CAUSALWAM: LEARNING ACTION-CONDITIONED VISUAL DYNAMICS FOR EFFICIENT WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) jointly model robot actions and future visual observations, leveraging future visual dynamics as dense supervisory signals to learn physically grounded action and alleviate the sparse action-supervision bottleneck in Vision-Language-Action (VLA) models. However, this paradigm suffers from *causal neglect*: although WAMs may generate plausible future observations, their predictions remain dominated by the current visual context and learned video priors, exhibiting only weak dependence to the conditioning action. To address this issue, we present **CausalWAM**, a causally grounded Mixture-of-Transformers (MoT) framework with a mixed training recipe. The recipe combines Action-Conditioned World Modeling (AC-WM), which uses clean action tokens to provide direct supervision for visual dynamics, with World Action Modeling (WAM) training, which preserves the joint action–video modeling pathway required for action generation. A progressive schedule is developed to emphasize AC-WM early to establish reliable action dependence and gradually shifts toward WAM to integrate the learned dynamics into action generation. By treating actions as conditioning variables rather than imitation targets, CausalWAM effectively exploits failed and suboptimal trajectories as transition-level supervision, thereby enhancing data efficiency. Experiments in simulation and on real robots demonstrate the effectiveness of CausalWAM in improving action dependence and policy performance, while system-optimized action-only inference reaches $85ms$ on an NVIDIA RTX 4090. The code will be publicly released.

1 INTRODUCTION

Recent advances in Vision-Language-Action (VLA) models have achieved remarkable success in embodied intelligence by learning to map visual observations and language instructions to robot actions from large-scale demonstrations (Black et al., 2024; Intelligence et al., 2025b;a; Team et al., 2025a; 2026a; Ni et al., 2026; Bjorck et al., 2025; Zhai et al., 2025; Wu et al., 2026; Ye et al., 2025). However, they suffer from a supervision bottleneck: unlike dense visual streams, action labels are sparse, low-dimensional signals that encode only commanded behaviors without capturing the resulting physical consequences. World Action Models (WAMs) address the limitation by jointly modeling robot actions and future visual observations, thereby leveraging dense visual dynamics as a complementary supervisory signal for robust policy learning (Wan et al., 2025; Agarwal et al., 2025; 2026; Team et al., 2025b; Liu et al., 2025a; Wang et al., 2024b).

Among these WAMs, action-conditioned variants (Ye et al., 2026) are particularly relevant. Through a carefully designed attention mask (Fig. 1 (a)), these models establish an intended one-way dependency and factorize the joint prediction such that future visual dynamics are explicitly conditioned on both the current observation and the predicted action. This design confers two complementary advantages. First, action-conditioned future prediction couples action generation with its visual consequences, allowing the model to learn action-relevant environmental dynamics. Aligning predicted and observed transitions provides dense, physically grounded supervision to the action branch beyond sparse behavioral cloning, promoting dynamics-aware policies. Second, under this factorization, action prediction does not depend on future-video generation, allowing the computationally heavy video decoder to be omitted at test time.

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.

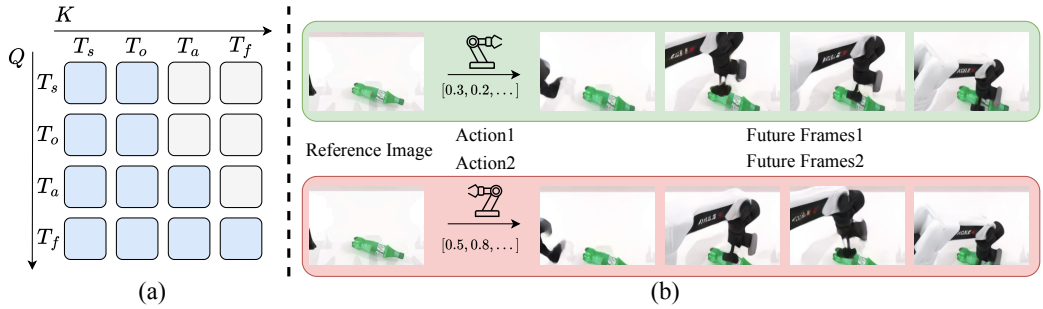

Figure 1: (a) The action-conditioned attention mask. It establishes the intended one-way dependency: action tokens (T_a) attend to the current observation (T_o & T_s), while future-video tokens (T_f) attend to both the observation (T_o & T_s) and action tokens (T_a), with reverse dependencies masked. (b) Counterfactual action intervention. Substantially different actions yield highly similar future-video predictions, revealing weak action sensitivity and causal neglect.

Despite its architectural appeal, the action-conditioned WAM does not guarantee that the intended causal influence of actions on future visual dynamics is actually learned. Although the attention mask permits the future-video branch to condition on the predicted action, it does not enforce this dependency. Under the standard joint denoising objective, both the action and video tokens are corrupted by noise simultaneously, while the reference image remains clean. This asymmetry enables a *visual shortcut*: the model tends to minimize the prediction loss by relying predominantly on the current visual context and learned visual priors, largely ignoring the noisy action signal. As shown in Fig. 1 (b), counterfactual interventions confirm this deficiency: altering the action while holding the observation fixed yields negligible changes in the predicted future. We term this failure mode *causal neglect*: the model fails to preserve the causal effect of actions despite its explicit action-conditioned prediction formulation. Consequently, gradients from future-video prediction cannot effectively supervise the action branch, leaving the original supervision bottleneck unresolved.

