PREDICTING SEMANTIC LATENTS FOR ROBUST AND EFFICIENT WORLD-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World-Action Models (WAMs) adapt pretrained video generative models for robot control by jointly predicting future pixel observations and actions. We argue that robot manipulation requires reasoning about high-level scene structure ad object dynamics, but does *not* require pixel-level fidelity, which can be sensitive to nuisance factors like lighting. Rather than predicting future pixels and actions, we train our model to predict future semantic latents (e.g., frozen DINO, SigLIP, or V-JEPA encodings of future frames) and future actions. We refer to our approach as **S**emantic **L**atent **A**ction **P**rediction (**SLAP**). Importantly, we find that video diffusion backbones can be efficiently post-trained to directly regress (in a single forward pass) future semantic latents, eliminating the need for expensive iterative video denoising. Semantic targets improves robustness to environmental pertubations (like lighting or even end-effector initial position) and requires far less robot data for post-training. We ablate several design choices and find that intermediate layers of the pretrained video transformer provide better features for control compared to the final layers, motivating an intermediate-readout design that further improves efficiency and performance. On standard simulation benchmarks like LIBERO and RoboTwin, SLAP remains competitive with strong VLA and WAM baselines that train on $50\times$ more robot data. On the perturbation-heavy LIBERO-Plus benchmark, SLAP substantially outperforms pixel-level prediction. Finally, we show that SLAP can be deployed for real-world robot manipulation with as little as 30 demonstrations. SLAP is $1.9\times$ faster than Fast-WAM, $2.2\times$ faster than Flex-π Joint, and $19\times$ faster than LingBot-VA, while remaining comparable in speed to $\pi_{0.5}$.

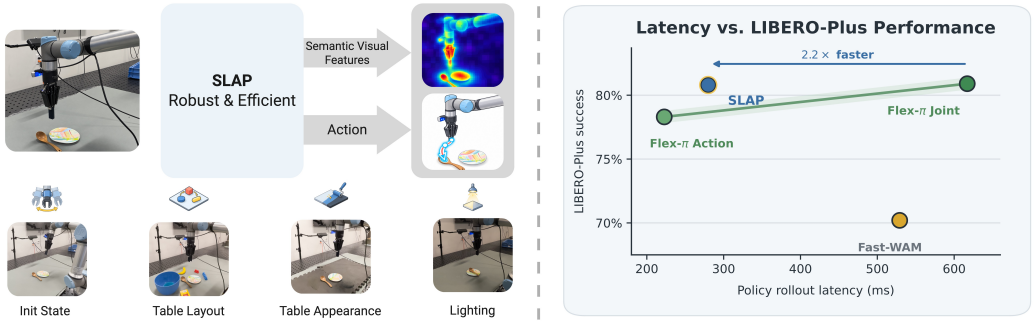


Figure 1: **Overview of SLAP.** (***left***) SLAP is a WAM that repurposes a pretrained video model to predict future semantic visual representations (e.g., frozen DINO embeddings) rather than future pixels. This design enables real-world manipulation that is robust to variations in the robot's initial state, table layout, table appearance, and lighting. (***right***) SLAP achieves a favorable performance–efficiency trade-off, which we show in LIBERO-Plus, SLAP matches the success rate of Flex-π(joint) while reducing latency by $2.2\times$.

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.

1 INTRODUCTION

Pretrained video diffusion models Wang et al. (2025a); Blattmann et al. (2023); Yang et al. (2024) have recently emerged as a compelling foundation for robot learning, as large-scale video pretraining captures rich priors over temporal dynamics, object interactions, and scene geometry. Building on these priors, a growing body of work repurposes video and image generative models for control, either by extracting predictive visual representations for downstream policies Hu et al. (2025); Pai et al. (2025); Du et al. (2023); Black et al. (2023), or by jointly modeling future observations and robot actions within World-Action Models (WAMs) Li et al. (2026); Ye et al. (2026); Yuan et al. (2026); Bi et al. (2025); Cheang et al. (2024). Together, these approaches suggest that generative visual pretraining provides a promising foundation for embodied control, complementary to conventional Vision-Language-Action (VLA) models Black et al. (2024); Intelligence et al. (2025); Kim et al.; 2025); Brohan et al. (2023); Zitkovich et al. (2023); Li et al. (2024).

