WanPolicy: Representation Alignment for Robust and Efficient World-Action Models

Abstract

World-Action Models (WAMs) repurpose pretrained video diffusion backbones for robot manipulation by jointly modeling future visual observations and actions. Existing approaches rely on video diffusion objectives to learn future dynamics, implicitly assuming that pixel-level reconstruction benefits downstream control. In this work, we challenge this paradigm. We present WanPolicy, which replaces the diffusion objective with a representation alignment loss that supervises intermediate DiT features using a frozen DINOv3 encoder; rather than training the DiT to predict future pixels, we train it to predict future DINOv3 features, which we show are more semantically relevant for robot control. Importantly, we fine-tune the DiT backbone as a regressor, eliminating the need for iterative denoising at inference, requiring only a single forward pass through the video backbone which results in significant speedups. On the LIBERO benchmark, WanPolicy matches standard vision language action (VLA) models and recent WAMs. Crucially, WanPolicy does so by training on much less robot data than prior work, OpenVLA-OFT (1M episodes in Open X-Embodiment vs 2K episodes in LIBERO), indicating that real-world human videos contain more robot-relevant priors than vision-language data. As such benchmarks appear saturated due to the alignment between test and training conditions, we further evaluate on LIBERO-Plus, a benchmark designed to evaluate generalization to environmental perturbations. Here, WanPolicy substantially outperforms prior methods with a success rate of 80% (compared to prior art of 69%).

1 Introduction

Pretrained video diffusion models (31; 5; 33) have recently emerged as a compelling foundation for robot learning because they encode rich priors over temporal dynamics, object interactions, and scene geometry. Building on this observation, a growing line of work repurposes video/image generative backbones for control, using them either to extract predictive visual representations (11; 25; 9; 4) or to jointly model future observations and robot actions (21; 34; 36; 2; 8). Collectively, these works suggest that video-based pretraining offers a promising alternative to conventional vision language action (VLA) backbones (3; 14; 18; 16; 6; 38; 22) for embodied control.

Despite their strong empirical performance, most existing WAMs still rely on the training objective of pixelwise video generation. We see two limitations to this dominant approach. First, pixel-space (or latent-frame) generation may be wasteful or "overkill" for control: manipulation depends primarily on representations that capture semantics, geometry, and action-relevant state, rather than pixel-perfect imagery. Second, iterative diffusion-based denoising is compute heavy and introduces substantial latency, limiting its applicability to closed-loop control.

In this paper, we revisit the role of future prediction in WAMs and ask a different question: does robot control really require predicting future pixels, or does it instead require learning action-relevant world representations? We present WanPolicy, a simple alternative to standard WAM training that replaces the video diffusion objective with a representation alignment objective (35; 19; 37). Instead

of supervising the model to generate future pixels, we supervise the model to generate future features from a frozen DINOv3 encoder (28). Our premise is that these features provide a more suitable target for policy learning: they are semantically structured and more invariant to irrelevant visual changes, properties that are more important for action prediction than pixel-accurate futures.

This change in objective leads to two key advantages. It directly optimizes the video backbone for control-oriented visual representations rather than high-fidelity pixel generation. Moreover, because we train the video branch with a regression-based DINO alignment loss, we can create predictive, action-relevant future states in a single forward pass of the video backbone, significantly speeding up inference. The result is a WAM-like policy that retains the priors from real-world pretraining while substantially simplifying deployment for real-time manipulation.

We show that WanPolicy matches both standard VLA baselines and WAM baselines on LIBERO (23), while achieving substantially stronger robustness on LIBERO-Plus (10) under environmental perturbations. WanPolicy does so while training on much less robot data than prior works, indicating that real-world human videos used to pretrain video generation models contain more robot-relevant priors than vision-language data. More broadly, our results suggest that the main value of video generation for robot learning may not lie in synthesizing future pixels, but in the structured world representations that can be aligned to control.

2 Related Work

Vision-Language-Action Models. Vision-Language-Action (VLA) models learn robot policies by mapping visual observations and language directly to actions, typically without explicitly modeling future scene evolution (18; 16; 3; 14; 6; 38; 22). This paradigm has enabled strong generalization through large-scale multimodal pretraining, but it often relies on backbones pretrained on static image-text data and therefore does not directly exploit predictive structure over future world dynamics.

World-Action Models and video-based policy learning. A parallel line of research explores how pretrained video generative models can serve as foundations for robot control by exploiting their learned representations of dynamics, interactions, and scene structure over time. One group of methods uses generated visual futures as an explicit intermediate for control. SuSIE (4) generates intermediate visual subgoals with an image-editing diffusion model and executes them with a separate low-level goal-conditioned policy, while UniPi and NovaFlow (9; 20) formulate policy learning as text-guided video generation and then derive actions with learned or heuristic inverse dynamics modeling. A second group uses video models primarily as predictive backbones for action learning. VPP and Mimic-Video (11; 25) condition policy learning on predictive visual representations extracted from a video diffusion model rather than acting directly from generated pixels. More recent world-action models couple future world modeling and action generation more tightly within a shared architecture: LingBot-VA (21) learns frame prediction and policy execution simultaneously in an autoregressive diffusion framework; DreamZero (34) jointly models future world states and actions with a pretrained video diffusion backbone; Fast-WAM (36) studies whether the benefits of WAMs come from joint video-action training or explicit future imagination at test time; and Motus (2) unifies video generation, action modeling, and related embodied capabilities within a latent action world model. Our work is most closely related to this latter line. However, unlike prior methods that retain a video generation objective during training, we replace future video reconstruction or denoising with direct representation alignment, training the video backbone to produce action-relevant world representations without iterative denoising at inference time.

3 Method

To provide a thorough understanding of WanPolicy, we first introduce the model design, including the architecture and the modified WAM formulation, in which the video branch is trained to predict future DINO representations while the action branch retains the standard flow-matching objective. We then discuss why this representation-level supervision is better aligned with action generation than pixel-level video prediction, and finally describe our intermediate read-out layer design, which further improves both performance and efficiency.

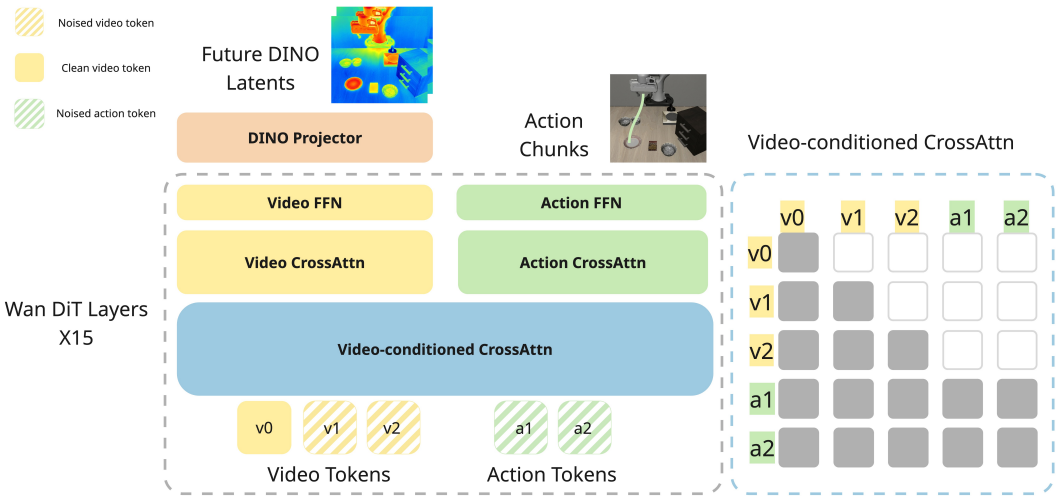


Figure 1: Overview of WanPolicy training pipeline. Some details are hidden for clarity. **Left**: video-conditioned cross attention and separate cross attention and FFN layers for video and action branch. Video branch is supervised by representation alignment loss with future DINO representation. Action branch is supervised by standard flow matching loss. We read out the features from the intermediate layer of DiTs and discard the later parts. **Right**: attention mask for video-conditioned cross attention. Video tokens attend to other video tokens but not action tokens. Action tokens attend to both video and action tokens.