To address *causal neglect*, we propose CausalWAM, a Causally-Grounded Mixture-of-Transformers (MoT) framework that reinstates actions as reliable drivers of future visual dynamics. Rather than treating action conditioning as a static architectural prior, CausalWAM decouples the learning of action-sensitive dynamics from the execution of the full observation–action–future pathway. This decoupling is realized through two complementary objectives within a shared MoT backbone. The first, Action-Conditioned World Modeling (AC-WM), supplies ground-truth actions as noise-free conditions to the future-video branch, enabling dense visual supervision to directly anchor the action-to-future dependency without interference from denoising noise. The second, World Action Modeling (WAM), retains the standard joint action-video modeling where the action branch predicts actions from observations and the video branch models their consequences, ensuring the video branch can be omitted for efficient action-only inference. These objectives are not trained uniformly; instead, we adopt a progressive curriculum that prioritizes AC-WM early to establish robust causal sensitivity, then gradually shifts weight toward WAM to bind this sensitivity to the policy generation pathway. This asymmetric schedule enables the joint action-video model to inherit the causal structure established under clean action conditioning, strengthening the complete observation–action–future chain while retaining efficient action-only inference.

Furthermore, the decoupled architecture enables quality-aware data routing, unlocking the latent dynamical value within heterogeneous trajectories. Suboptimal or failed trajectories are typically discarded in WAM since their actions do not reflect desirable policy behavior; yet they encode valid state–action–outcome transitions that remain invaluable for world modeling. CausalWAM exploits this distinction by assigning such trajectories exclusively to AC-WM, where actions serve as conditioning inputs rather than imitation targets, thereby extracting transition dynamics without risking policy corruption. Conversely, only high-quality trajectories are routed to WAM for joint policy and dynamics learning. This selective utilization transforms previously wasted data into pure supervisory signals for the world model, substantially expanding state–action coverage and improving overall sample efficiency.

In summary, our contributions are threefold:

- We identify *causal neglect* in action-conditioned world-action models: video predictions remain insensitive to actions despite an explicit causal structure. This failure impedes the propagation of dense video supervision to sparse action signals.
- We propose **CausalWAM**, a Causally-Grounded Mixture-of-Transformers framework by decoupling dynamics learning from policy generation. This design restores reliable action-conditioned visual dynamics while preserving action-only inference, and enables quality-aware data routing to leverage suboptimal trajectories for robust dynamics learning, substantially improving sample efficiency.
- Extensive experiments validate that CausalWAM improves action sensitivity, downstream policy performance, and data utilization, while maintaining efficient action-only inference ($85ms$ latency on RTX 4090).

2 Related Work

World models (LeCun, 2022; Zhu et al., 2024) have long been studied as predictive representations for decision making, and recent advances in large-scale video generation have renewed their relevance to robotic control (Agarwal et al., 2026; Team et al., 2026c; Wang et al., 2024a; Zhao et al., 2025). Rather than directly mapping observations to actions, world-model-based policies predict future scene evolution to provide temporal structure, dynamics priors, or intermediate plans.

Early video-based policy methods, such as UniPi (Du et al., 2023), formulate control as future-video generation followed by action recovery, while generative robot models such as GR-1 (Wu et al., 2024a) further combine large-scale video pretraining with action prediction to improve generalization in robotic manipulation.

World Action Models (WAMs) (Bi et al., 2026; Li et al., 2026; Shen et al., 2026; Ye et al., 2026) build on the video-generation paradigm by jointly modeling robot actions and future visual dynamics within a unified framework. By coupling action prediction with future visual modeling, WAMs provide dense temporal supervision and predictive dynamics priors beyond sparse demonstration actions. VideoVLA (Shen et al., 2026) adapts pretrained video-generation models to robotic manipulation by jointly predicting future visual outcomes and action sequences. Motus (Bi et al., 2026) introduces a unified latent action world model based on a Mixture-of-Transformers architecture and UniDiffuser-style scheduling, supporting multiple modeling modes including world modeling, inverse dynamics, policy learning, video generation, and joint video-action prediction. UWM (Zhu et al., 2025) similarly unifies video and action diffusion, demonstrating that heterogeneous video and robot data can support scalable policy learning. MotuBrain (Team et al., 2026c) further develops this direction with multiview modeling, cross-embodiment action representations, and deployment-oriented optimizations for real-time control. Furthermore, DiT4DiT (Ma et al., 2026) uses intermediate denoising features rather than decoded video frames for action prediction, while LaWAM (Chen et al., 2026) exposes future dynamics through compact latent visual subgoals instead of generating pixel-level future videos. Together, these works investigate the utility of latent world-model representations for policy learning.

However, WAMs that explicitly generate future video during inference incur substantial computational overhead, impeding real-time deployment. GigaWorld-Policy (Ye et al., 2026) and Fast-WAM (Yuan et al., 2026) explore more efficient ways to incorporate predictive world knowledge into action policies. Both methods use video-based predictive supervision during training while avoiding explicit future-video generation at test time. These works demonstrate that the benefits of world modeling do not necessarily require computationally expensive pixel-level future prediction during policy execution.

Building on these efforts, *CausalWAM* retains the action-centered WAM formulation and a lightweight action-only execution pathway for efficient inference, while addressing a complementary limitation: explicit action conditioning alone does not ensure that future predictions faithfully capture action-induced effects, thereby weakening the ability of dense visual prediction losses to provide informative supervision for action learning. CausalWAM strengthens the causal action-to-future dependency through a causal mixed training recipe, combining dense visual supervision with efficient action generation.

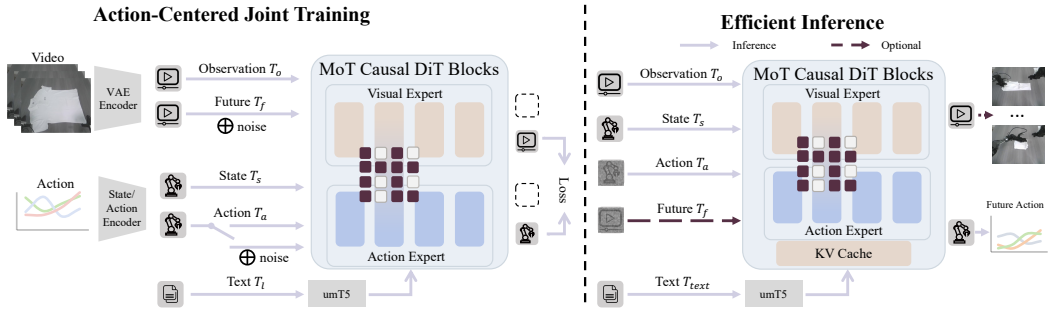

Figure 2: Overview of *CausalWAM*, an MoT-based action-centered World Action Model. The model consists of a visual expert and an action expert: the visual expert specializes in processing video tokens, while the action expert focuses on action-token modeling. The two experts are connected through multi-modal self-attention, which follows the same causal masking strategy as GigaWorld-Policy (Ye et al., 2026) to preserve action-centered dependency modeling.

