac{\|\hat{d}_k - d_\tau^*\|_1}{6}. \tag{9}$$

Using d_τ^* rather than the candidate-specific target d_k^* preserves the arm configuration associated with the demonstrated closure while allowing local variation in predicted closure timing. The objective thus aligns accumulated arm motion with a prediction-weighted grasp neighborhood.

Directional underreach. The displacement loss penalizes deviations on either side of the reference configuration. We additionally emphasize insufficient motion along the reference displacement direction, since a sequence of underestimated increments can leave the commanded arm short along that direction.

Let $r_\tau = \|d_\tau^*\|_2$ and $u_\tau = d_\tau^* / \max(r_\tau, 0.05)$. We define

$$\ell_U(\tau) = \mathbf{1}[r_\tau \geq 0.05] \sum_{k \in \mathcal{C}_\tau} \alpha_{\tau,k} \left[(d_\tau^* - \hat{d}_k)^\top u_\tau\right]_+. \tag{10}$$

Here $[z]_+ = \max(z, 0)$. This one-sided term penalizes projected shortfall in joint-displacement space. The directional term is disabled below a reference displacement norm of 0.05 rad, where

the direction is less informative. Overshoot and other configuration errors remain penalized by Eq. equation 9.

Each loss is averaged over reference closures from both arms within each of B windows, then across the batch; event-free windows contribute zero. The resulting $\mathcal{L}_{\text{Pred}}$ and $\mathcal{L}_U$ form PredU. Appendix C gives the indexed definitions and reduction.

3.6 Transition-Weighted Gripper Supervision

PredU supervises the arm configuration near closing events, but its detached selector does not directly train the gripper predictions through the event weights. We therefore add explicit gripper state supervision alongside the base action loss. The auxiliary covers the full action chunk and emphasizes neighborhoods of both opening and closing transitions.

For sample n and gripper $b \in \{1, 2\}$, the reference open-state label is $y_{n,k,b} = \mathbf{1}[g^*_{n,k,b} \geq 0.5]$. Each reference state change between rows $j-1$ and j marks

$$\{j-2, j-1, j, j+1\} \cap \{0, \ldots, H-1\}.$$

The union of these regions receives weight $w^g_{n,k,b} = 3$, with one elsewhere, independently for each gripper. We construct classification logits from the same continuous gripper prediction, giving

$$\mathcal{L}_{\text{grip}} = \frac{1}{2B} \sum_{n,b} \frac{\sum_k w^g_{n,k,b} \, \text{BCEWithLogits}(10(\hat{g}_{n,k,b} - 0.5), y_{n,k,b})}{\sum_k w^g_{n,k,b}}. \tag{11}$$

This objective requires no separate classification head. Normalization within each window and gripper prevents the number of weighted rows from directly setting its total loss weight. Transition weights affect only this auxiliary, leaving the base velocity loss unchanged.

The two auxiliary objectives address complementary aspects of interaction: PredU constrains where the commanded arm is near closure, while gripper classification supervises whether the gripper is open or closed throughout the chunk. In windows without a reference closure, the base objective and gripper classification remain active even though PredU contributes zero.

The complete task-adaptation objective is

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \mathcal{L}_{\text{Pred}} + 0.1\mathcal{L}_U + 0.1\mathcal{L}_{\text{grip}}. \tag{12}$$

No separate gripper edge-difference or Cartesian pose loss is used.

3.7 Inference and Execution

At inference, future visual latents and the relative-action chunk are jointly denoised from noise. Arm increments are denormalized and reconstructed from the measured starting joints using Eq. equation 2; gripper commands remain absolute.

We execute a prefix of $K \leq H$ commands, acquire fresh camera observations and measured joints, and replan. Each replanning cycle therefore conditions on the robot's observed outcome and anchors command reconstruction to its current measured configuration. Generated RGB images are not reused as observations.

4 Experiments

We evaluate Z-WAM on six real-robot tasks and examine pose supervision, gripper supervision, and cumulative pretraining on two tasks. The tables report preliminary results from five trials per task and configuration. Interaction ablations and generalization experiments remain pending.

4.1 Experimental Setup

We use a Galaxea R1-Lite equipped with a main camera and two wrist cameras. The six downstream tasks are fruit arrangement, book return, three-object placement, slipper arrangement, bimanual tray

Table 1: **Real-world task performance.** Each task reports successful trials out of five and mean completion (CR, %). Average reports successes out of 30 trials and mean CR across all six tasks.

