ADAPTING VIDEO MODELS FOR ROBOT CONTROL VIA MOTION-GUIDED ASYMMETRIC PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) leverage pretrained video generative models for robotic control by jointly predicting future visual observations and actions. The visual prediction objective plays a key role in adapting pretrained video representations to the target robot domain. Yet standard visual prediction inherited from video pretraining treats all visual regions uniformly, despite their differing relevance to robotic control. We therefore investigate the spatial allocation of visual supervision in WAMs and show that focusing supervision on dynamic regions can substantially improve policy performance. Building on this finding, we introduce *Motion-Guided Asymmetric Prediction (MoGAP)*, which augments WAM training with motion-focused visual prediction. MoGAP allocates noise asymmetrically across regions, training the model to reconstruct more heavily corrupted dynamic regions using less corrupted static regions as context. MoGAP is trained alongside standard video–action prediction through a training-only auxiliary prediction path within the shared video diffusion backbone. It uses motion cues derived directly from training videos, requiring neither an external teacher nor additional modalities. Across robot manipulation benchmarks and WAM architectures, MoGAP consistently improves policy performance, increasing success rates on the GR-1 tabletop benchmark from 38.8% to 47.5% for Fast-WAM (Wan2.2-5B) and from 52.3% to 62.3% for Cosmos3-Nano. We further demonstrate MoGAP's effectiveness in physical deployment through evaluation on a real humanoid robot.

1 INTRODUCTION

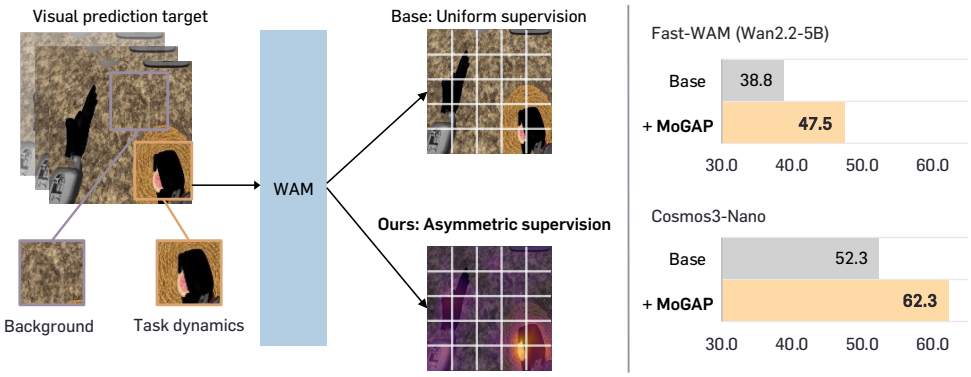
World Action Models (WAMs) have emerged as a promising approach to robotic foundation models, demonstrating that dense dynamics knowledge acquired through video pretraining can be adapted for robot control. Built on pretrained video diffusion backbones, WAMs jointly predict future visual observations and robot actions, extending video models' capacity for dense world modeling to action learning (Pai et al., 2026; Liang et al., 2025; Kim et al., 2026c; Ye et al., 2026b).

The visual prediction objective plays a central role in this adaptation by supervising world representation learning in the target domain. Removing it has been shown to substantially degrade control performance (Yuan et al., 2026), while jointly optimizing visual and action prediction further shapes these representations for control (Ma et al., 2026). Yet, the visual objective typically retains the formulation inherited from video pretraining, which was designed for visual generation (Lipman et al., 2023; Liu, 2022) rather than robot control.

Recent approaches seek to bridge this gap by introducing additional semantic (Qiu et al., 2026; Wang et al., 2026) or geometric supervision (Guo et al., 2026; Zhang et al., 2026; Zhao et al., 2026). However, these approaches primarily rely on external pretrained models or incorporate additional modalities such as depth or 3D information. We instead ask how visual supervision within the pretrained video backbone's native latent space can be tailored to better support action learning.

Our motivating study shows that shifting visual supervision toward dynamic regions can consistently improve policy performance across WAMs (Section 2.2). We interpret these gains as evidence that emphasizing dynamic regions can better align visual supervision with the demands of robot control. Dynamic regions tend to capture robot motion and object interactions, whereas static regions may contain visually complex but less action-relevant details, such as background textures. These findings motivate assigning distinct roles to dynamic and static regions in visual prediction for action learning.

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.



(a) Focusing visual supervision on task-relevant dynamics (b) Action policy performance

Figure 1: **Overview.** (a) We spatially reallocate visual supervision in World Action Models (WAMs) toward dynamic regions. This reallocation emphasizes robot motion and object interactions while downweighting static, task-irrelevant visual details, such as complex tabletop textures. (b) Our method, Motion-Guided Asymmetric Prediction (MoGAP), substantially improves policy performance across different WAM setups on GR-1 Tabletop, a challenging humanoid manipulation benchmark.

Building on this insight, we propose *Motion-Guided Asymmetric Prediction (MoGAP)*, a new visual prediction task for WAMs that treats dynamic regions as prediction targets and uses static regions as scene context. MoGAP establishes this distinction through asymmetric noise allocation (Chen et al., 2024; Chefer et al., 2026), applying higher noise levels to dynamic regions while preserving more information in static regions. The noise allocation is guided by latent motion scores directly computed from temporal differences in training videos, requiring neither external teacher models nor additional supervision modalities. MoGAP is jointly optimized with the standard WAM objective through a prediction path within the shared video diffusion backbone. This path is used only during training, leaving inference unchanged.

Through extensive experiments, we demonstrate that MoGAP consistently improves policy performance across WAM architectures in both simulated and real-world manipulation benchmarks. On GR-1 Tabletop, MoGAP increases the success rate of Fast-WAM (Wan2.2-5B) from 38.8% to 47.5% and that of Cosmos3-Nano from 52.3% to 62.3%, outperforming competitive VLA and WAM baselines as well as representation alignment methods using external teachers. On a real-world OpenArm humanoid, we find that MoGAP helps the policy better respond to task-relevant perturbations, such as target object relocation during execution, even without demonstrations of such perturbations.

In summary, our contributions are as follows:

- Through an intuitive motivating study, we show that the spatial allocation of visual supervision has a substantial impact on downstream control performance, establishing it as an important design consideration for WAMs.
- We propose *Motion-Guided Asymmetric Prediction (MoGAP)*, a visual prediction task that trains WAMs to predict more heavily corrupted dynamic regions from cleaner static context, without requiring external supervision models or additional modalities.
- Through extensive experiments, we demonstrate that MoGAP delivers consistent and substantial gains over strong WAM baselines across diverse tasks, embodiments, and environments.

2 METHOD

We introduce *Motion-Guided Asymmetric Prediction (MoGAP)*, a visual prediction task that promotes control-relevant representation learning in World Action Models (WAMs). We first describe the standard WAM framework in Section 2.1. In Section 2.2, we present a motivating study investigating how the visual prediction objective affects policy learning. Finally in Section 2.3, we present our *MoGAP*, with an illustrated overview in Figure 2.

