ViewDreamer: A Robot World Model beyond Fixed Camera Setups

Ye Mao[1,2,*], **Jinhui Xiong**[1], **Egor Larionov**[1], **Shiyang Lu**[1], **Krystian Mikolajczyk**[2], **Wei Ye**[1]

[1]Meta AI, [2]Imperial College London
[*]Work done at Meta

Video generation models are increasingly important for robot learning, yet most generate each camera view separately or tile all views in a fixed layout. Robot demonstrations, however, are captured by synchronized cameras whose number, placement, and order vary across embodiments. We present ViewDreamer, a single world model that handles variable sets of cameras by disentangling per-view generation from cross-view interaction. Each view follows the generation path of a pretrained single-view model, and views interact only through shared slot tokens. We stabilize multi-view training with a curriculum that gradually increases the number of views. Cross-View Slot Agreement uses visual foundation models to group geometrically and semantically similar regions across views into shared slots. On RoboView-Bench, which covers one to four cameras, ViewDreamer achieves the best generation quality among the models evaluated and improves cross-view consistency over independent per-view generation. These gains carry over to our world-action model (WAM) and action-conditioned world model (AC-WM). Our WAM outperforms the FastWAM-Joint baseline on LIBERO-Plus, RoboTwin 2.0, and RoboDojo, and on a real Franka robot. Both variants improve out-of-domain generalization to various camera settings and scenes.

Date: November 19, 2026
Correspondence: ye.mao21@imperial.ac.uk, weiye@meta.com
Project page: https://matchlab-imperial.github.io/ViewDreamer/

Meta

1 Introduction

Video generation models are increasingly used in robot learning. They can synthesize training data (Jang et al., 2025; Liu et al., 2025), evaluate policies without physical deployment (Guo et al., 2026; Zhu et al., 2024), and serve as policy backbones (Wu et al., 2024a; Ye et al., 2026; Yuan et al., 2026). Robots usually record a task with several synchronized cameras from complementary viewpoints. An embodied world model should predict what every camera will see, and these predictions must agree wherever views overlap. We call this property cross-view consistency.

Most video generation models, however, produce one video from one image (Blattmann et al., 2023; Wan Team et al., 2025). A common workaround tiles all views into one grid (Liu et al., 2025; NVIDIA, 2026; Ye et al., 2026). Tiling trades resolution against compute. A grid at the pretrained resolution shrinks each view, whereas an enlarged grid makes the computational cost of attention grow quadratically with the view count. Tiling also ties each camera to a fixed grid position and places the views side by side, a layout rare in single-camera pretraining videos. Other approaches attend across views or condition on camera geometry yet assume a fixed camera rig (Guo et al., 2026; Huang et al., 2025, 2026b; Jiang et al., 2025b; Liao et al., 2025). Real camera setups differ in number, placement, and order from robot to robot, and even between datasets of one robot. Bimanual robots typically carry more wrist cameras than single-arm robots, and humanoids carry more cameras than fixed arms (AgiBot-World Contributors, 2025; Jiang et al., 2025a; Khazatsky et al., 2024; Wu et al., 2025). Embodied world models therefore need a video backbone that stays cross-view consistent across such camera setups.

Our key idea is to disentangle per-view generation from cross-view interaction. A pretrained video generator models single-view appearance and motion, yet cannot keep multiple views consistent. We therefore process

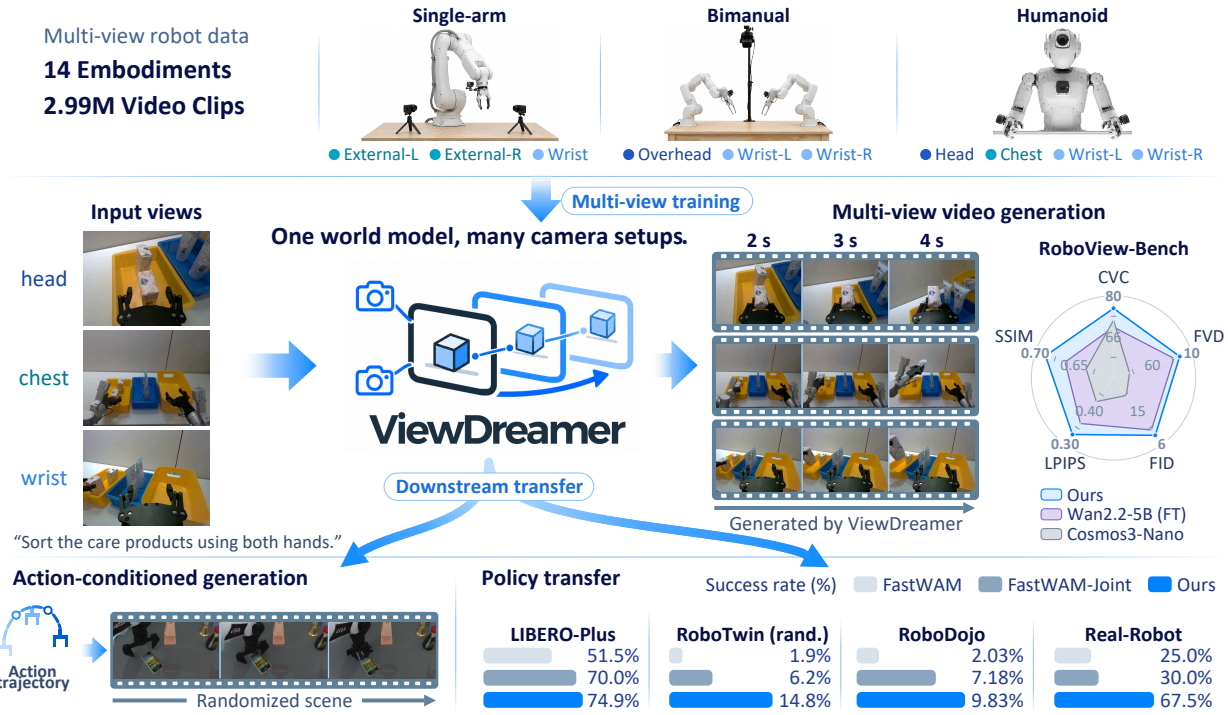


Figure 1 Robots often carry several cameras, whose number and placement vary widely, as in the single-arm, bimanual, and humanoid robots at the top. Pretrained on 2.99M robot video clips from 14 embodiments, ViewDreamer predicts cross-view-consistent futures for such camera sets from the first frame of every view and a task instruction (middle). It achieves the best three-view scores on RoboView-Bench. Its prior transfers to action-conditioned generation and policy learning (bottom) and improves out-of-domain generalization and camera robustness.

every view with the same pretrained backbone. Views communicate only through shared *slots*, which are learnable vectors that gather related regions from all views. We train the whole model jointly. The model retains per-view quality and accepts a variable number of views in any order at a computational cost linear in the view count. Without supervision, the slots tend to merge similar patches across views, such as those of one object, yet the grouping stays fragmented. We show that supervising this grouping improves cross-view consistency.

ViewDreamer implements this design with three components (Fig. 2). After every block of a pretrained video diffusion transformer, a Cross-View Interaction (CVI) module exchanges information among all views through the slots. Each CVI module starts as an identity mapping, and a View-Count Curriculum (VCC) begins with single views and adds more. The model thus grows from the pretrained single-view generator into a multi-view one. Cross-View Slot Agreement (CSA) provides this supervision by distilling region relations from frozen visual foundation models into the slot assignments. We train ViewDreamer on 2.99M clips, including real-robot recordings from 14 embodiments, and build RoboView-Bench from 500 held-out four-camera episodes. On RoboView-Bench with three and four views, ViewDreamer leads all compared generators in fidelity and improves cross-view consistency over independent per-view generation. In a controlled ablation, every component improves policy success and nearly all generation metrics. Better prediction thus appears to help control. CVI also produces better videos than full cross-view attention at a lower computational cost. As an action-conditioned world model, ViewDreamer scores highest among the compared models in both clean and randomized scenes. Our policies add CVI to FastWAM-Joint (Yuan et al., 2026) and are initialized from ViewDreamer. They outperform the original FastWAM-Joint in simulation with two or three cameras and on a real Franka robot. In simulation, these gains are largest in out-of-domain (OOD) settings, such as unseen scenes, new camera viewpoints, or sensor noise.

Our contributions are:

- ViewDreamer: a robot world model that predicts cross-view-consistent future videos for a variable number of cameras. It disentangles per-view generation on the pretrained single-view path from cross-view interaction through shared slot tokens.
- Cross-View Slot Agreement: a training-time alignment objective that makes geometrically and semantically similar regions in different views share a slot. A View-Count Curriculum complements it by growing the model from single-view to multi-view generation.
- RoboView-Bench: a multi-view world model benchmark where ViewDreamer achieves the best fidelity and improves cross-view consistency. Initializing a world-action model and an action-conditioned world model from ViewDreamer improves OOD generalization.

2 Related Work

Video generation in robot learning. Large video generators capture appearance and motion priors (Blattmann et al., 2023; Wan Team et al., 2025), and interactive models learn controllable dynamics (Bruce et al., 2024; Yang et al., 2023). Robot learning uses video generators as data engines that synthesize demonstrations (Guo et al., 2026; Huang et al., 2025; Jang et al., 2025; Jiang et al., 2025b; Liu et al., 2025) and as action-conditioned world models (AC-WMs) that evaluate policies without real robots (Guo et al., 2026; Jiang et al., 2025b; Liao et al., 2025; Rigter et al., 2024; Zhu et al., 2024). Video policies extract actions from planned future frames (Du et al., 2023), condition actions on video-model features (Hu et al., 2025), or jointly predict video and actions as world-action models (WAMs) (Li et al., 2025; Wang et al., 2026b; Wu et al., 2024a; Ye et al., 2026; Yuan et al., 2026; Zhen et al., 2026; Zhu et al., 2025). We develop a video generator for variable camera sets and validate it in an AC-WM and a WAM.

Multi-view robot world models. Genie-Envisioner, its simulator GE-Sim 2.0, and Ctrl-World concatenate the tokens of a fixed camera set (Guo et al., 2026; Liao et al., 2025; Qiu et al., 2026). Cosmos3, DreamZero, and RoboTransfer tile all views into one frame with a fixed layout (Liu et al., 2025; NVIDIA, 2026; Ye et al., 2026). EnerVerse, EVAC, PAIWorld, and MVG-WAM condition on calibrated or estimated camera geometry (Chen et al., 2026b; Huang et al., 2025, 2026b; Jiang et al., 2025b). These designs require a fixed camera set or camera geometry. Multi-agent world models support varying view counts. MultiWorld generates each view from a global 3D-aware state in games and multi-robot simulation (Wu et al., 2026). Multics and γ-World link game players through group or hub tokens (Huang et al., 2026a; Liu et al., 2026). These models simulate several agents with separate actions, whereas ViewDreamer targets one robot with a varying camera set. Unlike group or hub tokens inside self-attention, our slots sit between backbone blocks, which keeps the pretrained token layout and scales linearly with the view count. CSA routes each region's patches to one slot for cross-view consistency.

Multi-view video generation beyond robotics. Conditioned on target camera poses or trajectories, diffusion models generate static objects and scenes (Gao et al., 2024a; Shi et al., 2024), novel views across space and time (Watson et al., 2025), camera-controlled videos (He et al., 2025), and synchronized multi-camera videos (Bai et al., 2024; Kuang et al., 2024; Xu et al., 2024). ViewDreamer, by contrast, conditions only on each camera's first frame and a task instruction, without camera parameters. Driving models generate surround-view images or videos from layouts or ego actions (Gao et al., 2024b; Russell et al., 2025; Wang et al., 2024; Wen et al., 2024) and typically assume a fixed, outward-facing camera rig. Robot cameras instead face a shared workspace, and wrist cameras move with the arm. ViewDreamer assumes no camera layout and learns which regions correspond across views.

3 ViewDreamer

3.1 Overview

Given a task instruction c and the first frame from each of V synchronized cameras, ViewDreamer predicts future frames for all views. We build on a pretrained single-view video generator, which models individual views well but lacks cross-view consistency. ViewDreamer therefore disentangles per-view generation from

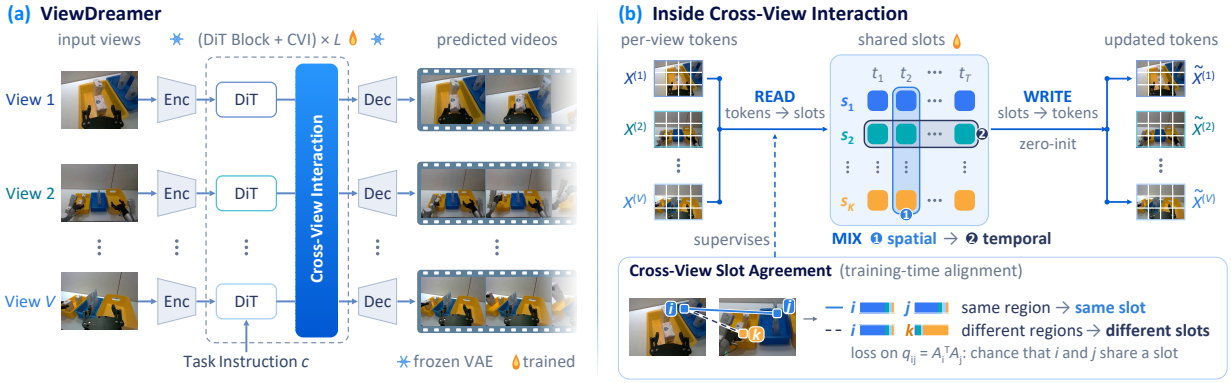


Figure 2 Overview of ViewDreamer. (a) Shared DiT blocks process each view, and a CVI module after every block is the only cross-view path. (b) CVI reads all view tokens into shared slots, mixes the slots over space and time, and writes the result back. CSA aligns the last READ attention with cross-view teacher relations: patches of the same region should share a slot, and patches of different regions should not.

cross-view interaction (Fig. 2a). Each view passes independently through L video DiT blocks (Peebles and Xie, 2023; Wan Team et al., 2025) with shared weights. After each block, a Cross-View Interaction (CVI) module exchanges information through K shared slots: learned tokens that collect features from related regions across views (Sec. 3.2). Our model requires no explicit 3D inputs, such as depth maps or camera parameters, because many robot datasets lack calibration, and off-the-shelf estimates are empirically too inaccurate to improve performance.