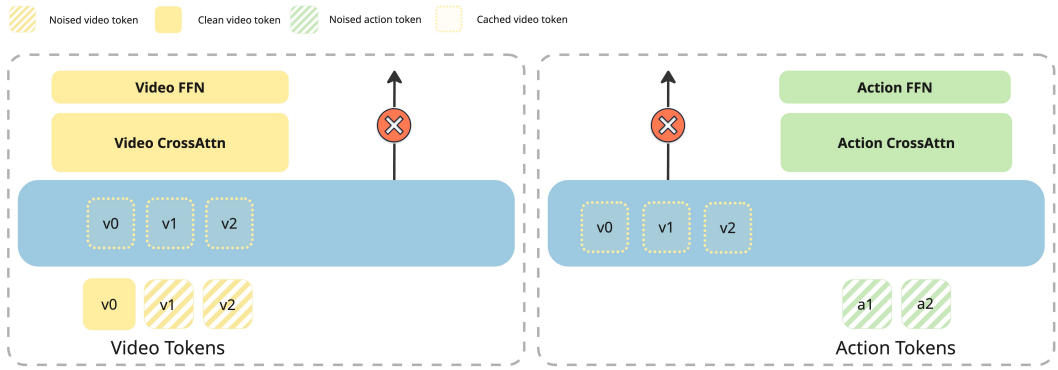


Figure 2: Inference pipeline

3.1 Model Design

Architecture.

We build upon WAN2.2 TI2V 5B (31), a video diffusion transformer pretrained on web-scale video data, and adopt a similar architecture design as Fast-WAM and Lingbot-VA (21; 36) that augments the backbone with action tokens. We modify the original self attention layer from WAN to design a video-conditioned cross attention for action tokens: under a structured mask, video tokens attend only to video tokens, which serve as the conditioning signal for action, while action tokens attend to both streams. After video-conditioned cross attention, separate cross-attention branches condition the video branch on T5 instruction embeddings (27), and the action branch on the same instruction embeddings together with encoded robot proprioceptive states.

Training Design. We start from the standard world-action model (WAM) formulation, in which a video backbone models future visual observations while an action branch predicts future action chunks. Prior WAMs typically train both branches with diffusion or flow-matching objectives, so that the model jointly denoises future video latents and actions. Our key modification is to retain the

standard flow-matching objective for action prediction while replacing the video-generation objective with a representation-alignment objective.

Let $\boldsymbol{v}_{1:T}$ denote the target future video latents. A standard video diffusion formulation constructs a noisy input by sampling Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ and a timestep $t \sim \mathcal{U}(0, 1)$, and defining

$$\boldsymbol{v}^t_{1:T} = (1 - t)\boldsymbol{v}_{1:T} + t\epsilon. \tag{1}$$

Standard WAMs then train the video branch to predict a denoising target from $\boldsymbol{v}^t_{1:T}$. In contrast, we supervise the video branch using frozen DINOv3 features extracted from the target future frames. In our setup, we always use $t = 1$, so the future video input is pure Gaussian noise, i.e., $\boldsymbol{v}^t_{1:T} = \epsilon$.

Formally, given initial observation o and language instruction l, the video branch predicts future visual representations, which are trained with the loss

$$\mathcal{L}_{\text{REP}} = 1 - \text{Avg}\big(\text{sim}\big(f_\theta(o, \boldsymbol{v}^t_{1:T}, t, l), \boldsymbol{d}_{1:T}\big)\big), \tag{2}$$

where f_θ denotes the video branch, and $\boldsymbol{d}_{1:T}$ denotes the frozen DINOv3 features of the target future frames. We compute cosine similarity at each spatiotemporal token and average across all frames and tokens.

For the action branch, we keep the standard flow-matching formulation. Let $\boldsymbol{a}_{1:H}$ denote a target action chunk. We define its noisy version as

$$\boldsymbol{a}^t_{1:H} = (1 - t)\boldsymbol{a}_{1:H} + t\epsilon, \tag{3}$$

and optimize

$$\mathcal{L}_{\text{Action}} = \mathbb{E}_{\boldsymbol{a},\epsilon,t}\left[\left\|g_\phi(\boldsymbol{a}^t_{1:H}, t, l, p) - (\epsilon - \boldsymbol{a}_{1:H})\right\|^2_2\right], \tag{4}$$

where g_ϕ denotes the action branch and p denotes the robot proprioceptive state.

The final training objective is

$$\mathcal{L} = \mathcal{L}_{\text{REP}} + \mathcal{L}_{\text{Action}}. \tag{5}$$

Inference Design. Our modification changes the inference procedure only for the video branch. In standard WAMs, future visual prediction is formulated as an iterative denoising process, which requires multiple sampling steps through the video backbone before obtaining features that can be used for action generation. In WanPolicy, the video branch is instead trained with a regression-based representation objective, so it directly predicts future action-conditioning representations from the initial observation, language instruction, and noisy video input in a single forward pass. This eliminates the need for iterative denoising in the video branch.

In practice, we first run the video branch once to obtain the predicted future visual representations as shown on the left hand side of Figure 2. Since the video-conditioned cross attention layers are only evaluated once for the video tokens, their key-value states can be cached and reused during subsequent action decoding. We then perform iterative denoising only in the action branch, shown on the right hand side of Figure 2, which is substantially smaller and therefore much faster than running the full video backbone for multiple denoising steps. This design preserves the benefits of action-space flow matching while significantly reducing overall inference latency.

3.2 Learning Action-Relevant World Representation

In this section, we shed light on the key design choice to replace the standard video diffusion objective with a regression-based representation objective for the video branch. The motivation is that, for action generation, reconstructing future frames in pixel or video latent space with pixel-accurate detail is wasteful. Instead, the model primarily needs future representations that preserve coarse but actionable information relevant to control, such as object configuration, scene geometry, and temporal evolution under interaction.

Based on this observation, we supervise the video branch using frozen DINO features extracted from future frames, rather than diffusion or flow-matching targets in the video latent space. This encourages the backbone to predict compact future representations that are more directly aligned with downstream action decoding, while reducing emphasis on low-level visual details that are less relevant for control.

This design also changes the inference procedure. In standard video diffusion models, future prediction requires iterative denoising, which introduces substantial inference latency. In contrast, because our video branch is trained with a regression objective, it predicts future representations in a single forward pass. We therefore avoid multi-step sampling in the video branch and use its predicted representations directly for action generation.