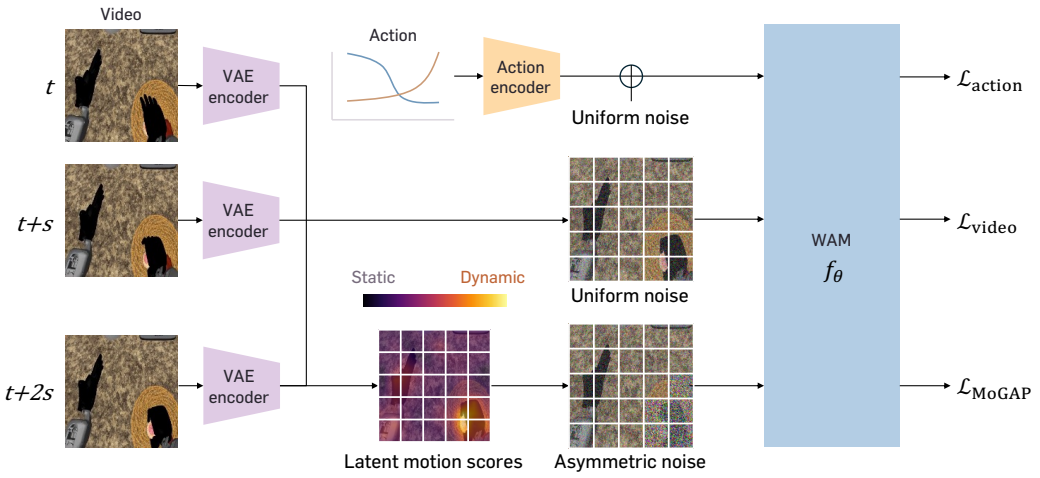


Figure 2: **Motion-Guided Asymmetric Prediction (MoGAP).** MoGAP is a visual prediction task that promotes control-relevant representation learning in World Action Models. MoGAP introduces regional asymmetry in noise levels, guided by temporal differences between video latents: dynamic regions receive higher noise levels, while less corrupted static regions serve as context for prediction.

2.1 Preliminaries: World Action Models

At timestep t, World Action Models (WAMs) predict future visual observations $\mathbf{o}_{t+1:t+H+1}$ and an action chunk $\mathbf{a}_{t:t+H}$ conditioned on the current observation $\mathbf{o}_t$, task instruction $\mathbf{c}$, and proprioceptive state $\mathbf{q}_t$. Here, slice notation denotes half-open intervals, so both prediction sequences have horizon H. The model f_θ learns the conditional joint distribution

$$p_\theta(\mathbf{o}_{t+1:t+H+1}, \mathbf{a}_{t:t+H} \mid \mathbf{o}_t, \mathbf{c}, \mathbf{q}_t)\,. \tag{1}$$

Building on pretrained video diffusion backbones, WAMs perform visual prediction in the latent space of a pretrained video autoencoder. We denote the encoded future observations by $\mathbf{z}_t = \mathcal{E}(\mathbf{o}_{t+1:t+H+1})$, where $\mathcal{E}$ is the video encoder; the corresponding decoder maps predicted latents back to the observation space.

Recent approaches (Kim et al., 2026c; Ye et al., 2026b; Yuan et al., 2026; Agarwal et al., 2026) train visual and action prediction jointly through flow matching (Lipman et al., 2023; Liu, 2022). At flow-matching timesteps $\tau_v, \tau_a \in [0, 1]$, noisy visual latents and actions are constructed as

$$\mathbf{z}_t^{\tau_v} = (1-\tau_v)\mathbf{z}_t + \tau_v \boldsymbol{\epsilon}_v, \quad \mathbf{a}_t^{\tau_a} = (1-\tau_a)\mathbf{a}_{t:t+H} + \tau_a \boldsymbol{\epsilon}_a, \tag{2}$$

where $\boldsymbol{\epsilon}_v, \boldsymbol{\epsilon}_a \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ are sampled independently. The model predicts visual and action velocity fields, denoted by $(\hat{\mathbf{u}}_v, \hat{\mathbf{u}}_a) = f_\theta(\mathbf{z}_t^{\tau_v}, \mathbf{a}_t^{\tau_a}; \mathbf{o}_t, \mathbf{c}, \mathbf{q}_t)$, and is trained with the flow-matching objectives:

$$\mathcal{L}_{\text{video}} = \mathbb{E}_{\mathbf{z},\mathbf{a}}\left[\|\hat{\mathbf{u}}_v - (\boldsymbol{\epsilon}_v - \mathbf{z}_t)\|_2^2\right], \tag{3}$$

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{\mathbf{z},\mathbf{a}}\left[\|\hat{\mathbf{u}}_a - (\boldsymbol{\epsilon}_a - \mathbf{a}_{t:t+H})\|_2^2\right], \tag{4}$$

The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\text{video}} + \lambda_{\text{action}}\mathcal{L}_{\text{action}}, \tag{5}$$

where $\lambda_{\text{action}} > 0$ is a weighting coefficient that balances the two objectives.

2.2 Motivating Study

To investigate how visual supervision affects action learning, we conduct a small experiment fixing the supervision budget, and redistributing it spatially across tokens. For this, we compute a motion proxy signal directly from temporal differences in the clean video latents $\mathbf{z}_t$. For temporal slice n and

spatial latent location p, we compute a motion score $m_{n,p} = \|\mathbf{z}_{t,n,p} - \mathbf{z}_{t,n-1,p}\|_2$. We normalize the scores by their spatial mean and define spatial reallocation weights as follows:

$$w_{n,p} = \frac{\varepsilon + \widetilde{m}_{n,p}^{\beta}}{\sum_{q=1}^{P} \left(\varepsilon + \widetilde{m}_{n,q}^{\beta}\right)/P}, \qquad \widetilde{m}_{n,p} = \frac{m_{n,p}}{\sum_{q=1}^{P} m_{n,q}/P}. \tag{6}$$

where $\widetilde{m}_{n,p}$ is the normalized score, and P is the number of spatial locations. Larger motion scores receive higher weights, directing more visual supervision toward regions with greater temporal change. Here, $\beta > 0$ controls the emphasis on the high-change regions, while $\varepsilon > 0$ maintains supervision in low-change regions.

We use these weights to define the dynamic-region prioritized visual objective:

$$\mathcal{L}'_{\text{video}} = \mathbb{E}_{\mathbf{z},\mathbf{a}}\left[\frac{1}{NP}\sum_{n=1}^{N}\sum_{p=1}^{P} w_{n,p}\,\mathcal{L}_{\text{video}}^{n,p}\right], \tag{7}$$

where $\mathcal{L}_{\text{video}}^{n,p}$ denotes the flow-matching error at temporal slice n and spatial location p, and N is the number of future latent slices.

For this study, we replace $\mathcal{L}_{\text{video}}$ in Equation (5) with $\mathcal{L}'_{\text{video}}$ to redistribute visual supervision toward dynamic regions, leaving the action prediction objective unchanged. As shown in Figure 3, this reallocation improves the action policy performance on GR-1 tabletop, a simulated humanoid manipulation benchmark. We interpret this finding as evidence that visual regions need not contribute equally to action learning: dynamic regions tend to capture robot motion and object interactions, whereas static regions may contain visually complex but less control-relevant details, such as tabletop textures. This suggests that visual prediction can better support control by assigning different learning roles to dynamic and static regions.

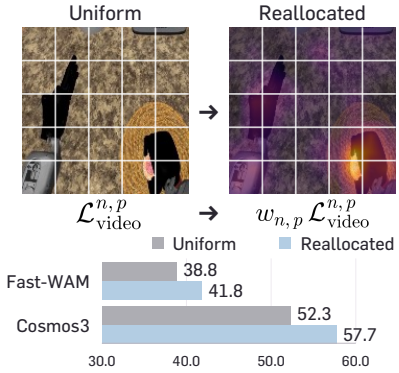


Figure 3: **Motivating study.** Redistributing supervision toward dynamic regions.

