WHAT VISUAL REPRESENTATIONS DO WORLD ACTION MODELS NEED?

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) inherit powerful priors from pretrained video generation models, but their variational autoencoder (VAE) representations are optimized for visual reconstruction rather than manipulation. Recent studies show that external visual representations improve manipulation performance and robustness. However, these methods select representations based on presumed capability gaps and use different integration mechanisms, making it difficult to disentangle the effects of representation choice and integration. To better understand how external representations strengthen WAMs, we conduct a systematic study through controlled comparisons of multiple visual representations, five integration mechanisms, and feature combinations. Our study yields three main findings. First, strong general-purpose visual representations provide broader robustness gains and can match or outperform representations pretrained for specific visual capabilities. Second, channel-wise joint prediction achieves the highest average success rate; further analyses suggest that future-feature prediction and the action expert's effective use of added visual information contribute to its gains. Third, additional sources should be selected for the task-relevant information they contribute beyond existing representations, as the same source can help one combination but harm another. However, even the best evaluated composition provides a smaller additional gain than the initial improvement from adding a strong general-purpose representation to the VAE-only baseline. We instantiate the resulting **R**epresentation **R**ecipe as R^2-WAM. It achieves state-of-the-art performance on RoboCasa GR1 Tabletop with an 84.7% success rate and reaches 92.8% on RoboTwin 2.0, improving over the VAE-only baselines by 5.9 and 2.2 percentage points, respectively. Taken together, these findings suggest that future work should prioritize stronger general-purpose visual representations and more effective mechanisms for integrating them into WAMs, with feature composition serving as a supplementary route to improvement.

1 INTRODUCTION

World Action Models (WAMs) adapt pretrained video generation models to robot control by jointly predicting future video and actions. This allows robot policies to draw on the visual structure and temporal dynamics learned through large-scale video pretraining, as demonstrated by DreamZero (Ye et al., 2026b), LingBot-VA (Li et al., 2026a), and Fast-WAM (Yuan et al., 2026). Along with these priors, WAMs inherit the latent space of their variational autoencoders (VAEs).

The inherited VAE is optimized for visual reconstruction (Wan Team et al., 2025), but how well its representations support robot manipulation remains unclear. Its reconstruction objective does not explicitly prioritize manipulation-relevant information, such as object semantics and spatial relationships.Recent studies augment WAMs with external visual features and report improvements in manipulation performance and robustness (Qiu et al., 2026; Yan et al., 2026b; Ma et al., 2026a). However, these methods select different representations based on prior assumptions about which visual capabilities WAMs lack. Systematic comparisons across representation types remain limited, leaving unclear which visual representations best support robot manipulation in WAMs.

Beyond representation choice, WAM performance also depends on the mechanism used to integrate external features into the model (Yang et al., 2026). A further question is whether combining more representations leads to continued performance gains. External features can be integrated through

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

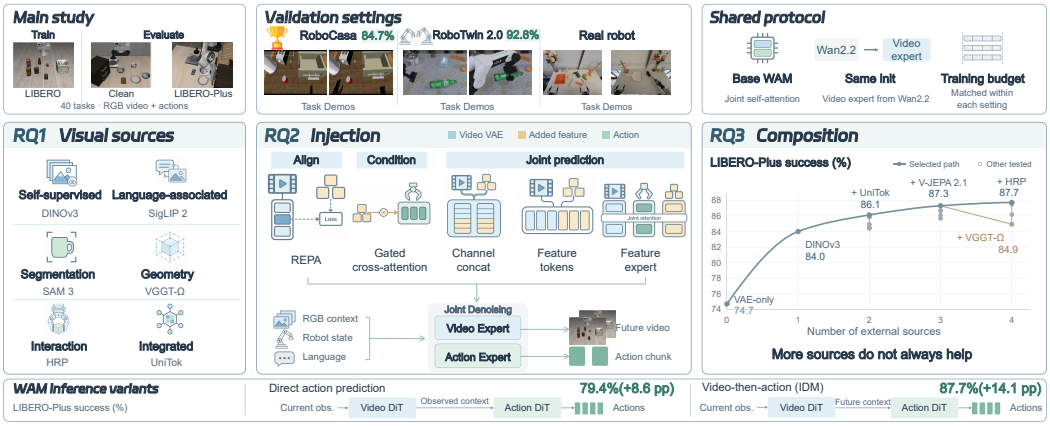

Figure 1: Overview of our systematic study of visual representations in WAMs. Under a shared training protocol, we compare visual feature sources (RQ1), five integration mechanisms (RQ2), and feature compositions (RQ3).

representation alignment (Yu et al., 2025; Qiu et al., 2026), action conditioning (Wang et al., 2026c), or joint prediction (Yan et al., 2026a). Existing studies often couple selected representations with specific integration designs (Qiu et al., 2026; Yan et al., 2026a; Zhang et al., 2026a), making the effects of representation choice and integration mechanism difficult to disentangle across methods. Besides, the diverse demands of manipulation—including semantics, geometry, object extent, motion, and interaction—suggest that different representations may be needed. Yet additional sources may also introduce redundant information or interference during optimization. Whether and when different feature combinations improve WAM performance therefore remains an empirical question.

To address these questions, we systematically study visual representations in WAMs, focusing on capabilities, integration mechanisms, and composition. Using a common WAM backbone built on a pretrained video generation model, we compare diverse visual representations, five integration mechanisms, and multiple source combinations under a shared training protocol. Our study addresses three research questions (RQs):

RQ1—Representations: Which visual representations improve WAM performance?

RQ2—Mechanisms: Which feature injection mechanism yields the greatest performance gains from external features?

RQ3—Composition: Does combining more feature sources consistently improve WAM performance?

Our controlled comparisons yield three findings that guide the selection, integration, and composition of external visual representations for WAMs. **(1) General-purpose visual representations should be the default choice for augmenting WAMs.** They can match or outperform representations pretrained for specific visual capabilities—even under perturbations relevant to those capabilities—while providing broader robustness gains. **(2) Channel-wise joint prediction yields the largest average gains across matched sources.** Our analyses suggest that future-feature prediction and effective use of the added visual information by the action expert contribute to these gains and should guide future integration designs. **(3) Additional sources should be selected for the task-relevant information they add beyond the existing representations.** The same source can improve one combination but provide little benefit or even reduce performance in another. Although feature composition can provide further gains, these are limited compared with the initial improvement. Feature composition should therefore serve as a supplementary route to improvement.

Our contributions are threefold:

- **A controlled study of visual representations in WAMs.** We systematically compare representation sources, five integration mechanisms, and feature combinations within a common WAM backbone and shared training protocol, enabling controlled comparisons of their effects on manipulation performance and robustness.

- **Empirical guidance for WAM representation design.** We derive practical recommendations for selecting, integrating, and combining external visual representations. These findings motivate future research on stronger general-purpose visual foundation models and more efficient and effective mechanisms for integrating their representations into WAMs.
- **R^2-WAM and validation beyond the controlled study.** We instantiate these findings as a **R**epresentation **R**ecipe for WAMs, termed R^2-WAM. We validate the effectiveness of the recipe across different WAM formulations on LIBERO-Plus, and further demonstrate improvements over the corresponding VAE-only baselines on RoboCasa GR1 Tabletop and RoboTwin 2.0, as well as in real-world robot experiments.

2 Related Work

World Action Models. WAMs learn action prediction alongside future visual dynamics. DreamZero (Ye et al., 2026b) transfers a pretrained video diffusion backbone to joint video–action generation, and LingBot-VA (Li et al., 2026a) combines autoregressive diffusion with closed-loop visual feedback. Fast-WAM (Yuan et al., 2026) shows that video co-training can benefit control even when future video generation is omitted at inference. Other studies modify the visual prediction space or its supervision. LDA-1B (Lyu et al., 2026) learns dynamics in DINO feature space, while LaWAM (Chen et al., 2026a) conditions action generation on predicted latent visual subgoals. Rep-WAM (Wang et al., 2026b) and LingBot-VA 2.0 (Zhang et al., 2026b) learn semantically aligned visual–action tokenizers. ST-WAM (Wang et al., 2026c) diagnoses failures under visual shifts and combines DINO future prediction and history retrieval with VAE dynamics. Robust-WAM (Yan et al., 2026b) aligns queries in the action stream with future semantic features while retaining the VAE generation pathway. Flex-π (Yan et al., 2026a) jointly denoises RGB latents, 3D pointmaps, DINO features, and actions, supporting different combinations of input and prediction streams at deployment. These studies demonstrate the effectiveness of specific visual representations within their respective WAM designs, but systematic comparisons are still needed to identify which features complement the capabilities that WAMs inherit from pretrained video generators.

Visual Representations. Pretraining objectives shape the information captured by visual representations. Visual self-supervision has produced encoders such as DINOv3 (Siméoni et al., 2026), VC-1 (Majumdar et al., 2023), LingBot-Vision (Fu et al., 2026), VideoMAE V2 (Wang et al., 2023), and V-JEPA 2.1 (Mur-Labadia et al., 2026). Language-associated approaches include image–text models such as CLIP (Radford et al., 2021) and SigLIP 2 (Tschannen et al., 2025), as well as multimodal language models such as Qwen3.5 (Qwen Team, 2026). Other pretraining objectives target segmentation with SAM 3 (Carion et al., 2025), geometry with VGGT-Ω (Wang et al., 2026a), and interaction-related information with HRP (Srirama et al., 2024) and DynaFLIP (Lee et al., 2026). UniTok (Ma et al., 2025) and C-RADIOv4 (Ranzinger et al., 2026) further combine multiple learning objectives or teacher models within a single representation. These approaches provide different but overlapping visual capabilities, but how their pretraining emphases translate into manipulation gains in WAMs remains an empirical question.

3 Study Design

3.1 Base WAM Architecture

Given RGB observations, a language instruction, and the current robot state, the base WAM jointly predicts future-video latents z^{vae} and an action chunk a. The model comprises a video expert initialized from Wan 2.2 (Wan Team et al., 2025) and a separate action expert. The experts interact through joint attention with an asymmetric mask: action tokens attend to both video and action tokens, whereas video tokens attend only to video tokens. Both experts attend to T5-encoded language features (Raffel et al., 2020) through cross-attention and are additionally conditioned on the robot state and flow time.