3.3 Read-out Layer Design

Another important design in WanPolicy is that the video and action predictions are not necessarily read from the final layer of the video diffusion transformer. Instead, both branches operate on intermediate backbone features from layer K. In practice, this means that layers after K are discarded during training and inference, and both the visual representation prediction and action prediction are computed from the features at layer K.

This design is motivated by the observation from Huang et al. (12), that the deepest layers of a video diffusion transformer are specialized toward the native video generation objective, and may therefore place greater emphasis on pixel-level reconstruction details than is necessary for control. By reading out from an intermediate layer, we aim to preserve higher-level scene structure and dynamics while avoiding over-specialization to appearance. We study the effect of K in Sec. 4.2.1 and show that the middle read-out layer yields the best performance.

4 Empirical analysis

We evaluate WanPolicy on both standard manipulation performance and robustness under environmental perturbations. We first compare against strong VLA and WAM baselines on LIBERO and LIBERO-Plus, showing that WanPolicy remains competitive on the standard benchmark while achieving substantially stronger robustness on the harder generalization setting. We then perform ablations on the two central design choices of our method, namely the read-out layer and the supervision target for the video branch, and conclude with an efficiency analysis that quantifies the inference-time benefit of removing iterative video denoising.

Implementation details We use the pretrained WAN2.2 TI2V 5B (31) model as video backbone. The action expert backbone is warm-started from the pretrained video DiT via variance-preserving linear interpolation of its weights to the target hidden dimension ($3072 \rightarrow 1024$), while task-specific input/output projections are randomly initialized. We use read-out layer $K = 15$ unless otherwise specified.

For data processing, we follow the Fast-WAM convention (36) and temporally downsample input videos by a factor of 4, resulting in 9 frames per chunk. Under this setup, the action horizon is 32. During training, we predict the full 32-step action chunk, and at inference time we execute the predicted actions over the entire horizon.

We use two camera views provided by the training data: an agent view capturing the overall scene and a gripper view mounted on the robot arm. The two views are concatenated into a single video input before being processed by the visual backbone.

All experiments are conducted on two types of hardware platforms: $8\times$ NVIDIA H100 GPUs and $8\times$ NVIDIA A6000 GPUs. We train all models using AdamW with a learning rate of 1×10^{-4}, weight decay of 0.01, and a cosine annealing learning rate scheduler. We train our model and all the related ablation variants for 20K with a total batch size of 96.

Datasets We evaluate WanPolicy on LIBERO (23), a standard benchmark for language-conditioned robot manipulation in simulation, and further assess its robustness on LIBERO-Plus (10), which extends LIBERO with a diverse set of environmental perturbations designed to stress-test policy generalization. For LIBERO, we rollout 50 trials for each task, creating a total of 2000 trials, and we measure success rate over all the trials. For LIBERO-Plus, we follow the standard evaluation protocol which rollout a total of around 10K trails with different types of perturbations, and we measure the success rate over all the trials.

Baselines We compare WanPolicy to a wide range of VLA and WAM methods. In addition to comparisons with both VLA and WAM baselines, we conduct a series of ablation studies to analyze the contribution of our design choices. Specifically, we first study the effect of the read-out layer

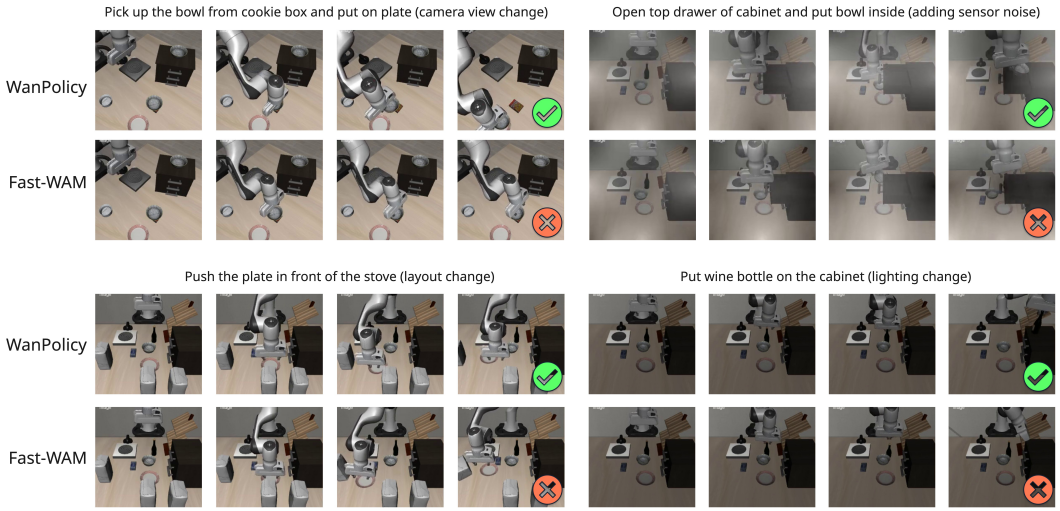


Figure 3: LIBERO-Plus qualitative comparisons between WanPolicy and the Fast-WAM model. Even though the two models use the same video backbone, WanPolicy's representation alignment objective makes it more robust under environmental perturbations. We visualize the rollout results for two models under the perturbation of camera view point, sensor noise, table-top layout, and lighting condition.

design by varying the layer from which visual features and action tokens are extracted. We then ablate the supervision target used for training the video backbone, comparing DINO features against alternative representations. These experiments highlight the advantage of DINO-based supervision, particularly in terms of robustness under distribution shifts.

4.1 Comparison to state-of-the-art

LIBERO. Table 1 shows that WanPolicy achieves competitive performance on LIBERO without any large-scale robot-data pretraining. This stands in clear contrast to several strong VLA baselines. OpenVLA is pretrained on 970k robot episodes from Open X-Embodiment (24), and OpenVLA-OFT inherits the same OpenVLA initialization before applying its optimized fine-tuning recipe on LIBERO (18; 16). Likewise, π_0 is pretrained on over 10K hours of robot data from a diverse cross-embodiment mixture, in addition to internet-scale VLM pretraining (3). In contrast, WanPolicy is trained only on LIBERO for 20k iterations, yet remains competitive with these heavily pretrained baselines.

These results support our hypothesis that the priors embedded in large video models—learned from web-scale human videos and transferred through our representation-alignment objective—provide a strong foundation for policy learning. In particular, the comparison indicates that video priors may offer a more effective source of action-relevant supervision than the static vision-language pretraining typically used in VLA backbones.

LIBERO-Plus. We next evaluate on LIBERO-Plus, which introduces diverse environmental perturbations and therefore provides a more direct test of robustness and generalization. As shown in Table 2, WanPolicy achieves the best overall success rate of 80.0, outperforming the prior state of the art, OpenVLA-OFT, by 10.4 points. We further outperform the concurrent work Fast-WAM by 9.8 points. This result is particularly notable because Fast-WAM is already a strong video-based policy baseline, suggesting that the gain comes not merely from using a video-pretrained backbone, but from learning a more action-relevant world representation.