2.3 Motion-Guided Asymmetric Prediction

Building on this insight, we introduce *Motion-Guided Asymmetric Prediction (MoGAP)*, a new visual prediction task for WAMs that uses less corrupted static regions as context for predicting more heavily corrupted dynamic regions. While our motivating experiment prioritizes dynamic regions by reweighting prediction errors, MoGAP extends this principle to the prediction task itself by controlling how much visual information is available in dynamic and static regions.

Designing the asymmetric prediction task. To construct this asymmetric prediction task, we exploit the interpretation of diffusion noising as partial masking (Chen et al., 2024), where higher noise levels reveal less information about the underlying clean context. Given the base visual timestep τ_v, we independently draw $\xi \sim \mathcal{T}_v$, where $\mathcal{T}_v$ is the video timestep sampling distribution, and rescale it to the interval $[\tau_v, 1]$:

$$\tau_v^{+} = \tau_v + (1 - \tau_v)\xi. \tag{8}$$

This preserves the base noise level while increasing corruption in dynamic regions, with the noise gap scaled by the remaining interval $1 - \tau_v$. We construct a binary motion mask $M_{n,p}$ by thresholding latent motion scores. The mask is mapped to the model's spatial token grid to construct a dual-timestep assignment $\boldsymbol{\tau}_M$:

$$M_{n,p} = \mathbb{I}[m_{n,p} > \bar{m}_n], \qquad \tau_{n,p} = \begin{cases} \tau_v^{+}, & M_{n,p} = 1, \\ \tau_v, & M_{n,p} = 0, \end{cases} \tag{9}$$

where $\bar{m}_n$ is the spatial mean. Using Gaussian noise $\boldsymbol{\epsilon}'_v \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ sampled independently of $\boldsymbol{\epsilon}_v$, we construct the asymmetrically noised visual latents as

$$\mathbf{z}_t^{\boldsymbol{\tau}_M} = \text{diag}(\mathbf{1} - \boldsymbol{\tau}_M)\mathbf{z}_t + \text{diag}(\boldsymbol{\tau}_M)\boldsymbol{\epsilon}'_v. \tag{10}$$

The corresponding flow-matching target is $\boldsymbol{\epsilon}'_v - \mathbf{z}_t$. Dynamic regions are thus predicted under greater corruption, with less corrupted static regions providing scene context.

For these noisy inputs, we apply the video flow-matching loss in Equation (3) only to moving tokens with $M_{n,p} = 1$. Within each latent slice, we normalize by the number of selected locations so that the moving-region fraction does not change the total loss weight. Let $\mathcal{S}_n = \{p \mid M_{n,p} = 1\}$, with all spatial locations selected when the mask is empty. The resulting objective is:

$$\mathcal{L}_{\text{MoGAP}} = \mathbb{E}_{\mathbf{z},\mathbf{a}} \left[\frac{1}{N} \sum_{n=1}^{N} \frac{1}{|\mathcal{S}_n|} \sum_{p \in \mathcal{S}_n} \left\| \hat{\mathbf{u}}^{\boldsymbol{\tau}^M}_{v,n,p} - (\boldsymbol{\epsilon}'_{v,n,p} - \mathbf{z}_{t,n,p}) \right\|_2^2 \right], \tag{11}$$

where $\hat{\mathbf{u}}^{\boldsymbol{\tau}^M}_{v,n,p}$ denotes the visual velocity prediction under the asymmetric timestep assignment.

Incorporating MoGAP into WAMs. As MoGAP provides complementary motion-guided supervision to standard video-action prediction, we incorporate it as an auxiliary training path that augments the original WAM objective. We process the dual-timestep-noised visual latents $\mathbf{z}_t^{\boldsymbol{\tau}^M}$ alongside the original WAM inputs in the shared video diffusion backbone, without introducing additional parameters. To keep the original WAM prediction path independent of the auxiliary tokens, we apply an asymmetric attention mask between the two paths. As shown in Figure 4, the original WAM tokens, including noisy future visual and action tokens, cannot attend to the MoGAP tokens. MoGAP tokens can attend to one another, the current observation, and noisy action representations, but not the original path's noisy future visual tokens. The task instruction $\mathbf{c}$ conditions both paths through the existing conditioning mechanism.

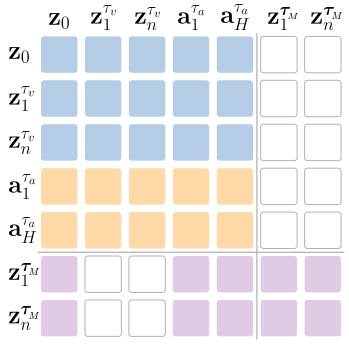


Figure 4: **Attention mask of MoGAP.** The MoGAP tokens $\mathbf{z}_t^{\boldsymbol{\tau}^M}$ are omitted at inference time.

We jointly optimize the standard WAM objectives with the MoGAP objective:

$$\mathcal{L} = \mathcal{L}_{\text{video}} + \lambda_{\text{action}} \mathcal{L}_{\text{action}} + \lambda_{\text{MoGAP}} \mathcal{L}_{\text{MoGAP}}, \tag{12}$$

where $\lambda_{\text{MoGAP}} > 0$ controls the contribution of motion-guided supervision. At inference time, the MoGAP tokens are omitted, leaving the original WAM inference procedure unchanged.

3 Experiments

In this section, we evaluate MoGAP across diverse manipulation settings, robot embodiments, and WAM architectures. We first describe the benchmarks and implementation details in Section 3.1. In Section 3.2, we study its impact on frontier WAM baselines, across multitask simulated manipulation benchmarks. We then assess MoGAP's performance in real-world physical deployment, on an OpenArm Humanoid equipped with Inspire hands, in Section 3.3. Finally, in Section 3.4, we present informative analyses to better understand the effects of MoGAP.

3.1 Setup

Benchmarks. We evaluate MoGAP on two multitask simulation benchmarks and a real-world manipulation setup, covering different embodiments, observation configurations, and environmental conditions.

- **RoboCasa GR-1 Tabletop** (Bjorck et al., 2025) consists of 24 tabletop hand manipulation tasks using the GR-1 humanoid. We use this benchmark for our main study, including comparisons with existing methods and ablation studies.
- **RoboCasa-Kitchen** (Nasiriany et al., 2024) comprises 24 manipulation tasks using a single-arm robot with a gripper in kitchen environments. We use this benchmark to evaluate MoGAP in multi-view observation settings.
- **Real-world evaluation** uses an OpenArm humanoid with two Inspire RH56F1 hands. We evaluate both in-domain tasks and diverse out-of-domain conditions to assess the abilities of MoGAP in physical deployment.

Table 1: **Comparison with state-of-the-art VLAs & WAMs.** Results include the success rate (%) of representative VLAs and WAMs on GR-1 tabletop. Results of Fast-WAM and Cosmos3 are reproduced while others are from the original work. Best results are highlighted in **bold**.