We train the model with a weighted sum of video and action flow-matching losses (Lipman et al., 2023):

$$\mathcal{L}_{\text{WAM}} = \mathcal{L}_{\text{FM}}(z^{\text{vae}}) + \lambda_{\text{act}}\mathcal{L}_{\text{FM}}(a), \tag{1}$$

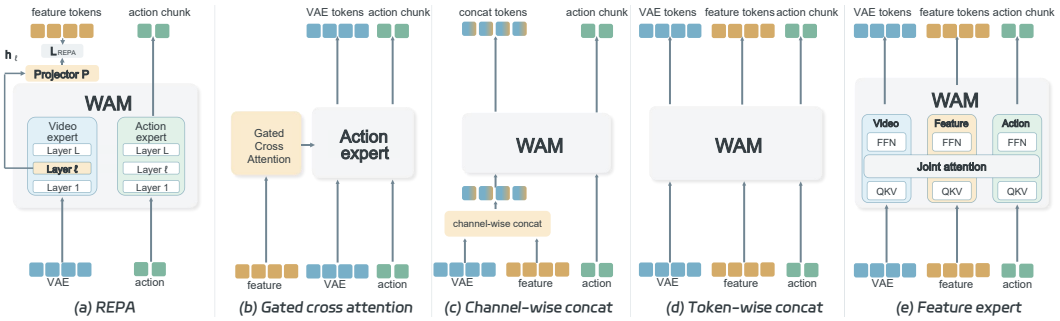

Figure 2: Injection mechanisms: (a) REPA, (b) gated cross attention, (c) channel-wise concat, (d) token-wise concat, and (e) feature expert.

where $\mathcal{L}_{\mathrm{FM}}$ denotes the modality-specific mean-squared error between predicted and target flow velocities, and λ_{act} weights the action loss. At inference, future-video latents and actions are initialized from noise and jointly denoised.

3.2 Visual Representations

We compare thirteen pretrained visual encoders, using their pretraining objectives as an initial indication of the information their representations capture and the visual capabilities they may support. All encoders remain frozen during WAM training and extract representations from the RGB observations available to the base WAM. We consider six pretraining emphases:

Visual self-supervision: VC-1 (Majumdar et al., 2023) and VideoMAE V2 (Wang et al., 2023) learn visual structure through masked image or video reconstruction. V-JEPA 2.1 (Mur-Labadia et al., 2026) predicts visual features to learn spatial and temporal relationships; DINOv3 (Siméoni et al., 2026) uses self-distillation to learn semantic and dense spatial features; and LingBot-Vision (Fu et al., 2026) emphasizes fine spatial structure through masked boundary modeling.

Language-associated pretraining: SigLIP 2 (Tschannen et al., 2025) combines image–text alignment with captioning and self-supervised objectives to learn semantic and dense visual features. Qwen3.5 (Qwen Team, 2026) uses multimodal language modeling for diverse visual understanding tasks.

Segmentation supervision: SAM 3 (Carion et al., 2025) learns object regions and boundaries through concept-conditioned detection and segmentation.

Geometric supervision: VGGT-Ω (Wang et al., 2026a) learns scene geometry through 3D reconstruction.

Interaction-oriented pretraining: HRP (Srirama et al., 2024) uses hand, object, and contact supervision to learn manipulation affordances. DynaFLIP (Lee et al., 2026) aligns images, language, and 3D motion to capture information relevant to interaction dynamics.

Multi-objective learning and multi-teacher distillation: UniTok (Ma et al., 2025) combines reconstruction and image–text learning to retain visual detail and semantics, while C-RADIOv4 (Ranzinger et al., 2026) distills DINOv3, SigLIP 2, and SAM 3.

Based on these pretraining objectives and capability emphases, we designate representations intended for broad transfer across visual or vision–language tasks as *general-purpose*, and those pretrained primarily for specific visual capabilities as *specialized*. Table 1 lists the sources and their group assignments.

3.3 Feature Integration Mechanisms

We compare five mechanisms that use external features for representation alignment, action conditioning, or joint prediction (Figure 2). The three joint-prediction mechanisms differ in how feature and VAE representations are fused and whether their Transformer parameters are shared.

Representation alignment (REPA): A trainable projector maps intermediate WAM hidden states into the external feature space. A cosine-distance loss aligns the projected states with spatially corresponding features from the frozen source encoder (Yu et al., 2025).

Gated cross-attention: An additional cross-attention module uses action hidden states as queries and projected features from the current observations as keys and values. A learnable gate scales the module's output before it is added to the action hidden states (Alayrac et al., 2022). The existing cross-attention to T5 language features is retained.

Channel-wise concatenation: We spatially align external features with the VAE latent grid and concatenate them along the channel dimension. The resulting representation is processed by the video expert, whose input and output projections are expanded to jointly predict VAE and external-feature flow velocities. The video-token layout and attention mask remain unchanged (Tan et al., 2026).

Token-wise concatenation. We represent external features as a separate token sequence with its own input projection and future-feature prediction head. Feature and VAE tokens share the video expert's Transformer layers (Yan et al., 2026a).

Feature expert: We introduce a dedicated Transformer expert to predict future external features. Unlike token-wise concatenation, this mechanism uses independent feature-processing parameters. Feature, video, and action tokens interact through masked joint attention, while each expert retains its own projections and feed-forward layers (Bi et al., 2026).

For REPA and the three joint-prediction mechanisms, the training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{WAM}} + \lambda_{\mathrm{aux}}\mathcal{L}_{\mathrm{aux}}, \tag{2}$$

where $\mathcal{L}_{\mathrm{aux}}$ is the cosine-distance alignment loss for REPA and the future-feature flow-matching loss for the joint-prediction mechanisms. Gated cross-attention is trained using $\mathcal{L}_{\mathrm{WAM}}$ alone.

We evaluate feature composition using channel-wise concatenation as the common integration mechanism. Features from each source are spatially aligned with the VAE latent grid and concatenated along the channel dimension.

4 Experiments

4.1 Experimental Setup

Training. We use the LIBERO(Liu et al., 2023) and RoboTwin 2.0 (Chen et al., 2026b) training datasets from Fast-WAM (Yuan et al., 2026) and the RoboCasa GR1 Tabletop (Nasiriany et al., 2024) dataset used in DIAL (Chen et al., 2026c). Across the LIBERO comparisons (RQ1–RQ3), we hold the training data, base observation inputs, action horizon, optimizer, and learning-rate schedule fixed while varying representation sources, integration mechanisms, or source combinations. Training configurations for simulation and real-robot experiments are provided in Appendix C.

Evaluation. We evaluate LIBERO (Liu et al., 2023) with 50 rollouts per task following Fast-WAM (Yuan et al., 2026); full results are reported in the Appendix D since performance is near saturation. The same models are evaluated on LIBERO-Plus (LP) (Fei et al., 2026) without further training, using three evaluation seeds and all 10,030 episodes per seed across seven perturbation categories. We report per-category and overall success rates. RoboCasa GR1 Tabletop (Nasiriany et al., 2024; Chen et al., 2026c) uses 50 trials per task across 24 tasks, while RoboTwin 2.0 (Chen et al., 2026b; Yuan et al., 2026) uses 100 trials per task across 50 tasks in each of the clean and randomized settings. We evaluate the VAE-only WAM and R^2-WAM on four real-robot tasks with 20 trials per task per model; metric definitions are provided in Appendix C.3.

4.2 RQ1: Which visual representations improve WAM performance?

We compare the visual representations introduced in Section 3.2. All representations are integrated using channel-wise concatenation, with the VAE-only WAM serving as the baseline. Table 1 reports success rates as mean $\pm$ standard deviation across three evaluation seeds for each of the seven LP perturbation categories and overall. We examine both overall performance and the range of perturbations under which each source improves robustness.

Table 1: RQ1: single-source success rates (%) across all seven LIBERO-Plus(Fei et al., 2026) perturbations under channel-wise concatenation. Sources are grouped as general-purpose or specialized, with subgroups indicating the dominant pretraining emphasis (Section 3.2). Results are reported as mean ± standard deviation across three evaluation seeds.

Source	Camera	Robot	Lang.	Light	Bkg.	Noise	Layout	Total
VAE-only	51.2±0.6	77.8±0.8	91.1±1.2	94.3±2.4	62.0±1.4	66.6±1.2	82.4±0.2	74.7±0.5
General-purpose representations								
Visual self-supervision								
DINOv3	**64.2±1.4**	82.5±0.9	93.8±0.8	**98.5±0.4**	83.9±0.4	86.0±0.3	83.6±0.8	**84.0±0.4**
V-JEPA 2.1	63.0±0.5	80.9±0.8	90.9±0.5	96.8±0.5	80.7±1.2	84.4±0.5	**84.2±0.5**	82.4±0.4
VC-1	60.8±1.0	78.1±0.5	93.3±0.4	98.4±0.4	82.0±0.6	86.7±1.3	83.0±0.9	82.5±0.4
LingBot-Vision	62.6±1.9	79.6±1.2	93.0±0.4	97.8±0.9	81.2±0.3	**87.7±0.5**	82.9±0.7	83.0±0.7
VideoMAE V2	53.6±1.4	81.4±0.8	91.3±1.1	95.4±0.3	68.0±1.0	73.1±0.5	82.7±0.5	77.5±0.6
Language-associated pretraining								
SigLIP 2	60.7±1.0	85.0±0.7	93.8±0.5	97.9±0.7	82.4±1.1	83.4±0.3	83.0±0.7	83.1±0.4
Qwen3.5	59.7±2.6	84.5±0.1	93.2±0.2	98.0±0.2	80.0±1.3	78.7±0.6	82.5±1.1	81.6±0.7
Multi-objective / multi-teacher learning								
UniTok	63.0±1.4	**85.2±0.6**	**93.9±1.0**	96.7±0.9	76.6±0.5	77.1±0.8	83.9±0.6	81.9±0.4
C-RADIOv4	62.6±1.5	81.0±1.3	92.2±1.0	97.3±1.2	**86.2±0.8**	86.6±0.7	83.6±0.8	83.5±0.6
Specialized representations								
Segmentation supervision								
SAM 3	57.8±1.3	80.8±1.5	93.7±0.7	95.6±1.2	75.3±1.0	81.5±1.0	82.4±0.6	80.6±0.4
Geometric supervision								
VGGT-Ω	62.4±0.7	78.8±0.8	92.7±0.4	95.8±0.6	75.7±0.2	79.6±0.5	82.5±0.9	80.6±0.3
Interaction-oriented pretraining								
HRP	52.6±1.0	77.8±1.0	91.9±0.3	96.9±1.0	82.2±1.3	81.2±0.8	82.4±0.6	79.8±0.4
DynaFLIP	55.2±0.9	83.1±1.0	92.3±0.6	93.9±1.4	69.9±0.1	77.5±0.8	82.1±1.3	78.8±0.3

Viewpoint, background, and noise perturbations expose key robustness gaps in the VAE-only WAM. The VAE-only WAM achieves high success rates under language perturbations (91.1%) and lighting changes (94.3%), yet performs substantially worse under viewpoint changes (51.2%), background changes (62.0%), and image noise (66.6%). This uneven robustness motivates examining whether external visual representations can address these weaknesses and improve performance across perturbation categories.