A closer look at the perturbation breakdown shows that WanPolicy delivers the strongest performance on Camera (90.4), Noise (94.8), Light (97.4), and Layout (78.4), while also remaining highly competitive on Language (79.2). These improvements are consistent with our design motivation: supervising the video branch with DINO features encourages invariance to nuisance visual changes while preserving the scene structure and dynamics needed for action generation. The overall LIBERO-

Model	Robot Pretraining	Spatial	Object	Goal	Long	Overall
OpenVLA (18)	✓	84.7	88.4	79.2	53.7	76.5
OpenVLA-OFT (16)	✓	97.6	98.4	97.9	94.5	97.1
Pi0 (3)	✓	96.8	98.8	95.8	85.2	94.1
Pi0.5 (14)	✓	98.8	98.2	98.0	92.4	96.9
LingBot-VA (21)	✓	98.5	99.6	97.2	98.5	98.5
Fast-WAM (36)	✗	98.2	100.0	97.0	95.2	97.6
Cosmos Policy (17)	✗	98.1	100.0	98.2	97.6	98.5
WanPolicy	✗	94.6	99.2	97.8	95.0	96.7

Table 1: Comparison on LIBERO benchmark suites. WanPolicy approaches the state-of-the-art on LIBERO while training on far fewer robot data; we trained for 20k iterations using solely LIBERO data, while past work pretrained on large robotic datasets before finetuning on LIBERO. Overall, many approaches saturate this benchmark since the test set is similar to the train set. We evaluate generalization on the more challenging LIBERO-Plus in Table 2.

Model	Camera	Robot	Language	Light	Background	Noise	Layout	Total
OpenVLA (18)	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
OpenVLA-OFT (16)	56.4	31.9	**79.5**	88.7	**93.3**	75.8	74.2	69.6
NORA (13)	2.2	37.0	65.1	45.7	58.6	12.8	62.1	39.0
WorldVLA (7)	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
UniVLA (32)	1.8	46.2	69.6	69.0	81.0	21.2	31.9	43.9
π_0 (3)	13.8	6.0	58.8	85.0	81.4	**79.0**	68.9	53.6
π_0-Fast (26)	**65.1**	21.6	61.0	73.2	73.2	74.4	68.8	61.6
RIPT-VLA (29)	55.2	31.2	77.6	88.4	**91.6**	73.5	74.2	68.4
Fast-WAM (36)	48.1	**72.5**	70.0	**96.7**	66.0	69.7	**78.1**	**70.2**
WanPolicy	**90.4**	**54.9**	**79.2**	**97.4**	65.4	**94.8**	**78.4**	**80.0**

Table 2: Comparison on LIBERO-Plus benchmark. WanPolicy achieves the best overall performance and outperforms prior methods across several generalization domains. Top-three results in each column are highlighted: best, second best, and third best.

Plus results show that the proposed representation objective yields substantially stronger robustness than prior VLA and WAM approaches, even without large-scale robot-data pretraining.

Qualitative Results. We provide visual comparisons between WanPolicy and the contemporary Fast-WAM on LIBERO-Plus under perturbations in camera viewpoint, sensor noise, table-top layout, and lighting condition, as shown in Figure 3. The results show that WanPolicy is more robust to these perturbations, which we attribute to the DINO feature supervision used during training. Concretely, the visual comparisons show that Fast-WAM suffers from issues such as getting stuck under camera viewpoint changes, unstable motion under layout changes, and reduced precision under sensor noise or lighting changes. In contrast, WanPolicy is generally more robust under these visual perturbations and completes the tasks successfully.

4.2 Ablations

4.2.1 Read-out Layer Ablation

A key design choice in WanPolicy is to read out the video features and action features from an intermediate DiT layer, rather than using the output from the final layer. Concretely, we truncate the backbone at layer K, extract the output feature for both video and action, and use them for training and inference. Table 3 shows that this design is important: using the middle layer ($K = 15$) yields the best performance on both LIBERO and LIBERO-Plus, outperforming both the final layer ($K = 30$) and a much earlier layer ($K = 8$).

Layer	Spatial	Object	Goal	10	LIBERO-Overall	LIBERO-Plus
30	90.6	99.6	97.0	91.0	94.6	75.0
20	94.2	99.0	96.8	94.0	96.0	78.3
15	94.6	99.2	97.8	95.0	96.7	80.0
8	91.2	96.8	94.6	82.8	91.4	70.3

Table 3: Ablating the read-out transformer layer. We find better results by defining the action head on features extracted from the 15th layer of the DiT rather than the final (30th) layer. We posit that "action semantics" can be recovered from early layers that by-pass later layers that may focus on detailed pixel generation.

We hypothesize that this behavior reflects a trade-off between semantic abstraction and task relevance. In video diffusion transformers, deeper layers are optimized to model fine-grained reconstruction details that are useful for pixel synthesis but less relevant for control. Continuing to optimize these late layers for policy learning may therefore waste both gradient and compute on features that do not directly improve action generation. In contrast, very early layers do not yet provide sufficiently rich visual processing or enough interaction between visual and action tokens, which limits their usefulness for downstream control. The intermediate layers strike a better balance: they preserve high-level scene structure and dynamics while avoiding over-specialization to pixel-level generation.

This observation is also consistent with recent findings in 3D understanding from video models. Huang et al. (12) show that, across several video diffusion backbones, the most informative features for downstream 3D read-out are consistently found in mid-layer representations rather than in the final layers. Their results suggest that intermediate features retain more structured geometric information, while the deepest layers become increasingly specialized for the native generation objective. Our ablation indicates a similar phenomenon in robot control: the best action-relevant representations are obtained not from the final DiT output, but from an intermediate stage where the model has developed sufficiently strong world representations without overcommitting to pixel reconstruction.

4.2.2 Supervision Representation Ablation.

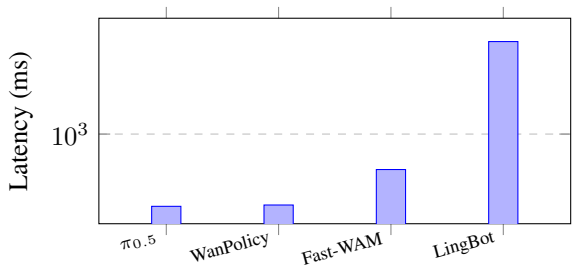
We next ablate the supervision target used for the video branch. In order to create a fair comparison, we keep other design choices intact. Concretely, we always use the same model architecture and read out the features of the 20_{th} layer for all candidate methods. During inference, we always run a single forward pass for the video backbone and perform iterative denoising on the action branch. As shown in Table 4, DINO-based supervision achieves the best overall performance across both benchmarks, yielding the strongest robustness on LIBERO-Plus while remaining competitive on standard LIBERO. Replacing DINO supervision with the standard video flow-matching objective slightly improves in-domain LIBERO performance, but leads to a noticeable drop on LIBERO-Plus.

We further consider an *action-only* variant, where the video branch receives no DINO or video flow-matching supervision and the model is trained only through the action objective. Removing video-side supervision degrades LIBERO performance more substantially, indicating that explicit future-oriented visual supervision is still important for strong in-domain policy learning. At the same time, the action-only variant performs better on LIBERO-Plus than the video flow-matching variant. This comparison is informative: although video flow matching helps fit the training distribution more closely, it also appears to encourage the backbone to model low-level visual details that are brittle under perturbations such as camera changes, noise, and layout shifts. We show more detailed Per-Perturbation breakdown at Appendix A.

Taken together, these results support our central claim that the choice of supervision target for the video branch matters more than simply adding an auxiliary visual objective. Standard video flow matching improves reconstruction-oriented in-domain performance, but DINO supervision yields representations that transfer better under distribution shift. We hypothesize that this is because DINO features emphasize higher-level semantic and structural information, whereas pixel-oriented video supervision pushes the model toward appearance details that are less relevant for action and less stable under perturbation.