Method	PnP	Articulated	24 Tasks Average
VLA-based methods			
GR00T-N1.6 (Bjorck et al., 2025)	55.4	24.2	47.6
StarVLA-α (Ye et al., 2026a)	58.4	54.0	57.3
RLDX-1 (Kim et al., 2026a)	59.0	57.7	58.7
WAM-based methods			
Fast-WAM (Yuan et al., 2026)	33.8	53.7	38.8
DiT4DiT (Ma et al., 2026)	50.9	50.3	50.8
Cosmos3 (Agarwal et al., 2026)	48.3	64.3	52.3
MoGAP (Ours)	**60.8**	**66.7**	**62.3**

Table 2: **Comparison to teacher-based supervision.** Results report the average success rate (%) on GR-1 tabletop, with each method applied to the same baseline WAM, Cosmos3-Nano (Agarwal et al., 2026). Best results are highlighted in **bold**.

Method	Teacher	24 Tasks Average
Cosmos3	–	52.3
w/ Geometry Forcing (Wu et al., 2026)	VGGT	53.9
w/ AGRA (Qiu et al., 2026)	DINOv2	55.4
w/ TSA (Kim et al., 2026b)	VideoMAEv2	56.8
w/ VideoREPA (Zhang et al., 2025)	VideoMAEv2	57.8
w/ VideoREPA (Zhang et al., 2025)	V-JEPA2.1-L	59.2
w/ MoGAP (Ours)	–	**62.3**

Implementation details. We adopt Cosmos3-Nano (Agarwal et al., 2026) as our primary WAM, generating 16 actions and 8 future video frames with a temporal stride of two in a single forward process. We train for 16K gradient steps for the multi-task simulation benchmarks and 8K steps for real-world experiments, respectively, with a global batch size of 256. We follow the original work for the training details, and set $\lambda_{\text{MoGAP}} = 1.0$. For the simulated benchmarks, we evaluate each method over 50 trials per task, for a total of 1,200 rollouts. Further details are described in Appendix B.5.

3.2 Simulated benchmark results

Comparison with state-of-the-art methods. Table 1 summarizes performance on RoboCasa GR-1 Tabletop. Cosmos3 already provides a strong baseline, outperforming the other WAM-based methods and remaining competitive with leading VLA-based methods. Despite this strong starting point, MoGAP improves the average success rate from 52.3% to 62.3%, a gain of 10.0 percentage points, surpassing all VLA and WAM-based methods in the comparison. The gains are particularly pronounced on pick-and-place tasks, which require precise manipulation of a wider variety of objects within each task, with success rates increasing from 48.3% to 60.8% (+12.5 percentage points). We attribute this improvement in part to MoGAP's emphasis on fine-grained motion that produces only subtle visual changes but is important for manipulation, especially for precise positioning during grasping and placing. We further visualize this effect in our qualitative analysis (see Figure 8).

Comparison to teacher-based supervision. We compare MoGAP with teacher-based representation learning methods that leverage visual foundation models (Oquab et al., 2024; Wang et al., 2023; Mur-Labadia et al., 2026) or 3D foundation models (Wang et al., 2025) as teachers, using the same Cosmos3-Nano baseline model. As shown in Table 2, all the other methods improve over the baseline, yet MoGAP outperforms them without an external teacher. While external representations provide useful supervision, MoGAP shows that motion-guided regional noise allocation offers an effective alternative for adapting the pretrained video backbone's native representations to robot control.

Table 3: **Ablation studies on MoGAP.** We study (a) the effectiveness of its key components through ablations and (b) the generalization of MoGAP across WAM architectures. Results report average success rates (%) on GR-1 Tabletop. Best results within each comparison are highlighted in **bold**.

Method	24 Tasks Average
MoGAP	**62.3**
w/o auxiliary path	54.4
w/o noise asymmetry	55.3
reversed direction	55.2

(a) Component analysis.

Method	Backbone	24 Tasks Average
Fast-WAM	Wan2.2-5B	38.8
+ MoGAP	Wan2.2-5B	**47.5**
Cosmos3	Cosmos3-Nano	52.3
+ MoGAP	Cosmos3-Nano	**62.3**

(b) Generalization across WAMs.

Component analysis. Table 3a studies individual components of MoGAP. As shown in the first two rows, replacing the main visual supervision with our method improves the success rate from 52.3% to 54.4%, while introducing it as an auxiliary branch yields a substantially larger gain. This suggests that the standard full-noise range supervision and our asymmetric supervision provide complementary learning signals. The third and fourth rows further show that removing the noise allocation asymmetry or reversing the asymmetry direction significantly reduces the success rate to 55.3% and 55.2%, respectively. These results demonstrate that asymmetric noise allocation toward predicting dynamic regions from static context is a crucial design choice for downstream performance.

Generalization across WAM architectures. We further assess MoGAP on Fast-WAM (Yuan et al., 2026), which combines a Wan2.2-5B (Wan et al., 2025) video diffusion backbone with a separately parameterized action expert and omits future visual prediction at inference. MoGAP improves the success rate from 38.8% to 47.5% (Table 3b). Together with the improvements on Cosmos3, these results demonstrate MoGAP's effectiveness across different WAM architectures and video backbones.

Adaptation to multi-view configuration. We also evaluate MoGAP on RoboCasa-Kitchen, a single-arm manipulation benchmark with three camera views. We apply motion-guided noise assignment to the two external camera views while assigning all wrist-camera tokens to the high-noise group, as camera motion can dominate the latent motion scores in the wrist view. With this adaptation, MoGAP consistently outperforms the Cosmos3 baseline while retaining the benefits of longer execution horizons (Figure 5). These results demonstrate that MoGAP extends to multi-view settings that include a moving wrist camera.

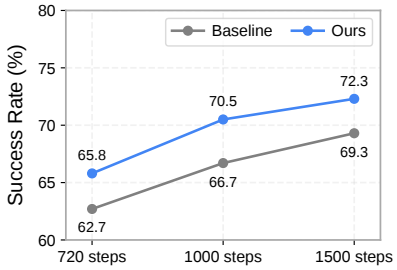


Figure 5: **RoboCasa-Kitchen success rate (%)** across execution timesteps.

3.3 Real-world experiment results

Real-world experimental setup. We conduct experiments on an OpenArm humanoid with two 7-DoF arms and two five-finger, 6-DoF hands and an egocentric stereo camera. We evaluate three task groups targeting different aspects of policy performance: *In-Domain* tasks for general manipulation, *OOD Objects & Environments* tasks for robustness to unseen objects and environments, and *OOD Object Relocation* tasks for evaluating the policy's ability to focus on task-relevant regions (Figure 7). For low-latency deployment, we use a single denoising step for both visual and action prediction without classifier-free guidance (Ho & Salimans, 2022). We precompute and cache text embeddings, removing the text encoder from the inference path, and omit the VAE decoder. Together, these choices enable Cosmos3-Nano, a 16B-parameter model, to run with an inference latency of 48ms on a single RTX 5090 GPU with 32GB of VRAM. For each method, we train a single policy on the same multitask dataset collected in our real-world setup. Further details are provided in Appendix C.