We jointly train the DiT blocks and CVI modules on examples with 1 to N views, with camera number, placement, and order varying across training batches. Cross-View Slot Agreement (CSA, Sec. 3.3) supervises which patches share a slot, while the View-Count Curriculum (VCC, Sec. 3.4) gradually increases the number of training views. Together with zero-initialized CVI modules, this curriculum lets the model start from single-view generation and gradually learn cross-view consistency. The base training objective is flow matching (Lipman et al., 2023). For view v with clean video latent $z_0^{(v)}$ and Gaussian noise $\epsilon^{(v)}$, the noisy latent at noise level r is $z_r^{(v)} = (1-r)z_0^{(v)} + r\epsilon^{(v)}$. The model learns to predict the velocity $u_r^{(v)} = \epsilon^{(v)} - z_0^{(v)}$. All views within an example share r but use independent noise, allowing them to interact at the same noise level. For each view, the clean first-frame latent replaces the first noisy frame as conditioning.

3.2 Cross-View Interaction

CVI enables cross-view interaction while keeping the token layout expected by the pretrained blocks. Joint self-attention across views or a tiled canvas would alter this layout. CVI instead keeps each view's tokens intact and routes cross-view exchange through K shared slots, a set bottleneck (Jaegle et al., 2021; Lee et al., 2019). At block ℓ, the DiT processes each view as $X_\ell^{(v)} = \mathrm{Block}_\ell(\tilde{X}_{\ell-1}^{(v)}, c, r)$, with its own 3D RoPE grid and first-frame condition. CVI then applies READ, MIX, and WRITE (Fig. 2b) to gather features into slots, mix them across slots and time, and return cross-view context to each token.

Let B, T, and P denote the batch size, latent frame count, and patches per frame. CVI projects the tokens of all views to width d and groups them by latent frame as $H_\ell \in \mathbb{R}^{B\times T\times VP\times d}$. The slots form $S_\ell \in \mathbb{R}^{B\times T\times K\times d}$ and start from learned seeds $S^0 \in \mathbb{R}^{K\times d}$ shared across frames. We write $\mathrm{CA}(Q, C)$ for cross-attention from queries Q to context C and $\mathrm{SA}(Z)$ for self-attention within Z. Each CVI module computes

$$
\begin{aligned}
\text{READ} \quad & S_\ell^{\mathrm{r}} = S^0 + \mathrm{CA}_\ell^{\mathrm{r}}\big(S^0, H_\ell\big), \\
\text{MIX} \quad & S_\ell^{\mathrm{s}} = S_\ell^{\mathrm{r}} + \mathrm{SA}_\ell^{\mathrm{space}}\big(S_\ell^{\mathrm{r}}\big), \\
& S_\ell^{\mathrm{t}} = S_\ell^{\mathrm{s}} + \mathrm{SA}_\ell^{\mathrm{time}}\big(S_\ell^{\mathrm{s}}; \Phi_T\big), \\
& S_\ell = S_\ell^{\mathrm{t}} + \mathrm{MLP}_\ell\big(S_\ell^{\mathrm{t}}\big), \\
\text{WRITE} \quad & \tilde{X}_\ell^{(v)} = X_\ell^{(v)} + \alpha_\ell\, U_\ell\big(\mathrm{CA}_\ell^{\mathrm{w}}(H_\ell^{(v)}, S_\ell)\big).
\end{aligned}
$$

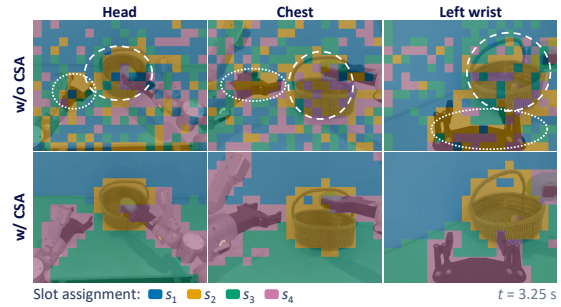


Figure 3 Last-block READ slot assignments (colors matched across rows) on a RoboView-Bench episode. Without CSA (VCC+CVI), slots partially group regions such as the basket (dashed) and gripper (dotted), but stay fragmented. With CSA (ViewDreamer), the groups are sharper and more consistent.

Algorithm 1 Teacher relations for one latent frame.

Require: unit DINO/VGGT-Ω descriptors $\mathbf{d}_i, \mathbf{o}_i$, $V \geq 2$, K
Ensure: labels y and cross-view pair sets $\mathcal{P}^+, \mathcal{P}^-$
$\mathcal{E} \leftarrow \varnothing$; initialize union-find with singleton groups
for each view pair $a < b$ **do**
  $\sigma_{ij} \leftarrow \frac{1}{2}(\mathbf{d}_i^\top \mathbf{d}_j + \mathbf{o}_i^\top \mathbf{o}_j)$ for $i \in a$, $j \in b$
  add (σ_{ij}, i, j) to $\mathcal{E}$ if i, j are mutual nearest neighbors
end for
for $(\sigma_{ij}, i, j) \in \mathcal{E}$ in descending σ_{ij} **do**
  merge the groups of i, j if distinct and view-disjoint
end for
$\tilde{\mathbf{d}}_i \leftarrow$ normalized mean DINO descriptor of i's group
$y \leftarrow$ deterministic spherical K-means on all $\tilde{\mathbf{d}}_i$
$\mathcal{P}^\pm \leftarrow$ cross-view pairs with $y_i = y_j$ (+) or $y_i \neq y_j$ (−)

READ aggregates the tokens of all views into the K slots, independently at each time step. MIX follows factorized space-time attention (Bertasius et al., 2021). It first applies self-attention across the K slots within each frame and then across the T frames of each slot. The sinusoidal temporal positions Φ_T enter only the queries and keys. All attention layers and the MLP apply LayerNorm to their inputs.

WRITE lets the tokens of each view, $H_\ell^{(v)}$, attend to the updated slots of the same frame. The zero-initialized output projection U_ℓ (Zhang et al., 2023) maps this context to the backbone width D, and the scalar α_ℓ starts at one. The model therefore behaves as the pretrained single-view generator at initialization.

CVI is permutation-equivariant over views, and its computational cost grows linearly with the view count. MIX operates on K slots per frame, regardless of the number of cameras. READ and WRITE add $O(VTPKd)$ per block, compared with $O(V^2T^2P^2D)$ for joint self-attention across views.

3.3 Cross-View Slot Agreement

Motivation. Slots decide which patches interact, so patches of one region should share a slot across cameras. Even with long training under flow matching alone, ViewDreamer tends to group similar patches, but the assignments stay fragmented: a region is often split across slots or merged with its surroundings (Fig. 3, top). A slot can therefore mix unrelated content into every view. CSA addresses this by supervising slot assignments with teacher relations: a teacher built from frozen visual foundation models links geometrically and semantically similar patches across cameras. Unlike object-centric models (Kipf et al., 2022; Locatello et al., 2020), CSA does not tie slots to objects.

Teacher relations. The teacher matches patches across viewpoints and groups them semantically. For each latent frame, Alg. 1 first matches patches across views with two complementary frozen teachers: DINOv2 (Oquab et al., 2024) for semantic similarity and VGGT-Ω (Wang et al., 2026a) for geometric correspondence. Mutual nearest neighbors are merged, by decreasing similarity, into groups of matched patches with at most one patch per view. Spherical K-means on the group-averaged DINOv2 features then assigns each patch one of K labels corresponding to the slots. Patch labels are precomputed offline, so neither teacher runs during training or inference.

Agreement loss. CSA supervises the last-block READ attention to follow the teacher's patch grouping, adding no parameters or inference-time computation. Let $M_{ki}^{(h)}$ be the attention from slot k to patch i in head h, normalized over patches. Renormalizing over slots and averaging over N_h heads yields:

$$A_{ik} = \frac{1}{N_\mathrm{h}} \sum_{h=1}^{N_\mathrm{h}} \frac{M_{ki}^{(h)}}{\sum_{k'=1}^{K} M_{k'i}^{(h)}}, \qquad \sum_{k=1}^{K} A_{ik} = 1.$$

Table 1 Multi-view video generation on RoboView-Bench with three and four views. Training never sees four views. Wan2.2-5B (FT) is fine-tuned on the same data as ViewDreamer and generates each view independently; the other baselines run zero-shot. Parentheses give the change from Wan2.2-5B (FT). Best in bold.

Method	3-view					4-view				
	SSIM ↑	LPIPS ↓	FID ↓	FVD ↓	CVC ↑	SSIM ↑	LPIPS ↓	FID ↓	FVD ↓	CVC ↑
Wan2.2-5B	0.510	0.537	23.92	272.02	56.166	0.509	0.540	22.31	259.90	56.284
Wan2.1-14B	0.484	0.570	24.99	261.73	58.314	0.483	0.572	23.75	254.20	58.715
Wan2.2-A14B	0.495	0.559	20.84	247.55	55.341	0.491	0.566	20.06	253.16	53.640
Genie-Envisioner	0.686	0.420	33.20	184.73	71.750	0.657	0.452	61.28	429.01	71.100
Cosmos3-Edge	0.590	0.530	40.85	392.89	53.760	0.646	0.460	28.42	309.27	65.128
Cosmos3-Nano	0.635	0.429	19.25	89.57	71.632	0.675	0.379	13.71	88.16	**77.714**
Wan2.2-5B (FT)	0.662	0.363	9.92	32.93	69.316	0.658	0.371	8.58	29.50	69.901
Ours	**0.687** (+0.025)	**0.330** (−0.033)	**8.47** (−1.45)	**25.41** (−7.52)	**75.693** (+6.377)	**0.688** (+0.030)	**0.332** (−0.039)	**7.13** (−1.45)	**21.26** (−8.24)	75.150 (+5.249)

For same-frame patches i, j in different views, $q_{ij} = A_i^\top A_j$ is the probability of sharing a slot. CSA boosts q_{ij} on same-label pairs $\mathcal{P}^+$ and lowers it on different-label pairs $\mathcal{P}^-$:

$$\mathcal{L}_{\text{CSA}} = \frac{1}{|\mathcal{P}^+|} \sum_{(i,j)\in\mathcal{P}^+} (1 - q_{ij}) + \frac{1}{|\mathcal{P}^-|} \sum_{(i,j)\in\mathcal{P}^-} q_{ij},$$

where empty pair sets are skipped. This is a class-balanced ℓ_1 distance between model and teacher co-association matrices, which only mark same-cluster pairs (Fred and Jain, 2005). It is thus invariant to slot and label permutations and, unlike set-prediction losses (Carion et al., 2020), needs no matching. Pairwise clustering and correspondence losses (Hamilton et al., 2022; Hsu et al., 2018, 2019), REPA (Yu et al., 2025), and PAIWorld (Huang et al., 2026b) supervise token features, whereas CSA supervises slot assignments. The total training loss is $\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_{\text{CSA}}\mathcal{L}_{\text{CSA}}$, where $\mathcal{L}_{\text{flow}}$ is the velocity MSE.

3.4 View-Count Curriculum

VCC stabilizes multi-view training by adding views gradually. Early steps use many single-view clips to adapt the generator to robot videos. At step s with horizon τ, examples keep $V_s = 1 + \text{Binomial}(N-1, p_s)$ uniformly sampled views, with $p_s = \min(s/\tau, 1)$, so $\mathbb{E}[V_s]$ rises linearly to N.

4 Experiments

4.1 Experimental Setting

Tasks. We evaluate ViewDreamer on multi-view video generation, action-conditioned video generation, and world-action policy learning. For the latter two, ViewDreamer initializes an action-conditioned world model (AC-WM) and a world-action model (WAM).

Data. We train ViewDreamer on 2,990,654 five-second clips from AgiBotWorld-Beta (AgiBot-World Contributors, 2025), Galaxea (Jiang et al., 2025a), RoboCOIN (Wu et al., 2025), RoboMIND (Wu et al., 2024b), and RoVid-X (Deng et al., 2026). Real-robot clips cover 14 embodiments, 12 of them with multiple views. Curation removes low-motion clips and invalid captions. The training set retains up to three views per clip and excludes all RoboView-Bench episodes.