Representation	DINO	Pixel	Action	Spatial	Object	Goal	10	LIBERO-Overall	LIBERO-Plus
DINOv3	✓	✗	✓	94.2	99.0	96.8	94.0	96.0	78.3
Video FM	✗	✓	✓	95.8	99.8	97.6	97.2	97.6	74.4
Action only	✗	✗	✓	93.8	99.4	91.0	92.2	94.1	77.2

Table 4: **Deva**: We ablate various supervised signals. Our default uses DINO and action supervision. When replacing DINO with pixel supervision (as much past WAM efforts do) performance on the challenging LIBERO-PLUS drops signifantly from 78 to 74. Interestingly, removing pixel supervision recovers some of that drop, producing a score of 77. Ablating supervision representations. We find that DINO feature supervision provides the best overall performance, especially on LIBERO-plus, indicating good robustness on environmental perturbations. Video flow matching slightly improves LIBERO performance but degrades robustness on LIBERO-Plus. The action-only variant underperforms on the training distribution, while also has slightly worse results on LIBERO-Plus.



Model	Latency(ms)
$\pi_{0.5}$	273.3
WanPolicy	279.9
Fast-WAM	529.1
LingBot-VA	5262.6

Figure 4: Rollout latency comparison. WanPolicy reduces rollout latency by nearly $19\times$ compared to LingBot-VA and about $1.9\times$ compared to Fast-WAM, while remaining comparable to the VLA baseline $\pi_{0.5}$.

4.3 Efficiency Analysis

A key advantage of WanPolicy is that it removes iterative denoising from the video branch at inference time. To quantify the resulting efficiency gain, we benchmark rollout latency under a unified protocol across WanPolicy, LingBot-VA, Fast-WAM, and the VLA baseline $\pi_{0.5}$. We measure the wall-clock GPU latency of one rollout chunk, defined as the time from receiving an observation to producing a chunk of actions. To ensure a fair comparison, all models are evaluated under the same input setting and timed with the same GPU synchronization protocol. We use the default inference setup of each method and report the average latency over 10 runs after warmup.

This benchmark directly reflects the systems-level difference between WanPolicy and prior WAMs. As shown in Figure 4, standard WAMs such as LingBot-VA repeatedly execute the full video backbone during iterative denoising, leading to 5262.6 ms of sampling time for a single action chunk. In contrast, WanPolicy predicts future visual representations in a single forward pass and performs iterative denoising only in the smaller action branch, resulting in a rollout latency of 279.9 ms. Fast-WAM also reduces video-side inference cost by avoiding repeated video-backbone denoising at test time, but still requires 529.1 ms per rollout chunk. By comparison, WanPolicy achieves an additional $\sim$1.9$\times$ speedup over Fast-WAM. Notably, the latency of WanPolicy is also comparable to the VLA baseline $\pi_{0.5}$, which runs at 273.3 ms. We attribute this further gain over Fast-WAM to our read-out layer design: instead of running the full DiT backbone, WanPolicy reads out from an intermediate layer, reducing both compute and inference cost while preserving strong policy performance.

5 Conclusion

We presented WanPolicy, a world-action model that replaces the standard video diffusion objective with a representation alignment objective for robot policy learning. Instead of training the video backbone to reconstruct future pixels, we supervise intermediate DiT features with frozen DINO representations, encouraging the model to learn visual features that are more directly relevant to control. This design also removes the need for iterative denoising in the video branch at inference time, enabling future representation prediction in a single forward pass.

Empirically, WanPolicy achieves competitive performance on LIBERO while requiring no large-scale robot-data pretraining, and substantially outperforms prior methods on LIBERO-Plus, where robustness under environmental perturbations is more directly tested. Our ablations further support two key conclusions: first, intermediate DiT layers provide the most useful read-out features for action generation; second, the choice of supervision target for the video branch is critical, with DINO-based supervision yielding better robustness than standard video flow matching.

Overall, our results suggest that the main value of pretrained video generative models for robot learning may not lie in pixel-level future synthesis itself, but in the structured world representations they contain and the ability to align these representations with downstream control objectives. This points to a simple alternative to standard WAM training: rather than optimizing video backbones for generation and hoping that useful control features emerge, one can directly optimize them for action-relevant future representation prediction.

There are several important directions for future work. First, due to compute limitations, we have not yet scaled WanPolicy to larger and more diverse robot datasets such as DROID (15) or BridgeData (30). Evaluating whether the same representation-alignment formulation continues to improve with broader robot data is an important next step. Second, our experiments only explore a limited set of supervision targets for the video branch. While DINO already yields strong results, it is likely not the optimal representation for control. More extensive investigation of alternative supervision spaces, including stronger predictive representation learning objectives such as JEPA-style targets (1), may further improve both in-domain performance and robustness. More broadly, we believe that identifying the most action-relevant supervision space for pretrained video backbones is a promising direction for building scalable and robust robot foundation policies.

References

[1] Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Komeili, M., Muckley, M.J., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., Arnaud, S., Gejji, A., Martin, A., Hogan, F.R., Dugas, D., Bojanowski, P., Khalidov, V., Labatut, P., Massa, F., Szafraniec, M., Krishnakumar, K., Li, Y., Ma, X., Chandar, S., Meier, F., LeCun, Y., Rabbat, M., Ballas, N.: V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. CoRR **abs/2506.09985** (2025)

[2] Bi, H., Tan, H., Xie, S., Wang, Z., Huang, S., Liu, H., Zhao, R., Feng, Y., Xiang, C., Rong, Y., Zhao, H., Liu, H., Su, Z., Ma, L., Su, H., Zhu, J.: Motus: A unified latent action world model. CoRR **abs/2512.13030** (2025)

[3] Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Shi, L.X., Tanner, J., Vuong, Q., Walling, A., Wang, H., Zhilinsky, U.: π_0: A vision-language-action flow model for general robot control. CoRR **abs/2410.24164** (2024)

[4] Black, K., Nakamoto, M., Atreya, P., Walke, H., Finn, C., Kumar, A., Levine, S.: Zero-shot robotic manipulation with pretrained image-editing diffusion models. CoRR **abs/2310.10639** (2023)

[5] Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)

[6] Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jackson, T., Jesmonth, S., Joshi, N.J., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, K., Levine, S., Lu, Y., Malla, U., Manjunath, D., Mordatch, I., Nachum, O., Parada, C., Peralta, J., Perez, E., Pertsch, K., Quiambao, J., Rao, K., Ryoo, M.S., Salazar, G., Sanketi, P.R., Sayed, K., Singh, J., Sontakke, S., Stone, A., Tan, C., Tran, H.T., Vanhoucke, V., Vega, S., Vuong, Q., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., Zitkovich, B.: RT-1: robotics transformer for real-world control at scale. In: Bekris, K.E., Hauser, K., Herbert, S.L., Yu, J. (eds.) Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023 (2023)

[7] Cen, J., Yu, C., Yuan, H., Jiang, Y., Huang, S., Guo, J., Li, X., Song, Y., Luo, H., Wang, F., Zhao, D., Chen, H.: Worldvla: Towards autoregressive action world model. CoRR **abs/2506.21539** (2025)