General-purpose representations show stronger robustness across perturbations. Across the sources evaluated in Table 1, general-purpose representations achieve a higher average success rate than specialized representations (82.2% vs. 80.0%). A general-purpose source also achieves the highest observed mean success rate in each of the seven perturbation categories, including camera, background, and image noise, where the VAE-only WAM is weakest.

Takeaway 1. General-purpose visual representations should be the default for augmenting WAMs. Rather than choosing a specialized encoder to remedy an assumed missing capability, start with a general-purpose visual representation. In our study, general-purpose representations provide broader robustness gains and can match or exceed specialized alternatives even where their specialization appears relevant.

4.3 RQ2: Which feature integration mechanism yields the greatest performance gains from external features?

We compare the five integration mechanisms in Section 3.3 across thirteen visual representations (Figure 3(a); per-source results in Appendix D). We then examine future-feature prediction, predicted

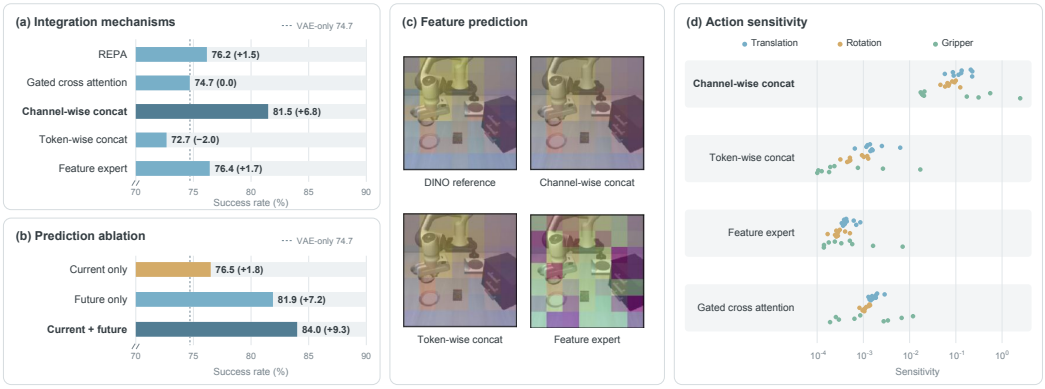

Figure 3: **Feature-integration performance and mechanism analysis.** (a) Average LP success rates across the thirteen representations in Table 1. (b) DINOv3(Siméoni et al., 2026) prediction ablations within channel-wise concat. (c) Shared-PCA features overlaid on future RGB observations. (d) Local sensitivity of final normalized actions to current DINOv3 features, inspired by gradient-based sensitivity and attribution (Simonyan et al., 2014; Hoffman et al., 2019; Yu et al., 2023). We evaluate action sensitivity at three selected time steps in each trajectory, corresponding to approach, grasp, and placement. Each point reports the sensitivity of one action group in the predicted action chunk at a selected time step. Metric and computation details are in Appendix D.3.

feature structure, and action sensitivity to investigate the gains from channel-wise concatenation (Figure 3(b–d)).

Channel-wise concatenation achieves the highest average success rate. Across the thirteen visual representations in Table 1, channel-wise concatenation achieves an average LP success rate of 81.5%, exceeding the next-best mechanism by 5.1 percentage points(Figure 3(a)).

Why channel-wise concatenation performs well. Figure 3(b) separates current-feature conditioning from future-feature prediction. Future-only prediction reaches 81.9%, close to the full configuration (84.0%) and above current-only conditioning (76.5%), supporting future-feature prediction as a major contributor.

The shared-PCA visualizations in Figure 3(c) show that the video expert recovers coarse spatial structure in future features. Token-wise predictions also retain recognizable structure, whereas the feature-expert prediction is coarser in the displayed example.

We analyze how action generation depends on external features (Figure 3(d)). Channel-wise concatenation exhibits higher translation and rotation sensitivities in these states, indicating stronger dependence of action generation on the added visual representations.

Takeaway 2. Use channel-wise concatenation as the default. For new designs, support future-feature prediction and encourage the action expert to make effective use of the added features. Check prediction quality and how added features affect generated actions, and verify that design changes improve task success rates.

4.4 RQ3: Does combining more feature sources consistently improve WAM performance?

Building on RQ1 and RQ2, we use DINOv3 (Siméoni et al., 2026) as the initial external representation and keep channel-wise concatenation fixed. The candidate additions span different pretraining emphases, with representatives from each category in Section 3.2: V-JEPA 2.1 (Mur-Labadia et al., 2026) for visual self-supervision, SigLIP 2 (Tschannen et al., 2025) for language-associated learning, SAM 3 (Carion et al., 2025) for segmentation, VGGT-Ω (Wang et al., 2026a) for geometry, HRP (Srirama et al., 2024) for interaction-oriented pretraining, and UniTok (Ma et al., 2025) and C-RADIOv4 (Ranzinger et al., 2026) for multi-objective / multi-teacher learning. We first add each

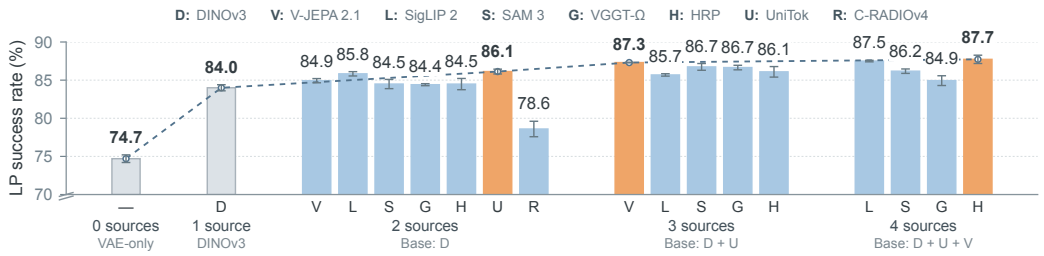

Figure 4: RQ3: progressive feature composition on LIBERO-Plus (mean ± SD over three evaluation seeds). Each bar in the two-, three-, and four-source groups adds its labeled source to the indicated base.

candidate separately to DINOv3, then evaluate selected larger combinations through single-source additions (Figure 4).

A general-purpose source provides the main gain. DINOv3 (Siméoni et al., 2026) alone improves LP success by 9.3 percentage points over VAE-only, while adding other features yields at most a further 3.7-point gain among the evaluated combinations.

The benefit of an additional source depends on the existing representations. The composition results suggest that overlap in pretraining objectives can limit complementary gains. Both UniTok (Ma et al., 2025) and SigLIP 2 (Tschannen et al., 2025) learn language-aligned visual representations. Adding SigLIP 2 improves DINOv3 (Siméoni et al., 2026) by 1.8 percentage points, but reduces success by 0.4 points once UniTok is included and adds only 0.2 points when V-JEPA 2.1 (Mur-Labadia et al., 2026) is also present. Spatial capabilities similarly overlap: DINOv3 already provides dense features supporting segmentation and depth estimation, while VGGT-Ω (Wang et al., 2026a) uses a DINOv3-initialized visual encoder. Adding VGGT-Ω or SAM 3 (Carion et al., 2025) to DINOv3 produces little change, although both improve some larger combinations. A direct teacher–student relationship occurs with C-RADIOv4 (Ranzinger et al., 2026), which distills DINOv3, SigLIP 2, and SAM 3: combining it with DINOv3 reduces LP success to 78.6%. Meanwhile, adding V-JEPA 2.1 (Mur-Labadia et al., 2026) to DINOv3 and UniTok improves performance, consistent with its video-based predictive pretraining providing complementary cues.

Takeaway 3. Additional sources should be selected for complementarity. Build on a strong general-purpose representation and use composition for selective refinement. When choosing additional sources, consider which task-relevant information they may contribute beyond the existing features, accounting for overlap in pretraining objectives and teacher models.

4.5 Representation Recipe and Validation

R^2-WAM combines DINOv3 (Siméoni et al., 2026), UniTok (Ma et al., 2025), and V-JEPA 2.1 (Mur-Labadia et al., 2026) through channel-wise concatenation. Adding HRP (Srirama et al., 2024) yields the highest mean LP success (87.7% versus 87.3%), but only a 0.4-percentage-point gain. We retain three sources to limit auxiliary encoders and feature-prediction targets.



Figure 5: Real-robot evaluation on four physical manipulation tasks.

Table 2: Validation Results across WAM formulations (a) and real-robot tasks (b). Real-robot evaluation uses 20 trials per task; metric definitions are in Appendix C.3.

(a) WAM formulations

Formulation	LIBERO	LP
Direct action prediction	96.8	70.8
+ recipe	**98.4**	**79.4**
Video-then-action (IDM)	98.6	73.6
+ recipe	**99.1**	**87.7**

(b) Real-robot tasks

Task	VAE-only	R^2-WAM
Air-fryer Food Preparation	50.0	**60.0**
Select and Pick-and-place	70.0	**80.0**
Sweep Rubbish	48.3	**66.7**
Push Cup to Target Position	65.0	**75.0**

Table 3: Comparison with selected reported methods on RoboCasa GR1 Tabletop (Nasiriany et al., 2024) and RoboTwin 2.0 (Chen et al., 2026b).