Despite their strong empirical performance, most existing WAMs retain the pixel-level video-generation objective of their pretrained backbone. In this formulation, future visual observations are modeled through diffusion or flow matching in pixel or latent-video space, implicitly assuming that accurate visual reconstruction provides a useful learning signal for downstream action generation. We identify two limitations to this design. First, pixel-accurate future prediction may be unnecessarily detailed for control. Manipulation depends primarily on properties such as object identity, scene geometry, object configuration, and interaction dynamics, while low-level appearance details such as texture, lighting, and sensor characteristics are often irrelevant to the desired action and can vary substantially across environments. Optimizing for these may therefore encourage representations that fit the training distribution without generalizing. Second, pixel-level prediction is expensive: diffusion-based WAMs typically require multiple evaluations of a large video backbone to iteratively denoise the visual future, introducing substantial inference latency for closed-loop control.

In this work, we ask a simple question: **should World-Action Models predict future pixels or a higher level of abstraction?** We introduce SLAP, which replaces future prediction of pixels with future prediction of semantic latents. Specifically, we post-train the video generation backbone to directly regress future features produced by frozen visual encoders such as DINO, SigLIP, and V-JEPA Siméoni et al. (2025); Tschannen et al. (2025); Assran et al. (2025), which eliminates the need for iterative video denoising. Semantic features are also less sensitive to environmental nuisance factors like lighting or background clutter. We further find that not all layers of the pretrained video transformer are necessary for effective control. The early half layers provide stronger action-relevant features than the generation-specialized final layers, motivating an intermediate-readout design that both improves policy performance and further reduces inference cost. While we are inspired by approaches (e.g., JEPA) argue for prediction of latents instead of pixels, our position is nuanced; we find pixels *are* the right pretraining objective with free and scalable supervision, but maybe the wrong post-training objective for robot control. Our approach is inspired by the LLM cookbook; pretrain on next-token prediction, but post-train on high-quality targets of interest (e.g., code generation).

Empirically, SLAP remains competitive with strong VLA and WAM baselines on simulation benchmarks like LIBERO and RoboTwin Liu et al. (2023); Chen et al. (2025). On the more challenging, perturbation-heavy LIBERO-Plus benchmark Fei et al. (2025), SLAP further achieves substantially stronger robustness under environmental distribution shifts. Finally, we show that this strong performance and robustness transfer to real world robot manipulations. We conduct a series of real world environmental perturbations (gripper initial position, table layout, table appearance, lighting), and SLAP demonstrates better performance comparing to the strong VLA baseline $\pi_{0.5}$.

Notably, SLAP achieves these results using no large-scale robot-data pretraining. This is particularly important compared with representative VLA models such as OpenVLA Kim et al. and π_0 Black et al. (2024), which rely on substantial robot-data pretraining, using approximately 970k robot episodes and 10k hours of robot data, respectively.

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 explic-

itly modeling future scene evolution. Representative approaches include OpenVLA Kim et al., OpenVLA-OFT Kim et al. (2025), π_0 Black et al. (2024), $\pi_{0.5}$ Intelligence et al. (2025), RT-1 Brohan et al. (2023), RT-2 Zitkovich et al. (2023), and RoboFlamingo Li et al. (2024). 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 Black et al. (2023) 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 Du et al. (2023); Li et al. (2025) 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 Hu et al. (2025); Pai et al. (2025) 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 Li et al. (2026) learns frame prediction and policy execution simultaneously in an autoregressive diffusion framework; DreamZero Ye et al. (2026) jointly models future world states and actions with a pretrained video diffusion backbone; Fast-WAM Yuan et al. (2026) studies whether the benefits of WAMs come from joint video-action training or explicit future imagination at test time; and Motus Bi et al. (2025) 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 pixel-level generation objective during training, we directly train the video backbone to regress action-relevant visual semantic latents without iterative denoising at inference time.

3 METHOD

3.1 MODEL DESIGN

Architecture. We build upon WAN2.2 TI2V 5B Wang et al. (2025a), a video diffusion transformer pretrained on web-scale video data. We adopt a similar architecture design as Fast-WAM and LingBot-VA Li et al. (2026); Yuan et al. (2026) that augments the backbone with action tokens. As shown in Fig. 2, we modify the original self-attention layer of WAN to create Vision-Action Joint Attention: under this structured attention mask, video tokens attend only to video tokens, while action tokens attend to both video and action tokens. The video stream therefore provides the visual context for action generation without receiving information from the action stream.