3 METHOD

We first review the formulation of world action models and then describe the MoT-based architecture of *CausalWAM*, followed by the proposed causal mixed-training recipe.

3.1 PRELIMINARIES

We formulate robotic manipulation as a sequential decision-making problem. At each time step t, the policy receives multi-view visual observations $O_t = \{o_t^{\text{left}}, o_t^{\text{front}}, o_t^{\text{right}}\}$, a state s_t, and a language instruction l. Conditioned on them, the policy predicts an action chunk of length p: $a_{t:t+p-1} = (a_t, a_{t+1}, \ldots, a_{t+p-1})$.

World Action Models (WAMs) augment action modeling with future visual prediction. In addition to the action chunk, the model predicts a sequence of future visual observations sampled with temporal stride Δ: $O_{t+\Delta:t+K\Delta} = (O_{t+\Delta}, O_{t+2\Delta}, \ldots, O_{t+K\Delta})$, $\quad K = \left\lfloor \frac{p}{\Delta} \right\rfloor$. Thus, the joint WAM formulation can therefore be written as:

$$(a_{t:t+p-1}, O_{t+\Delta:t+K\Delta}) \sim g_\theta \left(\cdot \mid O_t, s_t, l\right). \tag{1}$$

Action-conditioned WAMs further structure this joint modeling through an action-conditioning signal. Specifically, a unified model jointly parameterizes the action chunk and an associated action latent c_t:

$$(a_{t:t+p-1}, c_t) \sim g_\theta \left(\cdot \mid O_t, s_t, l\right), \tag{2}$$

where c_t denotes the action-conditioning signal produced by the action branch and exposed to the future-video branch. Conditioned on the same context and c_t, the model predicts the future:

$$O_{t+\Delta:t+K\Delta} \sim g_\theta \left(\cdot \mid O_t, s_t, l, c_t\right). \tag{3}$$

3.2 MODEL ARCHITECTURE

CausalWAM features a Mixture-of-Transformers (MoT) backbone that separates visual dynamics modeling from action generation while maintaining cross-modal information exchange. The overall architecture is illustrated in Fig. 2. The model takes multi-view visual observations O_t, states s, language instructions l, action tokens $a_{t:t+p-1}$, and future visual tokens $O_{t+\Delta,t+K\Delta}$ as inputs.

Input Tokens. For the visual input, we follow the multi-view composition strategy of Motus (Bi et al., 2026), arranging the three camera views $\{o_t^{\text{left}}, o_t^{\text{front}}, o_t^{\text{right}}\}$ into a single composite image to obtain O_t^{comp}. We then use the visual VAE from Wan2.2 (Wan et al., 2025) to encode the current observation O_t^{comp} and the future observation sequence $O_{t+\Delta:t+K\Delta}^{\text{comp}}$ into current-observation and future-visual latent tokens, denoted by T_o and T_f, respectively. The proprioceptive state s_t and the action chunk $a_{t:t+p-1}$ are projected into the shared visual hidden dimension using separate

multi-layer perceptrons (MLPs), producing state tokens T_s and action tokens T_a, respectively. The language instruction l is encoded by umT5 (Chung et al., 2023), and the resulting text embeddings are used as conditioning tokens for instruction-following control. Finally, we concatenate modality-specific tokens into a unified sequence as input:

$$T_{input} = [T_s, T_o, T_a, T_f]. \tag{4}$$

Mixture-of-Transformers (MoT) Backbone. Rather than using a fully shared Transformer for all modalities, *CausalWAM* adopts an MoT architecture that provides modality-specific processing for action generation and visual dynamics modeling. The architecture contains two experts: an action expert and a visual expert. The action expert specializes in processing action tokens and predicting the action chunk, while the visual expert processes current and future visual tokens and models the evolution of the visual scene in the latent space.

Each expert is equipped with modality-specific cross-attention and feed-forward network modules. These modules enable the action and visual streams to perform modality-specific transformations while preserving a shared hidden dimension for cross-modal interaction. The two experts exchange information through a shared multimodal self-attention module operating on the unified token sequence in Eq. 4. We apply the action-centered attention mask proposed by GigaWorld-Policy (Ye et al., 2026) to this module to regulate cross-modal attention and impose the intended one-way action-to-future information dependency. This design preserves distinct inductive biases for policy prediction and visual dynamics modeling while enabling controlled cross-modal information flow.

3.3 Causal Mixed Training

CausalWAM is trained with two complementary forms of world modeling over the shared MoT backbone. Both formulations use the flow-matching framework, while differing in how the action tokens are treated. We first introduce the common flow-matching formulation, followed by the objectives for *action conditioned world modeling* and standard *world action modeling*.

Flow-Matching Formulation. For either modality, let x denote the clean target representation, corresponding to either the action-token representation or the future-visual latent. We sample a flow time $u \sim \mathcal{U}(0, 1)$ and Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and construct the interpolated noisy variable

$$x^{(u)} = (1 - u)\epsilon + ux. \tag{5}$$

The corresponding target velocity is

$$\dot{x}^{(u)} = \frac{\mathrm{d}x^{(u)}}{\mathrm{d}u} = x - \epsilon. \tag{6}$$

Action-Conditioned World Modeling. In AC-WM, we instantiate the general flow-matching formulation in Eqs. 5–6 with the future-visual token sequence T_f as the clean target, yielding the target velocity $\dot{T}_f^{(u)}$. The action chunk is encoded into clean action tokens T_a and provided to the future-video branch as a conditioning signal, while only the future-visual tokens are flow-matched. The resulting objective is

$$\mathcal{L}_{\mathrm{AC-WM}} = \mathbb{E}_{u,\boldsymbol{\epsilon}_f}\left[\left\| G_\theta^f\left(T_f^{(u)}, u \mid T_s, T_o, T_a\right) - \dot{T}_f^{(u)}\right\|_2^2\right]. \tag{7}$$

In this formulation, G_θ^f denotes the visual expert, and the action tokens T_a remain clean and serve as conditioning inputs for future-visual prediction.

