COVIEW-WAM: A GEOMETRY-AWARE WORLD-ACTION MODEL WITH CROSS-VIEW CONSISTENT IMAGINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in World-Action Models (WAMs) have shown that future latent representations can effectively guide action prediction without explicitly decoding future videos or target-state images during inference. However, existing multi-view WAMs typically fuse image observations through implicit cross-view feature interaction without explicitly modeling geometric correspondences between views. As a result, complementary cues may be exchanged ineffectively across current observations, while future representations from different views may encode inconsistent scene geometry, ultimately weakening world-to-action guidance. In this paper, we introduce CoView-WAM, a geometry-aware cross-view WAM that incorporates geometric correspondence into both cross-view interaction and future representation learning. First, we develop a geometry-guided cross-view feature interaction mechanism that establishes spatial correspondences across camera views and enables feature exchange between corresponding regions. A gated interaction strategy is further introduced to suppress irrelevant cross-view information while preserving complementary cues. Second, we jointly model future image and depth representations and use camera extrinsics to encourage cross-view geometric consistency in 3D space, guiding future representations from different views to capture a shared underlying scene structure beyond appearance similarity. Future geometric interaction and depth supervision are used only during training. At inference time, CoView-WAM requires no explicit future image or depth decoding. Experiments on both simulated (LIBERO and CALVIN) and real-world multi-view manipulation tasks demonstrate the effectiveness of CoView-WAM. CoView-WAM achieves an overall success rate of 99.20% on LIBERO, 73.8% success rate on real-world tasks, and an average completed sequence length of 3.987 on CALVIN, with strong gains on tasks involving partial occlusion, precise placement, and cross-view spatial coordination.

1 INTRODUCTION

World Action Models (WAMs) have recently emerged as a promising paradigm for robotic manipulation by incorporating world foresight into action generation. Rather than relying solely on current visual observations, WAMs anticipate how the environment may evolve under a given task and exploit such foresight to guide subsequent action generation. Early approaches achieve this capability via explicit future video generation (Ye et al., 2026; Bi et al., 2026). More recent studies (Yuan et al., 2026; Zhang et al., 2026), however, have shown that effective world-action modeling does not necessarily require decoding complete future videos/images during inference. Instead, intermediate latent representations produced during future imagination can already provide informative world-to-action. This observation suggests that the effectiveness of WAMs may depend less on generating photorealistic futures than on learning structured or action-relevant future latent representations.

Despite recent progress, existing WAMs often fuse observations from different views via implicit feature interaction without explicitly modeling cross-view geometry relationships. Observations from different views are geometrically coupled: the appearance and location of the same 3D scene region across cameras are determined by relative camera poses and perspective projection. Ignoring such relationships may hinder effective information exchange and lead to unreliable multi-view fu-

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.

sion, particularly under large viewpoint changes or partial occlusion. Moreover, even when future images are not explicitly decoded, latent future predictions from different views typically lack explicit supervision to ensure consistency in 3D space. Consequently, WAMs may encode conflicting object locations or spatial relationships, resulting in inconsistent world-to-action guidance.

In this paper, we argue that although WAMs do not require explicit future visual decoding at inference, exploiting geometric correspondence between multi-view images and maintaining geometric consistency in future representations remain crucial for reliable world-to-action guidance. To this end, we introduce CoView-WAM, a geometry-aware cross-view World-Action Model that explicitly incorporates geometric correspondence into both cross-view interaction and future representation learning. First, we develop a geometry-guided cross-view feature interaction mechanism that uses camera calibration to identify geometrically corresponding regions across views. By constraining feature retrieval to these regions, the model enriches the latent representations used for action conditioning with complementary cross-view information while reducing interference from spatially unrelated regions. We further introduce a gated cross-view interaction strategy that adaptively modulates the injection of retrieved cross-view information, reducing the influence of unreliable matches while preserving useful complementary cues. This strategy is critical under partial occlusion or viewpoint-dependent visibility, where the quality of cross-view matches can vary substantially.

Second, we introduce geometry-aware predictive representation learning by jointly modeling future image and depth representations. Image-space prediction alone captures appearance evolution but provides limited constraints on the underlying 3D scene structure. We therefore perform bidirectional geometry-guided interaction between future-view representations and introduce depth supervision grounded in a shared underlying 3D scene geometry. In this way, predictive representations from different camera views are encouraged to capture coherent spatial structure of the same scene, rather than merely exhibiting similar visual appearance. Importantly, future geometric interaction and depth supervision are used only during training to shape the predictive representation. At inference time, CoView-WAM requires no explicit future image or depth decoding, retaining only the geometry-guided interaction over current observations for action conditioning.

We evaluate CoView-WAM on LIBERO (Liu et al., 2023a), CALVIN (Mees et al., 2022), and five real-world multi-view manipulation tasks. CoView-WAM achieves 99.20% overall success on LIBERO, 73.8% on five real-world tasks and an average completed sequence length of 3.987 on CALVIN. Ablations further demonstrate the effectiveness of explicit cross-view geometry for both current interaction and predictive representation learning. We further qualitatively compare geometric and semantic retrieval across camera configurations to assess explicit cross-view geometry in multi-view world-action learning.

Our main contributions are as follows:

- We identify under-modeled cross-view geometry as an important limitation of latent multi-view WAMs and empirically show that simply adding multiple views does not consistently improve the underlying WAM when cross-view interactions remain implicit.
- We propose a geometry-guided cross-view interaction mechanism that uses camera-parameter-based candidate selection, semantic retrieval, and gated feature interaction to enrich action-conditioning representations with complementary cross-view information.
- We introduce geometry-aware predictive representation learning through joint future image and depth modeling, encouraging cross-view geometric consistency in 3D space and thereby promoting more coherent latent world representations for world-to-action learning.

2 Related Work

2.1 World-Action Models and Predictive Visual Representations

Vision-language-action models map visual observations and language instructions to robot actions with large pretrained backbones (Brohan et al., 2023; Kim et al., 2024; Black et al., 2025b;a; NVIDIA, 2025b). To complement predominantly static image-text pretraining, predictive visual objectives have been introduced for control, with text-conditioned video generation (Du et al., 2023), predictive diffusion features (Hu et al., 2025), latent world models (Ha & Schmidhuber, 2018; Hafner et al., 2020; Wu et al., 2022) and interactive video models (Bruce et al., 2024; NVIDIA,

2025a). Recent world-action models further jointly model future visual prediction with action generation (Li et al., 2026; Bi et al., 2026; Cheang et al., 2024; Cen et al., 2025; Jang et al., 2025; Ye et al., 2026), while Fast-WAM (Yuan et al., 2026) and ImageWAM (Zhang et al., 2026) reduce or remove explicit future decoding at inference. Despite this progress (Qu et al., 2025; Wu et al., 2024; Zhou et al., 2024), how camera geometry should structure multi-view information for both current conditioning and predictive representation learning remains largely unexplored.