After video-conditioned attention, separate cross-attention and FFN layers are used for the two branches. The video branch is conditioned on T5 instruction embeddings Raffel et al. (2020), while the action branch is conditioned on the same instruction embeddings together with encoded current 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, jointly denoising future video pixels and actions. We retain the standard flow-matching objective for action prediction, but replace the video-generation objective with direct regression of semantic latents.

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

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

Standard WAMs then train the video branch to predict a flow matching target taking $\boldsymbol{v}^t$ as input.

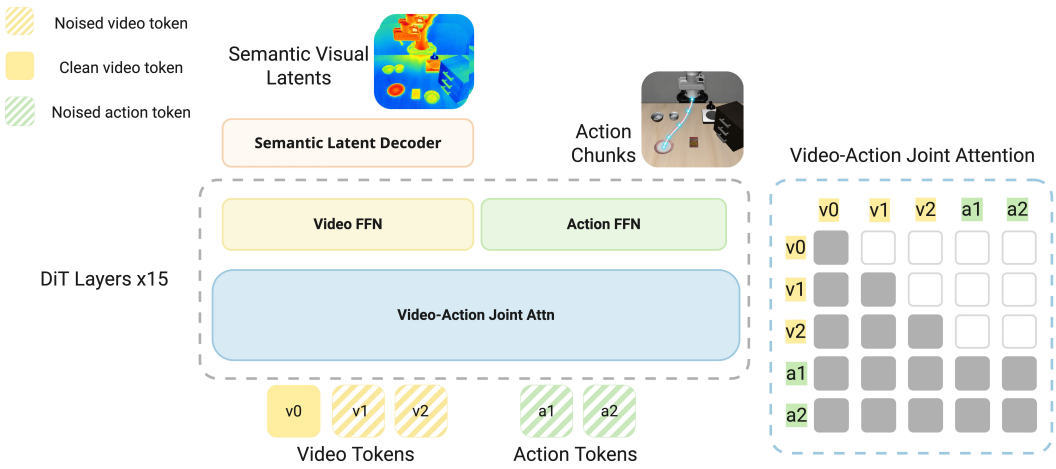


Figure 2: SLAP consists of a mixture of transformers (***left***) that conditions on the current visual frame token (v0) and denoises future video tokens (v1,v2) and future action tokens (a1,a2). Each transformer block is composed of joint attention (blue) followed by modality-specific attention and FFN layers for video (yellow) and action (green). Video tokens are supervised to predict semantic latents (e.g., frozen DINO) using a from-scratch trained decoder. Action tokens are trained with the standard flow-matching objective. Both predictions are read out from an intermediate DiT layer (layer 15). (***right***) To enable efficient inference, we make use of a sparse attention mask where visual tokens do not attend to actions tokens, but action tokens attend to both action and visual tokens. By training with full noise ($t = 1$) on video tokens but standard noise on action tokens $t \sim \mathcal{U}(0, 1)$, inference can be made efficient, requiring only a single iteration of video denoising (Fig. 3).

In SLAP, we always set $t = 1$ for the video branch, such that the future video input is pure Gaussian noise, i.e.,

$$\boldsymbol{v}^t = \boldsymbol{\epsilon}. \tag{2}$$

Instead of reconstructing the target future video, the model predicts future visual semantic latents.

Let $\boldsymbol{d}$ denote the target latents extracted from the future frames using a frozen vision encoder. We use DINOv3 Siméoni et al. (2025) as our default representation target and additionally study SigLIP Tschannen et al. (2025) and V-JEPA Assran et al. (2025). Given the initial observation o and language instruction l, we optimize the video branch with

$$\mathcal{L}_{\text{REP}} = 1 - \text{Avg}\big(\text{sim}\big(f_\theta(o, \boldsymbol{v}^t, t, l), \boldsymbol{d}\big)\big)\,, \tag{3}$$

where f_θ denotes the video branch and $\text{sim}(\cdot, \cdot)$ denotes cosine similarity. We compute the similarity at each spatiotemporal token and average across all frames and tokens.