World Action Modeling. Unlike AC-WM, WAM applies the general flow-matching formulation in Eqs. 5–6 to both action and future-visual token sequences. The two noisy token sequences are jointly processed by the shared MoT backbone, while the corresponding branch-wise objectives can be written as

$$\mathcal{L}_{\mathrm{action}} = \mathbb{E}_{u^a,\boldsymbol{\epsilon}_a}\left[\left\| G_\theta^a\left(T_a^{(u^a)}, u^a \mid T_s, T_o\right) - \dot{T}_a^{(u^a)}\right\|_2^2\right], \tag{8}$$

$$\mathcal{L}_{\mathrm{video}} = \mathbb{E}_{u^f,\boldsymbol{\epsilon}_f}\left[\left\| G_\theta^f\left(T_f^{(u^f)}, u^f \mid T_s, T_o, T_a^{(u^a)}\right) - \dot{T}_f^{(u^f)}\right\|_2^2\right]. \tag{9}$$

Here, G_θ^a denotes the action expert, which predict the velocity fields for the action token sequence, respectively. The action-condition attention mask (Ye et al., 2026) prevents the action branch from accessing future-visual tokens while allowing the future-video branch to attend to the action tokens. The overall WAM objective is

$$\mathcal{L}_{\mathrm{WAM}} = \mathcal{L}_{\mathrm{action}} + \mathcal{L}_{\mathrm{video}}. \tag{10}$$

Progressive Mixing Schedule. We progressively shift the training objective from AC-WM to WAM. Let $q \in [0, 1]$ denote the normalized training progress, and let $\rho_{\mathrm{AC}}(q)$ denote the fraction of training updates assigned to AC-WM:

$$\rho_{\mathrm{AC-WM}}(q) = \rho_{\min} + \frac{\rho_{\max} - \rho_{\min}}{2} \left(1 + \cos(\pi q)\right), \tag{11}$$

where $\rho_{\max}$ and $\rho_{\min}$ are the initial and final AC-WM ratios, respectively. The corresponding WAM ratio is

$$\rho_{\mathrm{WAM}}(q) = 1 - \rho_{\mathrm{AC}}(q). \tag{12}$$

In practice, $\rho_{\mathrm{AC}}(q)$ controls the probability of selecting AC-WM for a training update, or equivalently the relative weight of the two objectives in expectation:

$$\mathcal{L}_{\mathrm{total}}(q) = \rho_{\mathrm{AC-WM}}(q)\mathcal{L}_{\mathrm{AC-WM}} + \rho_{\mathrm{WAM}}(q)\mathcal{L}_{\mathrm{WAM}}. \tag{13}$$

The early emphasis on AC-WM establishes an action-sensitive action-to-future dependency under clean action conditioning. As training proceeds, the increasing contribution of standard WAM transfers this dependency to the joint action-generation pathway, thereby strengthening the complete observation–action–future dependency.

Supervision-Aware Trajectory Assignment. The two training objectives $\mathcal{L}_{\mathrm{AC-WM}}$ and $\mathcal{L}_{\mathrm{WAM}}$ also support different uses of heterogeneous trajectories. Let $\mathcal{D}_{\mathrm{trans}}$ denote trajectories with reliable observed state-action transitions, and $\mathcal{D}_{\mathrm{policy}} \subseteq \mathcal{D}_{\mathrm{trans}}$ denote trajectories whose actions are suitable as imitation targets. We formulate the data allocation as

$$\mathcal{D}_{\mathrm{AC}} = \mathcal{D}_{\mathrm{trans}}, \qquad \mathcal{D}_{\mathrm{WAM}} = \mathcal{D}_{\mathrm{policy}}.$$

Therefore, trajectories that are unsuitable for policy imitation can still contribute to AC-WM as transition supervision, while standard WAM training is reserved for policy-relevant trajectories.

3.4 Inference Acceleration

To support real-world deployment on edge computing platforms, we optimize the inference stack through KV caching, graph compilation, and a lightweight C++ runtime. During autoregressive action generation, the visual and language context remains unchanged across decoding steps; we therefore cache the key-value states of all attention layers after encoding the input context and reuse them for subsequent action-token predictions, avoiding redundant attention computation. We further apply `torch.compile` to optimize the inference graph, which reduces Python-level dispatch overhead and enables backend-level optimizations such as operator fusion and more efficient memory scheduling. Finally, we deploy the optimized policy in a C++ runtime that integrates image preprocessing, tensor construction, model execution, KV-cache management, and action post-processing into a unified native pipeline. This design removes the Python dependency, reduces runtime overhead, and simplifies integration with robot-control middleware, enabling low-latency closed-loop policy execution on edge devices.

4 Experiments

In this section, we evaluate *CausalWAM* against existing WAM methods in both simulation and real-world robotic settings, and conduct detailed ablation studies to analyze the contributions of its proposed components. Further details on the evaluation metrics and implementation settings are provided in the Appendix A.1–A.2.

Baselines. We compare *CausalWAM* with several representative baselines spanning two major paradigms. For VLA methods, we include $\pi_{0.5}$ (Intelligence et al., 2025b), which uses a large vision-language backbone to align visual observations with language instructions and decode low-level control actions through an action head. For WAM methods, we include Motus (Bi et al., 2026), FastWAM (Yuan et al., 2026), and GigaWorld-Policy (Ye et al., 2026), which incorporate visual dynamics or world-modeling objectives to support policy learning.