Method	Arrange fruits		Return the book		Place three objects		Arrange slippers		Load the tray		Hand over the object		Average	
ImageWAM	0/5	10.0	**1/5**	40.0	3/5	93.2	**3/5**	85.0	0/5	40.0	0/5	0.0	7/30	44.7
Fast-WAM	0/5	6.0	**1/5**	35.0	4/5	96.6	0/5	50.0	0/5	40.0	0/5	0.0	5/30	37.9
Motus	0/5	13.0	0/5	15.0	0/5	43.3	0/5	20.0	0/5	37.5	0/5	5.0	0/30	22.3
LingBot-VA	0/5	39.0	**1/5**	40.0	1/5	79.8	0/5	25.0	1/5	50.0	3/5	65.0	6/30	49.8
$\pi_{0.5}$	0/5	19.0	0/5	40.0	3/5	89.9	0/5	25.0	0/5	65.0	2/5	65.0	5/30	50.7
Z-WAM	**1/5**	**60.0**	**1/5**	**55.0**	**5/5**	**100.0**	**3/5**	**90.0**	**3/5**	**90.0**	**5/5**	**100.0**	**18/30**	**82.5**

loading, and object handover. We measure task success and partial completion; task illustrations and scoring rules are provided in Appendix A.1.

Z-WAM variants share the downstream data, normalization, and evaluation inputs. The complete model uses PredU and transition-weighted gripper BCE. Z-WAM uses 10 joint denoising steps at inference. The denoising-step ablation is reported in Appendix F.3. Dataset preparation, training exposure, and ablation controls are documented in Appendices A, B, and F.

We compare against ImageWAM (Zhang et al., 2026b), Fast-WAM (Yuan et al., 2026), Motus (Bi et al., 2026), LingBot-VA (Li et al., 2026), and $\pi_{0.5}$ (Intelligence et al., 2025). ImageWAM provides the closest editing-based reference, while the remaining methods provide video-based and direct-policy comparisons. Each method retains its documented input representation and sampler. These comparisons assess complete policies without assuming equal total compute; initialization, training exposure, and inference settings are documented separately in Appendix D. The planned generalization evaluation and qualitative visual comparisons are provided in Appendices G and H, respectively.

4.2 Real-Robot Evaluation

We evaluate five trials per method and task. Completion rate (CR) is the mean task-specific partial-credit score. For these preliminary records, a trial is counted as successful when its recorded completion reaches 100%; reconciliation with the full terminal-success criteria is discussed in Appendix E. Failures remain in the denominator. The overall CR weights the six tasks equally, and the aggregate success fraction equals the six-task mean success rate because each task has the same number of trials. Bold and underlined values in the tables indicate the best and second-best distinct results, respectively; ties share formatting.

Table 1 reports 18 successful trials out of 30 for Z-WAM, corresponding to a 60.0% success rate and 82.5% mean CR. The complete model achieves the highest observed CR on all six tasks, exceeding the next-highest six-task mean of 50.7% by 31.8 percentage points. With PredU and transition-weighted gripper BCE, it completes all five trials on both three-object placement and object handover. Fruit arrangement and book return remain challenging: their 60.0% and 55.0% CRs correspond to only one successful trial each. This gap indicates that substantial partial progress does not consistently translate into complete task execution.

4.3 Component Ablations

We evaluate component variants on three-object placement and object handover, retaining all six tasks during downstream fine-tuning. Each configuration is evaluated over five trials per task. The complete-configuration results are shared across the main comparison and both ablation tables; they represent the same physical trials.

Pose and gripper supervision. Table 2 compares three configurations: without PredU or auxiliary gripper BCE, with PredU alone, and with both PredU and transition-weighted gripper BCE. All

Table 2: **Pose and gripper supervision.** Each task reports successful trials out of five and mean completion (CR, %). Average reports successes out of ten and the two-task mean CR.

Pose (PredU)	Gripper Supervision	Place three objects		Hand over the object		Average	
×	×	**5/5**	**100.0**	1/5	50.0	6/10	75.0
✓	×	4/5	93.3	4/5	90.0	8/10	91.7
✓	✓	**5/5**	**100.0**	**5/5**	**100.0**	**10/10**	**100.0**

Table 3: **Cumulative training steps.** Each task reports successful trials out of five and mean completion (CR, %). Average reports successes out of ten and the two-task mean CR.