For the action branch, we retain the standard flow-matching formulation. Let $\boldsymbol{a}$ denote a target action chunk. We construct the noisy action input as

$$\boldsymbol{a}^t = (1 - t)\boldsymbol{a} + t\boldsymbol{\epsilon}, \tag{4}$$

where $t \sim \mathcal{U}(0, 1)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. The action branch is optimized with

$$\mathcal{L}_{\text{Action}} = \mathbb{E}_{\boldsymbol{a},\boldsymbol{\epsilon},t}\left[\left\|g_\phi(\boldsymbol{a}^t, t, l, p) - (\boldsymbol{\epsilon} - \boldsymbol{a})\right\|_2^2\right], \tag{5}$$

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

The final training objective is

$$\mathcal{L} = \mathcal{L}_{\text{REP}} + \mathcal{L}_{\text{Action}}. \tag{6}$$

Inference Design. Our modification changes the inference procedure only for the video branch. In standard WAMs, future visual prediction requires multiple denoising steps through the video backbone before or along with action generation. Since the video branch of SLAP is trained to predict the target representation directly from $t = 1$, the visual future can instead be obtained in a single forward pass.

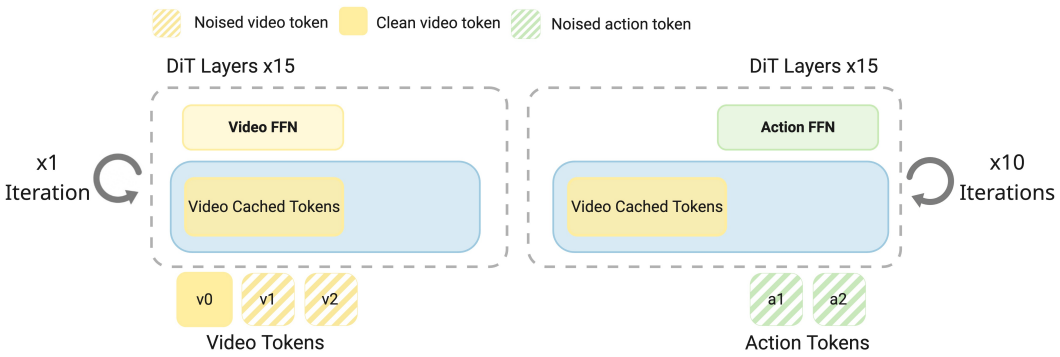


Figure 3: **SLAP is efficient for inference** because video tokens serve as a cacheable context for action prediction. Inference begins with a *single* (1x) forward pass of the video branch (***left***). We need not run the semantic latent decoder since it is not needed to predict actions. Rather, we KV-cache the 15 layers of transformed video tokens. These tokens serve as a conditioning context for an action diffusion policy (***right***), which makes use of many (10x) denoising iterations. Importantly, given V video tokens and A action tokens, joint attention becomes $O(A^2 + AV)$ instead of $O((A + V)^2)$ because one does not need to transform video tokens. This means that inference is effectively the same cost as self-attention within action tokens and cross-attention to video tokens.

As illustrated in Fig. 3, we first run the video branch once. Since video tokens do not attend to action tokens, their key-value states remain fixed throughout subsequent action generation and can therefore be cached. We then perform iterative denoising only in the action branch, reusing the cached visual states at every denoising step. This avoids repeatedly evaluating the substantially larger video backbone while retaining flow matching for action generation.

3.2 Intermediate Read-out Layer Design

Another important design choice in SLAP is the layer from which the video and action predictions are read out. Rather than using the final layer of the video diffusion transformer, we read out both branches from an intermediate layer K, as illustrated in Fig. 2. Layers after K are discarded during both training and inference, and both the visual representation prediction and action prediction are computed directly from the features at layer K.

This design is motivated by the observation from Huang et al. (2025) that the deepest layers of video diffusion transformers become increasingly specialized toward the native video-generation objective. The later layers may therefore emphasize fine-grained reconstruction details that are useful for pixel generation but less relevant for control. Intermediate layers, in contrast, can retain high-level scene structure and dynamics without fully specializing toward the final reconstruction objective.

The intermediate read-out also directly reduces computation by shortening the backbone used during both visual prediction and action generation. We study the effect of K in Sec. 4.4 and find that a middle-layer read-out achieves the best performance while further reducing inference cost.

4 Experiments

4.1 Implementation Details

We use the pretrained WAN2.2 TI2V 5B Wang et al. (2025a) as the 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 and output projections are randomly initialized. We use read-out layer $K = 15$ unless otherwise specified.