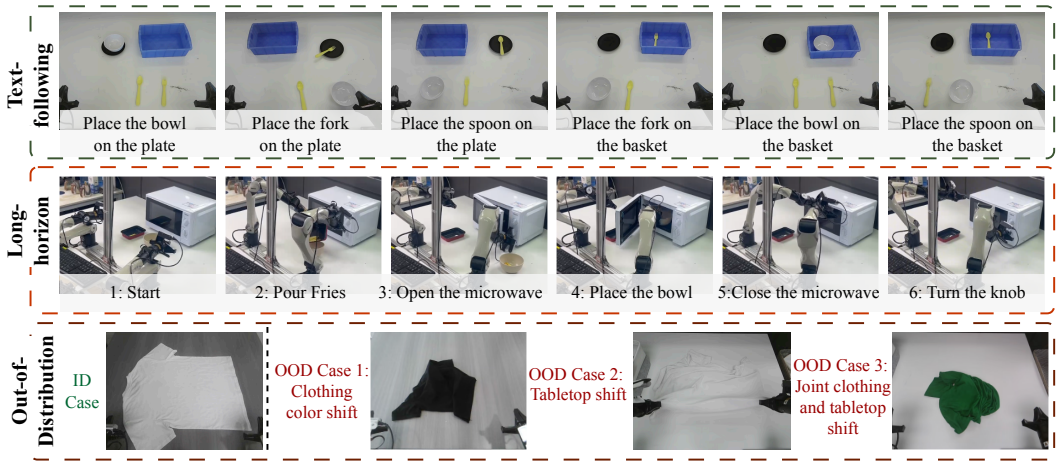

Figure 3: Demonstrations of real-world evaluation settings. Top: Text-following under a fixed scene configuration, covering six object-placement instructions. Middle: The long-horizon Food Heating task, which requires six temporally dependent manipulation steps. Bottom: The in-distribution clothing-folding scene and three OOD settings.

Table 1: Evaluation on RoboTwin 2.0 Simulation. Results are reported for representative tasks and averaged over all 50 tasks; complete per-task results are provided in the Appendix A.4. The best results are marked in **bold**, and the second-best results are underlined.

Setting	Methods	*Adjust Bottle*	*Scan Object*	...	*Open Microwave*	*Turn Switch*	Average
Clean	$\pi_{0.5}$ (Intelligence et al., 2025b)	79	42	...	35	5	42.98
	Motus (Bi et al., 2026)	89	67	...	**95**	**84**	88.66
	FastWAM (Yuan et al., 2026)	**100**	89	...	62	61	91.88
	GigaWorld-Policy (Ye et al., 2026)	**100**	60	...	74	82	86.36
	CausalWAM	**100**	**94**	...	88	68	**92.36**
Rand.	$\pi_{0.5}$ (Intelligence et al., 2025b)	83	38	...	37	6	43.84
	Motus (Bi et al., 2026)	93	66	...	**91**	78	87.02
	FastWAM (Yuan et al., 2026)	**100**	**92**	...	45	**95**	91.78
	GigaWorld-Policy (Ye et al., 2026)	**100**	64	...	66	84	85.04
	CausalWAM	**100**	72	...	72	74	**91.84**

4.1 Simulation Benchmark Experiments

We evaluate single-task performance on 50 representative manipulation tasks from RoboTwin 2.0 (Chen et al., 2025) under domain randomization. To evaluate multi-task performance and the effect of pretraining, we train all compared methods on a mixed dataset comprising 2,500 demonstrations from clean scenes (50 per task) and 25,000 demonstrations from heavily randomized scenes (500 per task). The randomization varies backgrounds, table clutter, table height, and lighting, thereby testing robustness to distribution shifts. As shown in Tab. 1, *CausalWAM* achieves the best overall performance in both clean and randomized settings, outperforming prior WAM-based methods such as Motus (Bi et al., 2026) and FastWAM (Yuan et al., 2026). These results demonstrate CausalWAM's strong performance across diverse manipulation tasks.

4.2 Real-World Experiment

We conduct gripper-based manipulation experiments on an AgileX PiPER 6-DoF robotic arm under three complementary settings: **text following**, **long-horizon manipulation**, and **out-of-distribution (OOD) generalization** as shown in Fig. 3. Text-following tasks evaluate language grounding by executing diverse instructions under a fixed scene configuration. Long-horizon tasks evaluate end-to-end sequential manipulation over temporally extended and multi-stage task sequences, with a

Table 2: Evaluation across three real-world tasks, with each individual setting evaluated over 10 independent trials. The best results are marked in **bold**, and the second-best results are underlined.

Tasks	Settings	Methods				
		$\pi_{0.5}$	Motus	FastWAM	GigaWorld-Policy	*CausalWAM*
Text-following	Pick up the *bowl* and place it on the *plate*	0.88	0.88	0.73	0.80	**0.93**
	Pick up the *fork* and place it on the *plate*	0.78	0.83	0.85	0.80	**0.88**
	Pick up the *spoon* and place it on the *plate*	0.73	0.83	0.80	0.75	**0.85**
	Pick up the *bowl* and place it into the *basket*	0.75	0.83	0.80	0.88	**0.95**
	Pick up the *fork* and place it into the *basket*	0.65	0.75	0.68	0.78	**0.85**
	Pick up the *spoon* and place it into the *basket*	0.80	0.85	0.75	0.83	**0.88**
	Average	0.77	0.83	0.77	0.81	**0.89**
Long-horizon	-	0.50	0.60	0.50	0.60	**0.80**
OOD	Clothing color shift	0.40	0.30	0.10	0.40	**0.50**
	Tabletop shift	0.50	0.60	0.50	0.60	**0.80**
	Joint clothing and tabletop shift	0.30	0.30	0.10	0.30	**0.40**
	Average	0.40	0.40	0.23	0.43	**0.57**

focus on maintaining coherent execution and recovering from intermediate failures. OOD tasks evaluate the model's generalization and robustness under distribution shifts.