[8] Cheang, C., Chen, G., Jing, Y., Kong, T., Li, H., Li, Y., Liu, Y., Wu, H., Xu, J., Yang, Y., Zhang, H., Zhu, M.: GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. CoRR **abs/2410.06158** (2024)

[9] Du, Y., Yang, S., Dai, B., Dai, H., Nachum, O., Tenenbaum, J., Schuurmans, D., Abbeel, P.: Learning universal policies via text-guided video generation. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 (2023)

[10] Fei, S., Wang, S., Shi, J., Dai, Z., Cai, J., Qian, P., Ji, L., He, X., Zhang, S., Fei, Z., Fu, J., Gong, J., Qiu, X.: Libero-plus: In-depth robustness analysis of vision-language-action models. CoRR **abs/2510.13626** (2025)

[11] Hu, Y., Guo, Y., Wang, P., Chen, X., Wang, Y., Zhang, J., Sreenath, K., Lu, C., Chen, J.: Video prediction policy: A generalist robot policy with predictive visual representations. In: Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., Zhu, J. (eds.) Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025. Proceedings of Machine Learning Research, PMLR / OpenReview.net (2025)

[12] Huang, Z., Li, X., Lv, Z., Rehg, J.M.: How much 3d do video foundation models encode? CoRR **abs/2512.19949** (2025)

[13] Hung, C., Sun, Q., Hong, P., Zadeh, A., Li, C., Tan, U., Majumder, N., Poria, S.: NORA: A small open-sourced generalist vision language action model for embodied tasks. CoRR **abs/2504.19854** (2025)

[14] Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Galliker, M.Y., Ghosh, D., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., LeBlanc, D., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Ren, A.Z., Shi, L.X., Smith, L., Springenberg, J.T., Stachowicz, K., Tanner, J., Vuong, Q., Walke, H., Walling, A., Wang, H., Yu, L., Zhilinsky, U.: $\pi_{0.5}$: a vision-language-action model with open-world generalization. CoRR **abs/2504.16054** (2025)

[15] Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M.K., Chen, L.Y., Ellis, K., Fagan, P.D., Hejna, J., Itkina, M., Lepert, M., Ma, Y.J., Miller, P.T., Wu, J., Belkhale, S., Dass, S., Ha, H., Jain, A., Lee, A., Lee, Y., Memmel, M., Park, S., Radosavovic, I., Wang, K., Zhan, A., Black, K., Chi, C., Hatch, K.B., Lin, S., Lu, J., Mercat, J., Rehman, A., Sanketi, P.R., Sharma, A., Simpson, C., Vuong, Q., Walke, H.R., Wulfe, B., Xiao, T., Yang, J.H., Yavary, A., Zhao, T.Z., Agia, C., Baijal, R., Castro, M.G., Chen, D., Chen, Q., Chung, T., Drake, J., Foster, E.P., Gao, J., Herrera, D.A., Heo, M., Hsu, K., Hu, J., Jackson, D., Le, C., Li, Y., Lin, R., Ma, Z., Maddukuri, A., Mirchandani, S., Morton, D., Nguyen, T., O'Neill, A., Scalise, R., Seale, D., Son, V., Tian, S., Tran, E., Wang, A.E., Wu, Y., Xie, A., Yang, J., Yin, P., Zhang, Y., Bastani, O., Berseth, G., Bohg, J., Goldberg, K., Gupta, A., Gupta, A., Jayaraman, D., Lim, J.J., Malik, J., Martín-Martín, R., Ramamoorthy, S., Sadigh, D., Song, S., Wu, J., Yip, M.C., Zhu, Y., Kollar, T., Levine, S., Finn, C.: DROID: A large-scale in-the-wild robot manipulation dataset. In: Kulic, D., Venture, G., Bekris, K.E., Coronado, E. (eds.) Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024 (2024)

[16] Kim, M.J., Finn, C., Liang, P.: Fine-tuning vision-language-action models: Optimizing speed and success. CoRR **abs/2502.19645** (2025)

[17] Kim, M.J., Gao, Y., Lin, T., Lin, Y., Ge, Y., Lam, G., Liang, P., Song, S., Liu, M., Finn, C., Gu, J.: Cosmos policy: Fine-tuning video models for visuomotor control and planning. CoRR **abs/2601.16163** (2026)

[18] Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E.P., Sanketi, P.R., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., Finn, C.: Openvla: An open-source vision-language-action model. In: Agrawal, P., Kroemer, O., Burgard, W. (eds.) Conference on Robot Learning, 6-9 November 2024, Munich, Germany. Proceedings of Machine Learning Research

[19] Leng, X., Singh, J., Hou, Y., Xing, Z., Xie, S., Zheng, L.: REPA-E: unlocking VAE for end-to-end tuning with latent diffusion transformers. CoRR **abs/2504.10483** (2025)

[20] Li, H., Sun, L., Hu, Y., Ta, D., Barry, J., Konidaris, G., Fu, J.: Novaflow: Zero-shot manipulation via actionable flow from generated videos. CoRR **abs/2510.08568** (2025)

[21] Li, L., Zhang, Q., Luo, Y., Yang, S., Wang, R., Han, F., Yu, M., Gao, Z., Xue, N., Zhu, X., Shen, Y., Xu, Y.: Causal world modeling for robot control. CoRR **abs/2601.21998** (2026)

[22] Li, X., Liu, M., Zhang, H., Yu, C., Xu, J., Wu, H., Cheang, C., Jing, Y., Zhang, W., Liu, H., Li, H., Kong, T.: Vision-language foundation models as effective robot imitators. In: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net (2024)

[23] Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: LIBERO: benchmarking knowledge transfer for lifelong robot learning. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 (2023)

[24] O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., et al.: Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 6892–6903. IEEE (2024)

[25] Pai, J., Achenbach, L., Montesinos, V., Forrai, B., Mees, O., Nava, E.: mimic-video: Video-action models for generalizable robot control beyond vlas. CoRR **abs/2512.15692** (2025)

[26] Pertsch, K., Stachowicz, K., Ichter, B., Driess, D., Nair, S., Vuong, Q., Mees, O., Finn, C., Levine, S.: FAST: efficient action tokenization for vision-language-action models. CoRR **abs/2501.09747** (2025)

[27] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 140:1–140:67 (2020)

[28] Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S.E., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: Dinov3. CoRR **abs/2508.10104** (2025)

[29] Tan, S., Dou, K., Zhao, Y., Krähenbühl, P.: Interactive post-training for vision-language-action models. CoRR **abs/2505.17016** (2025)

[30] Walke, H.R., Black, K., Zhao, T.Z., Vuong, Q., Zheng, C., Hansen-Estruch, P., He, A.W., Myers, V., Kim, M.J., Du, M., Lee, A., Fang, K., Finn, C., Levine, S.: Bridgedata V2: A dataset for robot learning at scale. In: Tan, J., Toussaint, M., Darvish, K. (eds.) Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA. pp. 1723–1736. Proceedings of Machine Learning Research, PMLR (2023)

[31] Wang, A., Ai, B., Wen, B., Mao, C., Xie, C., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Meng, X., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z., Liu, Z.: Wan: Open and advanced large-scale video generative models. CoRR **abs/2503.20314** (2025)

[32] Wang, Y., Li, X., Wang, W., Zhang, J., Li, Y., Chen, Y., Wang, X., Zhang, Z.: Unified vision-language-action model. CoRR **abs/2506.19850** (2025)