Baselines. The multi-view generation baselines are the Wan models (Wan Team et al., 2025), Genie-Envisioner (Liao et al., 2025), and Cosmos3 (NVIDIA, 2026), all run zero-shot from official checkpoints. Wan2.2-5B (FT) is fine-tuned on our data at our batch size and generates each view independently. For AC-WMs, we adapt Wan2.2-5B with our recipe and fine-tune both Cosmos3 models with the same batch size and number of epochs. The policy baselines are the VLAs π_0, $\pi_{0.5}$ (Black et al., 2025, 2024), StarVLA (StarVLA Community, 2026), and ABot-M0 (Yang et al., 2026), and the WAMs OpenWAM-α (Wang et al., 2026b),

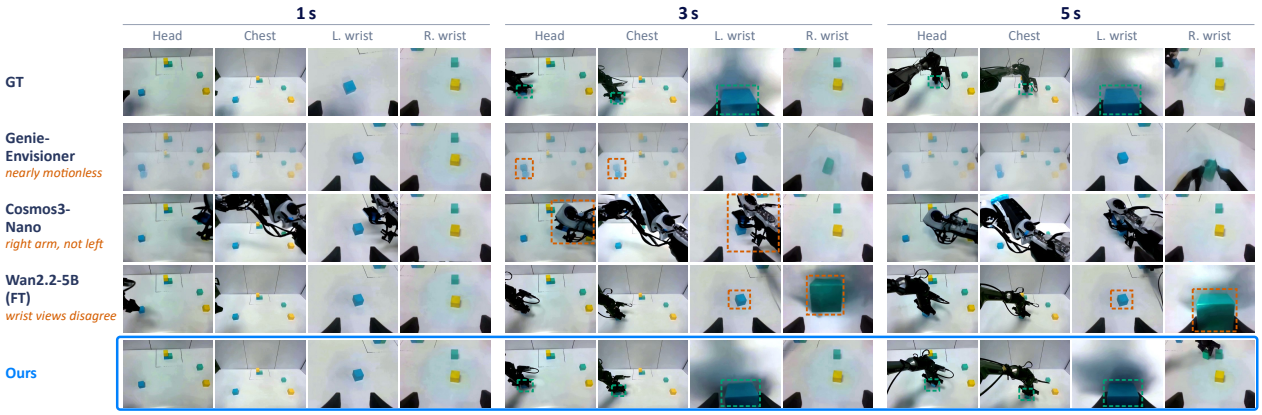


Figure 4 Four-view generation on RoboView-Bench (task: "Put the blue build block in the center of the table with left arm."); training never sees four views. Rows are methods; column groups show all views at 1, 3, and 5 s. Orange notes and boxes mark errors. Genie-Envisioner is nearly motionless, Cosmos3-Nano moves the right arm instead of the left, and the wrist views of Wan2.2-5B (FT) disagree with its other views. Only ViewDreamer keeps all views consistent, as in the ground truth (green boxes). More examples are in App. F.

Table 2 Action-conditioned generation on RoboTwin 2.0 with three views (higher is better). Clean (IID): 250 held-out episodes; Random (OOD): 250 randomized episodes of the same tasks. EWMScore (Shang et al., 2026) is the mean of 15 metrics. Offsets compare Ours with Wan2.2-5B.

Method	EWMScore	Visual quality	Motion quality	Content consistency	Physics adherence	3D accuracy	Controllability
Clean (IID)							
Cosmos3-Edge	47.70	47.63	35.98	**52.51**	20.28	75.86	57.43
Cosmos3-Nano	49.45	50.00	37.62	51.99	**23.67**	**78.61**	**59.17**
Wan2.2-5B	48.33	51.72	38.41	51.41	18.01	75.92	56.20
Ours	**50.09**	**54.72**	**40.74**	50.58	21.19	77.34	58.11
	(+1.76)	(+3.00)	(+2.33)	(−0.83)	(+3.18)	(+1.42)	(+1.91)
Random (OOD)							
Cosmos3-Edge	25.08	17.14	21.66	13.81	10.30	47.89	51.01
Cosmos3-Nano	31.39	29.36	26.78	19.77	11.38	58.36	51.79
Wan2.2-5B	43.08	36.24	40.41	**53.69**	13.08	**61.56**	52.98
Ours	**46.99**	**42.51**	**54.81**	52.73	**13.30**	61.05	**53.02**
	(+3.91)	(+6.27)	(+14.40)	(−0.96)	(+0.22)	(−0.51)	(+0.04)

FastWAM, and FastWAM-Joint (Yuan et al., 2026). FastWAM-Joint's action queries attend to all video tokens, whereas FastWAM's see only the first frame.

Implementation details. We initialize ViewDreamer from Wan2.2-TI2V-5B (Wan2.2-5B) (Wan Team et al., 2025) and train the DiT and CVI modules with the VAE and text encoder frozen. CVI uses $K = 4$ slots of width $d = 256$ and four attention heads. It adds 86.8M parameters, under 2% of the backbone. We set $\lambda_{\text{CSA}} = 0.1$. Each view has 81 frames of 640×480 at 16 FPS, which yields a $21 \times 15 \times 20$ grid of DiT tokens. We use AdamW with a learning rate of 10^{-5}, a weight decay of 0.01, 500 warmup steps, and cosine decay. We train ViewDreamer and Wan2.2-5B (FT) for 32K steps on 128 B200 GPUs with a global batch size of 1024. Ablation runs use 20K steps at a batch size of 256. ViewDreamer and the Wan baselines sample RoboView-Bench videos with 40 Euler steps, a guidance scale of 5, and a shift of 5.

4.2 Multi-View Video Generation

RoboView-Bench. We introduce RoboView-Bench to evaluate multi-view generation on 500 held-out RoboCOIN episodes covering 211 tasks across multiple robots. Camera configurations vary by robot and

include wrist-mounted, body-mounted, and external cameras. We evaluate each episode in both three- and four-view settings. Training uses at most $N = 3$ views per clip, and footage from the fourth RoboCOIN cameras is reserved for four-view evaluation. This setting tests generalization to both an unseen view count and held-out camera placements.

Setting. Given the first frame of every view and the instruction, each model predicts the remaining 80 frames, evaluated at 640×480. SSIM and LPIPS measure paired fidelity, and FID and FVD (Unterthiner et al., 2018) compare distributions. Our cross-view consistency (CVC) score adapts MEt3R (Asim et al., 2025) to simultaneous frames of two cameras. We use every camera pair at frames $t \in \{10, 20, \ldots, 80\}$, resized to 256×256. MEt3R returns a feature dissimilarity and an overlap fraction: (d_g, o_g) for the generated pair and (d_r, o_r) for the real pair. Each camera pair and frame with real overlap $o_r \geq 0.1$ scores

$$100\sqrt{\min\left(1, \tfrac{d_r+\delta}{d_g+\delta}\right)\min\left(1, \tfrac{o_g}{o_r}\right)}, \qquad \delta = 10^{-6},$$

which equals 100 when the generated pair is no more dissimilar and overlaps no less than the real pair. CVC takes the median over each pair's frames and the mean over pairs and episodes, and it needs no camera poses.

Results. ViewDreamer achieves the best SSIM, LPIPS, FID, and FVD of all compared generators with three and four views, and the best three-view CVC (Tab. 1). It also beats Wan2.2-5B (FT), trained on the same data, on every metric, and its fidelity gain grows from one to four views (Tab. 12). Even with a single view, ViewDreamer improves slightly. A VLM judge also rates its three-view consistency highest (Tab. 14). Wan2.2-5B (FT) generates each view alone, so its wrist and head views can disagree on where an object is (Fig. 4). With four views, ViewDreamer keeps a clear lead in fidelity, whereas Genie-Envisioner, trained only with three camera roles, more than doubles its FVD. Cosmos3-Nano reaches the highest four-view CVC, yet its FVD is more than four times ours. Nearly every method predicts wrist views worst, and ViewDreamer gains most there, cutting wrist LPIPS by 0.050 over Wan2.2-5B (FT) versus 0.019 elsewhere (Tab. 13). Other cameras likely supply context missing from fast-moving wrist views.

4.3 Action-Conditioned Video Generation

Setting. We fine-tune ViewDreamer and the official Wan2.2-5B into AC-WMs with the same EVAC-style recipe (Jiang et al., 2025b). ViewDreamer keeps its CVI modules, and Wan2.2-5B attends jointly over all views. Training uses 2,250 clean RoboTwin 2.0 episodes, 45 per task. Clean (IID) evaluation uses the 250 held-out clean episodes, and Random (OOD) evaluation uses 250 randomized-scene episodes of the same tasks. EWMScore (Shang et al., 2026) rates all three views.

Results. Our AC-WM obtains the highest EWMScore on both splits (Tab. 2). Its margin over Wan2.2-5B more than doubles under randomization, from 1.76 to 3.91 points. It also degrades least from Clean to Random, losing 3.10 EWMScore points versus 5.25 for Wan2.2-5B. The Cosmos3 models, which generate wrist views at a lower resolution, degrade far more. This robustness suggests that the ViewDreamer prior is less tied to the appearance of the fine-tuning scenes. The prior improves visual and motion quality on both splits. On Clean, it also improves physics adherence, 3D accuracy, and controllability.

4.4 World-Action Policy Transfer

Setting. Each WAM predicts an action chunk and future visual latents from the current multi-view observations. Our WAM adds CVI to FastWAM-Joint and initializes its video network (and, in simulation, its action network) from ViewDreamer. On LIBERO-Plus (Fei et al., 2025), we test the two-camera LIBERO policies zero-shot on all 10,030 instances of its seven perturbation types. RoboTwin 2.0 (Chen et al., 2025) has 50 three-camera bimanual tasks. Its Clean2Random setting trains on clean demonstrations and tests on 100 clean and 100 randomized episodes per task. The bimanual RoboDojo benchmark (Chen et al., 2026a) reports success rate (SR) and score.

Real-robot tests use a Franka in the DROID setup (Khazatsky et al., 2024) with two external cameras and one wrist camera. The robot puts a cup into a box, places two cubes into cups of matching color, and stacks three cups or three cubes. Each policy learns all four tasks from 50 demonstrations each, and test-time object

Table 3 Single-arm policy success (%). LIBERO tests in-distribution scenes; LIBERO-Plus tests the same policies zero-shot under seven perturbations. Ours builds on FastWAM-Joint; offsets are relative to it (green: better; red: worse). Best in bold, second best underlined.

Method	LIBERO Total ↑	LIBERO-Plus							
		Camera ↑	Robot ↑	Language ↑	Light ↑	Background ↑	Noise ↑	Layout ↑	Total ↑
π_0	94.4	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
$\pi_{0.5}$	96.9	**78.4**	<u>73.6</u>	80.8	96.2	<u>94.1</u>	**89.0**	**84.5**	**84.4**
StarVLA	96.6	52.5	49.8	88.5	95.7	**95.7**	73.0	76.9	74.1
ABot-M0	<u>98.6</u>	<u>60.4</u>	67.9	86.4	96.2	91.6	<u>86.4</u>	<u>82.6</u>	<u>80.5</u>
FastWAM	97.6	16.4	44.5	68.9	78.2	53.7	37.7	60.7	51.5
OpenWAM-α	**99.3**	33.8	**76.1**	88.0	**97.0**	87.1	39.8	77.5	69.2
FastWAM-Joint	98.5	40.7	71.5	**94.7**	<u>96.3</u>	58.0	55.3	78.8	70.0
Ours	<u>98.6</u> (+0.1)	53.6 (+12.9)	71.5 (0.0)	<u>93.3</u> (−1.4)	94.7 (−1.6)	63.6 (+5.6)	69.1 (+13.8)	81.0 (+2.2)	74.9 (+4.9)

Table 4 Bimanual success (%) on RoboTwin 2.0 and RoboDojo. Clean2Random trains on clean demonstrations only and tests on clean and randomized scenes. Ours builds on FastWAM-Joint, and offsets are relative to it. Best in bold, second best underlined.

Method	RoboTwin 2.0			RoboDojo	
	Clean	Rand.	Avg	SR	Score
π_0	46.4	16.3	31.4	1.53	3.48
$\pi_{0.5}$	70.7	<u>46.0</u>	<u>58.4</u>	6.93	11.44
StarVLA	46.5	3.2	24.9	3.24	6.40
ABot-M0	70.7	36.0	53.4	1.73	3.67
FastWAM	77.8	1.9	39.9	2.03	3.48
OpenWAM-α	**89.4**	**48.7**	**69.0**	**11.92**	**17.18**
FastWAM-Joint	80.9	6.2	43.5	7.18	11.35
Ours	<u>83.8</u> (+2.9)	14.8 (+8.6)	49.3 (+5.8)	<u>9.83</u> (+2.65)	<u>15.24</u> (+3.89)

Table 5 Real-robot success rate / score (%) over 10 trials per task. Best in bold, second best underlined.

Method	Cup into box	Cubes into cups	Stack cups	Stack cubes	Avg.
FastWAM	40 / –	<u>30</u> / 35	<u>10</u> / <u>35</u>	20 / <u>40</u>	25.0 / <u>36.7</u>
FastWAM-Joint	<u>60</u> / –	20 / <u>40</u>	<u>10</u> / 30	<u>30</u> / <u>40</u>	<u>30.0</u> / <u>36.7</u>
Ours	**80** / –	**90 / 95**	**40 / 65**	**60 / 75**	**67.5 / 78.3**

positions usually differ somewhat from the demonstrations. Multi-step tasks also get a score that counts one of two subtasks as half a success. Its average covers these three tasks. See App. I for details.

Simulation results. Compared with FastWAM-Joint, our WAM mainly improves OOD generalization (Tabs. 3 and 4). Both policies saturate standard LIBERO (Liu et al., 2023) (98.5% and 98.6%); on LIBERO-Plus, our WAM raises success from 70.0% to 74.9%. The largest gains come under sensor noise and camera viewpoint changes. The camera gain grows with the shift. When the third-person camera orbits the scene by more than 30°, success rises from 24.6% to 44.1% (Fig. 13). Background and layout also improve. On RoboTwin 2.0, our WAM more than doubles randomized success, from 6.2% to 14.8%, and improves clean

Table 6 Component ablation on RoboView-Bench (ablation recipe). LIBERO-Plus SR uses the same checkpoints. Best in bold.

Modules			3-view					4-view					LIBERO-Plus SR ↑
VCC	CVI	CSA	SSIM ↑	LPIPS ↓	FID ↓	FVD ↓	CVC ↑	SSIM ↑	LPIPS ↓	FID ↓	FVD ↓	CVC ↑	
Independent views			0.653	0.376	10.52	36.32	67.367	0.649	0.384	9.18	31.79	67.323	69.00
	✓		0.663	0.364	9.93	32.36	69.527	0.662	0.367	8.59	29.32	70.150	72.64
	✓	✓	0.672	0.351	9.69	31.30	70.538	0.670	0.359	8.44	28.38	71.065	73.89
✓			0.660	0.366	10.34	34.18	68.438	0.657	0.373	9.07	29.75	68.052	69.37
✓	✓		0.670	0.355	9.43	**29.55**	71.148	0.667	0.362	8.10	**26.38**	70.999	73.15
✓	✓	✓	**0.675**	**0.348**	**9.25**	30.59	**72.630**	**0.672**	**0.353**	**7.91**	26.98	**73.178**	**74.18**

Table 7 Cross-view designs from Wan2.2-TI2V-5B, all with VCC and no CSA (ablation recipe). Canvas tiles native-size views under one RoPE grid. Full attn. keeps per-view RoPE. Best in bold.