Real-world evaluation results. Figure 6 presents the results on the real-robot manipulation tasks. MoGAP consistently improves over the baseline policy across all three groups, demonstrating gains in both general manipulation and robustness to distribution shifts. Notably, we find that MoGAP yields particularly large gains on the object relocation tasks. In these trials, the baseline policy often continues its original trajectory and attempts to grasp at the object's previous location, whereas the

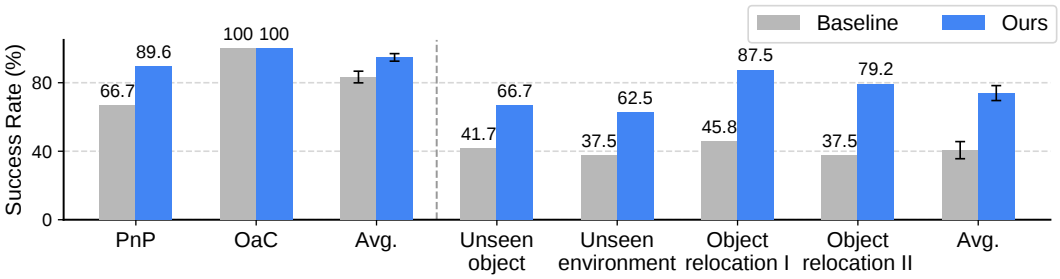


Figure 6: **Real-world experiment success rate (%).** Results on the OpenArm humanoid real-world benchmark. We report success rates with standard errors for each method across task suites.

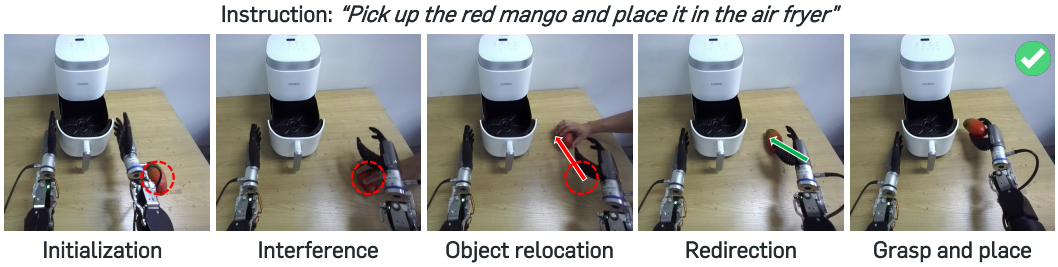


Figure 7: **Object relocation task example.** The target object's location is moved during execution. In set II, the object is relocated farther than in set I. MoGAP successfully adapts its trajectory.

MoGAP-trained policy more successfully adjusts its trajectory toward the relocated object. This adaptation emerges despite the absence of object relocation demonstrations during training. We attribute this behavior in part to MoGAP's emphasis on dynamic regions, which encourages the policy to respond more effectively to task-relevant visual changes in the environment.

3.4 Further Analysis

Motion score analysis. We analyze the motion score derived from temporal differences between video latents. In Figure 8, the scores highlight subtle arm and hand movements (rows 1 and 2) and remain focused on the robot and objects even under camera shifts caused by waist motion (row 3). We quantify this localization using region-wise AUROC, which evaluates how well the motion scores rank locations within each ground-truth region relative to the rest of the scene (Table 4a). The results show that the signal clearly assigns higher scores to the moving arm (including the hand) and the moving target object, while assigning relatively lower scores to background regions (i.e., the tabletop, surrounding environment, and distractor objects).

Trained policy analysis. Training with the latent motion score supervision encourages WAMs to attend more closely to dynamic regions during action generation and improves their prediction of these regions. In Figure 8, we visualize the action tokens' attention, and observe that it is concentrated more strongly on the moving hand and arm. In the generated examples, the baseline generation often appears blurred around the moving arm, while ours preserves its structure relatively more clearly (rows 2 and 3). Quantitative results in Table 4b support this attention shift, showing greater attention to the moving arm and hand and the manipulated object, and reduced attention to the background. Further details about the analysis metrics are provided in Appendix B.4.

Limitations. MoGAP introduces additional tokens during training, increasing wall-clock training time by a factor of 1.26 in exchange for improved policy performance. To assess this trade-off, we provide computational details and a compute-matched comparison in Appendix A.

4 Related Work

World Action Models (WAMs). WAMs adapt rich spatiotemporal priors from large-scale video data to robot control by learning to predict future visual observations and control actions together (Pai

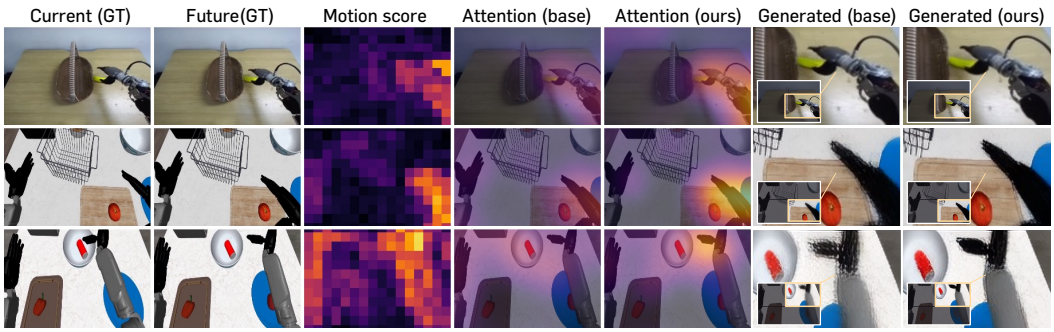


Figure 8: **Qualitative analysis.** The columns show, from left to right, current and future ground-truth observations, latent motion scores, action-to-vision attention maps for the baseline and MoGAP, and their generated future observations.

Table 4: **Quantitative analysis.** We conduct additional analyses of the latent motion signal, and the trained policy models on GR-1 Tabletop.

Evaluation Region	AUROC
Moving arm & hand	0.887
Manipulated object	0.834
Background	0.430

(a) Motion signal validity.

Evaluation Region	Attention Ratio Base $\rightarrow$ Ours
Moving arm & hand	$1.80 \rightarrow 2.07$
Manipulated object	$2.00 \rightarrow 2.29$
Background	$0.86 \rightarrow 0.82$

(b) Trained policy analysis.

et al., 2026; Liang et al., 2025; Kim et al., 2026c; Li et al., 2026; Ye et al., 2026b). However, visual reconstruction alone may not fully align with the needs of robot control (Zhou et al., 2025; Nilaksh et al., 2026). Recent studies incorporate supervision from semantic encoders (Qiu et al., 2026; Wang et al., 2026), or from depth and 3D geometry (Guo et al., 2026; Zhang et al., 2026; Zhao et al., 2026; Yang et al., 2026). Such approaches augment visual supervision through pretrained representation models (Oquab et al., 2024; Wang et al., 2025), or additional prediction targets derived from geometry and motion estimators (Lin et al., 2026; Morimitsu et al., 2025; Liu et al., 2026). Our work explores a complementary direction by introducing an asymmetric prediction task for action learning within the pretrained video diffusion model's native latent space, without external teachers or additional modalities.

Representation learning for video generative models. Recent work explores representation learning within video generative models to improve prediction quality. One line of work aligns diffusion and flow model representations with external pretrained encoders (Yu et al., 2025), encouraging physically plausible predictions (Zhang et al., 2025; Bhowmik et al., 2026) or incorporating 3D geometric structure (Wu et al., 2026). Another line improves representations through self-supervision without external teachers (Chefer et al., 2026; Pan et al., 2026). Our approach builds on these advances, drawing in particular on token-wise noise assignment (Chen et al., 2024; Chefer et al., 2026) to introduce motion-guided asymmetry tailored to action modeling for robot control.