2.2 Multi-View and Geometric Representations for Manipulation

Prior work has incorporated spatial information into robotic manipulation policies with 3D feature fields, voxel or point representations (Shridhar et al., 2022; Gervet et al., 2023; Ze et al., 2024), as well as reconstruction- or view-synthesis-based scene modeling (Ze et al., 2023; Goyal et al., 2023). In multi-camera settings, however, many methods still rely on multi-view concatenation, unified representations, or learned cross-view interaction (Zhang et al., 2026; Liu et al., 2024; Nilavadi et al., 2026), where cross-view geometric relations are learned implicitly from visual features. Recent approaches introduce explicit geometry cues, such as using wrist-pose or 4D reconstruction guidance (Qian et al., 2025), camera-aware cross-view attention (Huang et al., 2026), or depth- and correspondence-based representation learning (Yang et al., 2026). Related multi-view generation methods also exploit camera parameterization (Liu et al., 2023b; Shi et al., 2024; He et al., 2025), while geometry foundation models provide correspondence and depth priors (Wang et al., 2024; Leroy et al., 2024; Wang et al., 2025). In contrast, we directly use calibrated epipolar geometry to constrain cross-view feature retrieval and extend the same geometric relation to future representation learning with auxiliary depth prediction.

3 Method

3.1 Problem Formulation and Overview

We consider a robot manipulation model conditioned on current multi-view visual observations (*e.g.*, external and wrist images) and a task language instruction. At time t, the model uses external and wrist images I_t^e, I_t^w, a language instruction c, and robot proprioceptive state s_t, then predicts an action sequence $A_t = (a_t, \dots, a_{t+H-1})$. Meanwhile, we use calibrated multi-view geometry to establish geometry-guided cross-view feature interaction. The camera calibration parameters are denoted by $\mathcal{G}_t = \{K_t^v, T_t^v\}_{v\in\{e,w\}}$, where K_t^v is the intrinsic matrix and $T_t^v \in \mathrm{SE}(3)$ maps camera coordinate to the world coordinate, including rotation component R_t^v and camera center o_t^v. During training, we additionally use future images $I_{t_f}^v$, future calibration $\mathcal{G}_{t_f}$, and future depth targets $D_{t_f}^v$, where $t_f = t + H - 1$ is the future endpoint used by the training data sampler.

Our CoView-WAM follows a dual-expert WAM framework consisting of a world expert and an action expert. The world expert processes current multi-view observations to construct layer-wise conditioning key/value representations for the action expert. During training, a separate predictive pathway additionally processes noisy future visual inputs to regularize the learned world representations. CoView-WAM introduces geometry-guided cross-view retrieval and reliability-aware gated interaction to enrich current conditioning representations with complementary cross-view information. For predictive representation learning, we further apply bidirectional geometry-guided interaction and auxiliary depth supervision to future-view representations, encouraging 3D-aware cross-view geometric consistency. Each camera view contains N visual positions on an $H_p \times W_p$ grid, whose token centers are mapped to homogeneous pixel coordinates p_i^v in the calibrated image coordinate system consistent with K_t^v. At inference, only current observations and camera calibration are required, similar to ImageWAM (Zhang et al., 2026) and FastWAM (Yuan et al., 2026), while future interaction and depth supervision remain training-only.

3.2 Geometry-Guided Cross-View Feature Retrieval

We first introduce geometry-guided cross-view feature retrieval, which uses calibrated epipolar geometry to restrict feature matching to geometrically plausible regions across camera views. Given a source view a and a target view b, calibrated camera geometry defines a set of geometrically plausible source positions for each target position in view b. We refer to this retrieval direction as $a \to b$

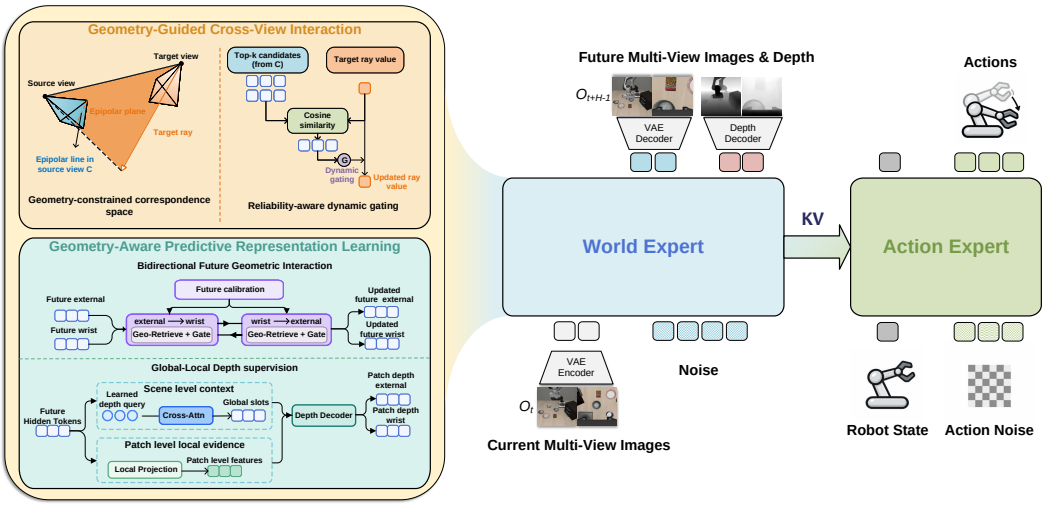


Figure 1: **Overview of CoView-WAM.** Geometry-guided cross-view interaction enhances current representations through epipolar-constrained retrieval and reliability-aware gating. During training, bidirectional future geometric interaction and global–local depth supervision structure predictive representations. The action expert generates actions conditioned on the current world-expert cache and robot state, without decoding future images or depth maps at inference.

and omit the direction index when it is clear from context. Let p_i^v denote the homogeneous image coordinate of position i in view v, with intrinsic matrix K_t^v, world-frame rotation R_t^v, and camera center o_t^v. Its world-frame viewing direction $\tilde{r}_i^v$ and the normal $\hat{n}_i$ of the epipolar plane defined by the target ray and camera baseline are as follows:

$$\tilde{r}_i^v = R_t^v (K_t^v)^{-1} p_i^v, \qquad \hat{n}_i = \frac{(o_t^a - o_t^b) \times \tilde{r}_i^b}{\|(o_t^a - o_t^b) \times \tilde{r}_i^b\|_2}. \tag{1}$$

Meanwhile, we assume calibrated camera pairs with well-defined epipolar geometry. For each source position j, let $\hat{r}_j^a = \tilde{r}_j^a / \|\tilde{r}_j^a\|_2$. We measure its deviation from the target epipolar plane and select the k closest source positions as follows:

$$\delta_{ij} = |\hat{n}_i^\top \hat{r}_j^a|, \qquad \mathcal{C}_i = \mathrm{argTopK}_{j \in \{1,\dots,N\}}(-\delta_{ij}, k). \tag{2}$$