For data processing, we follow the Fast-WAM convention Yuan et al. (2026) and temporally downsample input videos by a factor of 4, resulting in 9 frames per chunk. The action horizon is 32.

During training, we predict the full 32-step action chunk, and at inference time execute the predicted actions over the entire horizon.

We use all camera views provided by each dataset. In LIBERO and LIBERO-Plus, the observation consists of a third-person agent view and a gripper-mounted view. RoboTwin additionally provides two gripper-mounted cameras, one for each robot hand, together with the agent view. Our real-world setup follows the LIBERO configuration, using a third-person agent view and a single gripper-mounted view. For all settings, the available views are concatenated into a single frame before being processed by the visual backbone.

All experiments are conducted using either 8× NVIDIA H100 GPUs or 8× 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.

4.2 Experimental Setup

LIBERO and LIBERO-Plus. We evaluate on LIBERO Liu et al. (2023), a standard benchmark for language-conditioned robot manipulation, and LIBERO-Plus Fei et al. (2025), which evaluates robustness under controlled environmental perturbations. LIBERO contains four task suites—Spatial, Object, Goal, and Long—covering different forms of manipulation and compositional difficulty. For LIBERO, we evaluate 50 rollouts per task, resulting in 2,000 rollouts across the four suites, and report task success rate.

LIBERO-Plus perturbs the original LIBERO environments along seven dimensions: camera viewpoint, robot appearance, language, lighting, background, sensor noise, and object layout. We follow the official LIBERO-Plus evaluation protocol and report success rate over approximately 10K perturbed rollouts. Importantly, the same policy trained on LIBERO is evaluated directly on LIBERO-Plus without further adaptation, making the benchmark a direct measure of robustness to environmental distribution shift.

RoboTwin. We additionally evaluate on RoboTwin 2.0 Chen et al. (2025), a large-scale benchmark for bimanual manipulation with strong domain randomization. We follow the multi-task training and evaluation setup of Fast-WAM Yuan et al. (2026). Specifically, we train on a mixture of 2,500 demonstrations collected in clean scenes and 25,000 demonstrations collected under heavy scene randomization, spanning more than 50 bimanual manipulation tasks. For evaluation, we report the average success rate over 100 rollouts per task under both the *Clean* and *Randomized* settings. Due to space limit, we put the RoboTwin evaluation results in Appendix. B.

Real-Robot Setup. We further evaluate SLAP on a real-world manipulation platform consisting of a Universal Robots UR5e arm equipped with a Robotiq 2F-140 parallel gripper. Visual observations are provided by two cameras: an Intel RealSense camera mounted on the robot wrist and an external ZED camera providing a third-person view of the full tabletop workspace.

We construct four manipulation tasks spanning different levels of control difficulty. For rigid-object pick-and-place, the robot is instructed to pick up either a fork or a spoon and place it onto a plate, resulting in two separate tasks. To evaluate more precise manipulation, we include a cup-placement task in which the robot must grasp a cup near its rim and place it upright on a smooth plate surface without tipping over. Finally, we consider a longer-horizon deformable-object task in which the robot must pick up a cable, place it inside an open drawer, and subsequently close the drawer. Some task demonstrations are shown in Fig. 5

For each task, we collect 30 demonstrations for training. At evaluation time, we test the learned policy under both the original training configuration (**ID**) and four out-of-distribution perturbations (**OOD**) designed to probe robustness of the model. Specifically, we consider: (1) a large shift in the initial robot gripper position, (2) reduced ambient lighting, (3) modified tabletop layout with previously unseen distractor objects, and (4) a substantial change in tabletop appearance by placing a black mat over the workspace and positioning all task objects on top of it. For each task and each evaluation condition, we perform 5 independent rollouts and report task success rate.

Baselines. We compare SLAP against representative VLA and WAM approaches. For VLA baselines, we include strong pretrained policies such as OpenVLA-OFT Kim et al. (2025), $\pi_{0.5}$ Intelligence et al. (2025), and VLA-JEPA Sun et al. (2026). For WAM baselines, we primar-

ily compare against recent models, including Fast-WAM Yuan et al. (2026), LingBot-VA Li et al. (2026), and Flex-π Yan et al. (2026). Additional results from prior methods are included in the corresponding benchmark tables for completeness. we also indicate whether each method uses large-scale robot-data pretraining before downstream adaptation.