[33] Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024)

[34] Ye, S., Ge, Y., Zheng, K., Gao, S., Yu, S., Kurian, G., Indupuru, S., Tan, Y.L., Zhu, C., Xiang, J., Malik, A., Lee, K., Liang, W., Ranawaka, N., Gu, J., Xu, Y., Wang, G., Hu, F., Narayan, A., Bjorck, J., Wang, J., Kim, G., Niu, D., Zheng, R., Xie, Y., Wu, J., Wang, Q., Julian, R., Xu, D., Du, Y., Chebotar, Y., Reed, S., Kautz, J., Zhu, Y., Fan, L.J., Jang, J.: World action models are zero-shot policies. CoRR **abs/2602.15922** (2026)

[35] Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., Xie, S.: Representation alignment for generation: Training diffusion transformers is easier than you think. In: The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025. OpenReview.net (2025)

[36] Yuan, T., Dong, Z., Liu, Y., Zhao, H.: Fast-wam: Do world action models need test-time future imagination? CoRR **abs/2603.16666** (2026)

[37] Zhang, X., Liao, J., Zhang, S., Meng, F., Wan, X., Yan, J., Cheng, Y.: Videorepa: Learning physics for video generation through relational alignment with foundation models. CoRR **abs/2505.23656** (2025)

[38] Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., Vuong, Q., Vanhoucke, V., Tran, H.T., Soricut, R., Singh, A., Singh, J., Sermanet, P., Sanketi, P.R., Salazar, G., Ryoo, M.S., Reymann, K., Rao, K., Pertsch, K., Mordatch, I., Michalewski, H., Lu, Y., Levine, S., Lee, L., Lee, T.E., Leal, I., Kuang, Y., Kalashnikov, D., Julian, R., Joshi, N.J., Irpan, A., Ichter, B., Hsu, J., Herzog, A., Hausman, K., Gopalakrishnan, K., Fu, C., Florence, P., Finn, C., Dubey, K.A., Driess, D., Ding, T., Choromanski, K.M., Chen, X., Chebotar, Y., Carbajal, J., Brown, N., Brohan, A., Arenas, M.G., Han, K.: RT-2: vision-language-action models transfer web knowledge to robotic control. In: Tan, J., Toussaint, M., Darvish, K. (eds.) Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA. pp. 2165–2183. Proceedings of Machine Learning Research, PMLR (2023)

Appendices

A LIBERO-Plus Per-Perturbation Breakdown for Ablations

In Table 5, we provide the detailed success rate breakdown of perturbations for models trained with different supervision signals. The clearest trend is that standard video flow matching degrades robustness under visually disruptive perturbations, especially camera viewpoint and sensor noise. This is consistent with our hypothesis that pixel-oriented supervision encourages the backbone to model low-level appearance details that are brittle under visual shift. At the same time, video flow matching performs better on robot and language perturbations than DINO supervision. We do not have strong evidence for the exact cause of this effect, so we avoid over-interpreting it, but it suggests that different perturbation types stress different aspects of the learned representation. The action-only variant further supports this interpretation: it remains relatively robust on camera and noise perturbations, yet underperforms more clearly on language and in overall success rate. Taken together, these results suggest that DINO supervision provides the best overall balance, improving robustness to nuisance visual changes while preserving strong task performance.

In Table 6, we show success rate breakdown for models trained with different read-out layers. The per-perturbation breakdown shows that the advantage of intermediate read-out is consistent across multiple perturbation types. The middle layer ($K = 15$) gives the best overall result and is particularly strong under camera, light, and noise perturbations. In contrast, final-layer read-out ($K = 30$) degrades robustness substantially under camera perturbation, while very early read-out ($K = 8$) performs worse across most categories. This is broadly consistent with our hypothesis that late layers are increasingly specialized to pixel-level generation, whereas early layers are not yet sufficiently informative for control. We note, however, that some categories such as robot and background do not follow this trend perfectly, so we interpret the intermediate-layer advantage as an overall empirical pattern rather than a universal rule.

Representation	Camera	Robot	Language	Light	Background	Noise	Layout	Total
DINOv3	83.1	56.3	76.3	94.5	66.7	91.0	80.4	78.3
Video FM	50.9	74.5	90.6	95.9	58.9	72.0	79.0	74.3
action-only	85.7	58.0	64.0	91.5	71.4	92.6	78.5	77.2

Table 5: Per-perturbation LIBERO-Plus breakdown for different supervision representations.

Layer	Camera	Robot	Language	Light	Background	Noise	Layout	Total
30	59.1	66.3	78.7	90.6	78.8	79.9	77.0	75.0
20	83.1	56.3	76.3	94.5	66.7	91.0	80.4	78.3
15	90.4	54.9	79.2	97.4	65.4	94.8	78.4	80.0
8	82.2	51.9	70.0	90.6	40.6	81.5	73.4	70.3

Table 6: Per-perturbation LIBERO-Plus breakdown for the read-out layer ablation.

B Further Discussion

B.1 Limitations

Our study has several limitations. First, all experiments are conducted in simulation, primarily on LIBERO and LIBERO-Plus, so the conclusions have not yet been validated in real-robot settings. Although LIBERO-Plus provides a stronger robustness test than standard LIBERO, it still captures only a limited subset of the variability encountered in real-world deployment. Second, due to compute constraints, we have not yet scaled WanPolicy to larger and more diverse robot datasets such as DROID or BridgeData. It therefore remains an open question whether the same representation-alignment formulation will continue to improve under broader multi-domain training. Third, our experiments only explore a limited set of supervision targets for the video branch. While DINO supervision yields strong robustness and overall performance, it is unlikely to be the only or optimal representation space for control. Finally, although our results suggest that intermediate DiT features

are more action-relevant than the final generation-oriented layers, this interpretation is still primarily empirical and would benefit from deeper analysis.

B.2 Social Impact

This work studies foundation models for robot manipulation, which could have positive impact by improving the robustness and efficiency of embodied agents in applications such as household assistance, warehouse automation, and industrial manipulation. More reliable and efficient robot policies may reduce deployment cost and make robotic systems more practical in environments that require adaptation to changing visual conditions. At the same time, advances in robot autonomy may also introduce risks. More capable manipulation systems could be deployed in safety-critical settings without sufficient oversight, or be adapted for harmful or unintended uses. In addition, models pretrained on large-scale internet video may inherit biases from their pretraining data, which could affect behavior in downstream robotic settings. Our work does not directly address these issues, but we believe they are important considerations for future real-world deployment. We therefore encourage careful evaluation, human oversight, and appropriate safety constraints when transferring such systems beyond controlled research benchmarks.

C Visual Results for WanPolicy LIBERO-Plus Rollout

We provide additional qualitative results for WanPolicy to complement Figure 3 in the main paper.

Figure 5 shows successful rollouts across nine tasks spanning both LIBERO-Goal and LIBERO-Spatial. The first five rows cover perturbations, including camera viewpoint change, sensor noise, lighting change, and language variation along with tasks with diverse spatial references. WanPolicy reliably completes all nine tasks.