(a) RoboCasa GR1 Tabletop

Method	Success (%)
GR00T-N1.6 (NVIDIA GEAR Team, 2025)	47.6
DiT4DiT (Ma et al., 2026b)	50.8
FLARE (Zheng et al., 2025)	55.0
LDA-1B (Lyu et al., 2026)	55.4
DIAL (Chen et al., 2026c)	70.2
WALA (Liu et al., 2026)	75.2
LAWA (Li et al., 2026b)	80.8
VAE-only	78.8
R^2-WAM	**84.7** (+5.9)

(b) RoboTwin 2.0

Method	Avg. (%)
$\pi_{0.5}$ (Black et al., 2025)	79.8
Motus (Bi et al., 2026)	87.8
StarVLA-α (Ye et al., 2026a)	88.3
InternVLA-A1 (Cai et al., 2026)	89.5
Fast-WAM (Yuan et al., 2026)	91.8
LingBot-VA (Li et al., 2026a)	92.2
WALA-Wan (Liu et al., 2026)	92.3
VAE-only	90.6
R^2-WAM	**92.8** (+2.2)

Validation across WAM formulations. Table 2(a) compares VAE-only baselines with their recipe-augmented counterparts under two additional WAM formulations: direct action prediction (Yuan et al., 2026) and video-then-action prediction (IDM) (Li et al., 2026a). Implementation details are provided in Appendix B.1. These results support the benefit of the recipe across both evaluated formulations.

Cross-benchmark validation. R^2-WAM achieves a state-of-the-art success rate of 84.7% on RoboCasa GR1 Tabletop (Nasiriany et al., 2024). On RoboTwin 2.0 (Chen et al., 2026b), it reaches 92.8%. The gains over our VAE-only baselines are 5.9 and 2.2 percentage points, respectively, supporting the effectiveness of the same representation configuration across robot embodiments and environments.

Real-robot validation. Table 2(b) reports performance on the four physical tasks shown in Figure 5. R^2-WAM improves complete-task success by 10.0 percentage points on each of air-fryer food preparation, instructed pick-and-place, and cup pushing. On tool sweeping, the mean debris collection rate increases from 48.3% to 66.7%. These results extend the improvements from the selected representation recipe to physical manipulation, including sequential manipulation and tool use.

5 Conclusion

In this work, we systematically study how the selection, integration, and composition of external visual representations affect VAE-based WAMs. Our findings suggest a practical recipe: start with a strong general-purpose representation, integrate it through channel-wise concatenation, and retain additional sources when they improve task success rates in the intended combination. R^2-WAM instantiates this recipe and demonstrates gains across the evaluated robot embodiments and environments.

Our findings provide a starting point for further research on visual representations and their use in WAMs. Future work should focus on training stronger general-purpose visual representations and developing integration mechanisms that improve future-feature prediction and enable action experts to use visual information more effectively.

Reproducibility statement

Sections 3.1–3.3 describe the base WAM, representation sources, five injection mechanisms, and composition protocol. The experimental-design section defines training and evaluation settings, including benchmark denominators and aggregation. The appendix details feature construction and injection implementations.

AI use statement

Generative AI tools were used to assist with literature search, refinement of research questions and experimental designs, and revising portions of the manuscript. They also assisted with editing and formatting, LaTeX template migration, and the creation and revision of scientific figures. All AI-assisted outputs were reviewed by the authors. The authors take responsibility for the final content of the paper, including its text, claims, and artifacts.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, et al. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, 2022.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Kevin Black, Noah Brown, James Darpinian, et al. $\pi_{0.5}$: 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, 2025. URL https://proceedings.mlr.press/v305/black25a.html.

Junhao Cai, Zetao Cai, Jiafei Cao, et al. InternVLA-A1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026. URL https://arxiv.org/abs/2601.02456.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, et al. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Jialei Chen, Kai Wang, Kang Chen, et al. Lawam: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026a.

Tianxing Chen, Zanxin Chen, Baijun Chen, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *International Conference on Machine Learning*, 2026b.

Yi Chen, Yuying Ge, Hui Zhou, Mingyu Ding, Yixiao Ge, and Xihui Liu. DIAL: Decoupling intent and action via latent world modeling for end-to-end VLA. *arXiv preprint arXiv:2603.29844*, 2026c. URL https://arxiv.org/abs/2603.29844.

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: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Zelin Fu, Bin Tan, Changjiang Sun, Shaohui Liu, Kecheng Zheng, Yinghao Xu, Xing Zhu, Yujun Shen, and Nan Xue. Vision pretraining for dense spatial perception. *arXiv preprint arXiv:2607.05247*, 2026. URL https://arxiv.org/abs/2607.05247.

Judy Hoffman, Daniel A. Roberts, and Sho Yaida. Robust learning with Jacobian regularization. *arXiv preprint arXiv:1908.02729*, 2019. URL https://arxiv.org/abs/1908.02729.

Jusuk Lee, Seungjae Lee, Jonghun Shin, Hoseong Jung, Sungha Kim, Daesol Cho, H. Jin Kim, Jia-Bin Huang, and Furong Huang. DynaFLIP: Rethinking robotics perception via tri-modal-dynamics guided representation. *arXiv preprint arXiv:2605.30350*, 2026. URL https://arxiv.org/abs/2605.30350.

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

Xiang Li, Yupeng Zheng, Songen Gu, Huailiang Ma, Feng Yu, Yuhang Zheng, Xian Nie, Shanshuai Yuan, Yujie Zang, Weize Li, Shuai Tian, Moyang Liu, Ya-Qin Zhang, and Wenchao Ding. Latent action as intention enables efficient future imagination for world action models. *arXiv preprint arXiv:2608.24882*, 2026b. URL https://arxiv.org/abs/2608.24882.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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, Datasets and Benchmarks Track*, 2023.

Jiahao Liu, Zhongpu Xia, Shuai Tian, Huangrui Li, Yuhang Zheng, Ning Ma, Xin Fu, Xiaotian Liu, Jing Li, Yixian Li, ShangQing Zhou, Zebin Xing, Linbo Wang, Chaoyue Li, Haoran Li, and Dongbin Zhao. WALA learning executable latent actions from action-labeled demonstrations and action-free videos. *arXiv preprint arXiv:2607.11397*, 2026. URL https://arxiv.org/abs/2607.11397.

Jiangran Lyu, Kai Liu, Xuheng Zhang, et al. Lda-1b: Scaling latent dynamics action model via universal embodied data ingestion. In *Proceedings of Robotics: Science and Systems*, 2026.

Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. UniTok: A unified tokenizer for visual generation and understanding. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4306. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/bbe04d329d531e8814f1199098bd8fb6-Abstract-Conference.html.

Fulong Ma, Daojie Peng, Wenjun Yue, Jiahang Cao, Bintao Wang, Qiang Zhang, and Jun Ma. Geosemwam: Geometry- and semantic-aware world action models. *arXiv preprint arXiv:2606.03188*, 2026a.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. DiT4DiT: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026b. URL https://arxiv.org/abs/2603.10448.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0031. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/022ca1bed6b574b962c48a2856eb207b-Abstract-Conference.html.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mahmoud Assran, Koustuv Sinha, Michael Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. In *European Conference on Computer Vision*, 2026.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of household tasks for generalist robots. In *Proceedings of Robotics: Science and Systems*, 2024.

NVIDIA GEAR Team. GR00T N1.6: An improved open foundation model for generalist humanoid robots. NVIDIA Research, 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_6/.

Lu Qiu, Yizhuo Li, Yi Chen, Yuying Ge, Yixiao Ge, and Xihui Liu. Making foresight actionable: Repurposing representation alignment in world action models. *arXiv preprint arXiv:2606.12217*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Alec Radford, Jong Wook Kim, Chris Hallacy, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2020.

Mike Ranzinger, Greg Heinrich, Collin McCarthy, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. C-RADIOv4 (tech report). *arXiv preprint arXiv:2601.17237*, 2026.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, et al. Dinov3. *Transactions on Machine Learning Research*, 2026.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations Workshop*, 2014.

Mohan Kumar Srirama, Sudeep Dasari, Shikhar Bahl, and Abhinav Gupta. HRP: Human affordances for robotic pre-training. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024. doi: 10.15607/RSS.2024.XX.068. URL https://www.roboticsproceedings.org/rss20/p068.html.

Boming Tan, Xiangdong Zhang, Ning Liao, Yuqing Zhang, Shaofeng Zhang, Xue Yang, Qi Fan, and Yanyong Zhang. Dreamworld: Unified world modeling in video generation. *arXiv preprint arXiv:2603.00466*, 2026.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

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

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a.

Junke Wang, Qihang Zhang, Shuai Yang, Yiming Luo, Yujun Shen, Zuxuan Wu, Yu-Gang Jiang, and Yinghao Xu. Repwam: World action modeling with representation visual-action tokenizers. *arXiv preprint arXiv:2606.13674*, 2026b.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE V2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14549–14560, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Wang_VideoMAE_V2_Scaling_Video_Masked_Autoencoders_With_Dual_Masking_CVPR_2023_paper.html.

Mingxin Wang, Bin Hu, Bin Qian, et al. St-wam: Semantic-temporal world action model for robust manipulation under visual distribution shifts. *arXiv preprint arXiv:2607.28993*, 2026c.

Ge Yan, Jinghao Liu, Yuzhi Fan, Lei Cai, Minwen Liao, Jesse Zhang, and Dieter Fox. Flex-π: A multi-stream world-action model with compute flexibility. *arXiv preprint arXiv:2608.10860*, 2026a.

Haodong Yan, Junfeng Li, Junjie He, et al. Robust-wam: Bridging generative pretraining and semantic foresight in world-action models. *arXiv preprint arXiv:2608.05903*, 2026b.

Lishan Yang, Wenxuan Song, Xi Wang, Pingyue Sheng, Zheng Fang, Ziyang Zhou, Junjie He, Haodong Yan, Jiayi Chen, Nan Sun, Qiao Sun, Pengwei Wang, Lingqiao Liu, Yan Wang, Yuxiang Gao, Feras Dayoub, and Haoang Li. 4D-WAM: Infusing spatiotemporal awareness into world action models through trajectory fields. *arXiv preprint arXiv:2608.08023*, 2026. URL https://arxiv.org/abs/2608.08023.

Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. StarVLA-α: Reducing complexity in vision-language-action systems. *arXiv preprint arXiv:2604.11757*, 2026a. URL https://arxiv.org/abs/2604.11757.

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*, 2026b.

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.

Wanming Yu, Chuanyu Yang, Christopher McGreavy, Eleftherios Triantafyllidis, Guillaume Bellegarda, Milad Shafiee, Auke Jan Ijspeert, and Zhibin Li. Identifying important sensory feedback for learning locomotion skills. *Nature Machine Intelligence*, 5:919–932, 2023. doi: 10.1038/s42256-023-00701-w. URL https://www.nature.com/articles/s42256-023-00701-w.

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.

Jianjun Zhang, Jian Zhu, Taiyi Su, Chong Ma, Zitai Huang, Yi Xu, and Hanli Wang. Learning 4D geometric priors for inference-efficient world action models. *arXiv preprint arXiv:2607.05468*, 2026a. URL https://arxiv.org/abs/2607.05468.

Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, Jiaming Zhou, Yishu Shen, Yudong Jin, Fangyi Xu, Shuailei Ma, Jiaqi Liao, Guanxing Lu, Zifan Shi, Yongkun Wen, Yujie Zhao, Weixuan Tang, Xinyang Wang, Chaojian Li, Jiapeng Zhu, Ka Leong Cheng, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Native video-action pretraining for generalizable robot control. *arXiv preprint arXiv:2607.08639*, 2026b.

Ruijie Zheng, Jing Wang, Scott Reed, et al. FLARE: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025. URL https://arxiv.org/abs/2505.15659.

A Visual Representation Construction

Encoder configurations. We primarily use large (L) variants of the visual encoders. For Qwen3.5, we use Qwen3.5-4B.

Feature extraction. We use token features from the final layer of each encoder. For Qwen3.5, we extract the final-layer hidden states at visual-token positions from a single forward pass.

B Feature Integration Implementation Details

B.1 WAM Formulations

We implement three WAM formulations within our base architecture, differing in the visual information available to the action expert and the training and inference procedures (Figure 6).

Joint video–action denoising (default). Future visual tokens and action tokens are jointly denoised. As described in Section 3.1, action tokens attend to current-frame visual tokens, noisy future visual tokens, and action tokens, whereas visual tokens do not attend to action tokens. At inference, future visual tokens and actions are initialized from noise and jointly denoised.

Direct action prediction. Following the direct-prediction paradigm of Fast-WAM (Yuan et al., 2026), the attention mask restricts the action expert's visual context to current-frame tokens, preventing access to future visual tokens. At inference, actions are denoised conditioned on the current observation without generating future visual tokens.

Video-then-action prediction (IDM). Following the inverse-dynamics paradigm of LingBot-VA (Li et al., 2026a), the action expert predicts actions conditioned on current and future visual tokens. During training, it receives clean ground-truth future visual tokens. At inference, the video expert first predicts future visual tokens, which then serve as conditions for action denoising.

VAE-only baselines. For each formulation, VAE-only denotes the corresponding model using VAE representations without additional external visual representations. Its recipe-augmented counterpart incorporates DINOv3, UniTok, and V-JEPA 2.1 through channel-wise concatenation.

B.2 Integration Mechanisms

We denote current and future VAE latents by z_0 and z_+, and the corresponding external features by f_0 and f_+. Each integration mechanism adapts source features to its required spatiotemporal layout while preserving correspondence between camera views.

Representation alignment (REPA). The student representation is the output of the eighth video-expert block. A trainable three-layer MLP P maps this state to the source feature dimension, with SiLU after the first two linear layers. The teacher trajectory is aligned to the video-token grid: spatial resizing is bilinear, the observed temporal slice is preserved, and the eight future slices are averaged into two groups of four. Padding masks are aligned with the features and remove invalid locations. After ℓ_2 normalization, the alignment objective is

$$\mathcal{L}_{\mathrm{REPA}} = 1 - \frac{1}{|\Omega|} \sum_{i \in \Omega} \left\langle \widehat{P(h_i^8)}, \widehat{f}_i \right\rangle, \tag{3}$$

where Ω contains valid future tokens and excludes the observed slice. The frozen teacher and projector are used only during training and are absent from inference.

Gated cross-attention. An independent feature cross-attention branch is inserted into each of the 30 action-expert blocks. The normalized action state after mixed attention is used as the query, while normalized features from the current observation f_0 provide keys and values through an adapter to the action-expert width. The original T5 language cross-attention is retained, and both residual branches are computed from the same post-mixed-attention state $\bar{h}_a^\ell$:

$$\widetilde{h}_a^\ell = \bar{h}_a^\ell + \mathrm{CA}_{\mathrm{text}}^\ell(\bar{h}_a^\ell, c) + \tanh(g_\ell)\ \mathrm{CA}_{\mathrm{feat}}^\ell(\bar{h}_a^\ell, f_0). \tag{4}$$

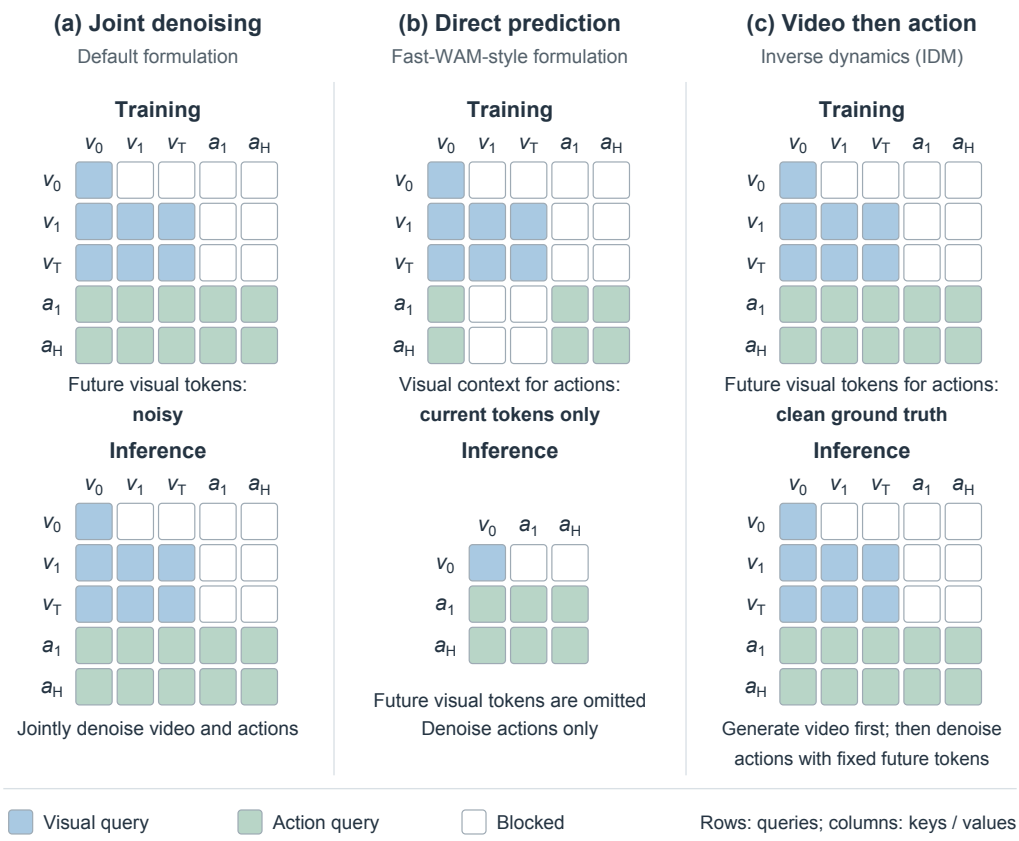

Figure 6: Attention masks for training and inference in our three WAM formulations. Rows are queries and columns are keys/values; filled cells permit attention and empty cells block it. v_0, $v_{1:T}$, and $a_{1:H}$ denote current visual, future visual, and action tokens, respectively. Representative temporal positions are shown, with each visual entry representing a block of spatial tokens. Direct prediction omits future visual tokens at inference. Joint denoising and IDM share the displayed connectivity but differ in future-token conditioning: jointly denoised tokens versus clean ground-truth tokens during IDM training and previously generated, fixed tokens during IDM action inference.

The resulting state is subsequently processed by the block FFN. Every layer has its own scalar gate, initialized with $g_\ell = 0$, so the added branch is initially neutral. This mechanism has no auxiliary feature loss. The frozen visual encoder and feature cross-attention branches remain active at inference.

Channel-wise concatenation. Following the channel-wise integration design of DreamWorld (Tan et al., 2026), we align external visual features with the VAE latent grid and concatenate them along the channel dimension. This preserves the video token grid and attention sequence length. The input and output projections are expanded to accommodate the added features while retaining the pretrained VAE weights. Observed features remain clean, whereas future VAE latents and external features are jointly noised and predicted. At inference, future features are initialized from noise and participate in joint denoising; only the VAE latents are decoded.

Token-wise concatenation. Dense external features are flattened into a separate token sequence, mapped from C_f to the video-expert hidden width by an independent input projection, and appended to the VAE tokens. Feature and VAE tokens share all video-expert Transformer blocks, while a separate prediction head maps feature hidden states back to the source dimension C_f. Features are flattened in (T, H, W) order into observed tokens F_0 and future tokens F_+. LayerNorm precedes both the input projection and the prediction head; their linear projections use Xavier-uniform weights and zero biases. The combined token ordering is $[V_0, V_+, F_0, F_+, A]$.

Feature tokens use the video expert's three-dimensional rotary positions, with feature-grid coordinates mapped to the video-token coordinate system. The attention mask allows observed video and feature tokens to interact only with observed visual tokens. Future video and feature tokens jointly attend to both observed and future video/feature streams. Action tokens attend to observed and noisy future video/feature tokens, as well as action tokens. Video and feature tokens do not attend to action tokens.

During training, F_0 remains clean and F_+ receives Gaussian noise sampled independently from the video noise. Feature and video flows use the same sampled timestep and the same scheduler, with feature velocity target $\epsilon_f - f$. The scheduler-weighted MSE is evaluated over valid future feature tokens only. At inference, F_0 is extracted only from the observed views, F_+ is initialized from Gaussian noise, and both are jointly denoised with video and action. The clean F_0 prefix is restored after every solver step, and the final predicted feature trajectory is discarded.