Tab. 2 summarizes the evaluation results across three real-world settings. **For the text-following setting**, we construct six instruction-conditioned subtasks by varying three object categories (bowl, fork, and spoon) and two destination locations (plate and basket). Each subtask requires the policy to ground the language instruction to the corresponding object and placement action. *CausalWAM* achieves the best performance on all six instructions, attaining an average success rate of 0.89 and outperforming the strongest baseline, Motus, by 0.06. This result indicates stronger language grounding of both object identities and spatial relations. **For the long-horizon setting**, we design a complex food-heating task involving sequential grasping, placing, pouring, poking, and knob-turning actions. *CausalWAM* achieves a success rate of 0.80, yielding an absolute improvement of 0.20 and a relative improvement of 33% over the strongest baseline. This gain highlights the effectiveness of our method in long-horizon execution, where successful completion requires maintaining coherent action sequences and accurately handling intermediate state transitions. It also suggests that the predictive representations learned through world modeling provide useful temporal and interaction-aware information for robotic manipulation. **For the OOD evaluation**, we train the policies on the in-distribution clothing-folding task and test them under three distribution shifts: a clothing color shift, a tabletop-background shift, and a joint shift involving both factors. *CausalWAM* achieves success rates of 0.50, 0.80, and 0.40 in the three OOD settings, respectively, obtaining the best performance in each case and an average success rate of 0.57. These results demonstrate improved robustness to appearance and background changes, including their combined effect.

4.3 Ablation Studies

We conduct comprehensive ablation studies on the components of *CausalWAM* to assess their contributions to both task performance and inference efficiency.

Effect of the pretraining strategy. We compare three pretraining strategies: WAM-only pretraining, AC-WM-only pretraining, and the proposed causal mixed pretraining. As shown in Fig. 4 (a), causal mixed pretraining achieves the best performance, reaching a success rate of 75%, compared with 71% for WAM-only and 59% for AC-WM-only pretraining. These results indicate that AC-

Table 3: Comparison of inference efficiency and real-robot performance. Lat. (A) and Lat. (R) denote inference latency on an NVIDIA A100 and an NVIDIA RTX 4090, respectively. SR denotes the average success rate across the three real-world tasks. The best results are highlighted in **bold**.

Method	Lat. (A) ↓	Lat. (R) ↓	SR ↑
$\pi_{0.5}$	225	110	0.56
Motus	3231	–	0.61
FastWAM	229	182	0.50
GigaWorld-Policy	360	293	0.61
CausalWAM	353	285	**0.75**
+ KV caching	301	241	**0.75**
++ graph compilation	189	110	**0.75**
+++ C++ deployment	**140**	**85**	**0.75**

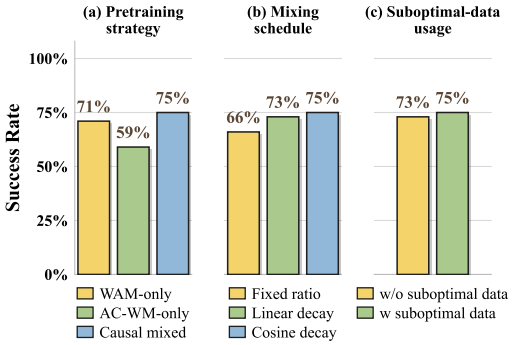

Figure 4: Ablation study of the components of *CausalWAM*.

WM is most effective when combined with WAM, which preserves the action-generation pathway, and supports the complementary roles of the two training objectives.

Effect of the mixing schedule. Fig. 4 (b) evaluates different schedules for causal mixed training. In the fixed-ratio setting, we maintain $\rho_{\mathrm{AC}} = 0.25$ throughout training. The linear and cosine schedules use the same initial and final mixing ratios, differing only in how the ratio changes over training. The fixed-ratio strategy achieves a lower success rate than both dynamic schedules, indicating that progressively shifting the training emphasis from AC-WM to WAM is important for effective policy learning. Among the two dynamic schedules, cosine decay performs slightly better than linear decay, suggesting that its smoother transition provides a more effective balance between early action-conditioned dynamics learning and later policy integration.

Effect of suboptimal-data usage. Fig. 4 (c) studies whether trajectories with suboptimal actions can provide useful supervision. Removing these trajectories reduces the success rate from 75% to 73%. This result shows that suboptimal trajectories can contribute valid state–action–outcome transition information when used as conditioning data, while avoiding their use as imitation targets. The improvement supports the proposed supervision-aware trajectory assignment and demonstrates its benefit for data utilization.

Inference efficiency. We compare CausalWAM with representative VLA and WAM methods in terms of inference latency and real-robot success rate, while also isolating the contribution of each acceleration component in *CausalWAM*. The resulting comparison is summarized in Table 3. The MoT-based action-only execution pathway already provides slightly lower latency than GigaWorld-Policy, and KV caching further reduces the latency. Graph compilation brings these values down to 189 ms and 110 ms, respectively. With the optimized C++ runtime, the latency is finally reduced to 140 ms on the A100 and 85 ms on the RTX 4090, 23% faster than $\pi_{0.5}$, demonstrating that *CausalWAM* combines stronger real-robot performance with efficient inference.

5 Conclusion

In this paper, we introduced *CausalWAM*, a causally grounded World Action Model that addresses causal neglect in action-conditioned future prediction. Built on an action-centered WAM formulation, *CausalWAM* retains future visual dynamics as a training-time supervisory signal while enabling action-only decoding during inference. Its mixed-training recipe combines Action-Conditioned World Modeling with standard WAM training, encouraging the model to capture action-induced visual changes and transfer them to action generation. Moreover, CausalWAM demonstrates a practical path toward faster and more deployable World Action Models for robot control. With C++ deployment on a local RTX 4090, it further achieves $85ms$ inference latency, improving the efficiency-performance trade-off of action-centered WAMs by retaining dense supervision from future visual dynamics while reducing the cost of action generation at inference time.