The resulting set $\mathcal{C}_i$ restricts semantic retrieval to geometrically plausible source regions, while backbone value similarities determine how candidate features are aggregated. Let $V_{\ell,h}^v \in \mathbb{R}^{N \times d_h}$ denote the value representations of view v at Transformer layer ℓ and attention head h. For each candidate $j \in \mathcal{C}_i$, we compute the cosine similarity as follows:

$$\begin{aligned} s_{\ell,h,ij} &= \cos(V_{\ell,h,i}^b, V_{\ell,h,j}^a), \\ \alpha_{\ell,h,ij} &= \frac{\exp(s_{\ell,h,ij}/\tau_s)}{\sum_{j' \in \mathcal{C}_i} \exp(s_{\ell,h,ij'}/\tau_s)}, \\ R_{\ell,h,i} &= \sum_{j \in \mathcal{C}_i} \alpha_{\ell,h,ij} V_{\ell,h,j}^a, \end{aligned} \tag{3}$$

where τ_s is the retrieval temperature and $R_{\ell,h,i}$ denotes the cross-view feature retrieved from source view a for target position i in view b through similarity-weighted aggregation over $\mathcal{C}_i$. The geometry-derived candidate sets $\mathcal{C}_i$ depend only on camera calibration and are therefore shared across Transformer layers and attention heads, while the retrieval weights $\alpha_{\ell,h,ij}$ are computed independently from the value representations at each layer and head, allowing semantic retrieval to adapt to the evolving representation space of the backbone.

3.3 Reliability-Aware Gated Feature Interaction

Geometry-guided candidate selection does not guarantee reliable cross-view retrieval, as occlusion, repeated appearance, limited view overlap, and large viewpoint changes can still produce ambiguous

or mismatched features. We therefore introduce a reliability-aware gating mechanism that adaptively controls how much retrieved cross-view information is injected into the target representation. For each target position, we define a detached retrieval residual and update the target value as

$$\Delta V_{\ell,h,i} = \mathrm{sg}(R_{\ell,h,i} - V^b_{\ell,h,i}), \qquad \widetilde{V}^b_{\ell,h,i} = V^b_{\ell,h,i} + g_\ell \Delta V_{\ell,h,i}. \tag{4}$$

Here, $\mathrm{sg}(\cdot)$ denotes stop-gradient and $g_\ell \in [0, g_{\max}]$ is one scalar gate per sample and Transformer layer, determining the intensity of cross-view residual injection. We parameterize the gate with two lightweight branches. The gate takes global summaries of the target and retrieved features, retrieval statistics capturing semantic and geometric reliability, and a learnable layer embedding as inputs. The need branch models the potential benefit of complementary cross-view information, while the trust branch models the reliability of the retrieved features, producing ν_ℓ and ζ_ℓ, respectively.

$$g_\ell = g_{\max}\sigma\big[b_g + s_g\big(\nu_\ell + \eta\tanh(\zeta_\ell)\big)\big], \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function, $b_g = \mathrm{logit}(g_{\mathrm{init}}/g_{\max})$ initializes the gate around g_{init}, s_g controls the logit scale, and η balances the contribution of the trust branch. Detailed gate inputs are provided in Appendix A.1. For current observations, we use the external view as the source and the wrist view as the target. Only the wrist-view values used for action conditioning are updated, while the native video-attention path retains the original values. The resulting geometry-enhanced wrist representations are directly consumed by the action branch. To stabilize optimization, we detach the retrieval residual and the representation statistics supplied to the gate. This prevents gradients from flowing through the internal retrieval computation, while the original target values remain trainable through the direct path in Eq. 4 and the gate parameters are still optimized through the interaction path. We use independent gate parameters for the current interaction and the two future interaction directions to accommodate their different feature distributions.

3.4 Geometry-Aware Predictive Representation Learning

During predictive learning, future representations from different views should preserve coherent spatial structure of the same future scene. We therefore combine bidirectional geometry-guided interaction with auxiliary depth supervision to encourage 3D-aware cross-view geometric consistency.

Bidirectional geometric interaction. The wrist camera moves together with the robot arm, and therefore the geometric relationship between external and wrist views changes over time. During training, we use the transformation relationship $\mathcal{G}_{t_f}$ at the time of predicted future image to construct candidate sets and calculate gate weights for both external-to-wrist and wrist-to-external interaction. Specifically, let $V^{F,v}_{\ell,h,i,\tau_v}$ denote the latent representation of future position i in view v at transformer layer ℓ, attention head h, and normalized visual denoising timestep τ_v. For either interaction direction from view a to view b, the target future value is updated as follows:

$$\widetilde{V}^{F,b}_{\ell,h,i,\tau_v} = V^{F,b}_{\ell,h,i,\tau_v} + g^{F,a\to b}_{\ell,\tau_v}\,\mathrm{sg}\Big(R^{F,a\to b}_{\ell,h,i,\tau_v} - V^{F,b}_{\ell,h,i,\tau_v}\Big). \tag{6}$$

Because the reliability of future representations varies with the visual denoising stage, the future gate trust branch additionally receives the timestep encoding:

$$\psi_{\tau_v} = [\tau_v, \sin(\pi\tau_v), \cos(\pi\tau_v)]. \tag{7}$$

This allows the interaction strength to adjust according to the noise level. To avoid order-dependent interaction, both directions compute retrievals from the original future values before either update is applied. The corresponding target-view future query outputs are then recomputed using the updated values, with full attention details provided in Appendix A.2.

Auxiliary depth supervision. Epipolar interaction constrains cross-view correspondence in the image plane but does not resolve geometric ambiguity along the viewing ray. We therefore introduce an auxiliary depth prediction objective to provide explicit 3D-aware spatial supervision for future visual representations. Let $H_{\tau_v} \in \mathbb{R}^{2N\times d}$ denote the final hidden representations of the two future views at visual timestep τ_v. We use a set of learned depth queries $Q^D \in \mathbb{R}^{M\times d_q}$ to aggregate context across the two future views:

$$X = \phi_{\mathrm{ctx}}(H_{\tau_v}) + E_{\mathrm{pos}}, \qquad Z^D = \mathrm{LN}\left[Q^D + \mathrm{MHA}(Q^D, X, X)\right]. \tag{8}$$

Here, ϕ_{ctx} normalizes and projects the future hidden representations, while E_{pos} combines spatial row, within-view column, and camera-view embeddings. The resulting query features Z^D aggregate

cross-view context that is subsequently constrained by the depth prediction objective. For each spatial position, a position-dependent readout aggregates the depth-query features and combines them with a local projection of the corresponding future hidden representation and its positional embedding. A lightweight decoder then predicts the standardized depth target $\widehat{y}_i^v$. Details of the readout and decoder are provided in Appendix A.3. The auxiliary depth objective combines masked Smooth-L_1 regression with a spatial-difference loss:

$$\mathcal{L}_{\text{depth}} = \tfrac{1}{2}\left(\mathcal{L}_{\text{reg}}^{e} + \mathcal{L}_{\text{reg}}^{w}\right) + \lambda_{\text{edge}}\mathcal{L}_{\text{edge}}, \tag{9}$$

where the regression term follows the weighting of the original visual objective. $\mathcal{L}_{\text{edge}}$ matches horizontal and vertical depth differences between adjacent valid patches, encouraging preservation of local geometric structure. The depth decoder is used only during training to regularize future visual representations and is discarded at inference.