Feature expert. This mechanism introduces an independent 30-block Transformer expert with hidden width 1024, FFN width 4096, 24 attention heads, and head dimension 128. Its input and output projections map between the source dimension C_f and the expert hidden width. These projections are newly initialized, while the remaining Transformer, language-conditioning, and time-conditioning parameters are initialized from the same pretrained ActionDiT backbone used by the action expert.

At every layer, the video, feature, and action experts independently form their Q/K/V tensors. The tensors are concatenated for masked joint attention, then split by modality before each expert applies its own attention output projection, text cross-attention, normalization layers, and FFN. Joint attention does not imply parameter sharing: the feature expert owns independent Transformer parameters. Its attention mask follows the same routing as token-wise concatenation: action queries attend to observed and noisy future video/feature tokens and action tokens, while video and feature queries cannot attend to action tokens. Feature noise is independent of video noise, while the feature and video flows share the same timestep and scheduler. The observed feature prefix remains clean, and only valid future features enter the flow-matching loss. During inference, future features are initialized from noise and participate in joint denoising; the observed prefix is restored after every step, and the predicted feature trajectory is discarded.

Unless otherwise specified, auxiliary feature-prediction losses use a weight of 1.0. The feature expert uses a feature-prediction loss weight of 0.1, and REPA uses an alignment weight of 0.1. Preliminary sensitivity experiments across five visual encoders compare REPA depths of 8, 15, and 22 at this weight, and weights of 0.1 and 0.5 at depth 8 (Table 5). Increasing the alignment weight does not improve the mean LIBERO-Plus success rate, and the mean varies by less than 0.1 percentage points across the four configurations, although individual encoders exhibit different preferences.

We compare five feature-integration mechanisms using a common WAM backbone and training budget. The feature preprocessing and auxiliary-loss settings for each mechanism are specified above.

Table 4: Implementation summary of the five feature-injection mechanisms.

Mechanism	**Representation interface**	**Future-feature prediction**	**Transformer parameters**	**Auxiliary objective**	**Feature pathway at inference**
REPA	hidden-state alignment	No	base WAM only	cosine alignment	removed
Gated cross-attention	current feature memory	No	added cross-attention	none	retained
Channel-wise concat.	feature channels	Yes	shared video expert	feature flow matching	retained during denoising
Token-wise concat.	feature tokens	Yes	shared video Transformer	feature flow matching	retained during denoising
Feature expert	feature tokens	Yes	independent expert	feature flow matching	retained during denoising

Table 5: **REPA sensitivity to alignment depth and loss weight.** Preliminary LIBERO-Plus success rates (%) across five visual encoders. Each entry reports one evaluation of 10,030 episodes using a single trained checkpoint per configuration. Mean is the unweighted average across the five encoders.

Encoder	$d = 8$ $\lambda = 0.1$	$d = 15$ $\lambda = 0.1$	$d = 22$ $\lambda = 0.1$	$d = 8$ $\lambda = 0.5$
CLIP	77.51	76.56	75.98	77.47
DINOv3	78.05	77.36	77.86	78.02
SigLIP 2	76.28	78.57	77.25	76.10
VGGT-Ω	76.43	76.00	76.85	78.23
V-JEPA 2.1	75.36	75.50	75.89	73.73
Mean	**76.73**	**76.80**	**76.77**	**76.71**

C Training and Evaluation Configurations

C.1 Simulation Training

Table 6 lists the common training settings, and Table 7 summarizes the benchmark-specific configurations. Within the main LIBERO comparisons (RQ1–RQ3), these settings are held fixed while the representation source, integration mechanism, or source set varies as specified in the main text.

Table 6: **Common training settings for the simulation benchmarks.**

Parameter	Value
Optimizer	AdamW
Initial learning rate	1×10^{-4}
Minimum learning rate	1×10^{-6}
Learning-rate schedule	Cosine decay
Warmup	Linear, first 5% of training
Weight decay	0.01
Adam betas	$(0.9, 0.95)$
Gradient clipping	1.0 (maximum norm)
Numerical precision	bfloat16
Action loss weight	1.0
Action horizon	32 steps
Video frames per view	9
Frame sampling offsets	$0, 4, \ldots, 32$ (33-step window)

C.2 Simulation Evaluation and Reporting

Benchmark coverage and aggregation. LIBERO uses 50 rollouts per task across 40 tasks, giving 2,000 rollouts per evaluation; we report the average across its four equally sized suites. A complete LIBERO-Plus evaluation contains 10,030 episodes across seven perturbation categories. Overall LP success is the total number of successes divided by the total number of evaluated episodes, rather than an unweighted average of category success rates. Each category uses its own denominator. RoboCasa GR1 Tabletop uses 50 trials per task across 24 tasks. RoboTwin 2.0 uses 100 trials per task across 50 tasks in each of the clean and randomized settings; its reported average weights these settings equally.

Instruction processing. The official LIBERO-Plus task instructions include appended descriptions of the perturbation types. We remove these descriptions for evaluation, while retaining the rewritten instructions in the language perturbation category.

Evaluation seeds and configuration matching. Results use one trained checkpoint per configuration. The RQ1 summaries and RQ3 composition results report means and sample standard deviations

Table 7: **Benchmark-specific training configurations.** Image sizes are per view.

Parameter	LIBERO	RoboCasa	RoboTwin 2.0
Training data	LIBERO (four suites)	GR1 Tabletop	Clean + randomized demonstrations
Number of tasks	40	24	50
Dataset size (demos)	1,712	24,000	27,500
RGB views	2	1 (ego)	3
Image resolution	224×224	224×224	240×320
Action dimension	7	47	14
Global batch size	128	1024	1024
Training epochs	10	5	5

Dataset sizes are counted before the validation split. The processed LIBERO datasets used by our runs contain 1,712 episodes; the original four-suite corpus contains 2,000 demonstrations (500 per suite) (Yuan et al., 2026). RoboCasa contains 1,000 demonstrations per task. RoboTwin combines 2,500 clean-scene and 25,000 randomized-scene demonstrations (Yuan et al., 2026).

across three evaluation seeds; these quantify evaluation variability, not independent training-seed variability. Other mechanism entries in Table 8 retain their recorded evaluations, as specified in its note. Qwen3.5 gated cross-attention, token-wise concatenation, and feature-expert LP evaluations cover 9,995, 10,008, and 10,008 recorded episodes, respectively, and use their actual denominators. Their incomplete coverage is identified in the result tables. The updated channel-wise Qwen3.5 results use the three-evaluation means in Table 1.

REPA uses layer 8 and auxiliary weight 0.1. Alternative readouts, auxiliary weights, and implementations define separate configurations. Perturbation breakdowns are reported only when they match the displayed configuration and total; missing matching entries remain marked with dashes.

C.3 Real-Robot Training and Evaluation

Robot setup. We conduct all four real-robot tasks on the same Galbot G1 platform, equipped with two arms and grippers. The model takes images from one head-mounted camera and two wrist-mounted cameras as input. Actions consist of joint positions and gripper states.

Training configuration. We use 200 demonstrations each for air-fryer food preparation and instructed pick-and-place, and 100 each for tool sweeping and cup pushing. Real-robot models are trained for 10 epochs with a batch size of 512.

Evaluation protocol. We evaluate the VAE-only WAM and R^2-WAM on the four real-robot tasks in Table 2, with 20 trials per task for each model. Complete-task success is reported for air-fryer food preparation, instructed pick-and-place, and cup pushing; tool sweeping is evaluated using the mean debris collection rate, as defined below.

Complete-task success. For air-fryer food preparation, instructed pick-and-place, and cup pushing, we report the percentage of trials in which the full task is completed. Air-fryer food preparation requires opening the air fryer, placing the corn cob inside, and closing it; instructed pick-and-place requires selecting the instructed object and placing it at the specified location; cup pushing requires pushing the paper cup next to the tissue box.

Debris collection rate. Each sweeping trial starts with six pieces of debris. Let $c_i \in \{0, \ldots, 6\}$ denote the number swept into the dustpan in trial i. We report the mean collection rate, $\frac{100}{20} \sum_{i=1}^{20} \frac{c_i}{6}$, as a percentage. This metric measures partial task completion rather than binary success, with each rollout treated as one evaluation trial. All reported scores are rounded to one decimal place.

D Feature Integration Results and Diagnostics for RQ2

D.1 Full Source–Mechanism Results

Table 8 reports LIBERO averages and LP perturbation-level and overall success rates for the thirteen main-study representations across all five integration mechanisms. The main-study channel-wise LP results match the means in Table 1.

Table 8: Detailed LIBERO and LIBERO-Plus results for REPA, gated cross-attention, channel-wise concatenation, token-wise concatenation, and the feature expert.