Figure 6 shows failure cases spanning two perturbation categories. Rows 1–3 show LIBERO-Plus Language perturbations: object descriptions deviate substantially from training vocabulary, e.g., a wine bottle referred to as *"glass container of fermented grapes"* or the bowl as a *"darkhued rounded food container."* These cases account for the gap in the Language score (79.2%). Rows 4–5 show failures under non-language perturbations: an extreme object layout displacement causes WanPolicy to miss the bowl placed far from its nominal position and an unusual robot initial configuration prevents successful bowl retrieval from the wooden cabinet.

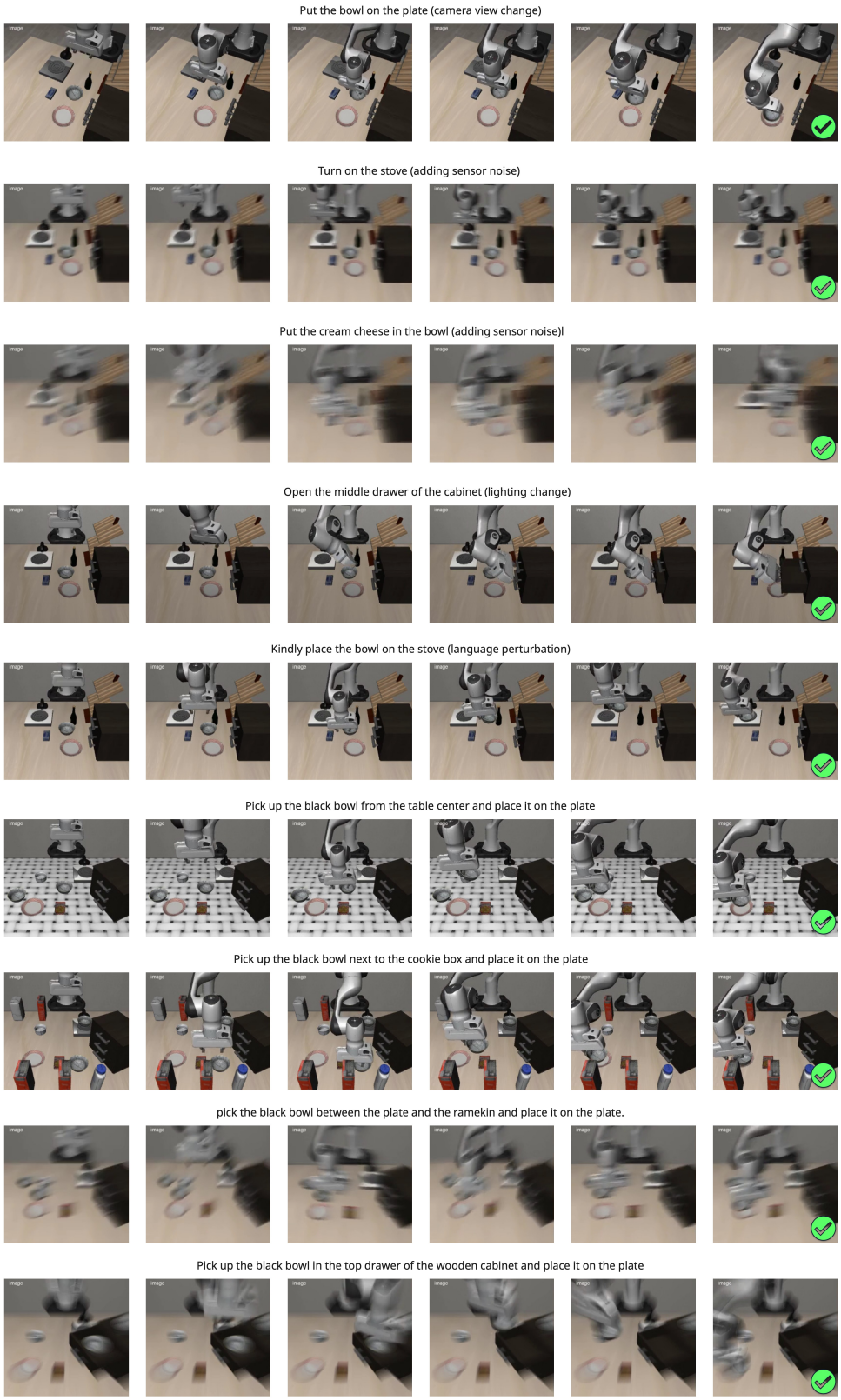


Figure 5: Successful rollouts of WanPolicy across nine perturbed tasks from LIBERO-Plus

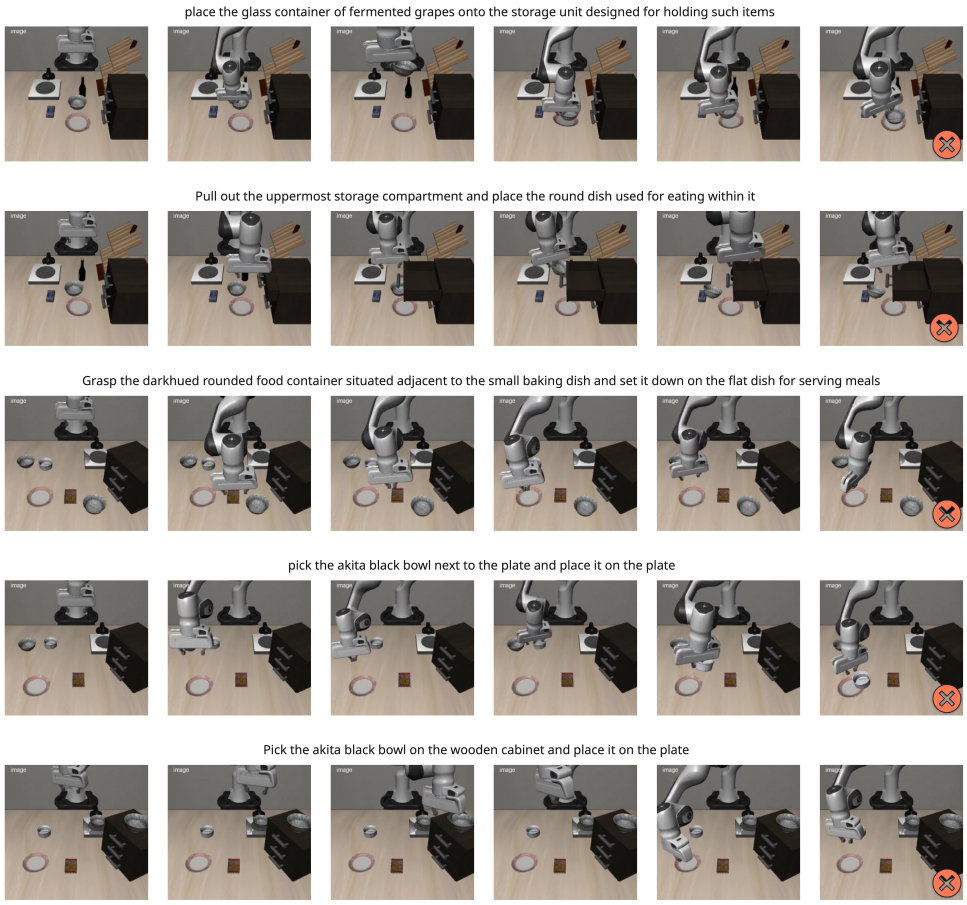


Figure 6: Failed rollouts of WanPolicy on LIBERO-Plus Language (rows 1–3), Objects Layout (row 4), and Robot Initial States (row 5) perturbations.