3.5 Training and Inference

We employ the image and action flow-matching objectives used in ImageWAM (Zhang et al., 2026) and add auxiliary depth supervision:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{action}}\mathcal{L}_{\text{action}} + \lambda_{\text{image}}\mathcal{L}_{\text{image}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}}. \tag{10}$$

The original scheduler targets and timestep weights are retained. Candidate sets are constructed analytically from camera calibration, requiring neither dense cross-view correspondence annotations nor a separate geometric matching objective. Appendix A lists all module hyperparameters. Current and future pathways are processed separately. Current cross-view interaction enhances the conditioned cache for the action branch, whereas future interaction and depth supervision operate only on predictive representations. Thus, predictive supervision affects action conditioning through shared parameters without transmitting sample-specific future information to action prediction. At inference, only current observations and camera calibration parameters are required. The external-to-wrist geometry-enhanced conditioning cache is reused throughout action denoising, while future interaction and depth prediction are removed.

4 Experiments

4.1 Experimental Setup

Benchmarks and evaluation protocols. On LIBERO (Liu et al., 2023a), we evaluate all 40 tasks across Spatial, Object, Goal, and Long with 50 rollouts per task, following ImageWAM's evaluation protocol, and report suite-wise and overall success rates. On CALVIN (Mees et al., 2022), we use the standard ABC→D protocol and evaluate the 1,000 official five-task sequences, reporting success rates for completing one to five consecutive tasks and the average completed sequence length $\mathbb{E}[L] = \sum_{k=1}^{5} \Pr(L \geq k)$. We follow ImageWAM (Zhang et al., 2026) to use the same initialization, data, optimizer, training budget, and checkpoint-selection protocol for fair comparison. We additionally report external baselines on LIBERO and CALVIN using their original training setups; result sources and CALVIN pretraining details are provided in Appendix C. CoView-WAM uses current camera calibration at inference and additionally uses future calibration and depth targets during training; other training and execution settings are provided in Appendix B.

Real-world platform and protocol. We use a Franka arm with an XHand dexterous hand, two external cameras, and one wrist camera, with all three views provided to the policy. For each of five tasks, we collect 50 teleoperated demonstrations and compare π_0, $\pi_{0.5}$, ImageWAM, and CoView-WAM using matched demonstrations and training steps. ImageWAM and CoView-WAM share the same foundation-model initialization, while CoView-WAM additionally uses offline Depth Anything 3 (Lin et al., 2026) depth targets during training. Each task is evaluated over five shared initial configurations with 20 trials each, totaling 100 trials. Task definitions are provided in Appendix B.1.

4.2 Main Results

Manipulation success on LIBERO. CoView-WAM improves overall success from 97.55% to 99.20% over the matched ImageWAM baseline, with gains on Spatial, Goal, and Long while maintaining near-saturated performance on Object. The largest gain occurs on Long, where success

Table 1: LIBERO success rates (%). †: external results with independent training setups. ImageWAM is our reproduced baseline. Best values are bold.

Method	Spatial	Object	Goal	Long	Overall
OpenVLA (Kim et al., 2024)†	84.7	88.4	79.2	53.7	76.5
π_0 (Black et al., 2025b)†	96.8	98.8	95.8	85.2	94.1
$\pi_{0.5}$ (Black et al., 2025a)†	**98.8**	98.2	98.0	92.4	96.9
LingBot-VA (Li et al., 2026)†	98.5	99.6	97.2	98.5	98.5
Motus (Bi et al., 2026)†	96.8	99.8	96.6	97.6	97.7
Fast-WAM (Yuan et al., 2026)†	98.2	**100.0**	97.0	95.2	97.6
ImageWAM (Zhang et al., 2026)	97.0	**100.0**	97.0	96.2	97.55
CoView-WAM	**98.8**	99.8	**99.0**	**99.2**	**99.20**

Table 2: CALVIN ABC→D results. Columns 1–5: consecutive-task success rates (%). †: external results; ‡: official VPP ablations. P.T.: additional manipulation-domain pretraining, excluding generic initialization (Appendix C).

Method	P.T.	1	2	3	4	5	Avg. length
GR-1 (Wu et al., 2024)†	✓	85.4	71.2	59.6	49.7	40.1	3.06
RoboUniView (Liu et al., 2024)†	✓	94.2	84.2	73.4	62.2	50.7	3.647
VPP (Hu et al., 2025)†	✓	**96.5**	**90.9**	**86.6**	**82.0**	**76.9**	**4.33**
w/o Internet data‡	✓	–	–	–	–	–	3.97
w/o CALVIN video‡	✓	–	–	–	–	–	3.31
ImageWAM (Zhang et al., 2026) (reproduced)	×	91.7	82.2	73.9	66.6	56.6	3.710
CoView-WAM	×	93.0	86.6	79.9	72.8	66.4	**3.987**

increases from 96.2% to 99.2%. CoView-WAM is also competitive with the externally reported results in Table 1, which use independent training setups.

Long-horizon execution on CALVIN. Under the same training protocol, CoView-WAM increases the average completed sequence length from 3.710 to 3.987 over ImageWAM, while the full five-task completion rate improves from 56.6% to 66.4%. Full VPP reports a higher average length of 4.33 with video pretraining on both Internet and CALVIN data. Notably, CoView-WAM numerically exceeds VPP's official ablations without Internet data (3.97) and without CALVIN video pretraining (3.31), despite using no additional manipulation-domain pretraining before CALVIN policy training.

Real-world manipulation. Across five real-world tasks, CoView-WAM improves overall success from 57.2% to 73.8% over the matched ImageWAM baseline. The largest gains occur on occluded fruit picking, pouring, and cup placement, where successful manipulation depends strongly on cross-view visibility, relative object geometry, and precise placement. CoView-WAM also outperforms $\pi_{0.5}$ overall under the same task-specific demonstration budgets, although the main controlled comparison remains ImageWAM.

4.3 Component Ablations

We ablate current cross-view interaction (C), future geometric interaction (F), and auxiliary depth supervision (D) on LIBERO. Starting from the reproduced ImageWAM baseline, we first evaluate current interaction alone and then add F and D individually or jointly, as shown in Table 4.