	LIBERO	LIBERO-Plus							
Source	Avg.	Camera	Robot	Lang.	Light	Bkg.	Noise	Layout	Total
VAE-only reference	97.9	51.2	77.8	91.1	94.3	62.0	66.6	82.4	74.7
REPA									
DINOv3	98.4	59.0	75.5	92.2	96.2	67.8	75.1	83.0	78.1
V-JEPA 2.1	97.9	52.7	76.3	91.0	92.6	64.9	71.6	80.9	75.4
VC-1	97.5	53.7	75.5	90.6	95.7	60.5	69.6	81.1	74.9
LingBot-Vision	98.0	57.2	76.2	91.8	95.3	63.1	75.1	81.0	76.9
VideoMAE V2	98.5	58.9	78.1	92.3	95.5	64.9	73.7	82.9	77.8
SigLIP 2	98.4	53.6	76.3	91.4	97.4	65.4	71.0	82.2	76.3
Qwen3.5	98.1	53.5	78.5	92.8	95.7	63.8	68.8	81.0	75.9
SAM 3	97.8	54.7	78.3	91.9	94.7	63.0	68.8	79.5	75.5
VGGT-Ω	98.2	53.4	78.6	93.0	96.5	62.6	72.4	80.7	76.4
HRP	98.0	53.4	75.1	90.3	96.2	60.8	73.0	81.6	75.5
DynaFLIP	97.9	53.5	75.2	92.1	96.1	66.4	76.2	81.2	76.8
UniTok	98.2	57.3	75.6	91.6	94.9	62.4	73.8	80.1	76.3
C-RADIOv4	97.9	53.0	75.9	90.8	95.9	59.4	67.7	80.1	74.4
Gated cross-attention									
DINOv3	97.9	51.1	74.3	89.9	91.9	61.3	69.0	82.0	73.9
V-JEPA 2.1	98.1	48.8	79.0	89.5	94.7	63.7	66.7	81.3	74.3
VC-1	97.9	53.7	78.7	89.9	95.1	64.9	70.1	81.1	75.8
LingBot-Vision	97.8	49.1	75.9	92.0	96.7	63.1	67.8	81.3	74.6
VideoMAE V2	98.3	53.0	77.9	90.6	94.6	63.8	70.9	82.6	75.8
SigLIP 2	97.9	53.4	77.2	89.2	95.7	64.6	68.5	82.3	75.4
Qwen3.5[†]	98.1	49.0	79.4	92.1	90.4	61.3	71.1	81.6	74.8
SAM 3	98.5	52.6	77.4	90.3	94.3	61.2	72.0	81.1	75.3
VGGT-Ω	98.6	50.6	77.9	92.9	96.8	61.2	67.1	82.0	75.1
HRP	98.2	51.2	76.1	89.9	92.1	59.8	67.6	80.5	73.6
DynaFLIP	98.3	52.5	75.3	87.6	95.3	63.9	71.1	79.9	74.6
UniTok	98.2	50.0	74.2	89.9	93.0	63.9	69.8	81.6	74.2
C-RADIOv4	98.3	49.1	74.1	88.9	94.9	62.6	66.0	82.2	73.5

Continued on next page

Table 8: Detailed LIBERO and LIBERO-Plus results (continued).

	LIBERO	LIBERO-Plus							
Source	Avg.	Camera	Robot	Lang.	Light	Bkg.	Noise	Layout	Total
Channel-wise concatenation									
DINOv3	98.4	64.2	82.5	93.8	98.5	83.9	86.0	83.6	84.0
V-JEPA 2.1	98.5	63.0	80.9	90.9	96.8	80.7	84.4	84.2	82.4
VC-1	98.5	60.8	78.1	93.3	98.4	82.0	86.7	83.0	82.5
LingBot-Vision	98.1	62.6	79.6	93.0	97.8	81.2	87.7	82.9	83.0
VideoMAE V2	98.7	53.6	81.4	91.3	95.4	68.0	73.1	82.7	77.5
SigLIP 2	98.5	60.7	85.0	93.8	97.9	82.4	83.4	83.0	83.1
Qwen3.5	98.5	59.7	84.5	93.2	98.0	80.0	78.7	82.5	81.6
SAM 3	98.3	57.8	80.8	93.7	95.6	75.3	81.5	82.4	80.6
VGGT-Ω	98.4	62.4	78.8	92.7	95.8	75.7	79.6	82.5	80.6
HRP	98.6	52.6	77.8	91.9	96.9	82.2	81.2	82.4	79.8
DynaFLIP	98.0	55.2	83.1	92.3	93.9	69.9	77.5	82.1	78.8
UniTok	99.1	63.0	85.2	93.9	96.7	76.6	77.1	83.9	81.9
C-RADIOv4	98.2	62.6	81.0	92.2	97.3	86.2	86.6	83.6	83.5
Token-wise concatenation									
DINOv3	97.5	49.8	77.7	92.8	95.6	68.3	70.8	81.7	76.1
V-JEPA 2.1	91.2	19.0	57.3	62.1	68.6	33.3	43.3	64.5	49.5
VC-1	97.9	55.0	77.4	92.1	95.5	60.3	69.8	81.8	75.8
LingBot-Vision	98.1	52.3	77.1	87.3	95.4	68.6	74.2	81.1	76.0
VideoMAE V2	97.9	52.7	77.0	91.3	93.3	65.9	73.5	81.1	76.1
SigLIP 2	94.3	35.6	68.6	84.2	89.5	59.7	56.1	76.9	66.4
Qwen3.5[†]	98.2	49.5	79.4	90.3	96.5	62.6	72.5	80.9	75.5
SAM 3	98.3	45.2	76.9	87.8	95.4	69.1	68.6	82.1	74.3
VGGT-Ω	96.7	48.0	73.8	88.8	89.7	74.0	70.5	81.3	74.4
HRP	97.9	54.2	75.2	89.6	94.2	66.2	69.7	82.1	75.4
DynaFLIP	97.5	54.1	74.0	86.3	94.1	60.5	71.0	80.2	74.0
UniTok	98.4	57.5	77.6	91.9	96.9	63.5	74.6	81.4	77.4
C-RADIOv4	96.8	49.9	72.5	84.5	93.5	67.2	72.0	80.9	73.8
Feature expert									
DINOv3	98.3	57.4	81.2	92.5	96.2	66.9	77.9	80.7	78.7
V-JEPA 2.1	98.8	47.8	78.6	90.0	94.1	59.0	70.3	79.2	73.9
VC-1	97.9	54.5	79.9	95.0	89.8	66.0	72.3	81.8	76.9
LingBot-Vision	98.2	54.8	78.6	95.7	91.6	61.6	76.1	80.4	77.0
VideoMAE V2	97.7	54.8	79.6	89.7	96.5	69.9	78.1	82.2	78.2
SigLIP 2	98.1	53.7	80.1	93.7	95.9	64.4	73.3	81.2	77.2
Qwen3.5[†]	97.0	45.9	77.3	91.0	90.1	60.7	69.2	81.0	73.3
SAM 3	98.3	57.3	76.8	91.3	95.7	61.5	74.1	81.8	76.8
VGGT-Ω	97.9	51.6	77.5	89.7	88.3	63.6	69.0	81.5	74.2
HRP	98.6	55.4	80.7	92.9	96.5	65.1	76.7	82.4	78.3
DynaFLIP	97.8	52.1	80.0	91.7	93.6	65.2	75.0	79.5	76.4
UniTok	98.9	50.5	79.2	94.0	94.6	62.2	70.8	81.3	75.8
C-RADIOv4	98.0	53.3	76.6	91.7	95.5	66.7	74.1	81.9	76.7

All values are success rates (%). LIBERO reports the four-suite average; LP Total is the full-benchmark micro-average. The LP results for the VAE-only reference and the thirteen main-study sources under channel-wise concatenation use the three-evaluation means in Table 1; other results retain their recorded evaluations. Results use one training seed. Each mechanism is evaluated with the thirteen main-study sources. [†]Qwen3.5 LP results cover 9,995 episodes for gated cross-attention and 10,008 for token-wise concatenation and the feature expert, out of 10,030; each category uses its recorded denominator. REPA uses layer 8 and loss weight 0.1.

Figure 3(a) averages LP success across the thirteen main-study representations with equal weight per representation. Averages and differences are computed from the reported one-decimal scores.

D.2 Current-Feature Conditioning and Future-Feature Prediction

We compare three DINOv3 configurations under channel-wise concatenation (Figure 3(b)):

Table 9: DINOv3 prediction ablations under channel-wise concatenation.

Configuration	Current-feature input	Future-feature prediction	LP success (%)
Current only	Yes	No	76.5
Future only	No	Yes	81.9
Current + future	Yes	Yes	84.0

The current-only configuration removes the future-feature channels and their prediction loss; no future features are generated at inference. The future-only configuration omits the current-frame DINOv3 features while retaining future-feature supervision and denoising. The full configuration includes both current-feature conditioning and future-feature prediction. All three configurations use the same attention-mask rule, training duration, remaining loss weights, and evaluation settings.

Feature prediction visualization. Following LDA-1B (Lyu et al., 2026), we visualize DINOv3 features by projecting them onto three principal components and mapping these components to RGB colors. A shared PCA projection and color scale are used for the reference and predictions. The resulting feature maps are overlaid on the ground-truth future RGB image to illustrate their spatial structure.

D.3 Local Action Sensitivity Analysis

We detail the gradient diagnostic in Figure 3(d). The analysis is inspired by gradient-based sensitivity and attribution methods (Simonyan et al., 2014; Hoffman et al., 2019; Yu et al., 2023); our statistic uses local Jacobians, rather than integrated gradients along a baseline-to-input path.

Differentiation target and sampling path. Let F denote the raw current-frame DINOv3 features from the agent-view and wrist cameras. Each view has a 14×14 grid of 1,024-dimensional features. For mechanism m, we write the final normalized action chunk as $A_m(F;\xi) \in \mathbb{R}^{32\times 7}$, where ξ collects the fixed non-feature inputs and realized sampling noise. We differentiate through the full ten-step denoising process, including each mechanism's transformations of the raw features, while holding the original VAE latents and other inputs fixed. The derivative therefore measures total local dependence on the current external-feature input across the sampling computation. It does not isolate dependence on a generated future-feature state.

We analyze channel-wise concatenation, token-wise concatenation, the feature expert, and gated cross-attention. REPA has no external-feature input at inference, so this diagnostic is not applicable to it. We do not assign it a zero sensitivity.

Action groups and gradient estimator. We flatten translation, rotation, and gripper outputs into separate vectors a_g with dimensions $d_g = 96$, 96, and 32, respectively. Within each camera, each non-overlapping 2×2 block of raw feature locations forms one cell p on a 7×7 diagnostic grid. Let $F_{c,p} \in \mathbb{R}^{4096}$ contain all feature channels in this block, and let

$$J_{g,c,p} = \frac{\partial a_g}{\partial F_{c,p}} \tag{5}$$

be its action–feature Jacobian. The local score is the Frobenius norm of this block Jacobian divided by $\sqrt{d_g}$, i.e., the root-mean-square input-gradient norm over the action outputs in group g.