General Editing Initialization	Robot-Domain Visual Adaptation	Joint World–Action Pretraining	Task-Specific Fine-Tuning	Place three objects		Hand over the object		Average	
✓	×	×	✓	0/5	16.6	0/5	0.0	0/10	8.3
✓	✓	×	✓	0/5	13.3	0/5	10.0	0/10	11.7
✓	✓	✓	✓	**5/5**	**100.0**	**5/5**	**100.0**	**10/10**	**100.0**

three use a downstream training budget equivalent to 20,000 optimizer updates at batch size 128. Removing auxiliary gripper BCE retains gripper prediction and execution.

Without auxiliary gripper BCE, PredU improves handover from 1/5 successes and 50.0% CR to 4/5 and 90.0%, while three-object placement CR decreases from 100.0% to 93.3%. This task-dependent effect is consistent with handover's sensitivity to grasp alignment: small accumulated motion errors can misalign the gripper with a slender brush handle. PredU constrains cumulative joint displacement near closure, although task scores do not directly measure alignment error. Adding transition-weighted gripper BCE yields 5/5 successes and 100.0% CR on both tasks, raising mean CR from 91.7% to 100.0%. These results support combining closure-related arm supervision with gripper-transition supervision on these tasks, but do not isolate transition weighting from uniformly weighted BCE.

Cumulative pretraining. Table 3 compares three initialization histories, each followed by six-task fine-tuning: general editing initialization, additional robot-domain visual adaptation, and additional joint world–action pretraining. All three downstream configurations use PredU and transition-weighted gripper BCE. The two-task mean CR is 8.3% with general editing initialization, 11.7% after adding robot-domain visual adaptation, and 100.0% after also adding joint world–action pretraining. The complete configuration succeeds in all ten trials across the two tasks. These results compare cumulative training stages, including their additional data and computation; they do not isolate a pretraining objective at equal total training cost.

Interaction controls and failure analysis. The planned interaction ablation removes action-to-visual feedback during downstream adaptation from a common bidirectionally pretrained checkpoint. Both branches use PredU without auxiliary gripper BCE; results remain pending. Failure analysis will use the same physical trials as the task scores, including timeouts and interventions. Detailed controls and optional command and visual diagnostics are specified in Appendices F and H.

5 Conclusion and Limitations

Z-WAM combines staged adaptation, coupled image–action generation, and interaction-conditioned supervision for bimanual control. Preliminary evidence includes mean completion scores from available records, a three-task pretraining/pose ablation, and qualitative visual comparisons. Repeated comparisons and remaining ablations are pending; overall superiority and the causal benefit of action-to-visual coupling remain unestablished.

REPRODUCIBILITY STATEMENT

The method, stage definitions, and evaluation interfaces are specified in Sections 3–4. Appendix A documents the self-collected dataset and quality control; Appendix D records available-checkpoint training provenance and inference settings, with unresolved deployment choices marked; Appendix E documents selection and interpretation of the preliminary records and the planned repeated-trial protocol. Appendix H specifies visual-output provenance, denoising schedules, spatial annotations, and timing measurements for the planned analysis.

ETHICS STATEMENT

Real-robot evaluation involves physical manipulation and potential contact with objects and the surrounding workspace. Object handover in this task suite occurs between the robot's two grippers.

AI USE STATEMENT

Generative AI was used to assist manuscript organization and language polishing.

REFERENCES

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*, 2024a.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 33431–33452, 2024b.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2023.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

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, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

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.

Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. In *International Conference on Learning Representations*, 2024.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *International Conference on Learning Representations*, 2023.

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.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, 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.

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.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025a.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025b.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

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

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8871–8879, 2024.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

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.

Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 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, 2023.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26125–26135, 2025.

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.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems*, 38: 24195–24228, 2026a.

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? *arXiv preprint arXiv:2606.19531*, 2026b.

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. *arXiv preprint arXiv:2504.02792*, 2025.