	3-view			4-view		
	SSIM ↑	FVD ↓	CVC ↑	SSIM ↑	FVD ↓	CVC ↑
Canvas	0.658	35.32	65.210	0.648	30.38	67.800
Full attn.	0.645	37.48	67.380	0.639	33.58	68.138
CVI	**0.670**	**29.55**	**71.148**	**0.667**	**26.38**	**70.999**

Table 8 Effect of the slot count K (shaded: full model). The other settings follow the full model (ablation recipe). Best in bold.

K	3-view			4-view		
	SSIM ↑	FVD ↓	CVC ↑	SSIM ↑	FVD ↓	CVC ↑
2	0.677	32.92	70.244	**0.677**	28.97	69.814
4	0.675	**30.59**	**72.630**	0.672	**26.98**	**73.178**
8	**0.678**	33.71	71.815	0.676	28.56	71.445

Table 9 Effect of λ_{CSA} and the teacher (shaded: full model). All runs use VCC and CVI (ablation recipe). w/o VGGT-Ω: DINO-only teacher, $\lambda_{\text{CSA}} = 0.1$. Best in bold.

λ_{CSA}	3-view			4-view		
	SSIM ↑	FVD ↓	CVC ↑	SSIM ↑	FVD ↓	CVC ↑
0	0.670	**29.55**	71.148	0.667	**26.38**	70.999
0.1	0.675	30.59	72.630	0.672	26.98	**73.178**
0.5	0.678	35.09	**72.995**	0.677	30.65	72.532
1.0	**0.680**	40.92	69.713	**0.679**	36.22	70.703
w/o VGGT-Ω	0.672	31.14	72.088	0.669	26.74	71.639

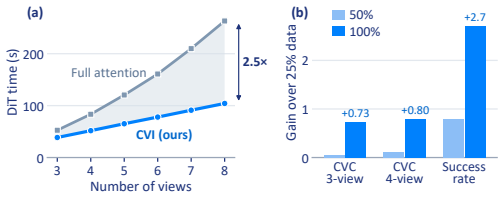


Figure 5 (a) DiT GPU time per sample (80 forward passes, no VAE) on one B200. (b) Gains in three- and four-view CVC and LIBERO-Plus success over the nested 25% subset (ablation recipe).

success. It ranks second on RoboDojo. The policies that lead on some benchmarks, $\pi_{0.5}$, ABot-M0, and OpenWAM-α, pretrain on thousands of hours of embodied data with actions, whereas our pretraining uses only videos, without actions.

Real-robot results. Our WAM outperforms both FastWAM variants on every real-robot task and raises the average success rate from 30.0% (FastWAM-Joint) to 67.5% (Tab. 5). The gain is largest on cubes into cups, from 20% to 90%. On the two stacking tasks, our WAM doubles or quadruples the success of FastWAM-Joint (60% vs. 30% for cubes and 40% vs. 10% for cups). It also completes more subtasks: its average score over the three multi-step tasks rises from 36.7% to 78.3%. All three policies use the same 200 demonstrations, the same three cameras, and the same trial protocol.

4.5 Ablation Study

Tab. 6 reports runs of 20K steps at a global batch size of 256. Each checkpoint initializes a WAM fine-tuned on LIBERO as in Sec. 4.4. Every component improves SSIM, LPIPS, FID, and CVC, and the full model is best on LIBERO-Plus.

Cross-View Interaction. Adding CVI to the independent-view baseline improves policy success and every generation metric; three-view FVD drops from 36.32 to 32.36. Given the same views and data and the small single-view gain (Tab. 12), most of the gain likely comes from cross-view exchange. Without CVI in pretraining or policy, LIBERO-Plus success is 69.0%, close to FastWAM-Joint's 70.0%.

Cross-view design. Tab. 7 compares CVI with canvas tiling and full self-attention, which also use no view embedding. Both alternatives trail CVI on every reported metric, likely because they disturb the token layout of the pretrained blocks. CVI also requires less computation. Each extra view adds a nearly constant 13 s of DiT time per sample, whereas full-attention time grows superlinearly (Fig. 5a).

View-Count Curriculum. Runs without VCC use all native views of every clip. VCC helps every variant and lowers the three-view FVD of CVI from 32.36 to 29.55. It helps even without CVI. With CVI, it lets cross-view interaction grow out of the single-view solution.

Cross-View Slot Agreement. CSA improves the SSIM, LPIPS, FID, and CVC of VCC+CVI and raises LIBERO-Plus success from 73.2% to 74.2%. Its CVC gain grows from 1.5 points with three views to 2.2 with four, where more views must agree. CSA also sharpens the slot assignments (Fig. 3). Without CSA, a region's patches spread over several slots. With CSA, the same-slot share of adjacent patches rises from 39% to 87% (27% and 42% if shuffled).

Slot count, agreement weight, and teacher. Four slots give the best FVD and CVC with both view counts (Tab. 8). A DINO-only teacher (Tab. 9, last row) improves CVC over no CSA yet trails our teacher by 0.5 and 1.5 points with three and four views. Geometric correspondences may matter more as the view count grows.

Data scaling. CVC and LIBERO-Plus success improve with more training data, mostly from half to full data (Fig. 5b). Even with a quarter of the data, the full model beats the all-data independent-view baseline in CVC (71.9 vs. 67.4) and LIBERO-Plus success (71.5% vs. 69.0%).

5 Conclusion

We introduce ViewDreamer, a novel multi-view robot world model that improves consistency across views by running a pretrained single-view generator per camera and linking cameras through shared slots. Extensive evaluations show the best generation quality and stronger downstream policies in simulation and on a real robot, underscoring the potential of one model for robots with diverse camera setups. The gains are largest out of domain, including under new camera viewpoints, scenes, and sensor noise.

Limitations and Future Work. Generating consistent wrist-camera views remains challenging, as these cameras move quickly. Our design restricts neither the backbone nor the camera setup, yet truly arbitrary setups remain unverified. Our experiments cover a limited range of viewpoints and camera types, at most four cameras, and only Wan2.2-5B. Broader placements, more cameras, and larger backbones remain to be tested, and future work could extend ViewDreamer to multi-agent games and autonomous driving.

References

AgiBot-World Contributors. AgiBot World Colosseo: A Large-scale Manipulation Platform for Scalable and Intelligent Embodied Systems. *arXiv preprint arXiv:2503.06669*, 2025. https://arxiv.org/abs/2503.06669.

Mohammad Asim, Christopher Wewer, Thomas Wimmer, Bernt Schiele, and Jan Eric Lenssen. MEt3R: Measuring multi-view consistency in generated images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. https://arxiv.org/abs/2501.06336.

Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Xiao Fu, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. SynCamMaster: Synchronizing multi-camera video generation from diverse viewpoints. *arXiv preprint arXiv:2412.07760*, 2024. https://arxiv.org/abs/2412.07760.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning*, 2021. https://arxiv.org/abs/2102.05095.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Conference on Robot Learning*, 2025.

Kevin Black et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023. https://arxiv.org/abs/2304.08818.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, et al. Genie: Generative interactive environments. *arXiv preprint arXiv:2402.15391*, 2024. https://arxiv.org/abs/2402.15391.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, et al. RoboDojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026a.

Tianxing Chen 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.

Wenbo Chen, Tianfu Li, Haoxuan Xu, et al. MVG-WAM: Multiple view geometry-aware world-action modeling for robotic manipulation. *arXiv preprint arXiv:2609.37793*, 2026b. https://arxiv.org/abs/2609.37793.

Yufan Deng et al. Rethinking Video Generation Model for the Embodied World. *arXiv preprint arXiv:2601.15282*, 2026. https://arxiv.org/abs/2601.15282.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning Universal Policies via Text-Guided Video Generation. In *Advances in Neural Information Processing Systems*, 2023. https://arxiv.org/abs/2302.00111.

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

Ana L. N. Fred and Anil K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. CAT3D: Create Anything in 3D with Multi-View Diffusion Models. In *Advances in Neural Information Processing Systems*, 2024a. https://arxiv.org/abs/2405.10314.

Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. MagicDrive: Street View Generation with Diverse 3D Geometry Control. In *International Conference on Learning Representations*, 2024b. https://arxiv.org/abs/2310.02601.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A Controllable Generative World Model for Robot Manipulation. In *International Conference on Learning Representations*, 2026. https://arxiv.org/abs/2510.10125.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *International Conference on Learning Representations*, 2022.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling Camera Control for Video Diffusion Models. In *International Conference on Learning Representations*, 2025. https://arxiv.org/abs/2404.02101.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. In *International Conference on Learning Representations*, 2018.

Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. Multi-class classification without multi-class labels. In *International Conference on Learning Representations*, 2019.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, et al. Video Prediction Policy: A Generalist Robot Policy with Predictive Visual Representations. In *International Conference on Machine Learning*, 2025. https://arxiv.org/abs/2412.14803.

Kaiyi Huang, Wangbo Yu, Yukun Huang, Jiwen Yu, Kunhao Liu, Donglin Yang, Wenbo Hu, Ying Shan, and Xihui Liu. Multics: Generative Multi-Player World Model with Hierarchical Communication. *arXiv preprint arXiv:2610.17201*, 2026a. https://arxiv.org/abs/2610.17201.

Siyuan Huang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Zhengkai Jiang, Yue Hu, et al. EnerVerse: Envisioning Embodied Future Space for Robotics Manipulation. In *Advances in Neural Information Processing Systems*, 2025. https://arxiv.org/abs/2501.01895.

Yuhang Huang, Xuan Lv, Junyan Xu, Zhiyuan Yu, Jiazhao Zhang, et al. PAIWorld: A 3D-Consistent World Foundation Model for Robotic Manipulation. *arXiv preprint arXiv:2606.18375*, 2026b. https://arxiv.org/abs/2606.18375.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, 2021. https://arxiv.org/abs/2103.03206.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, et al. DreamGen: Unlocking Generalization in Robot Learning through Video World Models. *arXiv preprint arXiv:2505.12705*, 2025. https://arxiv.org/abs/2505.12705.

Tao Jiang et al. Galaxea Open-World Dataset and G0 Dual-System VLA Model. *arXiv preprint arXiv:2509.00576*, 2025a. https://arxiv.org/abs/2509.00576.

Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, et al. EnerVerse-AC: Envisioning Embodied Environments with Action Condition. *arXiv preprint arXiv:2505.09723*, 2025b. https://arxiv.org/abs/2505.09723.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. https://arxiv.org/abs/2403.12945.

Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. In *International Conference on Learning Representations*, 2022.

Zhengfei Kuang, Shengqu Cai, Hao He, Yinghao Xu, Hongsheng Li, Leonidas Guibas, and Gordon Wetzstein. Collaborative Video Diffusion: Consistent Multi-video Generation with Camera Control. In *Advances in Neural Information Processing Systems*, 2024. https://arxiv.org/abs/2405.17414.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. In *International Conference on Machine Learning*, 2019. https://arxiv.org/abs/1810.00825.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified Video Action Model. In *Robotics: Science and Systems*, 2025. https://arxiv.org/abs/2503.00200.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint arXiv:2508.05635*, 2025. https://arxiv.org/abs/2508.05635.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. https://arxiv.org/abs/2210.02747.

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*, 2023.

Fangfu Liu, Kai He, Tianchang Shen, Tianshi Cao, Sanja Fidler, Yueqi Duan, Jun Gao, Igor Gilitschenski, Zian Wang, and Xuanchi Ren. Gamma-World: Generative multi-agent world modeling beyond two players. *arXiv preprint arXiv:2605.28816*, 2026.

Liu Liu, Xiaofeng Wang, Guosheng Zhao, Keyu Li, Wenkang Qin, Jiagang Zhu, et al. RoboTransfer: Controllable Geometry-Consistent Video Diffusion for Manipulation Policy Transfer. *arXiv preprint arXiv:2505.23171*, 2025. https://arxiv.org/abs/2505.23171.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, 2020. https://arxiv.org/abs/2006.15055.

NVIDIA. Cosmos 3: Omnimodal World Models for Physical AI. Technical report, NVIDIA, 2026. https://research.nvidia.com/labs/cosmos-lab/cosmos3/.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. https://arxiv.org/abs/2304.07193.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *IEEE International Conference on Computer Vision*, 2023. https://arxiv.org/abs/2212.09748.

Boxiang Qiu, Liliang Chen, Yue Liao, Nan Wang, Lintao Wang, et al. GE-Sim 2.0: A Roadmap Towards Comprehensive Closed-loop Video World Simulators for Robotic Manipulation. *arXiv preprint arXiv:2605.27491*, 2026. https://arxiv.org/abs/2605.27491.

Marc Rigter, Tarun Gupta, Agrin Hilmkil, and Chao Ma. AVID: Adapting Video Diffusion Models to World Models. *arXiv preprint arXiv:2410.12822*, 2024. https://arxiv.org/abs/2410.12822.

Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. GAIA-2: A Controllable Multi-View Generative World Model for Autonomous Driving. *arXiv preprint arXiv:2503.20523*, 2025. https://arxiv.org/abs/2503.20523.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, et al. WorldArena: A unified benchmark for evaluating perception and functional utility of embodied world models. *arXiv preprint arXiv:2602.08971*, 2026.

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. MVDream: Multi-view Diffusion for 3D Generation. In *International Conference on Learning Representations*, 2024. https://arxiv.org/abs/2308.16512.

Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. StyleGAN-V: A continuous video generator with the price, image quality and perks of StyleGAN2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

StarVLA Community. StarVLA: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards Accurate Generative Models of Video: A New Metric & Challenges. *arXiv preprint arXiv:1812.01717*, 2018. https://arxiv.org/abs/1812.01717.