AI USE STATEMENT

In this work, we used generative AI tools to assist with English writing and polishing, including grammar, wording, readability, technical exposition, and LaTeX formatting. The authors take full responsibility for the final content of this work, including all text, scientific claims, experimental results, analyses, code, and other artifacts produced with the assistance of generative AI.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

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.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, et al. Lawam: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Hyung Won Chung, Xavier Garcia, Adam Roberts, Yi Tay, Orhan Firat, Sharan Narang, and Noah Constant. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. In *The Eleventh International Conference on Learning Representations*, 2023.

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.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi_{0.6}$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025a.

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*, 2025b.

Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 2022.

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

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7353–7363, 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.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. Dit4dit: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026.

Chaojun Ni, Cheng Chen, Xiaofeng Wang, Zheng Zhu, Wenzhao Zheng, Boyuan Wang, Tianrun Chen, Guosheng Zhao, Haoyun Li, Zhehao Dong, et al. Swiftvla: Unlocking spatiotemporal dynamics for lightweight vla models at minimal overhead. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13474–13485, 2026.

Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. Videovla: Video generators can be generalizable robot manipulators. *Advances in neural information processing systems*, 38:95597–95621, 2026.

Hengkai Tan, Yao Feng, Xinyi Mao, Shuhe Huang, Guodong Liu, Zhongkai Hao, Hang Su, and Jun Zhu. Anypos: Automated task-agnostic actions for bimanual manipulation. *arXiv preprint arXiv:2507.12768*, 2025.

GigaBrain Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jie Li, Jiagang Zhu, Lv Feng, et al. Gigabrain-0: A world model-powered vision-language-action model. *arXiv preprint arXiv:2510.19430*, 2025a.

GigaBrain Team, Boyuan Wang, Bohan Li, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. Gigabrain-0.5 m*: a vla that learns from world model-based reinforcement learning. *arXiv preprint arXiv:2602.12099*, 2026a.

GigaWorld Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jiagang Zhu, Kerui Li, Mengyuan Xu, et al. Gigaworld-0: World models as data engine to empower embodied ai. *arXiv preprint arXiv:2511.19861*, 2025b.

GigaWorld Team, Angyuan Ma, Boyuan Wang, Bohan Li, Chaojun Ni, Guo Li, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, et al. Gigaworld-1: A roadmap to build world models for robot policy evaluation. *arXiv preprint arXiv:2607.02642*, 2026b.

MotuBrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, et al. Motubrain: An advanced world action model for robot control. *arXiv preprint arXiv:2604.27792*, 2026c.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European conference on computer vision*, pp. 55–72. Springer, 2024a.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Boyuan Wang, Xinze Chen, and Jiwen Lu. Worlddreamer: Towards general world models for video generation via predicting masked tokens. *arXiv preprint arXiv:2401.09985*, 2024b.

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, 2024a.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024b.

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.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. Gigaworld-policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026.

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.

Andy Zhai, Brae Liu, Bruno Fang, Chalse Cai, Ellie Ma, Ethan Yin, Hao Wang, Hugo Zhou, James Wang, Lights Shi, et al. Igniting vlms toward the embodied space. *arXiv preprint arXiv:2509.11766*, 2025.

Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. Drivedreamer-2: Llm-enhanced world models for diverse driving video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10412–10420, 2025.

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.

Zheng Zhu, Xiaofeng Wang, Wangbo Zhao, Chen Min, Bohan Li, Nianchen Deng, Min Dou, Yuqi Wang, Botian Shi, Kai Wang, et al. Is sora a world simulator? a comprehensive survey on general world models and beyond. *arXiv preprint arXiv:2405.03520*, 2024.

A Appendix

A.1 Evaluation Metrics.

We primarily report the Success Rate (SR) to measure task completion. In simulation, SR is defined as a binary indicator: $\mathrm{SR} = 1$ if the task is successfully completed and $\mathrm{SR} = 0$ otherwise. For real-world evaluation, we consider three settings: **text-following tasks**, **long-horizon tasks**, and **out-of-distribution (OOD) tasks**. For text-following tasks, we use a graded SR to better capture partial task completion: each trial is scored by four equally weighted stages, with 0.25 assigned to each stage, including reaching the target object, grasping it, moving to the target location, and successfully placing it. For long-horizon and OOD tasks, SR follows the same binary definition as in simulation: $\mathrm{SR} = 1$ only when the complete task sequence is successfully executed, and $\mathrm{SR} = 0$ otherwise.

A.2 Implementation Details.

We initialize the visual expert from the pretrained GigaWorld-1 (Team et al., 2026b), allowing the model to inherit strong visual generative representations. The action expert is initialized from the visual expert by copying the corresponding weights. For layers with mismatched dimensions, we retain the compatible submatrices and truncate the remaining entries to match the action-expert

configuration. The visual expert uses a hidden dimension of 3072 and an FFN dimension of 14336, while the action expert is configured with a smaller hidden dimension of 1024 and an FFN dimension of 4096.

The original resolution of images is 640×480. The action chunk length is set to 48, and the temporal stride for future observations follows the setting of GigaWorld-Policy (Ye et al., 2026). For flow matching, the action and future-visual modalities use the same modality-specific timestep transformation and linear noising procedure as GigaWorld-Policy. The flow times u^a and u^f are sampled independently for the action and future-visual tokens, respectively.

The progressive mixing schedule uses $\rho_{\max} = 1$ and $\rho_{\min} = 0$, where ρ_{AC} denotes the fraction of training updates assigned to AC-WM. In AC-WM, the action tokens remain clean and are used only as conditioning signals for future-video prediction; consequently, the action-prediction loss is removed from this training objective. WAM training flow-matches both the action and future-visual token sequences using their corresponding modality-specific flow times.

A.3 Training Details

We train *CausalWAM* in two stages: robot-data pretraining and target-robot post-training.

Robot-data pretraining. During pretraining, we use approximately $2,000$ hours of filtered open-source robot data from (Bu et al., 2025; Liu et al., 2025b; Wu et al., 2024b; Tan et al., 2025), together with internally collected real-robot data.