Current cross-view interaction. Adding current geometric interaction improves overall success from 97.55% to 98.65%, showing that explicitly organizing cross-view information in the action-conditioning cache is beneficial even without additional predictive supervision.

Joint predictive supervision. Adding future geometric interaction or depth supervision individually yields 98.50% and 98.60%, respectively, neither exceeding the 98.65% Current-only model. In contrast, combining both reaches 99.20%, a further 0.55-point gain over Current-only, with the

Table 3: Real-world success rates (%) with matched demonstrations and training steps per task. Overall averages five tasks; best results are bold.

Method	Cup stacking	Cup placement	Cup pouring	Orange placement	Occluded fruit picking	Overall
π_0 (Black et al., 2025b)	52	23	41	84	44	48.8
$\pi_{0.5}$ (Black et al., 2025a)	59	29	66	86	52	58.4
ImageWAM (Zhang et al., 2026)	63	34	47	86	56	57.2
CoView-WAM	**77**	**52**	**68**	**93**	**79**	**73.8**

Table 4: LIBERO component ablations (%). C: current cross-view interaction; F: future geometric interaction; D: depth supervision. Best values are bold.

Configuration	C	F	D	Spatial	Object	Goal	Long	Overall
Baseline	—	—	—	97.0	**100.0**	97.0	96.2	97.55
Current only	✓	—	—	**99.0**	99.8	98.8	97.0	98.65
Current + future	✓	✓	—	98.4	**100.0**	**99.0**	96.6	98.50
Current + depth	✓	—	✓	**99.0**	99.8	98.4	97.2	98.60
Full model	✓	✓	✓	98.8	99.8	**99.0**	**99.2**	**99.20**

largest improvement on LIBERO-10. This result supports using future geometric interaction and depth supervision jointly rather than as independent auxiliary objectives.

4.4 Analysis

Geometric candidate selection. As shown in Table 5, we compare Geometry Top-32 with Global Semantic Top-32 under the Current-only configuration, keeping the candidate count, semantic aggregation, gating, and training setup fixed while changing only how candidates are selected. Geometry Top-32 reaches 98.65% overall success versus 98.20% for global semantic selection, supporting the use of epipolar geometry to restrict the candidate space before semantic retrieval.

Cross-view retrieval visualization. Figure 2 shows that geometry-based selection can focus on candidates within epipolar-consistent regions, while the subsequent semantic weights determine which candidates are most related to retrieval. In contrast, global semantic selection produces more spatially dispersed candidates and can retrieve visually similar but geometrically unrelated regions.

Camera-view configurations. To test whether the gains arise simply from using multiple views, we separately train ImageWAM with External-only, Wrist-only, and External+Wrist inputs under matched settings. The native dual-view model achieves 97.55% overall success, slightly below Wrist-only at 97.80%, showing that adding an external view does not consistently improve performance under implicit fusion. In contrast, CoView-WAM reaches 99.20%, supporting the importance of explicitly organizing cross-view information rather than merely increasing the number of views.

Table 5: Candidate selection under Current only, measured by success rate (%).

Candidate strategy	Spatial	Object	Goal	Long	Overall
Geometry Top-32	99.0	99.8	98.8	97.0	98.65
Global semantic Top-32	98.0	99.4	98.8	96.6	98.20

Table 6: ImageWAM (Zhang et al., 2026) success rates (%) with separately trained camera-view configurations.

View configuration	Spatial	Object	Goal	Long	Overall
External only	96.6	98.8	96.2	92.2	95.95
Wrist only	97.6	99.6	98.2	95.8	97.80
External + wrist	97.0	100.0	97.0	96.2	97.55



Figure 2: Cross-view retrieval comparison. Columns show the wrist-view query; the 32 external-view candidates selected by epipolar deviation; their retrieval weights, with darker colors indicating larger weights; and the 32 candidates selected globally by representation similarity. The fourth column comes from the separately trained Global semantic Top-32 model. Each row corresponds to a different sample, with the same observations used across all four columns.

5 Conclusion

We propose CoView-WAM, a geometry-aware World-Action Model that explicitly incorporates calibrated cross-view geometry into current action conditioning and predictive representation learning. Geometry-constrained retrieval and reliability-aware gated interaction organize complementary information across current views, while bidirectional future interaction and auxiliary depth supervision jointly encourage coherent spatial structure in predictive representations. Across LIBERO, CALVIN, and real-world manipulation, CoView-WAM consistently improves over the matched ImageWAM baseline; ablations further show that current geometric interaction provides the main gain and that joint future geometry and depth supervision brings additional improvement. Future interaction and depth prediction are training-only, so inference requires only current observations and camera calibration. Overall, our results highlight explicit cross-view geometry as a practical inductive bias for multi-view world-action modeling.

AI USE STATEMENT

Generative AI tools, including ChatGPT, were used to assist with literature search and summarization, language editing and drafting, and feedback on the presentation of the methodology and experimental design. All AI-assisted literature-related content was checked against the original sources, and all methodological descriptions, experimental results, analyses, and scientific claims were reviewed and verified by the authors. Generative AI was not used to generate experimental data or evaluation results. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies robotic manipulation and does not involve experiments on human subjects or the use of personally identifiable information. Real-world demonstrations are collected solely for training and evaluating robotic manipulation policies. We are not aware of additional ethical concerns specific to this work beyond the general safety considerations associated with robotic manipulation systems.

REPRODUCIBILITY STATEMENT

We provide the model formulation and training objectives in Section 3, with detailed gate parameterization, future interaction, depth supervision, and hyperparameters in Appendix A. Training and execution settings for LIBERO, CALVIN, and the real-world experiments are provided in Appendix B, including optimization settings, training budgets, action horizons, and evaluation protocols. CALVIN baseline sources and pretraining configurations are documented in Appendix C. These details are intended to support reproduction of both the proposed method and the reported evaluations. Anonymous source code and configuration files are provided in the supplementary material.

REFERENCES

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35101–35113, 2026.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025b. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Anthony Brohan, Noah Brown, Justice Carbajal, et al. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems (RSS)*, 2023. doi: 10.15607/RSS.2023.XIX.025. URL https://www.roboticsproceedings.org/rss19/p025.html.

Jake Bruce, Michael J. Dennis, Ashley Edwards, et al. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Jun Cen, Chaohui Yu, Hangjie Yuan, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint*, 2025.

Chilam Cheang, Guangzeng Chen, Ya Jing, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

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. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1d5b9233ad716a43be5c0d3023cb82d0-Abstract-Conference.html.

Théophile Gervet, Xian Zhou, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. RVT: Robotic view transformer for 3D object manipulation. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 694–710. PMLR, 2023. URL https://proceedings.mlr.press/v229/goyal23a.html.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Lei Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

Hao He, Yinghao Xu, Yuwei Guo, et al. Cameractrl: Enabling camera control for video diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025.

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. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24328–24346, 2025. URL https://arxiv.org/abs/2412.14803.