Wan Team, Ang Wang, Baole Ai, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. https://arxiv.org/abs/2503.20314.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. *arXiv preprint arXiv:2605.15195*, 2026a.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the Future: Multiview Visual Forecasting and Planning with World Model for Autonomous Driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. https://arxiv.org/abs/2311.17918.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, Yue Chen, Xuemin Chi, Donghao Zhou, Qize Yu, Yu-Kai Wang, Yuhan Rui, Shenzhe Yao, Zhen Yuan, Zhenhao Shen, Kefei Zhu, Zijie Zhu, Ning Gao, Xiaowei Chi, Guanqi He, Shanghang Zhang, Hao Dong, Lin Shao, and Hang Zhao. OpenWAM: An open, modular exploration towards systematic world–action model pretraining. *arXiv preprint arXiv:2609.07398*, 2026b. https://arxiv.org/abs/2609.07398.

Daniel Watson, Saurabh Saxena, Lala Li, Andrea Tagliasacchi, and David J. Fleet. Controlling Space and Time with Diffusion Models. In *International Conference on Learning Representations*, 2025. https://arxiv.org/abs/2407.07860.

Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, et al. Panacea: Panoramic and Controllable Video Generation for Autonomous Driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. https://arxiv.org/abs/2311.16813.

Haoyu Wu, Jiwen Yu, Yingtian Zou, and Xihui Liu. MultiWorld: Scalable Multi-Agent Multi-View Video World Models. *arXiv preprint arXiv:2604.18564*, 2026. https://arxiv.org/abs/2604.18564.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, et al. Unleashing Large-Scale Video Generative Pre-training for Visual Robot Manipulation. In *International Conference on Learning Representations*, 2024a. https://arxiv.org/abs/2312.13139.

Kun Wu et al. RoboMIND: Benchmark on Multi-embodiment Intelligence Normative Data for Robot Manipulation. *arXiv preprint arXiv:2412.13877*, 2024b. https://arxiv.org/abs/2412.13877.

Shihan Wu et al. RoboCOIN: An Open-Sourced Bimanual Robotic Data COllection for INtegrated Manipulation. *arXiv preprint arXiv:2511.17441*, 2025. https://arxiv.org/abs/2511.17441.

Dejia Xu, Yifan Jiang, Chen Huang, Liangchen Song, Thorsten Gernoth, Liangliang Cao, Zhangyang Wang, and Hao Tang. Cavia: Camera-controllable multi-view video diffusion with view-integrated attention. *arXiv preprint arXiv:2410.10774*, 2024. https://arxiv.org/abs/2410.10774.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023. https://arxiv.org/abs/2310.06114.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. ABot-M0: VLA foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, et al. World Action Models are Zero-shot Policies. *arXiv preprint arXiv:2602.15922*, 2026. https://arxiv.org/abs/2602.15922.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation Alignment for Generation: Training Diffusion Transformers Is Easier Than You Think. In *International Conference on Learning Representations*, 2025. https://arxiv.org/abs/2410.06940.

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. https://arxiv.org/abs/2603.16666.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE International Conference on Computer Vision*, 2023. https://arxiv.org/abs/2302.05543.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

Haoyu Zhen, Zixian Gao, Qiao Sun, et al. Action images: End-to-end policy learning via multiview video generation. *arXiv preprint arXiv:2604.06168*, 2026. https://arxiv.org/abs/2604.06168.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified World Models: Coupling Video and Action Diffusion for Pretraining on Large Robotic Datasets. In *Robotics: Science and Systems*, 2025. https://arxiv.org/abs/2504.02792.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A Fine-Grained World Model for Robot Manipulation. *arXiv preprint arXiv:2406.14540*, 2024. https://arxiv.org/abs/2406.14540.

Appendix

This appendix follows the order of the main paper. The training data, the model, and CSA come first (Secs. A–C). We then define the cross-view design baselines and the evaluation protocol (Secs. D and E) and add generation results (Sec. F). The setups of the downstream experiments close the appendix (Secs. G–I). Image sizes in this appendix are width×height, and latent and token grids list height first.

A Training Data and Curation

Table 10 Training clips per source after holdout removal and the three-view cap. Counts are clips, not demonstrations.

Source	1-view	2-view	3-view	Total
AgiBotWorld-Beta	0	0	764,058	764,058
Galaxea	0	0	121,735	121,735
RoboCOIN	6,464	0	375,864	382,328
RoboMIND	67,527	21,738	58,988	148,253
RoVid-X	1,574,280	0	0	1,574,280
Total	1,648,271	21,738	1,320,645	2,990,654

We first describe how we build the training clips and how training draws views from them.

Sources. Tab. 10 lists the clips per source. The real-robot clips come from 14 embodiments: AgiBot G1/A2D, Galaxea R1 Lite, Leju Kuavo, AgileX Cobot Magic, AlphaBot 2, Tien Kung, Realman RMC-AIDA-L, Galbot G1, Unitree G1, single-arm Franka, UR5e, AIRBOT MMK2, dual-arm Franka FR3, and Tianqin A2. They span single-arm, dual-arm, and humanoid robots with different camera rigs. RoboMIND simulation and RoVid-X supply the remaining clips.

Curation. Curation removes clips that would teach the model little. A clip counts as near-static when the median adjacent-frame mean absolute luma difference of its primary view is below 0.25. Captions are clip-level for 2,620,273 clips and episode- or task-level for 370,381, and deterministic rules normalize them and reject invalid ones. These two filters keep 2,992,673 of 3,054,470 candidate clips. Removing the 2,019 clips of the RoboView-Bench episodes then leaves 2,990,654. Frames are resized with antialiasing to cover 640×480 and center-cropped.

View counts and sampling. Tab. 10 also splits the clips by view count. All RoVid-X clips have one view, whereas most clips of the other sources have three. VCC (Sec. 3.4) then decides how many of a clip's views each sample uses. Its horizon τ is 20K steps, and each of the three view counts receives one third of the curriculum steps in expectation. Runs without VCC sample clips uniformly and group each step by view count.

Data-scaling subsets. The nested 25% and 50% subsets of Fig. 5b keep whole episodes. Stratification by data source and by the native view count of each clip keeps the full data mix in every subset.

B Model and Training Details

The details below complement Sec. 3: the backbone, the CVI module, and the training objective.

Backbone. ViewDreamer builds on Wan2.2-TI2V-5B (Wan Team et al., 2025), a DiT with 30 blocks of width 3072 and 24 attention heads. Its VAE maps each view to a $48 \times 21 \times 30 \times 40$ latent, and 2×2 patches give 6,300 tokens per view. All views share the UMT5 encoding of one prompt: "A video recorded from a robot's point of view executing the following instruction:" followed by the instruction. Sampling uses the default Wan negative prompt for classifier-free guidance.

CVI modules. Fig. 6 traces the data flow of one module. The module stays small because it works at width $d = 256$ instead of 3072. Its input LayerNorm has no affine parameters, and the MIX feed-forward network

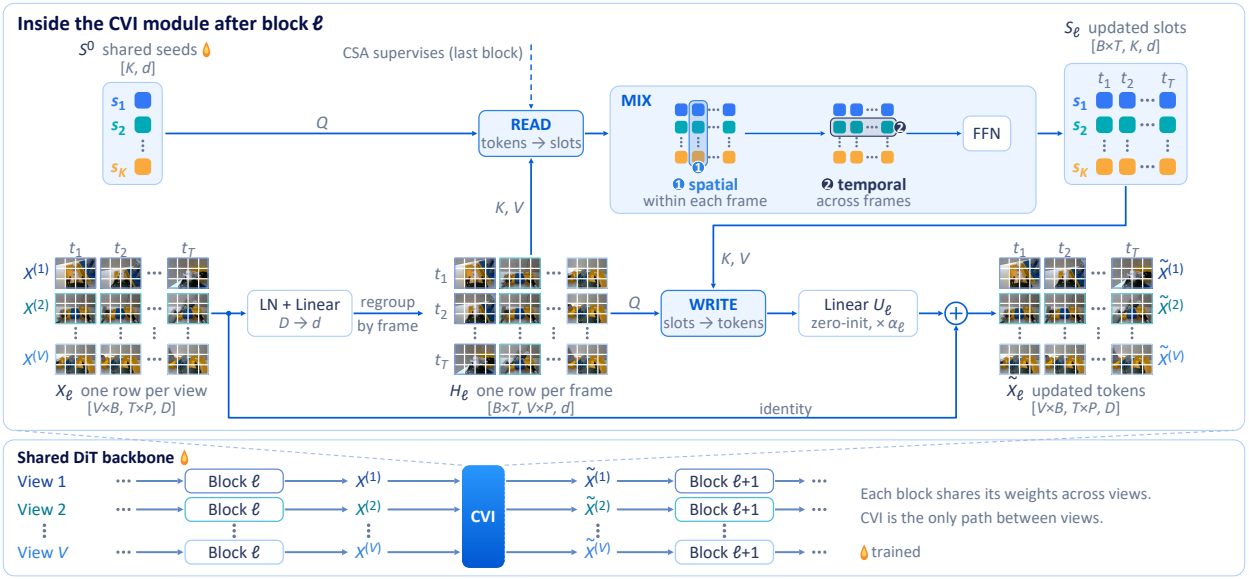


Figure 6 Data flow of one CVI module. Bottom: each backbone block runs on every view with shared weights, and a CVI module follows every block. Top: one CVI module (Sec. 3.2). X_ℓ has one row per view, and H_ℓ has one row per latent frame, holding all views. Slot colors follow Fig. 2. B, T, P, V, and K denote the batch size and the numbers of latent frames, patches per frame, views, and slots; $D = 3072$ and $d = 256$.

has hidden width 512 and uses GELU. Every block has its own READ, MIX, and WRITE weights. All blocks share the slot seeds S^0, drawn from $\mathcal{N}(0, 0.02^2)$. One module has 2.894M parameters.

Training objective. The flow-matching loss averages the squared error over all latent elements of the active views except their conditioning frames. Wan2.2-5B (FT) and the ablations start from the same backbone and use the optimizer settings of Sec. 4.1.

C Cross-View Slot Agreement Details

CSA relies on a teacher that links similar regions across views. This section describes the teacher, shows why it uses VGGT-Ω, and explains how the loss is computed.

Teacher features. The teacher needs features that can match patches across views. We extract them at pixel frames $0, 4, \ldots, 80$, one per latent frame. DINOv2 ViT-L/14 without registers (Oquab et al., 2024) encodes each view separately. The VGGT-Ω aggregator (Wang et al., 2026a) processes all views of one frame jointly, and we take its patch tokens after the global attention of the last block. Each teacher grid is pooled onto the 15×20 token grid, weighting every teacher patch by its pixel overlap with each token footprint. The pooled features are ℓ_2-normalized. VGGT-Ω only decides which patches correspond in Alg. 1, and its image-position cues never enter the clustered DINOv2 features. The spherical K-means step of Alg. 1 initializes its cluster centers by farthest-point sampling.

Why the teacher uses VGGT-Ω. Appearance alone can put one surface into different clusters in different views. A DINO-only teacher runs the same spherical K-means on the DINOv2 features of all views, without correspondences. On a fixed sample of 131 multi-view training clips (about 0.01%), it keeps 82.0% of the mutual matches of Alg. 1 in one cluster. Our teacher keeps 99.2%. Fig. 7 shows typical failures on wrist views.

Loss computation. The two means of $\mathcal{L}_{\text{CSA}}$ pool all labeled cross-view pairs of a microbatch. We compute them exactly from sums over the tokens of each label and view instead of enumerating the pairs. READ uses standard attention normalized over patches, so each slot reads a weighted mean of patch features. CSA renormalizes this attention over slots only to obtain each patch's slot distribution; the forward pass is unchanged.

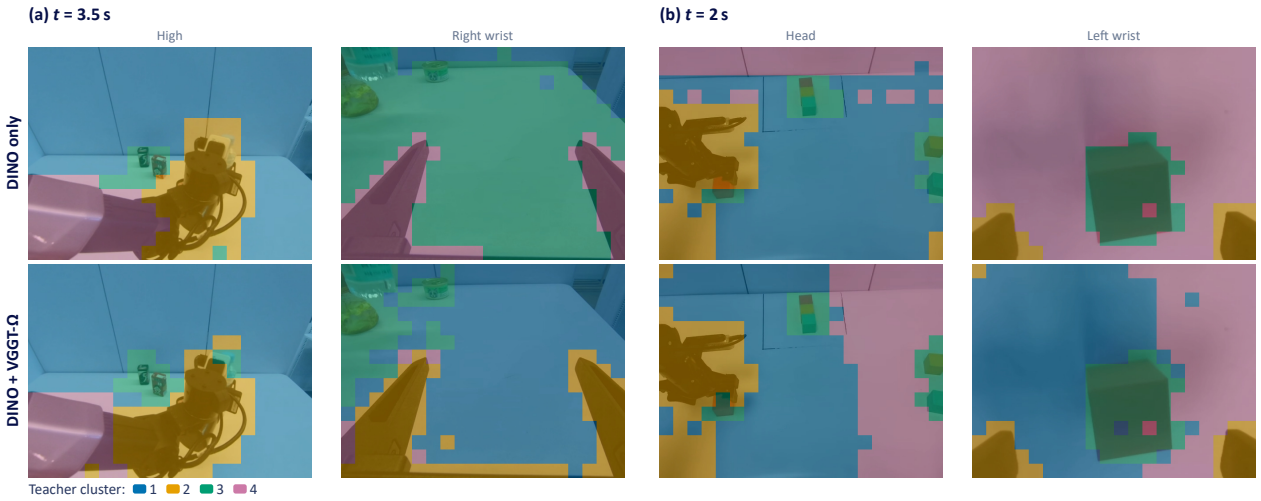


Figure 7 CSA teacher labels from DINOv2 alone (top) and with VGGT-Ω (bottom) on two RoboView-Bench episodes, each in a wrist view and a non-wrist view. Without VGGT-Ω, the table seen at close range by the wrist camera falls into a different cluster than in the non-wrist view. With VGGT-Ω, the table falls into the same cluster in both views.