4.3 Main Results

LIBERO. As shown in Tab. 1, SLAP achieves competitive performance on LIBERO despite using no robot data pretraining. Whereas OpenVLA and π_0 are pretrained on approximately 970K robot demonstrations and over 10K hours of robot data, respectively O'Neill et al. (2024); Kim et al.; Black et al. (2024), SLAP is trained directly on LIBERO for 20K iterations. This result demonstrates that large-scale video pretraining provides strong transferable priors for downstream manipulation.

LIBERO-Plus. On LIBERO-Plus, SLAP achieves 80.8% overall success, outperforming Fast-WAM by 10.6 percent and achieving comparable performance with Flex-π(Joint) despite using the same family of video-pretrained backbone without large scale robot pretraining. It performs best under Camera (90.2), Light (97.1), and Noise (95.3) perturbations while remaining competitive across the other categories. These results support our hypothesis that semantic future representation prediction produces policies that are less sensitive to environmental variation than pixel-level prediction.

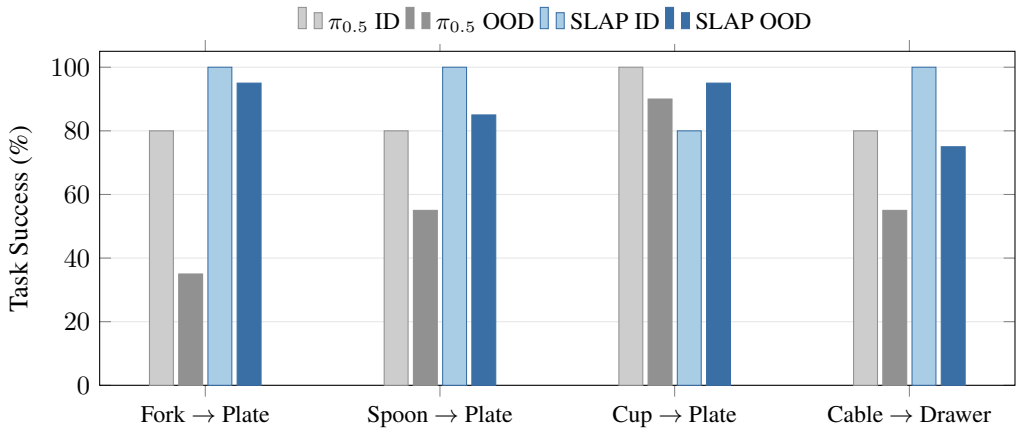


Figure 4: Real-robot evaluation under original training configuration (ID) and out-of-distribution (OOD) settings. OOD aggregates the four perturbation conditions. While both SLAP and $\pi_{0.5}$ achieve good performance under ID setting, SLAP is more robust to environmental perturbations under different OOD evaluation conditions.

Real Robot Results. We next examine whether the robustness observed in simulation transfers to real-world manipulation. Across four tasks, SLAP achieves 95.0% success under in-distribution conditions and 87.5% under OOD perturbations, compared with 85.0% and 58.8% for the robot-pretrained $\pi_{0.5}$ baseline Intelligence et al. (2025) (Fig. 4). This corresponds to a substantially smaller OOD degradation of 7.5 versus 26.2 percentage points. Qualitatively, SLAP recovers more effectively from execution failures, frequently retrying after dropped objects, whereas $\pi_{0.5}$ often stalls or fails to resume. We identify this retry behavior as one of the major reasons causing the difference in real robot experiments, especially when environmental perturbations are introduced. These results demonstrate that high-level future representation prediction enables robust real-world control without large-scale robot-data pretraining.

4.4 Ablation Study

Visual Supervision Space. We first study how the supervision target of the visual branch affects policy performance. All variants use the same architecture and action supervision, differing only in the visual training target. We compare high-level representations from DINOv3, SigLIP, and V-JEPA against conventional pixel-level video flow matching and an action-only variant with no explicit visual objective.