Yuhang Huang, Xuan Lv, Junyan Xu, Zhiyuan Yu, Jiazhao Zhang, Ruizhen Hu, Wancheng Feng, Shilong Zou, Hewen Xiao, Ziqiao Zhou, Kaiyun Huang, Zhiyu Peng, Juzhan Xu, Hang Zhao, Chenyang Zhu, Renjiao Yi, Yifei Huang, Douhui Wu, Yan Zhang, Kexu Cheng, Chunhe Song, Yunzhi Xue, Xiuhong Zhang, Leitao Guo, Yunji Chen, Bin Wu, Haibin Yu, and Kai Xu. PAIWorld: A 3d-consistent world foundation model for robotic manipulation. *arXiv preprint arXiv:2606.18375*, 2026. URL https://arxiv.org/abs/2606.18375.

Joel Jang, Seonghyeon Ye, Zongyu Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2024. URL https://proceedings.mlr.press/v270/kim25c.html.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 71–91, 2024. doi: 10.1007/978-3-031-73220-1_5.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Proceedings of Robotics: Science and Systems*, 2026. doi: 10.15607/RSS.2026.XXII.016. URL https://roboticsproceedings.org/rss22/p016.html.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yirunib8l8.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Fanfan Liu, Feng Yan, Liming Zheng, Chengjian Feng, Yiyang Huang, and Lin Ma. RoboUniView: Visual-language model with unified view representation for robotic manipulation. *arXiv preprint arXiv:2406.18977*, 2024. URL https://arxiv.org/abs/2406.18977.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9264–9275, 2023b.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022. doi: 10.1109/LRA.2022.3180108.

Nisarga Nilavadi, Ralf Römer, Moritz Reuss, Michael Krawez, Tobias Jülg, Angela P. Schoellig, Rudolf Lioutikov, and Wolfram Burgard. DUET-DINO: Simultaneous cross-view world modeling for latent planning in robot manipulation. *arXiv preprint arXiv:2609.10506*, 2026. URL https://arxiv.org/abs/2609.10506.

NVIDIA. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025a.

NVIDIA. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025b.

Zezhong Qian, Xiaowei Chi, Yuming Li, Shizun Wang, Zhiyuan Qin, Xiaozhu Ju, Sirui Han, and Shanghang Zhang. WristWorld: Generating wrist-views via 4d world models for robotic manipulation. *arXiv preprint arXiv:2510.07313*, 2025. URL https://arxiv.org/abs/2510.07313.

Delin Qu, Haoming Song, Qizhi Chen, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint*, 2025.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *International Conference on Learning Representations (ICLR)*, 2024.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-Actor: A multi-task transformer for robotic manipulation. In *Proceedings of the Conference on Robot Learning (CoRL)*, volume 205, pp. 785–799, 2022.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709, 2024.

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*, 2024. URL https://arxiv.org/abs/2312.13139.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Proceedings of the Conference on Robot Learning (CoRL)*, volume 205, pp. 2226–2240, 2022.

Jiarui Yang, Yehao Lu, Yuning Su, Yufeng Xie, Yu Zhong, Haiyu Lan, Tianjing Hao, Kaixiang Lu, Peiwen Lin, Chuang Wang, Enyu Li, and Junwei Liang. Multi-view unified camera fields: Geometry-shaped action-facing representations for RGB-only multi-camera VLA policies. *arXiv preprint arXiv:2608.01826*, 2026. URL https://arxiv.org/abs/2608.01826.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. URL https://arxiv.org/abs/2602.15922.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. GNFactor: Multi-task real robot learning with generalizable neural feature fields. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 284–301. PMLR, 2023. URL https://proceedings.mlr.press/v229/ze23a.html.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Robotics: Science and Systems (RSS)*, 2024. doi: 10.15607/RSS.2024.XX.067. URL https://www.roboticsproceedings.org/rss20/p067.html.

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*, 2026. URL https://arxiv.org/abs/2606.19531.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

A Implementation Details of CoView-WAM

A.1 Detailed Gate Parameterization

For each sample and layer, the gate produces one scalar shared across all spatial positions and attention heads of the target view. It uses sample-level representation summaries and retrieval statistics to control the cross-view interaction strength at that layer. We first average the target and retrieved representations over spatial positions and then concatenate the attention heads:

$$\begin{aligned} \overline{V}_\ell^b &= \mathrm{Concat}_h\left(\frac{1}{N}\sum_i V_{\ell,h,i}^b\right), \\ \overline{R}_\ell &= \mathrm{Concat}_h\left(\frac{1}{N}\sum_i R_{\ell,h,i}\right). \end{aligned} \tag{11}$$

The gate also receives sample-level retrieval statistics u_ℓ: the maximum retrieval weight, normalized entropy, the margin between the two largest retrieval weights, retrieved–target cosine similarity, the residual-to-target RMS ratio, and the mean and maximum geometric deviations. These statistics describe retrieval concentration, representation agreement, and geometric compatibility.

Given a learned layer embedding $\mathbf{e}_\ell$, two lightweight branches model the need for complementary information and the trustworthiness of the retrieval:

$$\begin{aligned}\nu_\ell &= f_{\text{need}}\big(P_V\,\text{sg}(\overline{V}_\ell^b), \mathbf{e}_\ell\big),\\ \zeta_\ell &= f_{\text{trust}}\big(P_R\,\text{sg}(\overline{R}_\ell), \text{sg}(u_\ell), \mathbf{e}_\ell\big),\end{aligned} \tag{12}$$

where P_V and P_R are learnable projections. These branches drive the gate g_ℓ defined in Eq. 5.

A.2 Future Interaction Attention Update

For future bidirectional interaction, both external-to-wrist and wrist-to-external retrievals and residuals are computed from the original, unmodified future values before either update is applied, avoiding order-dependent information propagation within a Transformer layer. For an interaction direction $a \to b$, let $\mathbf{V}_\ell^b$ denote the full value tensor in which only the future positions of target view b are replaced by their geometry-enhanced values from Eq. 6, and let $\mathbf{Q}_\ell^b$ denote the corresponding future queries. We recompute their attention outputs as

$$\widetilde{O}_\ell^b = \text{Attn}(\mathbf{Q}_\ell^b, \mathbf{K}_\ell, \mathbf{V}_\ell^b; \mathsf{M}^b), \tag{13}$$

where $\mathbf{K}_\ell$ retains the original keys and M^b contains the corresponding rows of the original attention mask. External-to-wrist interaction therefore recomputes only the future wrist-query outputs, while wrist-to-external interaction recomputes only the future external-query outputs. The two directions use independently updated value tensors and are supervised through the future visual and auxiliary depth objectives.