Slot assignments over time. CSA acts on each frame separately, yet its grouping persists over time. In Fig. 8, the slots with CSA stay stable while the arms move. The shared seeds and the temporal MIX likely cause this stability.

D Cross-View Design Baselines

Tab. 7 compares CVI with two other ways to couple the views. This section defines them and the latency measurement of Fig. 5a. All three designs share the data, the optimizer, and the flow-matching objective, and none uses camera embeddings. CVI denotes the VCC+CVI model of the ablation study (Tab. 6).

Table 11 DiT GPU time on one B200 (lower is better). Per forward pass: median kernel time. Per sample: 80 passes. Best in bold.

		Per forward pass (ms)			Per sample (s)		
Views	Tokens	Canvas	Full attn.	CVI	Canvas	Full attn.	CVI
3	18,900	659	656	**481**	52.7	52.5	**38.5**
4	25,200	1,037	1,033	**642**	82.9	82.6	**51.4**

Full attention. Full attention lets every token attend to all views. Every self-attention layer runs over the concatenated tokens of all active views, and its computation thus grows quadratically with the view count. As in ViewDreamer, every view keeps its own 3D RoPE grid, and tokens at the same position in different views share their rotary phases. The VAE still encodes each view separately with its own first-frame condition.

Pixel canvas. The canvas baseline instead turns all views into one wide video. The views are tiled left to right in sampler order into one $640V \times 480$ video, which the frozen VAE encodes once. The first canvas frame is the image condition, and evaluation cuts the generated canvas back into 640×480 views. Each tile yields 15×20 tokens per latent frame. The canvas thus holds as many tokens as full attention over all views.

Latency. Tab. 11 and Fig. 5a report DiT GPU time per sample. A sample takes 80 forward passes: 40 Euler steps with two guidance branches each. The time per forward pass is the median of five repeats and excludes the text encoder. For three to eight views, one sample takes 38.5 to 104 s with CVI and 52.5 to 263 s with full attention. Canvas requires nearly as much computation as full attention, as its equal token count suggests.

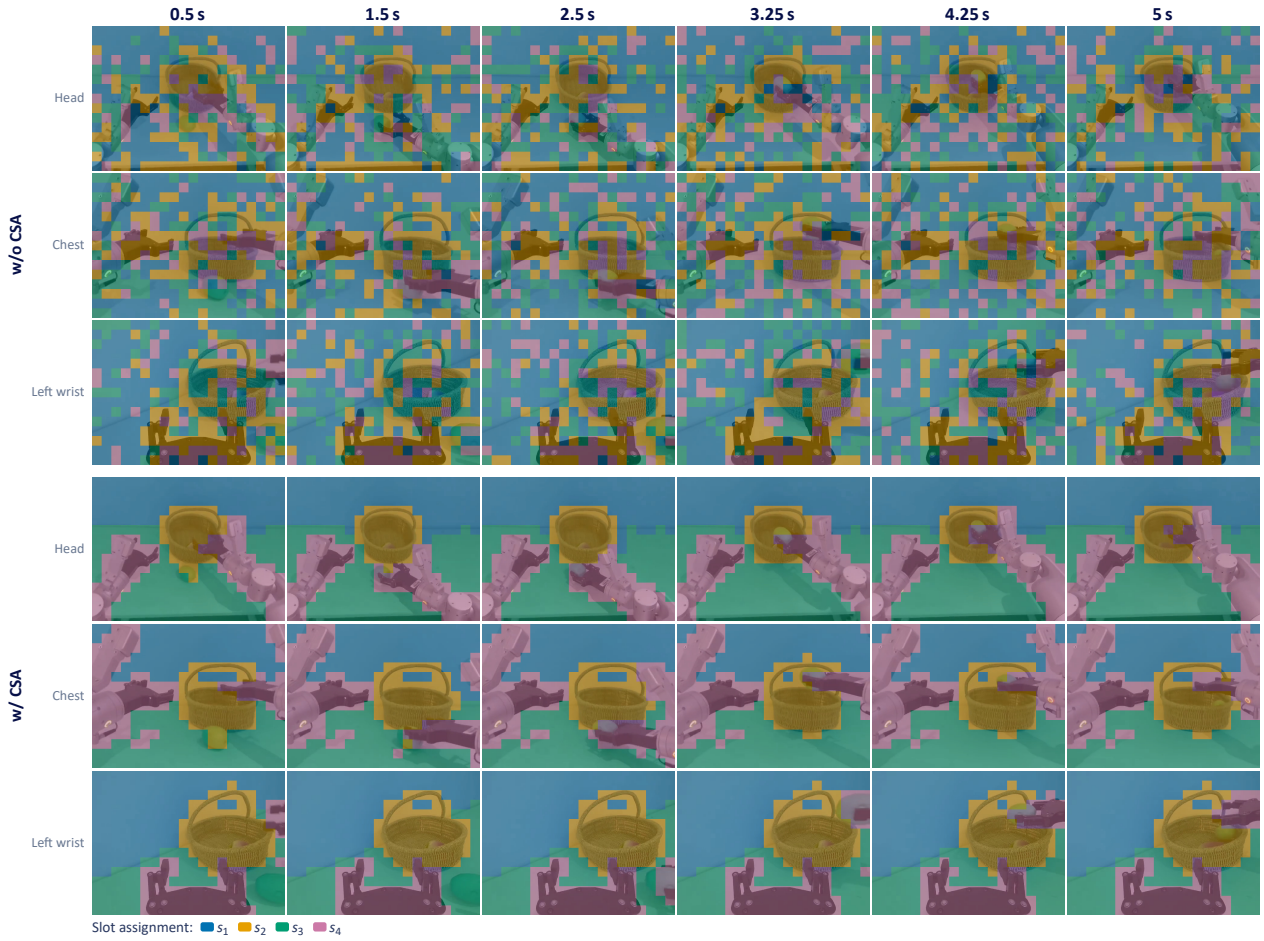


Figure 8 Slot assignments of the last READ attention over 5 s for the episode of Fig. 3 (same colors). The 3.25 s column is the frame of Fig. 3. Without CSA (VCC+CVI), slots tend to merge similar regions, yet the grouping stays fragmented. With CSA (ViewDreamer), the basket, the arms, the table, and the background keep their slots across views and over time.

E Evaluation Protocol

The RoboView-Bench numbers of Tab. 1 depend on a few protocol choices, which this section fixes.

View selection. The three-view setting keeps the first listed view and picks two of the other views by a fixed pseudo-random ranking. Genie-Envisioner instead needs a scene camera and both wrist cameras, so its three views are these cameras. The four-view setting uses all four views. It tests an unseen view count and new camera placements. The ground truth uses the resize and center crop of training.

Fidelity metrics. SSIM and LPIPS (Zhang et al., 2018) with AlexNet features compare each generated frame with its ground truth. We average them over the frames of a view, then over the views of an episode, and then over episodes. FID and FVD instead compare distributions. FID (Heusel et al., 2017) uses the 2048-D Inception pool features of pytorch-fid on every frame, resized to 299×299. FVD uses the StyleGAN-V (Skorokhodov et al., 2022) implementation of I3D (Carreira and Zisserman, 2017) on non-overlapping 16-frame clips of each view and drops the incomplete tail. Both pool all views and episodes into one dataset-level statistic, and views are never tiled into one image. The radar in Fig. 1 plots the three-view results of Tab. 1 on linear axes, and lower LPIPS, FID, and FVD lie farther out. From center to rim, the axes span CVC 52 to 80, FVD 110 to 10, FID 24 to 6, LPIPS 0.50 to 0.30, and SSIM 0.60 to 0.70, so every value lies inside.

Table 12 RoboView-Bench results with one and two views. A single view has no camera pair and thus no CVC. Genie-Envisioner needs a scene camera and both wrist cameras and is omitted. Parentheses give the change of ViewDreamer from Wan2.2-5B (FT). Best in bold.

Method	1-view				2-view				
	SSIM ↑	LPIPS ↓	FID ↓	FVD ↓	SSIM ↑	LPIPS ↓	FID ↓	FVD ↓	CVC ↑
Wan2.2-5B	0.523	0.505	38.87	357.03	0.518	0.527	27.40	291.52	56.30
Wan2.1-14B	0.511	0.524	31.99	294.16	0.496	0.556	26.82	266.73	57.39
Wan2.2-A14B	0.534	0.490	27.28	245.34	0.508	0.539	22.46	233.82	56.56
Cosmos3-Edge	0.617	0.422	44.01	406.40	0.578	0.517	42.79	377.79	45.70
Cosmos3-Nano	0.677	0.332	22.06	104.23	0.666	0.370	19.54	100.75	75.17
Wan2.2-5B (FT)	0.702	0.292	16.46	63.93	0.675	0.342	12.24	39.55	69.97
Ours	**0.705** (+0.003)	**0.283** (−0.009)	**15.78** (−0.68)	**60.23** (−3.70)	**0.693** (+0.018)	**0.318** (−0.024)	**10.89** (−1.35)	**34.03** (−5.52)	**75.93** (+5.96)

Table 13 RoboView-Bench fidelity by camera type. Wrist views have "wrist" or "hand" in the camera name. The three-view setting of Genie-Envisioner uses two wrist views per episode. Best in bold.

Method	3-view				4-view			
	SSIM ↑		LPIPS ↓		SSIM ↑		LPIPS ↓	
	Wrist	Other	Wrist	Other	Wrist	Other	Wrist	Other
Wan2.2-5B	0.497	0.521	0.586	0.497	0.497	0.521	0.583	0.497
Wan2.1-14B	0.458	0.506	0.625	0.526	0.458	0.508	0.624	0.521
Wan2.2-A14B	0.448	0.533	0.647	0.487	0.447	0.536	0.647	0.485
Genie-Envisioner	**0.684**	0.692	0.436	0.388	**0.691**	0.622	0.430	0.474
Cosmos3-Edge	0.587	0.593	0.557	0.509	0.651	0.641	0.478	0.442
Cosmos3-Nano	0.605	0.660	0.468	0.397	0.670	0.680	0.404	0.354
Wan2.2-5B (FT)	0.624	0.692	0.439	0.301	0.627	0.689	0.435	0.306
Ours	0.664	**0.706**	**0.389**	**0.282**	0.669	**0.708**	**0.384**	**0.279**

CVC. CVC builds on the official MEt3R (Asim et al., 2025) code. MEt3R splats FeatUp-upsampled DINO ViT-S/16 features of both frames into the first camera with MASt3R point maps. The dissimilarity is the mean cosine distance over the region that both frames cover, and the overlap is the fraction of the image this region fills. Frames are resized without cropping, and each unordered camera pair is scored once in camera order. A pair without an eligible frame is left out.

F Additional Generation Results

We extend the generation results of Secs. 4.2 and 4.3 with fewer views, more qualitative examples, camera types, a reference-free VLM judge, and action-conditioned rollouts.

Fewer views. Tab. 12 tests whether the gains hold with one or two views. Both settings use the same 500 episodes, and their cameras are a subset of the three-view cameras. The single view is always a body-mounted or external camera. Two-view CVC covers the 484 episodes whose camera pair reaches the overlap threshold. ViewDreamer achieves the best value of every metric with both view counts. With a single view, ViewDreamer still improves slightly over Wan2.2-5B (FT). Multi-view training thus does not seem to hurt single-view generation. The two-view CVC gain over Wan2.2-5B (FT) is 6.0 points, close to the three- and four-view gains (6.4 and 5.2).

More examples. Figs. 9 and 10 add five episodes from three embodiments in the layout of Fig. 4. ViewDreamer again follows the ground-truth motion and keeps the views consistent, whereas the wrist and head views of Wan2.2-5B (FT) often disagree with each other.

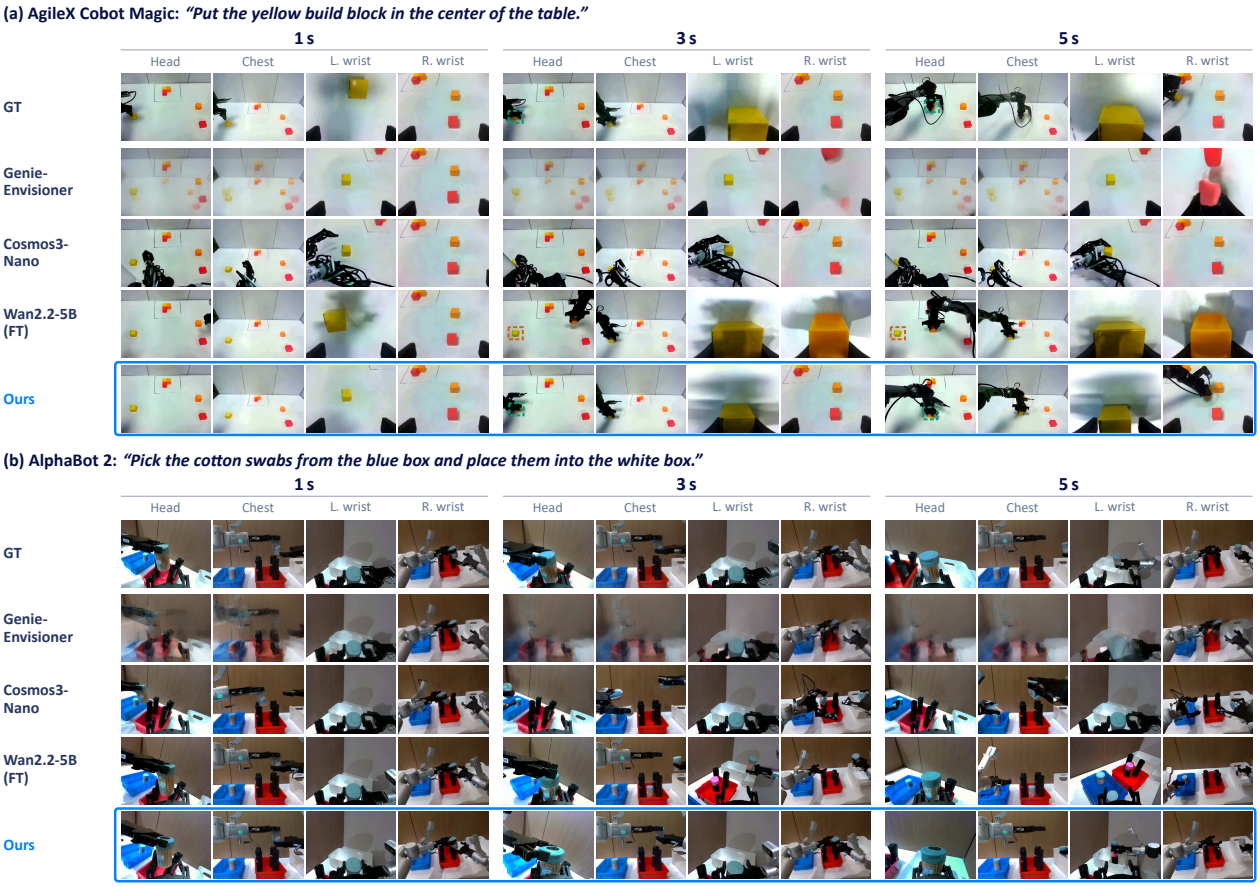


Figure 9 Four-view RoboView-Bench episodes in the layout of Fig. 4. Green boxes mark the block in the left gripper, and orange boxes mark errors of Wan2.2-5B (FT). In (a), its head view leaves the yellow block on the table, while its chest and left wrist views show it grasped.