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

A Dataset and Task Protocol

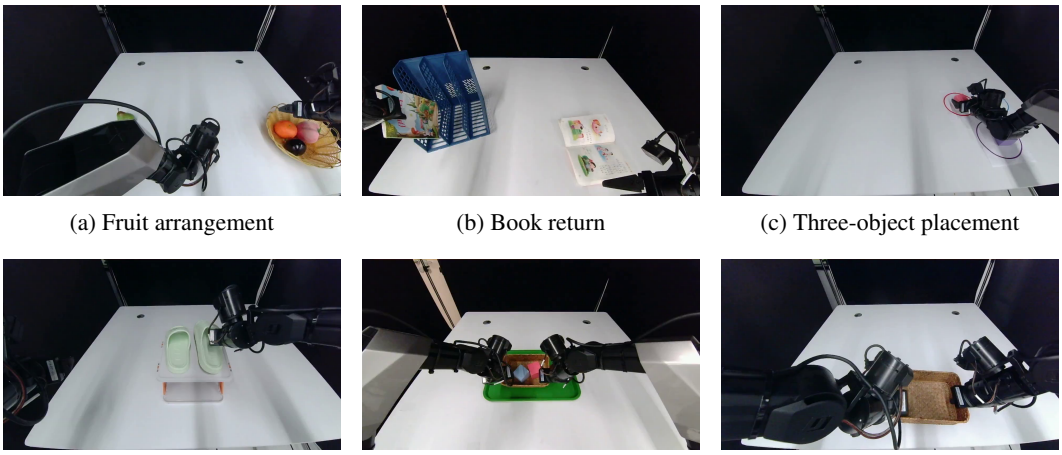
The downstream dataset contains demonstrations collected on a Galaxea R1-Lite with one main camera and two wrist cameras. The task suite comprises fruit arrangement, book return, three-object placement, slipper arrangement, bimanual tray loading, and object handover. Drawer loading and brush placement are excluded from downstream training and evaluation. A brush is nevertheless used as the transferred object in the handover task. Upstream robot-visual and joint-pretraining data are recorded separately from this downstream collection.

A.1 Task procedures and completion scores

Figure 3 illustrates the six task settings. Fruit arrangement collects five target fruits in a basket. Book return places one book in a rack, and three-object placement assigns each object to its designated circle. Slipper arrangement places both slippers on a shoe rack. Bimanual tray loading first places two blocks in a basket, then moves the basket into a tray using both arms. Object handover picks up a brush with the left arm, transfers it to the right arm, and puts it down.

Table 4 specifies partial-completion credits. For the same object, pickup-only and completed-placement credits are alternatives and are not added together. Book return and handover use the explicitly additive stage credits shown in the table. The fruit task uses five targets with 20% placement and 10% pickup-only credit per fruit.

The three-object task sheet records 33.3% and 16.6% credits. These printed values are retained here; they are not used to reconstruct or overwrite the supplied trial scores. Fully completed trials in the reported records are labeled 100%. The precise normalization and rounding of intermediate scores, and the scoring-version linkage of earlier records, remain to be reconciled.



(a) Fruit arrangement (b) Book return (c) Three-object placement

(d) Slipper arrangement (e) Bimanual tray loading (f) Object handover

Figure 3: **Six downstream manipulation tasks.** Frames illustrate the task settings from the supplied demonstrations; they are not policy-evaluation outcomes. Object handover transfers a brush between the robot's grippers.

A.2 Episode screening and split construction

Demonstrations are screened before episode-level split assignment and extraction of overlapping training windows. The screening protocol covers required streams, minimum sequence length, camera and arm record counts, nonfinite states, backward timestamps, gripper variation, and start/end recording poses. A rejection excludes an episode; a warning alone does not. These checks assess recording consistency and do not establish task success, visual quality, or precise synchronization between cameras.

Overlapping windows from one episode remain in the same split. The relevant Edit2Act variants share downstream manifests, normalization, and evaluation inputs. Repeatedly sampling an episode

Table 4: **Task-specific completion rules.** Credits are percentages of full task completion. Pickup-only and completed-placement credits for the same object are alternatives.

Task	Completion credit
Fruit arrangement	Five fruits: 20% per fruit picked up and placed in the basket; 10% if picked up but not placed.
Book return	One book: 50% for pickup and a further 50% for placement in the rack.
Three-object placement	33.3% per object placed in its assigned circle; 16.6% if picked up but not placed. Placement requires more than two thirds of the object to lie inside the circle.
Slipper arrangement	50% per slipper picked up and placed on the shoe rack; 25% if picked up but not placed.
Bimanual tray loading	25% per block placed in the basket, for two blocks. Basket transport adds 50% for bimanual pickup and placement in the tray, 25% for one-handed pickup and placement, or 12.5% for one-handed pickup without placement.
Object handover	25% for pickup, a further 50% for left-to-right transfer, and a further 25% for putting the object down.

does not increase the number of independent demonstrations. Final six-task episode counts, split manifests, active screening thresholds, and rejection/warning counts remain to be documented. Earlier eight-task collection totals are not used as the current six-task dataset statistics.

B Training and Implementation Details

The complete training sequence starts from general editing initialization, applies robot-domain visual adaptation without action labels, introduces joint world–action pretraining, and finishes with downstream adaptation. The visual adaptation stage trains current–future observation prediction. Joint pretraining optimizes the editing stream, action stream, and reciprocal interaction modules on paired observations and controls. Downstream adaptation initializes these components from the joint checkpoint and adds PredU and transition-weighted gripper BCE.