A.3 Depth Readout and Loss Definitions

Let H_{τ_v} denote the concatenated future tokens from both views. The readout combines a global path that aggregates context across positions with a local path that retains evidence at each position. The global path uses M learned future depth queries $Q^D \in \mathbb{R}^{M\times d_q}$ to read all future visual tokens through cross-attention in Eq. 8. Here, ϕ_{ctx} consists of LayerNorm followed by a linear projection to d_q dimensions. The positional embedding E_{pos} is the sum of row, within-view column, and view embeddings; its component at position i in view v is denoted by $\mathbf{e}_i^v$. The query outputs Z^D are read at each prediction position using position-dependent weights:

$$f_{\text{global},i}^v = \text{softmax}\left(\frac{\mathbf{e}_i^v[\kappa(Z^D)]^\top}{\sqrt{d_q}}\right) Z^D, \tag{14}$$

where $\kappa(\cdot)$ is a learnable affine mapping. Let $H_{\tau_v,i}^v$ denote the future hidden token at position i in view v. The local path directly transforms that token without aggregating across positions:

$$f_{\text{local},i}^v = \phi_{\text{local}}(H_{\tau_v,i}^v), \tag{15}$$

where ϕ_{local} uses a separate normalization and linear projection. A lightweight MLP decoder D_ψ predicts depth in the standardized target space from the sum of the two paths and the positional embedding:

$$\widehat{y}_i^v = D_\psi\big(f_{\text{global},i}^v + f_{\text{local},i}^v + \mathbf{e}_i^v\big)\,. \tag{16}$$

Depth targets are generated offline, aligned with the future endpoint and the visual token grid, and transformed using fixed per-camera clipped-log standardization. We denote the resulting target and validity mask by $y_{q,i}^v$ and $m_{q,i}^v$, respectively, where q indexes samples in the batch. Predictions $\widehat{y}_{q,i}^v$ lie in the same standardized space rather than representing depth directly in meters. Neither the intermediate readout features nor the predicted depth values are written back to the video or action backbone. The auxiliary loss propagates gradients to the future hidden representations and their upstream computation.

For each view, the regression loss uses the nonnegative scheduler weight ω_q associated with the sample's video noise timestep:

$$\mathcal{L}_{\text{reg}}^v = \frac{\sum_{q,i}\omega_q m_{q,i}^v \rho_\beta(\widehat{y}_{q,i}^v - y_{q,i}^v)}{\sum_{q,i}\omega_q m_{q,i}^v}, \tag{17}$$

Table 7: Hyperparameters of the geometric interaction and auxiliary depth modules.

Symbol	Description	Value
—	Video expert transformer blocks (double / single)	25 (5 / 20)
k	Geometric candidates per direction	32
τ_s	Semantic retrieval temperature	0.1
λ_{action}	Action objective coefficient	1.0
λ_{image}	Image objective coefficient	0.5
λ_{depth}	Auxiliary depth objective coefficient	0.02
λ_{edge}	Spatial-difference coefficient within the depth objective	0.1
g_{init}	Initial gate value	0.01
$g_{\max}$	Gate upper bound	0.1
s_g	Gate logit scale	4
η	Trust-branch contribution	0.5
M	Learned future depth queries	8
d_q	Depth query dimension	256
—	Depth cross-attention layers / heads	1 / 4
β	Smooth-L_1 threshold for regression and spatial differences	0.1

where ρ_β is the Smooth-L_1 loss. This term is defined as zero when the total weighted validity is zero.

In addition to patch-wise regression, we supervise local spatial changes in standardized log-depth by matching depth differences between adjacent patches. Let $\mathcal{E}_{v,d}$ contain adjacent patch pairs within the batch for which both positions are valid, where $d \in \{\text{hor}, \text{ver}\}$ denotes the horizontal or vertical direction. We define

$$\mathcal{L}_{\text{edge}} = \frac{1}{|\mathcal{S}|} \sum_{(v,d)\in\mathcal{S}} \frac{1}{|\mathcal{E}_{v,d}|} \sum_{(q,i,j)\in\mathcal{E}_{v,d}} \rho_\beta\left[(\widehat{y}^v_{q,j} - \widehat{y}^v_{q,i}) - (y^v_{q,j} - y^v_{q,i})\right], \tag{18}$$

where $\mathcal{S} = \{(v,d) : |\mathcal{E}_{v,d}| > 0\}$. The loss is defined as zero when $\mathcal{S}$ is empty. This spatial-difference loss does not use video timestep weights. The complete objective is given in Eq. 9.

A.4 Hyperparameters

Table 7 lists the module hyperparameters used in our training and evaluation configuration.

The spatial-difference loss has an effective coefficient of 0.002 in the total objective, obtained by multiplying $\lambda_{\text{depth}} = 0.02$ and $\lambda_{\text{edge}} = 0.1$.

A.5 Gate Architecture and Interaction Layers

Cross-view value updates are applied at all 25 transformer blocks of the video expert. Layer identity is encoded by a learned layer embedding. The target and retrieved summaries each have dimension $d = 24 \times 128 = 3072$, obtained by spatial averaging followed by concatenation of all attention heads. Separate linear projections map these summaries to 64 dimensions before concatenation with the other gate inputs.

The need branch has an input dimension of $64 + 16 = 80$, and the current trust branch has an input dimension of $64 + 7 + 16 = 87$. Here, 7 is the number of retrieval statistics and 16 is the layer embedding dimension. The future trust branch additionally receives a three-dimensional noise timestep encoding, giving an input dimension of $64 + 7 + 3 + 16 = 90$. Each branch uses the architecture

$$\text{LayerNorm} \to \text{Linear}(\cdot, 64) \to \text{SiLU} \to \text{Linear}(64, 1). \tag{19}$$

The weights and bias of the final linear layer are initialized to zero, so $g = g_{\text{init}}$ at initialization.

The current gate, the future external-to-wrist gate, and the future wrist-to-external gate have three independent parameter sets. Parameters within each instance are shared across layers.

Table 8: Image inputs and visual token grid.

Item	Value
Image resolution per view	224×224
Two-view concatenation	Horizontal, 224×448
Patch size	16×16
Positions per view, N	196 (14×14)
Visual tokens across both views, $2N$	392

Table 9: Depth target preprocessing configuration.

Item	Value
Rendered depth resolution per view	128×128
Offline storage precision	float16
Statistics sampling	Per-camera priority reservoir
Reservoir capacity / random seed	2×10^6 / 42
Clipping percentiles	0.5 / 99.5
Target resolution	14×14
Downsampling	Area interpolation

A.6 Inputs and Spatial Grid

A.7 Depth Target Generation and Standardization

During offline preprocessing, the 0.5th and 99.5th percentiles of each camera's reservoir samples define the clipping bounds $\mathrm{clip}_{\min}$ and $\mathrm{clip}_{\max}$. The mean $\log_\mathrm{mean}$ and standard deviation $\log_\mathrm{std}$ are computed from

$$\log(\mathrm{clip}(d, \mathrm{clip}_{\min}, \mathrm{clip}_{\max}))\,. \tag{20}$$

These four constants are stored separately for each camera and remain fixed during training.