5 Conclusion

In this work, we present *Motion-Guided Asymmetric Prediction (MoGAP)*, a visual prediction task that adapts pretrained video diffusion models for robot control through motion-guided regional noise allocation. Our experiments demonstrate consistent improvements over strong WAM baselines in simulated and real-world manipulation settings, including improved adaptation to target object relocation tasks during execution. These findings highlight the importance of tailoring visual supervision for robot policy learning, particularly by emphasizing regions that capture task-relevant dynamics. We hope this work encourages further exploration of control-oriented visual prediction to translate pretrained video models' world knowledge into more capable robotic policies.

AI USE STATEMENT

In this work, we used generative AI tools for polishing writing, making earlier versions of the figures, and generating a portion of the research code. We have not used generative AI tools for research ideas, methodology, or experiment designs. We have reviewed all AI-assisted work. For example, all LLM-generated code and text were directly reviewed and revised for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the implementation details and full hyperparameter settings (Section 3.1, Appendix B.5), and further training and evaluation protocols in Appendix B.2 and Appendix C.3 for reproducibility.

REFERENCES

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

Aritra Bhowmik, Denis Korzhenkov, Cees G Snoek, Amirhossein Habibian, and Mohsen Ghafoorian. MoAlign: Motion-centric representation alignment for video diffusion models. In *International Conference on Learning Representations*, 2026.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Hila Chefer, Patrick Esser, Dominik Lorenz, Dustin Podell, Vikash Raja, Vinh Tong, Antonio Torralba, and Robin Rombach. Self-supervised flow matching for scalable multi-modal synthesis. In *International Conference on Machine Learning*, 2026.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, 2024.

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4d world action modeling from video priors with asynchronous denoising. *arXiv preprint arXiv:2604.26694*, 2026.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Dongyoung Kim, Huiwon Jang, Myungkyu Koo, Suhyeok Jang, Taeyoung Kim, Beomjun Kim, Byungjun Yoon, Changsung Jang, Daewon Choi, Dongsu Han, et al. RLDX-1 technical report. *arXiv preprint arXiv:2605.03269*, 2026a.

Manjin Kim, Suha Kwak, and Minsu Cho. Tempered self-similarity alignment for physically plausible video generation. *arXiv preprint arXiv:2605.24962*, 2026b.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning, 2026c.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Robotics: Science and Systems*, 2026.

Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations*, 2026.

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

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

Xinhang Liu, Yuxi Xiao, Donny Y Chen, Jiashi Feng, Yu-Wing Tai, Chi-Keung Tang, and Bingyi Kang. Trace Anything: Representing any video in 4d via trajectory fields. In *International Conference on Learning Representations*, 2026.

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

Henrique Morimitsu, Xiaobin Zhu, Roberto M Cesar, Xiangyang Ji, and Xu-Cheng Yin. DPFlow: Adaptive optical flow estimation with a dual-pyramid framework. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, and Nicolas Ballas. 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 *Robotics: Science and Systems*, 2024.

Nilaksh, Saurav Jha, Artem Zholus, and Sarath Chandar. Reconstruction or semantics? what makes a latent space useful for robotic world models. *arXiv preprint arXiv:2605.06388*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Jonas Pai, Liam Achenbach, Oliver Sanchez, Stefanos Charalambous, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond vlas. In *Robotics: Science and Systems*, 2026.

Zirui Pan, Xin Wang, Yipeng Zhang, Yuwei Zhou, and Wenwu Zhu. Temporal-aware flow matching for video generation with temporally coherent motion. In *International Conference on Machine Learning*, 2026.

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.

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

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

Mingxin Wang, Bin Hu, Bin Qian, Kaitao Jiang, Haoning Wu, Feng Yan, Bowen Jing, Ruiyang Hao, Enyi Wang, Kangning Niu, et al. ST-WAM: Semantic-temporal world action model for robust manipulation under visual distribution shifts. *arXiv preprint arXiv:2607.28993*, 2026.

Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. In *International Conference on Learning Representations*, 2026.

Lishan Yang, Wenxuan Song, Xi Wang, Pingyue Sheng, Zheng Fang, Ziyang Zhou, Junjie He, Haodong Yan, Jiayi Chen, Nan Sun, et al. 4D-WAM: Infusing spatiotemporal awareness into world action models through trajectory fields. *arXiv preprint arXiv:2608.08023*, 2026.

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. In *European Conference on Computer Vision*, 2026a.

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. In *Conference on Robot Learning*, 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.

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*, 2026.

Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. Videorepa: Learning physics for video generation through relational alignment with foundation models. In *Advances in Neural Information Processing Systems*, 2025.

Haoyu Zhao, Xingyue Zhao, Siteng Huang, Xin Li, Deli Zhao, and Zhongyu Li. Rynnworld-4d: 4d embodied world models for robotic manipulation. *arXiv preprint arXiv:2607.06559*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning*, 2025.

A Computational Details

Experiments are conducted on 4 $\times$ NVIDIA H200 Tensor core GPUs. In this setup, training Cosmos3 for our GR-1 Tabletop experiments takes 14.7 hours for 16K optimization steps with a global batch size of 256. MoGAP introduces additional tokens during training, increasing training time by a factor of 1.26 to 18.5 hours. Caching the VAE latents reduces the training times to 10.9 and 15.0 hours, respectively. In the main experiments, Cosmos3 and MoGAP are both trained for 16K steps with the same batch size and therefore see the same number of samples. For a compute-matched comparison we additionally train Cosmos3 for 22K steps (6K more steps) and report its average success rate in Table 5.

Method	24 Tasks Average
Cosmos3 (22K)	55.8
MoGAP (16K)	62.3

Table 5: Compute-matched comparison

B Simulation Experiments Details

B.1 Dataset

For RoboCasa GR-1 tabletop (Bjorck et al., 2025) and RoboCasa-Kitchen (Nasiriany et al., 2024), we use the publicly available dataset[1]. We use 1000 demonstrations per task for GR-1 tabletop, and 300 demonstrations per task for RoboCasa-Kitchen. We provide a visualization of example tasks of our simulated benchmarks in Figure 9.

B.2 Training and evaluation details

For Cosmos3-Nano (Agarwal et al., 2026) and Wan2.2-5B (Wan et al., 2025), we employ the publicly available checkpoints[2][3]. For Cosmos3 experiments, we train 16K optimization steps with a global batch size of 256 using AdamW with weight decay of 0.05, 500 steps of warmup to a peak learning rate of 2×10^{-5}, followed by linear decay to zero. We train to generate 16 actions and 8 future video frames with a temporal stride of two, given a single observation frame. For Fast-WAM (Yuan et al., 2026) experiments, we train for 100K optimization steps with a global batch size of 64 using AdamW with weight decay of 0.01 and a peak learning rate of 10^{-4}, with linear warmup over the first 5% of training. The input and output frames and the action horizon are identical to those of Cosmos3. For RoboCasa-Kitchen, the three views (exterior left, exterior right, wrist) are stacked in a rectangular canvas, shown in Figure 9.

For the evaluation protocol, we evaluate each method over 50 trials with fixed seeds for reproducibility. RoboCasa GR-1 Tabletop and RoboCasa-Kitchen both consist of 24 task suites, making 1200 evaluation trials for all models. For Cosmos3, we use all 16 actions generated in a single forward process at execution without using classifier-free guidance, and we use 12 actions for Fast-WAM following the original work.