Rather than discarding all imperfect trajectories from the internally collected data, we retain a subset of suboptimal trajectories for dynamics modeling. We define suboptimal trajectories as demonstrations in which human errors, unexpected events, or execution failures require additional actions outside the nominal task specification to restore the scene, or prevent the scene from being fully recovered. For example, in a fries-pouring task, an excessively large motion may cause a substantial number of fries to spill onto the tabletop. Subsequent actions for collecting the spilled fries are not part of the original task objective and are therefore labeled as suboptimal behavior. We preserve such trajectories and use them with the AC-WM objective to learn action-conditioned environmental dynamics, without treating the recovery actions as target policy behavior.

Target-robot post-training. For each target task, we collect several hundreds of real-robot demonstrations on the target robot. All evaluated models are post-trained on the same task-specific data for five epochs under an identical training protocol. We then evaluate the resulting policies on the target robot using the evaluation settings described in the main paper.

A.4 More Details on the Simulation Experiments

Table 4 reports the complete per-task results on all 50 RoboTwin 2.0 simulation tasks. We compare *CausalWAM* with $\pi_{0.5}$ (Intelligence et al., 2025b), Motus (Bi et al., 2026), FastWAM (Yuan et al., 2026), and GigaWorld-Policy (Ye et al., 2026) under both clean and randomized settings.

Table 4: Per-task success rates on RoboTwin 2.0 under clean and randomized settings. The best results are marked in **bold**, and the second-best results are underlined.

Simulation Task	$\pi_{0.5}$		Motus		FastWAM		GigaWorld-Policy		*CausalWAM*	
	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.
Adjust Bottle	79	83	89	93	**100**	**100**	**100**	**100**	**100**	**100**
Beat Block Hammer	63	50	95	88	**99**	**97**	86	86	94	92
Blocks Ranking RGB	43	35	99	97	**100**	**100**	92	96	96	98
Blocks Ranking Size	8	14	75	63	**94**	**98**	44	48	78	78
Click Alarmclock	97	93	**100**	**100**	**100**	**100**	**100**	**100**	98	**100**
Click Bell	75	76	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Dump Bin Bigbin	30	42	95	91	97	96	92	**100**	**98**	96
Grab Roller	90	89	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Handover Block	18	19	86	73	**95**	81	80	80	94	**88**
Handover Mic	28	18	78	63	99	**100**	72	72	**100**	**100**
Hanging Mug	3	3	38	38	**58**	**62**	16	12	44	48
Lift Pot	0	0	96	99	**100**	**100**	98	98	**100**	**100**
Move Can Pot	29	27	34	74	**90**	88	76	78	88	**92**
Move Pillbottle Pad	33	29	93	96	**100**	**99**	90	90	**100**	96
Move Playingcard Away	59	67	**100**	96	**100**	**100**	78	72	**100**	98
Move Stapler Pad	16	18	83	**85**	77	64	**92**	82	74	74
Open Laptop	19	35	95	91	98	**100**	96	98	**100**	**100**
Open Microwave	35	37	**95**	**91**	62	45	74	66	88	72
Pick Diverse Bottles	5	3	**90**	**91**	80	85	82	70	88	86
Pick Dual Bottles	10	6	96	90	**100**	**96**	86	86	96	90
Place A2B Left	62	60	88	79	95	**93**	94	88	**96**	90
Place A2B Right	62	57	91	87	93	**99**	90	92	**94**	94
Place Bread Basket	48	56	**91**	94	**91**	93	82	82	90	**96**
Place Bread Skillet	38	46	86	83	90	**93**	**94**	90	84	88
Place Burger Fries	66	70	**98**	98	96	**99**	**98**	96	96	94
Place Can Basket	19	25	**81**	**76**	71	69	78	74	70	74
Place Cans Plasticbox	40	47	98	94	99	96	**100**	**100**	**100**	**100**
Place Container Plate	71	78	98	99	96	**100**	98	96	**100**	**100**
Place Dual Shoes	12	7	93	87	94	**88**	**96**	84	78	82
Place Empty Cup	75	86	99	98	**100**	**100**	90	90	98	96
Place Fan	25	36	91	87	**96**	**96**	92	94	**96**	94
Place Mouse Pad	21	26	66	68	83	89	88	90	**98**	**96**
Place Object Basket	43	36	81	87	89	88	90	**92**	**92**	88
Place Object Scale	40	49	88	85	90	**97**	88	80	**94**	88
Place Object Stand	74	65	98	97	90	94	**100**	**98**	96	96
Place Phone Stand	49	53	87	86	**97**	**99**	82	72	94	98
Place Shoe	57	61	99	97	96	**99**	98	96	**100**	98
Press Stapler	80	70	93	**98**	90	97	**96**	96	**96**	**98**
Put Bottles Dustbin	12	9	81	79	**95**	**90**	72	70	94	86
Put Object Cabinet	24	15	88	71	**94**	**89**	74	74	88	88
Rotate QRcode	47	56	89	73	**93**	89	90	84	88	**92**
Scan Object	42	38	67	66	89	92	60	64	**94**	**96**
Shake Bottle Horizontally	96	**100**	**100**	98	**100**	**100**	**100**	98	**100**	**100**
Shake Bottle	91	**100**	**100**	97	**100**	**100**	**100**	**100**	**100**	**100**
Stack Blocks Three	15	16	91	95	95	97	70	78	**98**	**100**
Stack Blocks Two	48	56	**100**	98	**100**	**100**	**100**	94	**100**	**100**
Stack Bowls Three	33	35	79	**87**	80	81	70	72	**90**	86
Stack Bowls Two	78	66	**98**	**98**	92	**98**	96	92	**98**	96
Stamp Seal	36	23	93	92	90	94	**96**	**98**	92	96
Turn Switch	5	6	**84**	78	61	59	82	**84**	68	74
Average	42.98	43.84	88.66	87.02	91.88	91.78	86.36	85.04	**92.36**	**91.84**