Training uses the corresponding camera's clipping and standardization constants. A depth pixel is valid if it is finite and positive. Invalid pixels are initially replaced with $\mathrm{clip}_{\min}$. Let z denote standardized pixel depth, m its validity mask, and $\mathcal{A}$ the area interpolation operator. We compute

$$a = \mathcal{A}(mz), \qquad w = \mathcal{A}(m). \tag{21}$$

At positions with $w > 0$, the patch target is the validity-normalized average $y = a/w$, with numerical protection applied to the denominator in the implementation. Remaining positions are set to zero, and the patch validity mask is $\mathbf{1}[w > 0]$. Invalid pixels therefore do not contribute to the average over valid depth values.

A.8 Scheduler Weights

The weight ω_q is supplied by the video training scheduler at the noise timestep t_q of sample q. It is the same weight tensor used by the video objective:

$$\begin{aligned} y(t) &= \exp\left[-2\left(\frac{t-500}{1000}\right)^2\right], \\ \omega_q &= \frac{y(t_q) - y_{\min}}{\overline{y - y_{\min}} + \epsilon}. \end{aligned} \tag{22}$$

Here, $\epsilon = 10^{-10}$ provides numerical protection, and $t \in [0, 1000]$. The constants $y_{\min} \approx 0.6065307$ and $\overline{y - y_{\min}} \approx 0.1894955$ are precomputed by evaluating the scheduler on an evenly spaced grid in u, using its mapping $u \mapsto t$ with shift 5.0. They remain fixed during training. The spatial-difference loss $\mathcal{L}_{\mathrm{edge}}$ does not use these weights.

Table 10: Training and execution settings for ImageWAM and CoView-WAM. Real-world optimizer steps vary with the amount of data per task.

Setting	LIBERO	CALVIN	Real world
Training budget	34,720 optimizer steps	5 epochs	30 epochs; approximately 3,510–9,960 optimizer steps
Number of GPUs	8	8	8
Batch size per GPU	2	10	4
Gradient accumulation	5	1	4
Global batch size	80	80	128
Pretraining initialization	Unified foundation-model weight initialization	Foundation-model weight initialization consistent with LIBERO	Foundation-model weight initialization consistent with LIBERO and CALVIN
Action horizon	16	16	16
Actions per replanning cycle	12	12	12
Evaluation scale	50 rollouts per task; 2,000 total	1,000 official five-task sequences	100 trials per task

Table 11: Real-world manipulation tasks. Each task uses 50 demonstrations and 100 evaluation trials across five initial configurations.

Task	Objective
Cup stacking	Stack cups in sequence.
Cup placement	Place three cups of different sizes into their corresponding holders.
Pouring	Transfer the contents of one cup into another.
Orange placement	Retrieve an orange from a basket and place it in a cup.
Occluded fruit picking	Pick an apple and an orange from opposite sides outside the basket and place both inside it.

B Training and Execution Settings

The settings in Table 10 apply to ImageWAM and CoView-WAM, which use the same foundation-model weight initialization across LIBERO, CALVIN, and the real-world experiments. Module hyperparameters are provided in the Method appendix. All experiments use AdamW with an initial learning rate of 1×10^{-4} and a cosine schedule. Gradient norms are clipped at 1.0. Training uses BF16, except for the real-world pouring task, which uses `mixed_precision=no` with FP32 parameters.

B.1 Real-World Task Definitions

We evaluate five real-world manipulation tasks designed to cover stacking, precise placement, relative-pose coordination, object retrieval, and occlusion.

Table 12 summarizes the main observed failures. In cup stacking and occluded fruit picking, failures are especially common when target visibility is limited. Cup placement and orange placement primarily exhibit placement errors. Pouring requires coordinating the relative positions and orientations of the two cups, and complete transfer of their contents remains challenging.

C CALVIN Baselines and Pretraining Sources

The P.T. column in Table 2 identifies an additional manipulation-domain pretraining stage, such as manipulation video prediction, 3D occupancy representation learning, or policy pretraining. It distinguishes these stages from generic foundation-model initialization and from the separate question of whether data outside CALVIN are used. ImageWAM and CoView-WAM initialize from a pre-

Table 12: Observed failure modes in the five real-world tasks.

Task	Observed failure
Cup stacking	Failure to nest a cup inside a larger one, with localization errors more frequent under occlusion.
Cup placement	Failure to place a cup accurately into its corresponding holder.
Pouring	Incomplete transfer of contents into the target cup.
Orange placement	Failure to place the orange accurately inside the cup.
Occluded fruit picking	Lower grasp success for the target on the occluded side.

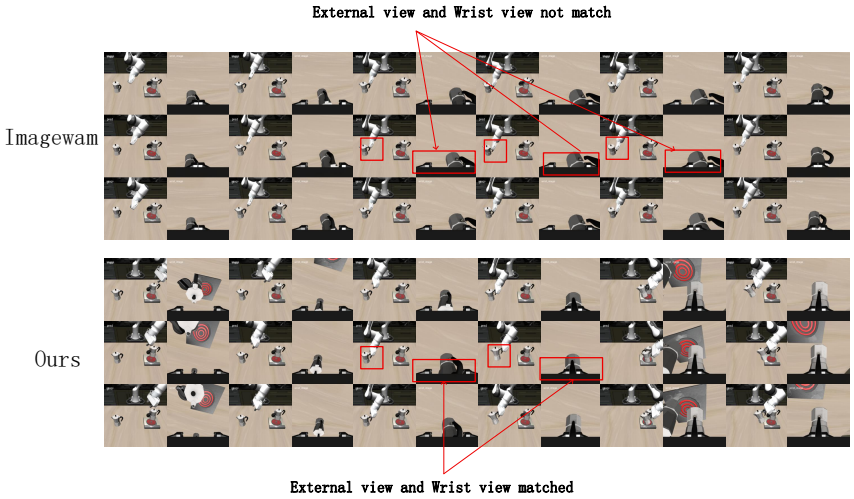


Figure 3: **Qualitative comparison of cross-view prediction consistency.** ImageWAM (top) and CoView-WAM (bottom) predict future observations from external and wrist views. Red boxes highlight cross-view discrepancies in ImageWAM's predictions and better-aligned object configurations in CoView-WAM's predictions.

trained image-editing foundation model and proceed directly to joint training on CALVIN; both are therefore marked $\times$.

VPP pretraining. VPP adopts a two-stage training procedure. Before downstream CALVIN policy training, its manipulation-focused video prediction model is fine-tuned using large-scale Internet human and robot manipulation data together with downstream CALVIN ABC videos. The official VPP ablations report an average completed sequence length of 4.33 for the full configuration, 3.97 when Internet manipulation data are removed, and 3.31 when CALVIN videos are excluded from video-model pretraining. We include these values in Table 2 to distinguish the contribution of VPP's additional manipulation-domain video pretraining from downstream policy training.