Camera types. Wrist views are the hardest views for most generators (Tab. 13). With three views, every method has lower SSIM and higher LPIPS on wrist views than on the other views. With four views, Genie-Envisioner scores better on wrist views in both SSIM and LPIPS, and Cosmos3-Edge reaches a higher wrist SSIM. ViewDreamer has the lowest LPIPS on both camera types in both settings.

VLM judge. SSIM and LPIPS need the ground-truth video, and none of the metrics above checks physical plausibility directly. We therefore also score generations with a VLM judge that needs no reference. Gemini 4 Argon sees one composite image per episode. Its rows are the views, and its columns are frames 0, 16, 32, 48, 64, and 80, where the first column is the given frame. We judge all 500 three-view RoboView-Bench episodes of every generator and of the real videos, which serve as a reference (Tab. 14).

The judge rates four criteria from 1 to 5 against a fixed rubric. Instruction following asks whether the robot visibly performs, or clearly progresses on, the instructed task. First-frame grounding asks whether the rollout keeps the scene, objects, and robot of the given frame. Physical plausibility asks whether the robot and the objects keep their identity and shape, move only after contact, and never pass through each other. Cross-view consistency asks whether all views of a column show the same object positions, gripper state, and task stage. The overall score is the mean of the four.

ViewDreamer scores highest among the generators on every criterion and overall. Its gains over Wan2.2-5B (FT) are 0.21 in first-frame grounding, 0.21 in physical plausibility, 0.26 in cross-view consistency, and 0.18 overall. All four 95% paired bootstrap intervals exclude zero. The zero-shot Wan models generate each view alone and receive a cross-view score of 2 in nearly every episode. Real videos score 3.42 overall and stay well above every generator.

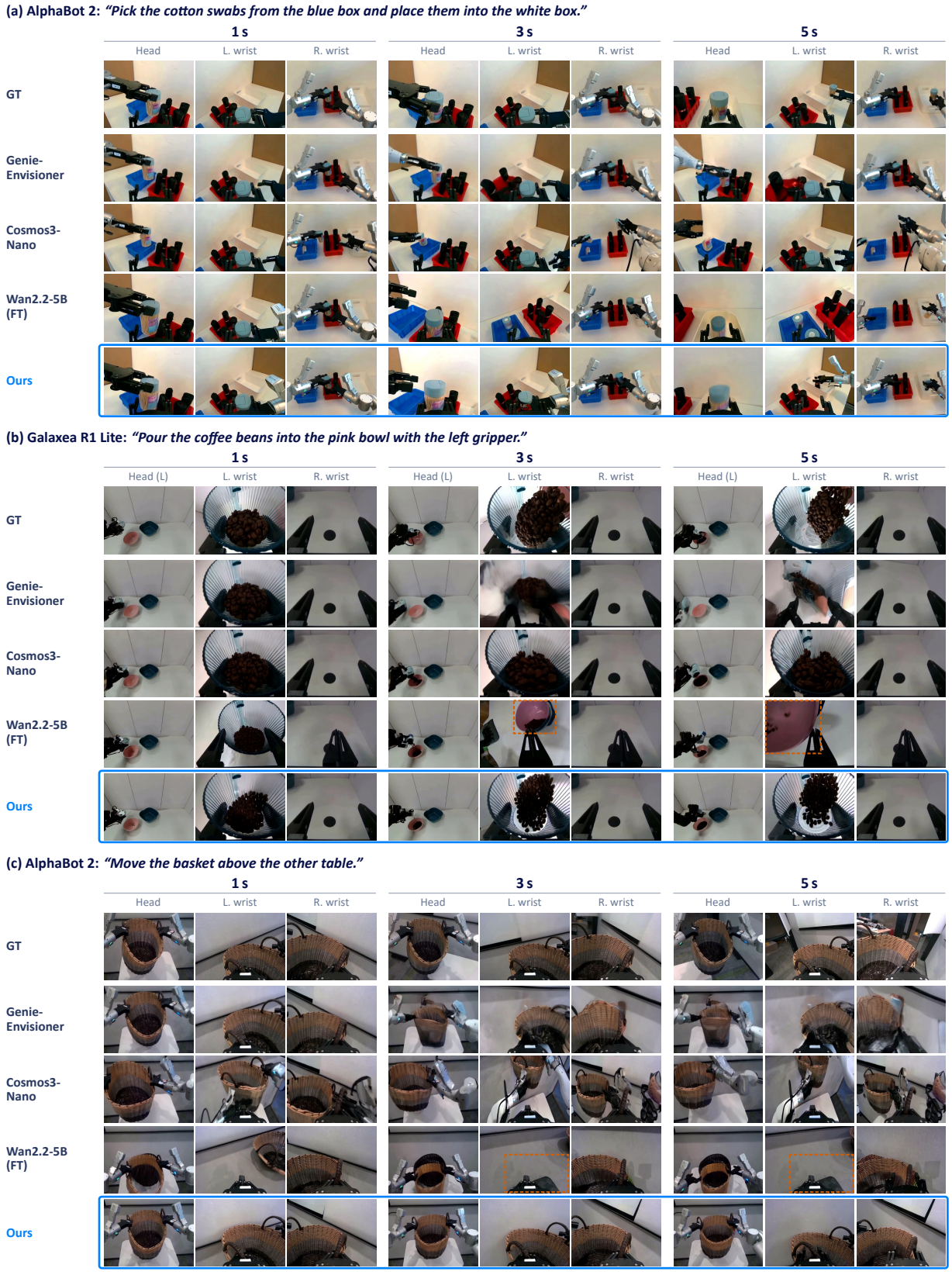


Figure 10 Three-view RoboView-Bench episodes in the layout of Fig. 4. Orange boxes mark errors of Wan2.2-5B (FT), and Head (L) is the left head camera. In (b), it replaces the glass bowl of coffee beans in the left wrist view with a pink bowl. In (c), its left wrist view loses the basket, although its head view still shows it.

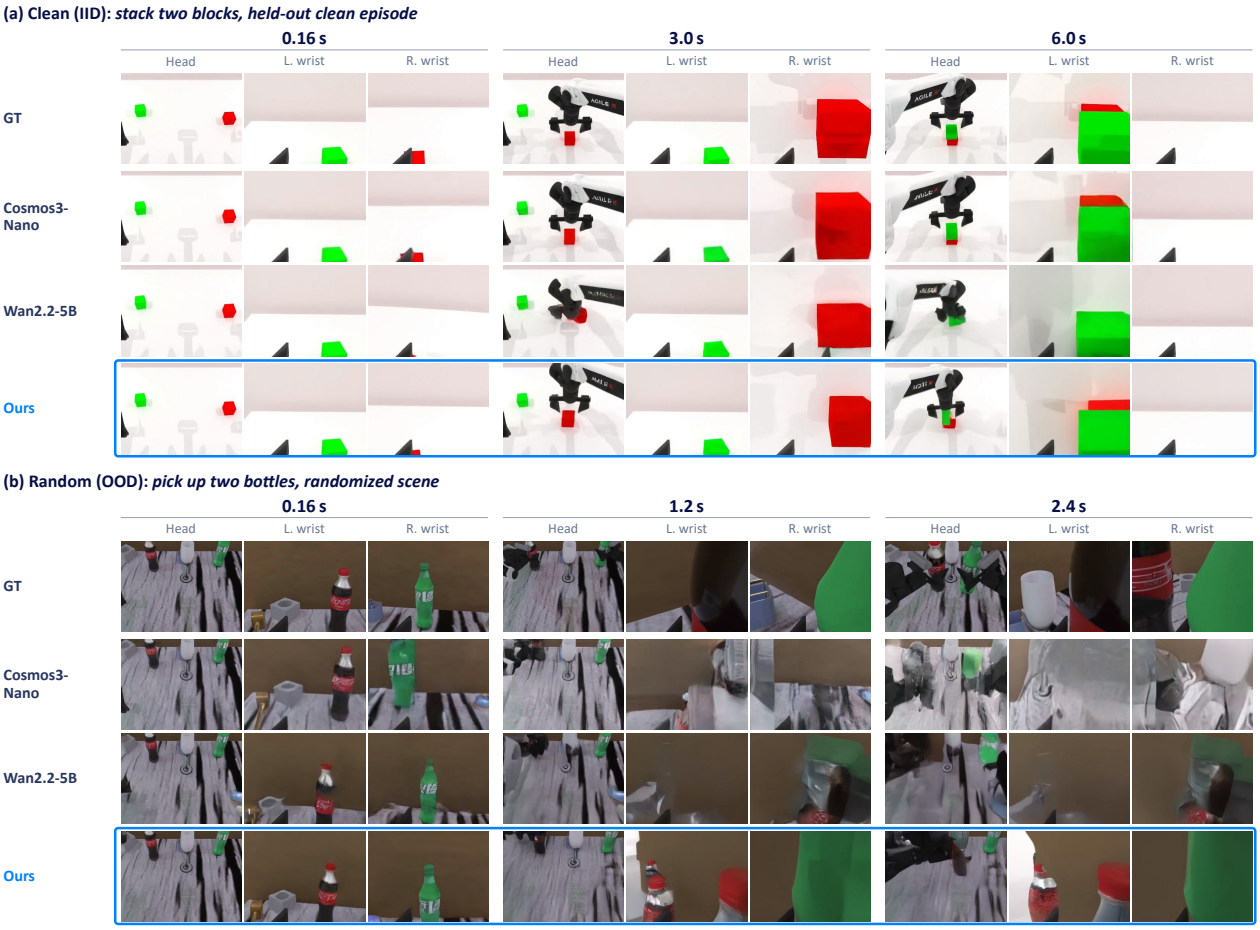


Figure 11 Action-conditioned video generation on RoboTwin 2.0 after four epochs of fine-tuning. (a) A held-out Clean episode (stack two blocks) at 0.16, 3.0, and 6.0 s. (b) A Random episode (pick up two bottles) at 0.16, 1.2, and 2.4 s. All models start from the real first frame of each view and follow the same recorded trajectory. Under randomization, Wan2.2-5B blurs, Cosmos3-Nano washes out the wrist views, and our AC-WM keeps the wrist views sharp. Episodes are selected for illustration.

Action-conditioned rollouts. Fig. 11 illustrates the randomization gap of Tab. 2. All three models follow both arms in all three views of the Clean episode, and the moving wrist views of our AC-WM stay close to the ground truth. Cosmos3-Nano washes out both wrist views of the Random episode within about one second. For display, we upsample the 160×128 wrist views of Cosmos3-Nano.

G Action-Conditioned Generation Details

Sec. 4.3 fine-tunes each video model into an AC-WM. The paragraphs below describe this conversion, and Tab. 15 summarizes the four models.

Data. Each RoboTwin 2.0 task (Chen et al., 2025) is recorded by a head camera and two wrist cameras. Training clips have 17 frames at 25 FPS: the current frame and 16 future frames, and each future frame is paired with one recorded action.

Action conditioning. Wan2.2-5B and our AC-WM share the EVAC-style conditioning (Jiang et al., 2025b) and differ only in the video backbone. The current frame of each view serves as a clean condition. Action maps render the end-effector poses into every camera, and the VAE encodes them. These maps are the only input that uses camera calibration. The networks receive no camera parameters, and, unlike EVAC, we use

Table 14 VLM judge scores on RoboView-Bench with three views (1–5, higher is better). Overall is the mean of the four criteria, and real videos give a reference. Parentheses give the change of ViewDreamer from Wan2.2-5B (FT). Best generated value in bold.

Method	Instruction following ↑	First-frame grounding ↑	Physical plausibility ↑	Cross-view consistency ↑	Overall ↑
Wan2.2-5B	2.126	2.120	2.000	2.010	2.064
Wan2.1-14B	2.300	2.152	2.012	2.000	2.116
Wan2.2-A14B	2.642	2.206	2.024	2.002	2.219
Genie-Envisioner	2.068	2.966	2.216	2.394	2.411
Cosmos3-Edge	2.052	2.254	2.044	2.046	2.099
Cosmos3-Nano	2.338	3.070	2.504	2.392	2.576
Wan2.2-5B (FT)	2.942	2.944	2.434	2.218	2.635
Ours	**2.976** (+0.034)	**3.154** (+0.210)	**2.648** (+0.214)	**2.474** (+0.256)	**2.813** (+0.178)
Real videos	3.226	3.824	3.208	3.418	3.419

Table 15 Action-conditioned world models on RoboTwin 2.0. All four receive three camera views and numeric controls but no text, train for four epochs at a batch size of 512 and a base learning rate of 4×10^{-6}, and sample 50 steps with shift 5 and no guidance. All parameters include frozen modules such as the VAE, which trainable parameters exclude.