Table 5 summarizes the available training records. The identified joint-pretraining lineage uses 20,636 OpenGalaxea episodes and 60,000 completed optimizer updates. The three reported pose/gripper variants use a downstream budget equivalent to 20,000 updates at effective batch size 128. Effective batch size denotes episode windows per optimizer update, including gradient accumulation. Checkpoint-name suffixes are not used to infer a different comparable training budget.

Table 5: **Recorded training settings.** Joint-pretraining settings describe the identified lineage. The downstream budget is the confirmed equivalent budget for the three pose/gripper variants.

Item	Setting
Joint-pretraining data	20,636 OpenGalaxea episodes
Completed joint updates	60,000
Joint effective batch size	128 episode windows per update
Joint optimizer	AdamW
Joint learning rate	2×10^{-5}
Joint warm-up	100 updates
Joint computation precision	BF16
Pose/gripper adaptation budget	Equivalent to 20,000 updates at batch size 128
Downstream action channels	Twelve arm and two gripper channels
Predicted action horizon	48 transitions at a recorded rate of 15 Hz
Complete-model auxiliaries	PredU and transition-weighted gripper BCE
Final downstream manifest	Pending run-specific registration

The recorded architecture uses an editing backbone with hidden dimension 3840, 30 Transformer layers, and 30 attention heads, and an action stream with hidden dimension 512 and eight heads. Attention preserves the valid-token masks for text, images, and actions. Downstream action supervision selects twelve arm and two gripper channels. The action horizon is 48 control transitions at the recorded rate of 15 Hz; each window spans 49 observation timestamps, including its start and endpoint.

The exact stage input/output checkpoints, final downstream manifests, normalization statistics, augmentations, optimizer schedule, trainable-module settings, and checkpoint-selection records remain to be associated with each evaluated run. The execution prefix K and deployment postprocessing must also be recorded for the physical trials.

C Additional Method Definitions

This section gives the event indexing and loss reduction used by PredU. Let $n \in \{1, \dots, B\}$ index training windows, $b \in \{1, 2\}$ index arms and their associated grippers, and $k \in \{0, \dots, H-1\}$ index commands. Predicted and reference cumulative displacements are

$$\hat{d}_{n,k,b} = \sum_{i=0}^{k} \widehat{\delta q}_{n,i,b}, \qquad d^*_{n,k,b} = \sum_{i=0}^{k} \delta q^*_{n,i,b}. \tag{13}$$

Each displacement has six joint coordinates and is expressed in physical command units. The shared starting configuration cancels when computing their difference.

Reference closing events are

$$\mathcal{E}_n = \{(\tau, b) : 1 \le \tau < H,\ g^*_{n,\tau-1,b} \ge 0.5,\ g^*_{n,\tau,b} < 0.5\}. \tag{14}$$

No event is synthesized at the first row. For each event, use $\mathcal{C}_\tau = \{\tau-1, \tau, \tau+1\} \cap \{1, \dots, H-1\}$. The prediction-dependent weights are

$$\begin{aligned} s_{n,k,b} &= \mathrm{sigmoid}(10(\hat{g}_{n,k-1,b} - 0.6))\,\mathrm{sigmoid}(10(0.6 - \hat{g}_{n,k,b}))\,, \\ \alpha_{n,\tau,k,b} &= \mathrm{stopgrad}\left(\frac{s_{n,k,b} + 0.05}{\sum_{j \in \mathcal{C}_\tau}(s_{n,j,b} + 0.05)}\right). \end{aligned} \tag{15}$$

All candidates use the reference target at τ:

$$\ell_{\mathrm{Pred}}(n, \tau, b) = \sum_{k \in \mathcal{C}_\tau} \alpha_{n,\tau,k,b} \frac{\|\hat{d}_{n,k,b} - d^*_{n,\tau,b}\|_1}{6}. \tag{16}$$

Let $r_{n,\tau,b} = \|d^*_{n,\tau,b}\|_2$ and $u_{n,\tau,b} = d^*_{n,\tau,b} / \max(r_{n,\tau,b}, 0.05)$. The directional term is

$$\ell_U(n, \tau, b) = \mathbf{1}[r_{n,\tau,b} \ge 0.05] \sum_{k \in \mathcal{C}_\tau} \alpha_{n,\tau,k,b} \left[(d^*_{n,\tau,b} - \hat{d}_{n,k,b})^\top u_{n,\tau,b}\right]_+. \tag{17}$$