B.3 Comparison Methods Details

AGRA details. AGRA (Qiu et al., 2026) applies REPA-style (Yu et al., 2025) per-token cosine alignment to patch features from a frozen DINOv2 encoder (Oquab et al., 2024). With the future latents fixed at the pure-noise level, we extract vision-token hidden states after the 8th transformer block and project them through a three-layer MLP with a hidden width of 2048. For each temporal latent group, we use its first RGB frame as input to DINOv2 ViT-L/14 at 448×448 resolution,

[1] https://huggingface.co/datasets/nvidia/PhysicalAI-Robotics-GR00T-X-Embodiment-Sim
[2] https://huggingface.co/nvidia/Cosmos3-Nano
[3] https://huggingface.co/Wan-AI/Wan2.2-TI2V-5B

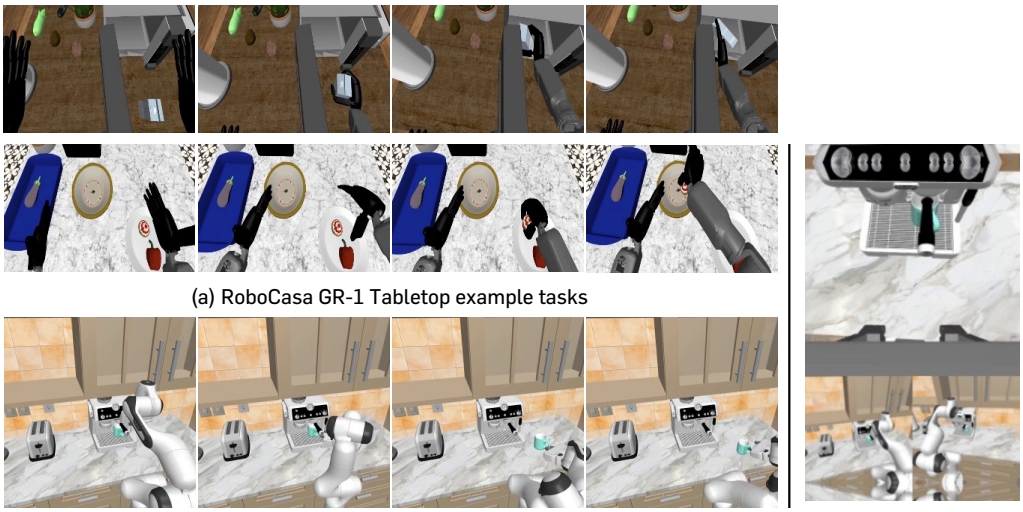


Figure 9: Task examples of (a) RoboCasa GR-1 Tabletop, (b) RoboCasa-Kitchen, and (c) the multi-view observation input configuration.

following the original work's implementations. The resulting 32×32 patch-feature grid is bilinearly resized to the 8×8 latent grid for token-wise alignment. We set the alignment loss weight to 0.5.

Geometry Forcing details. Geometry Forcing (Wu et al., 2026) aligns intermediate video-denoiser features with aggregator tokens from a frozen VGGT model (Wang et al., 2025). We adopt the angular alignment objective from the original work. Specifically, we extract the value projections of vision tokens from the 8th transformer block and project them through a three-layer MLP with SiLU activations and a hidden width of 2048. We minimize the mean per-token cosine distance $(1 - \cos)$ between the projected features and the output tokens of the final aggregator layer of VGGT-1B. Following the original work, we set the alignment loss weight to 0.5 and jointly train the projector and backbone.

VideoREPA details. VideoREPA (Zhang et al., 2025) aligns the pairwise cosine-similarity matrix of projected intermediate denoiser features with that of a frozen video foundation model using a margin-based hinge loss. We extract hidden states after the 21st of 36 transformer blocks and project them through a three-layer MLP with a hidden width of 2048. We retain the future video latent tokens and trilinearly resize the $2 \times 8 \times 8$ student feature grid to match the $4 \times 16 \times 16$ teacher tubelet grid. The teacher receives clean future frames at 256×256 resolution. We use VideoMAEv2 ViT-B (Wang et al., 2023) with a margin of 0.1 and V-JEPA 2.1 ViT-L (Mur-Labadia et al., 2026) with a margin of 0.05. Following the original work, we set the alignment loss weight to 0.5, half the flow-matching loss weight.

Tempered Self-similarity Alignment details. Tempered Self-similarity Alignment (TSA) (Kim et al., 2026b) converts pairwise cosine similarities into correspondence distributions using a temperature-scaled softmax over the spatial tokens of each target frame. We use its masked variant (M-TSA), which restricts queries to the top 20% of tokens in each temporal slice, ranked by patch-wise ℓ_1 frame differences in the teacher input. We reuse the VideoREPA pipeline with VideoMAEv2 ViT-B, replacing the alignment objective with the KL divergence from the teacher distribution to the student distribution, averaged over selected queries and target frames. We set the temperature parameter to $\tau = 0.1$ and set the alignment loss weight to 0.5.

B.4 Analysis Details

Motion score analysis details. We evaluate how well the latent motion score identifies each region in Table 4a using the area under the receiver operating characteristic curve (AUROC). For each region, we treat locations inside the region as positives and those outside as negatives, using the motion

score as the prediction score. Higher AUROC indicates that the score preferentially highlights the region, with 0.5 corresponding to chance-level discrimination. The ground-truth evaluation regions are obtained from the segmentation masks provided by the simulator, where the moving arm, hand, and objects are selected based on centroid displacements of at least two pixels.

Attention ratio. We define the attention ratio metric used in Table 4b as the mean attention share assigned to a region divided by its mean spatial coverage:

$$\mathrm{AttentionRatio}(R) = \frac{\mathbb{E}\left[\sum_{i=1}^{N} a_i c_{R,i}\right]}{\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} c_{R,i}\right]},$$

where a_i denotes action-token attention normalized over the N visual tokens within each temporal block, and $c_{R,i} \in [0, 1]$ is the fraction of token i's spatial area covered by region R. A value of one corresponds to uniform attention, while larger values indicate greater concentration relative to region size. We aggregate across evaluation windows and the conditioning and two future-latent blocks. We select the top four layers with the highest dynamic-region attention in the baseline model, and evaluate both models on the same layers to observe the dynamic region attention change.

B.5 Hyperparameters and further details

For the motivation study in Figure 3, we set the temperature parameter β=2. For Motion-Guided Asymmetric Prediction (MoGAP), we set the auxiliary loss weight to $\lambda_{\mathrm{MoGAP}} = 1.0$. For backbones using the Wan2.2 causal VAE tokenizer (Wan et al., 2025), the tokenizer first encodes a single-frame chunk and then encodes the remaining frames in four-frame chunks. Because the first chunk has a different temporal extent, we compute the motion score for the first predicted latent using the forward difference $m_{n,p} = \|\mathbf{z}_{t,n+1,p} - \mathbf{z}_{t,n,p}\|_2$. To avoid division by zero when all motion scores vanish, we lower-bound the normalization denominator by $\delta = 10^{-8}$. This yields $\widetilde{m}_{n,p} = 0$ and uniform loss weights. We did not encounter this case in our experiments.