To avoid constructing the full Jacobian, we use random output projections (Hoffman et al., 2019). For each group, draw independent Rademacher vectors $z_r \in \mathbb{R}^{d_g}$ with entries $\pm 1/\sqrt{d_g}$ and compute vector–Jacobian products by automatic differentiation:

$$\widehat{s}_{g,c,p} = \left[\frac{1}{R} \sum_{r=1}^{R} \left\| \frac{\partial (z_r^{\mathsf{T}} a_g)}{\partial F_{c,p}} \right\|_2^2 \right]^{1/2}. \tag{6}$$

Since $\mathbb{E}[z_r z_r^\top] = I/d_g$, the quantity inside the square root estimates $\|J_{g,c,p}\|_F^2/d_g$. We then average the local scores over the 49 cells and two camera views:

$$\widehat{S}_{m,g}(o) = \frac{1}{2 \cdot 49} \sum_{c=1}^{2} \sum_{p=1}^{49} \widehat{s}_{m,g,c,p}(o), \tag{7}$$

where o identifies the observation state. Averaging occurs after the square root; this statistic is a mean of local sensitivity estimates, not the Frobenius norm of the entire camera Jacobian. No feature-update magnitude is multiplied into the score.

Observation states and plotted points. All four mechanisms use the same nine fixed observation states: approach, grasp, and placement in each of three LIBERO trajectories. The recorded frame indices are $(28, 36, 93)$ for `spatial_e000`, $(34, 42, 136)$ for `object_e000`, and $(24, 32, 84)$ for `goal_e114`, in approach–grasp–placement order. The `spatial_e000_f028_approach` state uses $R = 16$ probes per action group; the other eight states use $R = 8$, consistently across mechanisms.

Each point in Figure 3(d) is one $\widehat{S}_{m,g}(o)$ value. Colors distinguish translation, rotation, and gripper outputs, giving nine points per color and 27 points per mechanism. The horizontal axis is logarithmic; vertical offsets separate overlapping points and have no quantitative meaning. Points show variation across observation states, not independent training replications or confidence intervals.

Interpretation and numerical scope. For fixed other inputs and sampling noise, a small feature change satisfies $\Delta a_g \approx J_g \Delta F$ to first order. Larger scores therefore indicate stronger local action sensitivity in the measured feature coordinates. They do not measure task success, the usefulness of the affected information, or a causal contribution to control performance. The original finite-difference checks did not calibrate the gradient magnitudes under native numerical precision, so the plotted values should be interpreted as local gradient diagnostics rather than validated finite-perturbation effect sizes.

E Feature Composition Protocol and Results for RQ3

E.1 Feature Construction and Training Objective

For source set $\mathcal{K}$, channel-wise concat uses aligned feature states in a fixed source order:

$$z_\tau^+ = [z_\tau^{\text{vae}} \|_{\text{ch}} z_\tau^{\text{vis}}], \qquad z_0^{\text{vis}} = \text{Concat}_{\text{ch}}\{z_0^k : k \in \mathcal{K}\}. \tag{8}$$

Current features follow the same composition order; future features start from noise at inference. Each source contributes a future-feature loss:

$$\mathcal{L}_{\text{aux}} = \sum_{k \in \mathcal{K}} \mathcal{L}_{\text{aux}}^k, \qquad \mathcal{L}_{\text{aux}}^k = \mathcal{L}_{\text{FM}}(z_0^k). \tag{9}$$

The feature losses are evaluated on valid future targets; observed feature slices remain clean and are excluded from the loss. Source-specific processing and concatenation order are held fixed between training and evaluation.

E.2 Progressive Composition and Conditional Gains

DINOv3 is the initial external source. The seven candidate additions are V-JEPA 2.1, SigLIP 2, SAM 3, VGGT-Ω, HRP, UniTok, and C-RADIOv4. Figure 4 first evaluates each candidate added to DINOv3. The next group uses DINOv3 plus UniTok as its parent, and the four-source group uses DINOv3 plus UniTok plus V-JEPA 2.1. Each bar adds one source not already in its parent set. These comparisons evaluate a progressive path rather than all possible subsets.

For source F, existing source set S, and evaluation condition c, we define the marginal contribution and its change between two source sets as

$$\Delta_F(S, c) = p(S + F, c) - p(S, c), \qquad I_F(c) = \Delta_F(S_1, c) - \Delta_F(S_0, c). \tag{10}$$

Here $p(S, c)$ is the success rate, in percent, of source set S under evaluation condition c, so Δ_F and I_F are measured in percentage points. A source is compared with its own parent configuration. These conditional gains describe performance interactions; they do not by themselves identify redundancy or optimization interference.

E.3 Per-Seed Results and Evaluation Settings

Table 10 reports the evaluations underlying the composition means and standard deviations. Each combination uses one fixed trained checkpoint, evaluated with seeds 41, 42, and 43. Every seed covers all 10,030 LP episodes. The mean is computed from the three full-benchmark success rates, and the standard deviation uses the sample convention (denominator $3 - 1$). Statistics are computed before rounding. The VAE-only and single-DINOv3 references in Figure 4 retain the RQ1 means and standard deviations in Table 1.

These composition evaluations use ten inference steps, replanning every ten steps, 30 initial waiting steps, no action ensembling, text guidance scale 1.0, CPU noise generation, and binarized gripper outputs. The LP evaluation protocol follows Appendix C.2.

Table 10: RQ3 composition success rates on LIBERO-Plus across three evaluation seeds. All values are percentages; the standard deviation reflects evaluation-seed variability for a fixed trained checkpoint. Two decimal places are retained here to show small standard deviations.

Source combination	Seed 41	Seed 42	Seed 43	Mean $\pm$ SD
$D + V$	85.13	84.62	85.03	84.93 ± 0.28
$D + L$	86.01	85.50	85.97	85.83 ± 0.28
$D + S$	84.93	83.80	84.75	84.49 ± 0.61
$D + G$	84.38	84.29	84.55	84.40 ± 0.13
$D + H$	85.31	83.91	84.23	84.48 ± 0.74
$D + U$	86.05	85.85	86.47	86.12 ± 0.32
$D + R$	79.74	77.82	78.23	78.59 ± 1.01
$D + U + V$	87.32	87.34	87.26	87.30 ± 0.04
$D + U + L$	85.51	85.82	85.71	85.68 ± 0.16
$D + U + S$	87.25	86.49	86.47	86.74 ± 0.44
$D + U + G$	86.99	86.53	86.43	86.65 ± 0.30
$D + U + H$	86.83	85.97	85.46	86.09 ± 0.69
$D + U + V + L$	87.60	87.40	87.51	87.50 ± 0.10
$D + U + V + S$	85.84	86.38	86.30	86.17 ± 0.29
$D + U + V + G$	85.62	84.83	84.36	84.94 ± 0.64
$D + U + V + H$	87.56	87.29	88.33	87.73 ± 0.54

D = DINOv3, V = V-JEPA 2.1, L = SigLIP 2, S = SAM 3, G = VGGT-Ω, H = HRP, U = UniTok, and R = C-RADIOv4. The three groups correspond to the two-, three-, and four-source groups in Figure 4.

F Task-Level RoboCasa Results

Table 11: Task-level success rates (%) of R^2-WAM on RoboCasa GR1 Tabletop. Each task is evaluated with 50 trials. Task names are abbreviated for readability; the 18 transfer tasks without a closing step use novel objects from Split A.

Task	Success (%)	Task	Success (%)
Cup to drawer (close)	54.0	Placemat to bowl	86.0
Potato to microwave (close)	86.0	Placemat to plate	88.0
Milk to microwave (close)	78.0	Placemat to tiered shelf	68.0

Continued on next page

Table 11: Task-level success rates on RoboCasa GR1 Tabletop (continued).

Task	Success (%)	Task	Success (%)
Bottle to cabinet (close)	88.0	Plate to bowl	90.0
Wine to cabinet (close)	76.0	Plate to cardboard box	76.0
Can to drawer (close)	96.0	Plate to pan	88.0
Cutting board to basket	94.0	Plate to plate	92.0
Cutting board to cardboard box	78.0	Tray to cardboard box	94.0
Cutting board to pan	98.0	Tray to plate	98.0
Cutting board to pot	88.0	Tray to pot	92.0
Cutting board to tiered basket	80.0	Tray to tiered basket	82.0
Placemat to basket	88.0	Tray to tiered shelf	74.0
Overall (1,016/1,200)			**84.7**

G TASK-LEVEL ROBOTWIN RESULTS

Table 12: Task-level success rates (%) of R^2-WAM on RoboTwin 2.0. Each of the 50 tasks is evaluated with 100 trials in each of the clean and randomized (Rand.) settings. Task identifiers are formatted with spaces for readability.

Task	Clean	Rand.	Task	Clean	Rand.
Adjust bottle	99.0	99.0	Place container plate	98.0	98.0
Beat block hammer	97.0	94.0	Place dual shoes	95.0	96.0
Blocks ranking RGB	100.0	97.0	Place empty cup	99.0	99.0
Blocks ranking size	66.0	87.0	Place fan	91.0	97.0
Click alarmclock	98.0	97.0	Place burger fries	100.0	99.0
Click bell	100.0	100.0	Place mouse pad	93.0	94.0
Dump bin bigbin	86.0	82.0	Place object basket	89.0	87.0
Grab roller	100.0	100.0	Place object scale	98.0	100.0
Handover block	99.0	96.0	Place object stand	99.0	99.0
Handover mic	100.0	100.0	Place phone stand	90.0	96.0
Lift pot	100.0	100.0	Move pillbottle pad	100.0	100.0
Move can pot	99.0	100.0	Place shoe	92.0	99.0
Move playingcard away	100.0	100.0	Press stapler	91.0	97.0
Move stapler pad	88.0	87.0	Put bottles dustbin	89.0	96.0
Hanging mug	79.0	75.0	Put object cabinet	89.0	81.0
Open laptop	95.0	98.0	Rotate QR code	88.0	81.0
Open microwave	56.0	45.0	Scan object	97.0	93.0
Pick diverse bottles	92.0	92.0	Shake bottle	100.0	99.0
Pick dual bottles	100.0	100.0	Shake bottle horizontally	100.0	99.0
Place A2B left	98.0	96.0	Stack blocks three	96.0	92.0
Place A2B right	97.0	98.0	Stack blocks two	100.0	100.0
Place bread basket	97.0	92.0	Stack bowls three	81.0	87.0
Place bread skillet	94.0	92.0	Stack bowls two	94.0	96.0
Place can basket	67.0	73.0	Stamp seal	94.0	98.0
Place cans plasticbox	100.0	100.0	Turn switch	57.0	66.0
Overall	**92.5**	**93.0**		Combined: 92.8	