Here $[z]_+ = \max(z, 0)$, and the displacement threshold is 0.05 rad. For $j \in \{\mathrm{Pred}, U\}$, reduction is

$$\mathcal{L}_j = \frac{1}{B} \sum_{n=1}^{B} \frac{\sum_{(\tau,b) \in \mathcal{E}_n} \ell_j(n, \tau, b)}{\max(|\mathcal{E}_n|, 1)}. \tag{18}$$

Event-free windows contribute zero. Thus, averaging first occurs over both arms' events within each window and then over windows. Stopping the selector gradient blocks its direct path to gripper predictions; it does not remove shared-parameter interactions between arm and gripper prediction.

For the gripper auxiliary, each reference state change between $j-1$ and j marks $\{j-2, j-1, j, j+1\} \cap \{0, \dots, H-1\}$. Overlapping regions are merged independently for each gripper. Weights are three inside this union and one elsewhere, and the weighted BCE is normalized within each window and gripper. The logits are $10(\hat{g} - 0.5)$ and the open-state labels are $\mathbf{1}[g^* \ge 0.5]$. These weights affect only the auxiliary BCE. The coefficients of $\mathcal{L}_{\mathrm{Pred}}$, $\mathcal{L}_U$, and $\mathcal{L}_{\mathrm{grip}}$ are 1, 0.1, and 0.1, respectively.

D Baseline Configurations and Comparison Scope

The comparison includes ImageWAM, Fast-WAM, Motus, LingBot-VA, and $\pi_{0.5}$. Each method retains its documented observation representation and sampling procedure. Differences in backbone, pretraining, action interface, and inference computation are part of the complete-policy comparison; equal total training or inference cost is not assumed.

For each evaluated policy, the reproducibility record should identify its initialization and resume chain, downstream manifest, completed updates, effective batch size, normalization, and deployed checkpoint. Deployment records should specify camera layout and resolution, predicted and executed horizons, sampler, denoising steps, precision, command rate, and gripper postprocessing. Earlier checkpoint snapshots and training allocations are not substituted for the current six-task deployment records.

Latency should measure the complete observation-to-action path, including visual computation required to produce commands. RGB decoding used only for diagnostic export should be timed separately. Final run-specific baseline records and measured inference costs remain to be completed.

E Real-Robot Evaluation and Trial Records

The main comparison reports five designated trials per method and task. For task t with n_t trials, let $p_{ti} \in [0, 1]$ denote recorded completion and $s_{ti} \in \{0, 1\}$ denote a binary success label. The metrics are

$$\mathrm{SR}_t = \frac{100}{n_t} \sum_{i=1}^{n_t} s_{ti}, \qquad \mathrm{CR}_t = \frac{100}{n_t} \sum_{i=1}^{n_t} p_{ti}. \tag{19}$$

For the preliminary tables, success counts follow the supplied records with $s_{ti} = \mathbf{1}[p_{ti} = 1]$. The full terminal-success protocol additionally requires completion within the task deadline without assistance or a safety violation. Reconciliation of those conditions with the recorded binary outcomes and scoring versions remains pending.

The six-task mean gives each task equal weight. With five trials per task, the aggregate success fraction also equals the macro success rate. The current complete-model records contain 18 successful trials out of 30 and a mean CR of 82.5%. The three-object placement and handover records are reused in the component and cumulative-pretraining tables; they are not additional independent trials.

The controlled evaluation protocol uses matched initial configurations, randomized method order, and frozen checkpoints and execution settings. Timeouts, model failures, manual takeovers, and safety stops remain in the denominator. Completion is scored before assistance. Exclusion requires a documented external fault and a retained trial record. Task deadlines, placement tolerances, stability criteria, and the actual ordering and reset records remain to be finalized or linked to the reported trials.

Future uncertainty estimates will use reconciled individual trial records, including failures. The current five-trial comparisons and repeated presentation of the same complete-model trials do not establish statistical significance.

F Ablation Protocols and Supporting Diagnostics

Component comparisons evaluate three-object placement and object handover while retaining all six tasks during downstream fine-tuning. The reported pose/gripper configurations are $(\mathrm{PredU}, \mathrm{BCE}) = (\times, \times)$, $(\checkmark, \times)$, and $(\checkmark, \checkmark)$. All three use an equivalent downstream budget of 20,000 updates at effective batch size 128. Removing auxiliary BCE retains gripper prediction and execution through the base action objective.