C Real-Robot Experiments Details

C.1 Hardware setup

We use the OpenArm humanoid[4], which consists of two 7-DoF arms, each equipped with a 6-DoF Inspire RH56F1 hand[5]. The robot head is mounted on a 2-DoF neck supporting pitch and yaw motions, with a ZED 2i stereo camera mounted on the neck. During data collection, both the left and right egocentric camera views are recorded, while only the left view is used as visual input for training and evaluation. We use ExoArm-7, a wearable exoskeleton compatible with OpenArm, to teleoperate both robot arms, and Manus Pro gloves to track fingertip positions, from which finger joint angles are computed via inverse kinematics. The table height is kept fixed throughout data collection and evaluation.

C.2 Real-Robot Tasks

We evaluate real-robot manipulation across six settings, comprising two in-domain settings and four generalization settings. The in-domain settings evaluate basic manipulation capabilities through pick-and-place and air-fryer manipulation. The generalization settings evaluate robustness to unseen objects, unseen environments, and dynamic object relocation under human interference.

In-domain tasks. We consider two in-domain settings: pick-and-place and air-fryer manipulation. For pick-and-place, we evaluate three object–destination pairs: banana to basket, mango to air fryer, and apple to air fryer. The robot is instructed to pick up the target object and place it at the specified destination. Each pair is evaluated from three predefined initial object locations. For air-fryer manipulation, the robot is instructed to either open or close the air fryer, with the two operations evaluated separately (Figure 10).

OOD objects. We evaluate object-level generalization by replacing the in-domain objects with four unseen objects which are not included in the training set: a box, a doll, a cube, and a tennis ball. Note that the doll used in this evaluation differs in shape and texture from the dolls seen during training. The robot is instructed to pick up each object and place it into either the basket or the air fryer. We use the same three predefined initial locations as in the in-domain pick-and-place setting (Figure 11).

OOD environments. We evaluate environment-level generalization by varying the visual background while keeping the manipulation setup otherwise unchanged. Specifically, we change the tabletop covering to introduce backgrounds not observed in the in-domain setting (Figure 12).

OOD object relocation. We evaluate robustness to dynamic object displacement by introducing human interference during the in-domain pick-and-place tasks. During execution, a human relocates the target object, requiring the robot to adapt to its updated position. We consider two difficulty levels. In set I, the object is relocated by a moderate distance at an earlier stage of the robot's approach. In set II, the object is displaced by a larger distance and the relocation occurs when the robot hand is closer to the object, requiring stronger adaptation to task-relevant visual changes (Figure 13).

C.3 Real-World Training and Evaluation Details

Dataset. We use a single dataset of teleoperated expert demonstrations across 21 tasks on our real-world setup, including the object pick-and-place and air fryer open-and-close in-domain tasks. We visualize the verbs and objects of our OpenArm dataset in Figure 14.

Training details. For real-robot experiments, the dataset is substantially smaller than that of the simulated benchmark, and we find that conditioning on historical observations and applying image augmentations help mitigate overfitting. We adopt a video-to-video generation setup conditioned on four past frames sampled with a stride of two. For image augmentation, we randomly crop 5% of the image and resize back to its original size. We also uniformly sample brightness, contrast, saturation factors, and hue shifts from $[0.7, 1.3]$, $[0.6, 1.4]$, $[0.5, 1.5]$, and $[-0.08, 0.08]$, respectively. We train a single model for each method, for 8K optimization steps with a global batch size of 256.

[4] https://openarm.dev/

[5] https://en.inspire-robots.com/product/rh56f1/

Evaluation details. We evaluate each policy over 24 trials per task, except for in-domain pick-and-place, where we use 48 trials to cover a more diverse set of objects, yielding 168 trials per method. For pick-and-place, a trial is successful if the robot grasps the target object and places it at the designated destination. For air-fryer opening and closing, success requires moving the appliance to the fully open or closed state, respectively.

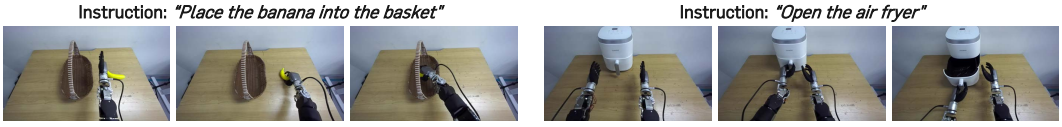


Figure 10: In-domain task examples.

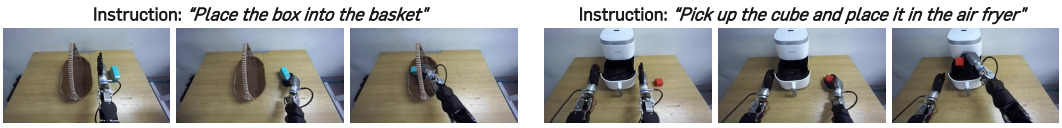


Figure 11: Unseen objects task examples.

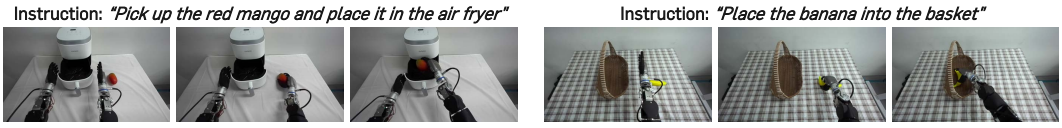


Figure 12: Unseen environments task examples.

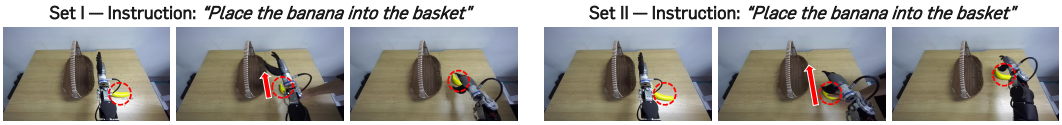


Figure 13: Object relocation task examples (left: Set I; right: Set II)

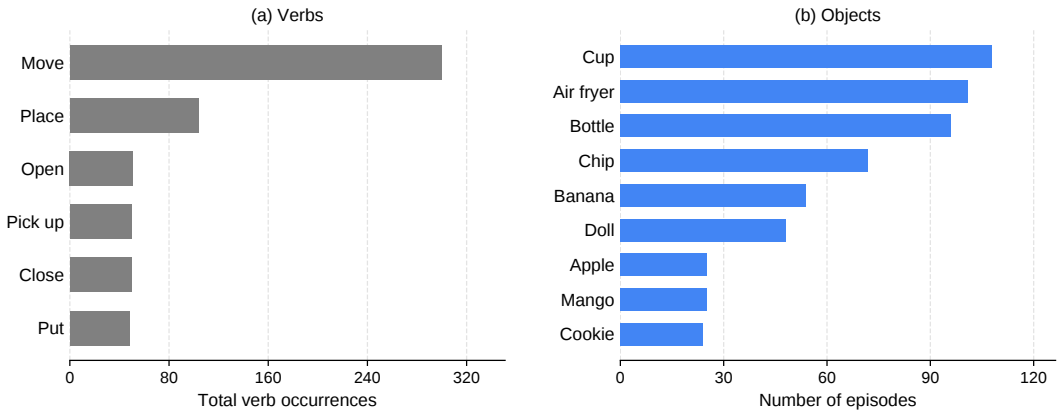


Figure 14: Visualization of the OpenArm dataset distribution.