Setting	Cosmos3-Edge	Cosmos3-Nano	Wan2.2-5B	**Ours**
Initialization	official checkpoints		Wan2.2-TI2V-5B	ViewDreamer
All parameters	4.07B	15.88B	5.76B	5.84B
Trainable parameters	1.42B	6.97B	5.05B	5.14B
Added parameters	none		51.1M (EVAC)	
Views	one head-over-wrists mosaic		three per-view latents	
Head / wrist size (W×H)	320 × 256 / 160 × 128		320 × 256 / 320 × 256	
Cross-view exchange	attention within the mosaic		joint self-attention	CVI (86.8M)
Actions per clip	16 joint targets (14-D)		16 end-effector deltas (14-D)	
Action maps	none		added to the patch tokens	
Action tokens	16, in self-attention		16 from a resampler, in cross-attention	
State token	none		1, in cross-attention	
Optimizer	AdamW $(0.9, 0.99)$, weight decay 0.05		AdamW $(0.9, 0.95)$, weight decay 0.01	
Sampler	UniPC (Zhao et al., 2023)		Euler	

no ray maps. A Perceiver resampler (Jaegle et al., 2021) encodes the 16 end-effector deltas as tokens that replace the text tokens in cross-attention, together with one token for the robot state.

Training and rollout. Wan2.2-5B and our AC-WM update the whole DiT and the conditioning modules. The Cosmos3 models instead follow the official forward-dynamics recipe and update their generation tower, which processes the video and action tokens. Their equally large understanding tower processes only text and stays frozen, as do all VAEs. At test time, each model rolls out the whole episode in 17-frame chunks. Each chunk starts from the last generated frame, without resetting frames to the ground truth. At the start of each chunk, Wan2.2-5B and our AC-WM also receive the robot's joint state from the recorded episode, the same source as the actions; the Cosmos3 models instead receive absolute joint targets as actions.

H World-Action Model Details

Our WAM adds CVI to FastWAM-Joint (Yuan et al., 2026). This section describes the architecture of the policy, its initialization, and its settings on each simulation benchmark.

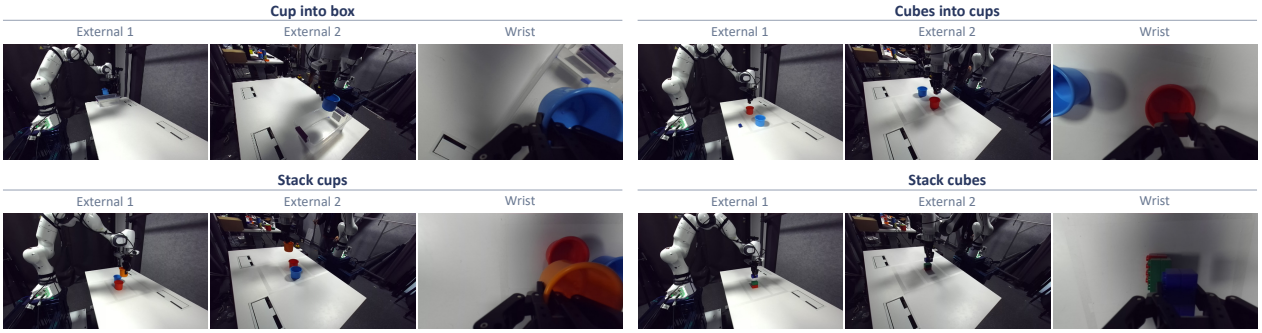


Figure 12 The four real-robot tasks in the three camera views, each shown at one instant of a teleoperated demonstration. Cup into box was recorded only at 320×180 and is upsampled for display.

Architecture. As in FastWAM-Joint, the video and action networks form an asymmetric mixture of transformers. CVI follows every block of the video network. The action queries of every layer attend to the video keys and values of all views after the CVI module. Policy training uses no CSA, and the policy adds no other fusion parameters. FastWAM and FastWAM-Joint keep their original design, tiling all views into one frame.

Initialization. The video network loads every DiT tensor of ViewDreamer, including the CVI modules. The action network mirrors the Wan blocks at width 1024. In simulation, its blocks start from the DiT weights of ViewDreamer, linearly interpolated to width 1024, just as FastWAM-Joint derives its action network from Wan2.2-TI2V-5B. Its encoder and head start from random weights. On the real robot, it keeps the official FastWAM-Joint initialization.

Table 16 Hyperparameters of our WAM on LIBERO. LIBERO-Plus evaluation uses the same inference settings.

Hyperparameter	Value
Demonstrations (tasks)	1,712 (40)
Views (W×H); frames per view	2 × 224×224; 9 (stride 4)
Action space; chunk length	7-D end-effector delta; 32
Global batch; epochs (steps)	128; 10 (21,700)
Learning rate; schedule	10^{-4}; 5% warmup, cosine
Optimizer (β_1, β_2); weight decay	AdamW (0.9, 0.95); 0.01
Gradient clipping	1.0
Augmentation	None
Loss weights (video / action)	1 / 1
Flow shift, video / action (train; test)	5 / 1; 5 / 5
Denoising steps; executed actions	10; 10 of 32

LIBERO and LIBERO-Plus. Tab. 16 lists the LIBERO settings. The 10,030 LIBERO-Plus instances comprise 1,599 Camera, 1,550 Robot, 1,537 Language, 1,142 Light, 1,076 Background, 1,601 Noise, and 1,525 Layout instances. The total success rate divides all successes by 10,030 rather than averaging categories. In Tabs. 3 and 4, the rows other than FastWAM-Joint and ours are published results (Chen et al., 2026a, 2025; Wang et al., 2026b), so their totals follow the sources, and Avg is computed before rounding. We reproduce FastWAM-Joint with its official code, except for its published LIBERO result. The ablation checkpoints in Tab. 6 use the same full benchmark.

Camera viewpoints. LIBERO-Plus moves only the third-person camera, and the wrist camera stays fixed. Fig. 13 splits the 1,599 Camera instances by the type and size of the change. The gain of our WAM over FastWAM-Joint grows with the shift. The two policies are on par for orbits up to 30°. Beyond 30°, our WAM raises success from 24.6% to 44.1%. It also gains 21.3 points when the camera is more than 1.5 times farther away and 9.2 points when the camera only rotates in place.

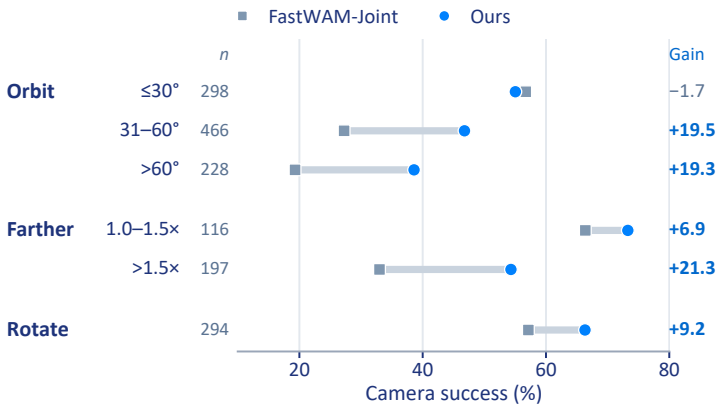


Figure 13 LIBERO-Plus Camera success by perturbation (n per row). The policies see only the default cameras in training. At test time, the third-person camera orbits the scene (binned by the larger of the azimuth and elevation offsets), moves away, or rotates in place. Gain: ours minus FastWAM-Joint.

Table 17 Separating CVI from the ViewDreamer prior (success, %). Wan + CVI has our architecture and recipe but starts from Wan2.2-TI2V-5B, like FastWAM-Joint. Best in bold.

	FastWAM-Joint	Wan + CVI	**Ours**
Initialization	Wan2.2-5B	Wan2.2-5B	ViewDreamer
CVI		✓	✓
LIBERO	98.5	**98.7**	98.6
LIBERO-Plus	70.0	70.1	**74.9**
Camera	40.7	46.0	**53.6**
Robot	**71.5**	61.8	**71.5**
Language	**94.7**	87.2	93.3
Light	**96.3**	89.6	94.7
Background	58.0	62.3	**63.6**
Noise	55.3	**69.4**	69.1
Layout	78.8	78.2	**81.0**

Video initialization. Our WAM differs from FastWAM-Joint in two ways: it adds CVI, and it starts from ViewDreamer. To separate the two, we train Wan + CVI with our architecture and recipe but the initialization of FastWAM-Joint. Both of its networks start from Wan2.2-TI2V-5B, and its CVI modules start as at the beginning of pretraining, with zero WRITE projections. CVI alone leaves LIBERO-Plus success nearly unchanged (70.1% vs. 70.0%) yet raises Camera success from 40.7% to 46.0% (Tab. 17). Starting from ViewDreamer adds 4.8 points overall and 7.6 points on Camera. The multi-view prior thus provides most of the policy gain.

RoboTwin 2.0 and RoboDojo. Tabs. 18 and 19 list the Clean2Random and RoboDojo settings. Both benchmarks use their official protocols and task suites.

I Real-Robot Setup

The real-robot experiment of Sec. 4.4 tests whether the gains carry over to a physical robot. This section describes its hardware, demonstrations, policies, and trial protocol.

Table 18 Hyperparameters of our WAM for Clean2Random on RoboTwin 2.0. ColorJitter strengths are 0.2, except 0.05 for hue.

Hyperparameter	Value
Demonstrations (tasks)	2,500 clean (50)
Views (W×H); frames per view	3 × 320×256; 9 (stride 4)
Action space; chunk length	14-D joint position; 32
Global batch; epochs (steps)	1,024; 5 (2,685)
Learning rate; schedule	10^{-4}; 5% warmup, cosine
Optimizer (β_1, β_2); weight decay	AdamW (0.9, 0.95); 0.01
Gradient clipping	1.0
Augmentation	ColorJitter, $p = 0.8$
Loss weights (video / action)	1 / 1
Flow shift, video / action (train; test)	5 / 1; 5 / 5
Denoising steps; executed actions	10; 24 of 32

Table 19 Hyperparameters of our WAM on RoboDojo. One percent of the 3,500 demonstrations is held out for validation.

Hyperparameter	Value
Demonstrations (tasks)	3,465 (35)
Views (W×H); frames per view	3 × 320×256; 9 (stride 4)
Action space; chunk length	14-D joint position; 32
Global batch; epochs (steps)	256; 16 (114,704)
Learning rate; schedule	10^{-4}; 5% warmup, cosine
Optimizer (β_1, β_2); weight decay	AdamW (0.9, 0.95); 0.01
Gradient clipping	1.0
Augmentation	None
Loss weights (video / action)	1 / 1
Flow shift, video / action (train; test)	5 / 1; 5 / 5
Denoising steps; executed actions	10; 24 of 32

Hardware. We use a Franka Research 3 arm with a Robotiq 2F-85 gripper in the DROID setup (Khazatsky et al., 2024). Two ZED 2i cameras view the workspace from the left and the right, and a ZED Mini is mounted on the wrist. The policies use all three streams at 15 Hz and 320 × 180 pixels.

Tasks and demonstrations. Tab. 20 lists the four tasks, and Fig. 12 shows them in all three views. They range from one pick-and-place to two-step stacking. All demonstrations are teleoperated with a Meta Quest controller. For each task, we keep the 50 recordings closest to the median length, which gives 200 demonstrations and 58,794 frames (about 65 minutes). Median lengths are 14.2 s for cup into box, 19.5 s for cubes into cups, 19.8 s for stack cups, and 23.0 s for stack cubes. The two stacking tasks start from randomized positions of the objects.

Policies. All three policies train on the same 200 demonstrations with the settings of Tab. 21. They see the same three views, resized to 320 × 192. FastWAM and FastWAM-Joint receive them tiled side by side into one 960 × 192 frame, whereas our WAM keeps them as separate views. Each policy executes 24 of its 32 predicted joint-position actions at 15 Hz and then queries again. The arm holds still while the next chunk is computed.

Trial protocol. Before each trial, the arm returns to a fixed home pose with the gripper open, and the operator places the objects. Their positions vary around the demonstration layouts. A trial ends at success or after 60 s. We run 10 trials per task and policy, and Tab. 20 defines success and score.

Table 20 Real-world task suite and success criteria. Every policy receives the same instruction. A trial succeeds when its success criterion holds within 60 s. The score of a two-step task counts a trial that completes one of its two subtasks as half a success.

Task	Instruction	Success criterion
Cup into box	Put the blue cup in the box.	The blue cup comes to rest inside the box.
Cubes into cups	Put the red cube in the red cup and the blue cube in the blue cup.	Each cube rests inside the cup of its color, placed in either order. *Subtasks:* red cube into red cup; blue cube into blue cup.
Stack cups	Stack the orange cup in the red cup, then the red cup in the blue cup.	All three cups end in one nested stack, with the orange cup inside the red cup inside the blue cup. *Subtasks:* orange cup into red cup; then red cup into blue cup.
Stack cubes	Stack the green block on the red block, then the blue block on top.	The three cubes stand as one stack, with red at the bottom, green in the middle, and blue on top. *Subtasks:* green cube onto red cube; then blue cube onto green cube.

Table 21 Hyperparameters of our WAM on the real robot. Only its video network starts from ViewDreamer.

Hyperparameter	Value
Demonstrations (tasks)	200 (4)
Views (W×H); frames per view	3 × 320×192; 33 (stride 1)
Action space; chunk length	8-D joint position; 32
Global batch; epochs (steps)	128; 10 (4,600)
Learning rate; schedule	10^{-4}; 5% warmup, cosine
Optimizer (β_1, β_2); weight decay	AdamW (0.9, 0.95); 0.01
Gradient clipping	1.0
Augmentation	None
Loss weights (video / action)	1 / 1
Flow shift, video / action (train; test)	5 / 1; 5 / 1
Denoising steps; executed actions	10; 24 of 32 (15 Hz)