Table 6 records the supplied individual completion scores. The PredU-only configuration is available as aggregate results: 4/5 successes and 93.3% CR on placement, and 4/5 successes and 90.0% CR on handover. Its individual trial scores are not reconstructed from those means. Exact checkpoint identifiers and trial-to-checkpoint mappings remain to be recorded.

Table 6: **Available individual completion scores.** Entries list five trial scores in percent. A dash indicates unavailable individual records; the PredU-only aggregates are reported in Table 2.

Pose (PredU)	Gripper Supervision	Place three objects	Hand over the object
×	×	100, 100, 100, 100, 100	25, 75, 100, 25, 25
✓	×	—	—
✓	✓	100, 100, 100, 100, 100	100, 100, 100, 100, 100

The cumulative-pretraining comparison contains three initialization histories, each followed by six-task fine-tuning with PredU and transition-weighted BCE. Robot-domain visual adaptation and joint world–action pretraining are added successively to general editing initialization. Each row therefore has a trained action stream. The comparison includes the additional data and exposure introduced by these stages and does not hold total training cost fixed.

F.1 Pending component controls

The planned interaction comparison starts both branches from a common bidirectionally pretrained checkpoint. One branch removes action-to-visual feedback during downstream adaptation while retaining visual-to-action conditioning and the visual objective. Both branches use PredU without auxiliary BCE. This comparison tests retaining feedback after shared pretraining, rather than changing the directionality of pretraining itself.

Additional pose controls can compare supervision at the reference closure, prediction-weighted alignment, and PredU under otherwise matched settings. A uniform-neighborhood control is needed to separate predicted weighting from the use of a local neighborhood. A uniformly weighted BCE control with the same auxiliary coefficient is needed to isolate transition weighting. These finer comparisons remain pending.

F.2 Failure analysis and command diagnostics

Failure analysis will annotate the same physical attempts used for the task scores, including timeouts and interventions. Categories include missed grasps, premature release, incorrect object selection, and unstable transport. Command diagnostics will measure cumulative joint error, directional under-reach, overshoot, and gripper-transition timing using common reference events and reporting event coverage. These measurements support specific mechanism analyses and do not replace physical success and completion scores.

F.3 Effect of Denoising Steps

We compare 1, 5, 10, and 25 denoising steps on three-object placement using the same Z-WAM checkpoint with PredU and transition-weighted gripper BCE. Each setting reports five real-robot trials. The default setting uses 10 steps and reuses the complete-model three-object placement trials reported in Table 1. Table 7 summarizes successful trials and mean completion, computed from the recorded integer percentages. The one-step trial interrupted by a disconnection is retained with its recorded completion of 66%.

The default 10-step setting achieves 5/5 successes and 100.0% mean completion, the highest observed values among the evaluated budgets. One and five steps achieve 89.8% and 86.4% mean completion, respectively, but yield fewer successful trials. Increasing the budget to 25 steps does not improve the observed performance: it yields 2/5 successes and 89.8% mean completion. These results support retaining the default 10-step setting for this task. With only five trials per setting, they do not establish a generally optimal denoising budget.

Table 7: **Effect of denoising steps.** Each setting reports successful trials out of five followed by mean completion (CR, %) on three-object placement. A trial succeeds at 100% completion. Bold values mark the best results.

Denoising steps	Place three objects	
1	3/5	89.8
5	2/5	86.4
10 (default)	**5/5**	**100.0**
25	2/5	89.8

G Generalization within Learned Tasks

We plan to evaluate frozen policies under held-out object instances, scene layouts, and visual backgrounds. Each condition changes one factor while preserving task instructions and terminal goals. Object substitutions remain within trained semantic categories and feasible size, mass, and graspability ranges. Layouts vary object positions within the robot workspace. Background changes affect non-contact visual context while keeping cameras, illumination, and manipulation surfaces fixed.

All methods will retain the deployment checkpoints, normalization, samplers, and execution settings fixed for the main comparison. No additional training or test-time parameter adaptation will use the held-out conditions. Candidate instances and configurations will be checked against downstream demonstrations and identifiable upstream robot data. Unknown overlap in inherited foundation-model data limits claims of absolute novelty.

Each shifted condition will be paired with a nominal configuration that keeps the remaining factors unchanged. Trial counts, conditions, and scoring rules will be fixed before inspecting outcomes. We will report absolute SR/CR and the paired change from nominal performance, with equal weighting across the six tasks and per-task records retained. These experiments remain pending; no generalization gain is claimed from the nominal five-trial results.

H Qualitative Visual Analysis and Inference Diagnostics

H.1 Visualizations across training stages

Figure 4 illustrates visual predictions for fruit placement across the stages in Section 3.3. Stage I uses the general instruction-conditioned image editor. Stage II applies robot-domain visual adaptation without action supervision. Stage III introduces joint world–action pretraining, and Stage IV applies downstream task adaptation. The input and ground-truth future observation provide references for inspecting the predicted changes.

The supplied panel is retained without altering its images or embedded stage labels. Its exact checkpoint identifiers, episode and target timestamps, sampling settings, and training-split membership remain to be linked. In particular, the Stage-IV image is not independently verified as an output of the checkpoint used in the main performance table. The panel therefore provides qualitative context rather than a controlled comparison of the final configurations.

H.2 Cross-method visual comparisons

Figure 5 compares predicted future observations across methods on slipper arrangement and book return. Each panel combines the three camera views. The examples share inputs and target times, without ground-truth future actions.

The original panel images and labels are retained. Their exact checkpoint and sampling records require the same provenance checks as the stage panel. Differences in camera composition, display size, and selected regions should not be interpreted as quantitative image-quality rankings.

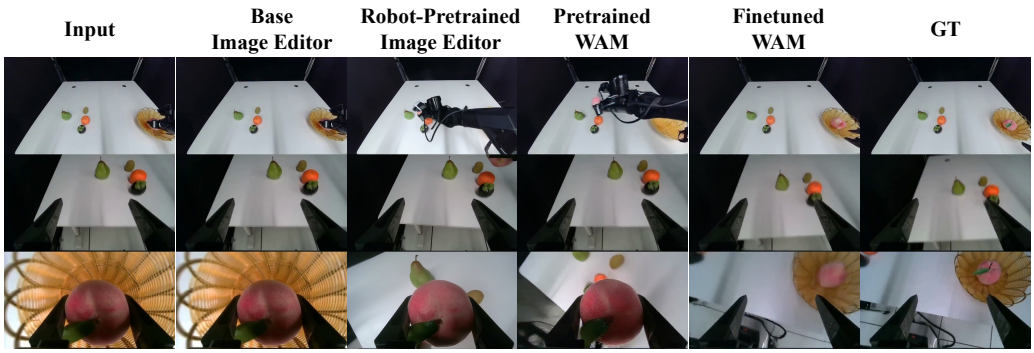


Figure 4: **Visual predictions across training stages.** Columns show the input, base image editor (I), robot-pretrained image editor (II), pretrained WAM (III), fine-tuned WAM (IV), and ground-truth reference (GT). Each column stacks three camera views. Stage definitions follow Section 3.3.

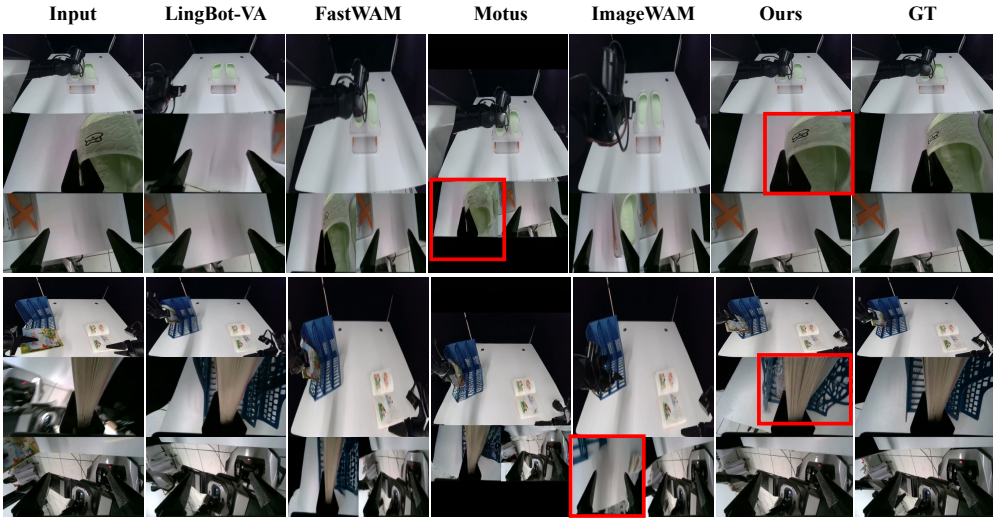


Figure 5: Qualitative future-observation comparisons on *Arrange slippers* (top) and *Put the book back* (bottom). Each panel combines three camera views. Methods share inputs and target times, without ground-truth future actions. Red boxes highlight local details; Ours denotes Z-WAM and GT denotes ground truth.