Method	Robot PT	Orig.	Plus	Cam.	Robot	Lang.	Light	BG	Noise	Layout
VLAs										
OpenVLA Kim et al.	Yes	76.5	15.6(↓60.9)	0.8	3.5	23.0	8.1	34.8	15.2	28.5
OpenVLA-OFT Kim et al. (2025)	Yes	97.1	69.6(↓27.5)	56.4	31.9	79.5	88.7	93.3	75.8	74.2
NORA Hung et al. (2025)	Yes	87.9	39.0(↓48.9)	2.2	37.0	65.1	45.7	58.6	12.8	62.1
UniVLA Wang et al. (2025b)	Yes	95.2	42.9(↓52.3)	1.8	46.2	69.6	69.0	81.0	21.2	31.9
π_0 Black et al. (2024)	Yes	94.1	53.6(↓40.5)	13.8	6.0	58.8	85.0	81.4	79.0	68.9
π_0-Fast Pertsch et al. (2025)	Yes	85.5	61.6(↓23.9)	65.1	21.6	61.0	73.2	73.2	74.4	68.8
RIPT-VLA Tan et al. (2025)	Yes	97.5	68.4(↓29.1)	55.2	31.2	77.6	88.4	91.6	73.5	74.2
VLA-JEPA Sun et al. (2026)	Yes	97.2	79.5(↓17.7)	63.3	67.1	**85.4**	95.6	93.6	66.3	**85.1**
$\pi_{0.5}$ Intelligence et al. (2025)	Yes	96.9	**84.7**(↓12.2)	72.2	**76.8**	**85.1**	96.2	**95.5**	87.0	**86.9**
ELAN4D He et al. (2026)	Yes	97.0	78.2(↓18.8)	63.7	70.7	77.8	89.8	91.4	79.9	81.4
WAMs										
WorldVLA Cen et al. (2025)	No	79.1	25.0(↓54.1)	0.1	27.9	41.6	43.7	17.1	10.9	38.0
Fast-WAM Yuan et al. (2026)	No	97.6	70.2(↓27.4)	48.1	**72.5**	70.0	**96.7**	66.0	69.7	78.1
Flex-π(full joint) Yan et al. (2026)	Yes	**98.5**	**80.9**(↓17.6)	**73.7**	53.1	84.3	91.9	**93.8**	**93.4**	82.9
Flex-π(action-only) Yan et al. (2026)	Yes	**98.4**	78.3(↓20.1)	**71.9**	50.7	83.3	91.9	91.9	91.5	80.4
SLAP(WAN2.2)	No	97.3	80.8(↓16.5)	**90.2**	55.5	81.2	**97.1**	65.4	**95.3**	78.4

Table 1: **Evaluation results on LIBERO and LIBERO-Plus.** Success rates are reported in %. **Orig.** denotes performance on the original LIBERO benchmark, while **Plus** denotes the overall LIBERO-Plus success rate. The detailed LIBERO-Plus categories are listed as the right most 7 columns. Absolute performance drops from LIBERO to LIBERO-Plus are shown in parentheses. **Robot PT** indicates whether the method uses large-scale robot-data pretraining prior to LIBERO fine-tuning. Top-three results in each column are highlighted: best, second best, and third best. On the challenging LIBERO-Plus, SLAP matches prior art (Flex-π) while being trained on $50\times$ less robot data and being $2.2\times$ faster.

(a) Visual Supervision Space

Representation	LIBERO	LIBERO-Plus
DINO	97.3	80.8
SigLIP	96.7	79.9
V-JEPA	96.9	81.8
Video FM	97.6	74.3
Action Only	94.1	77.2

(b) Intermediate Read-out Layer

Layer	LIBERO	LIBERO-Plus
30	96.1	75.8
20	96.0	78.3
15	97.3	80.8
8	91.4	70.7

Table 2: **Ablations** of visual supervision and intermediate read-out layer. (***left***) Semantic latents improve robustness on LIBERO-Plus compared with pixel-level prediction. (***right***) Reading out an intermediate DiT layer (15) outperforms the final layer while being $2\times$ faster.

The left side of Tab. 2 reveals a clear difference between in-distribution performance and robustness. Pixel supervision achieves the strongest LIBERO score (97.6), but its performance drops substantially on LIBERO-Plus to 74.3. In contrast, DINO, SigLIP, and V-JEPA achieve 80.8, 79.9, and 81.8 on LIBERO-Plus, respectively, while retaining similar performance on standard LIBERO.

Interestingly, the action-only model reaches 77.2 on LIBERO-Plus, outperforming the pixel-supervised variant despite having no explicit visual prediction objective. This suggests that pixel prediction can itself introduce sensitivity to low-level visual details. At the same time, action-only training reduces standard LIBERO performance to 94.1, indicating that predictive visual supervision remains beneficial when the supervision space is appropriately chosen.

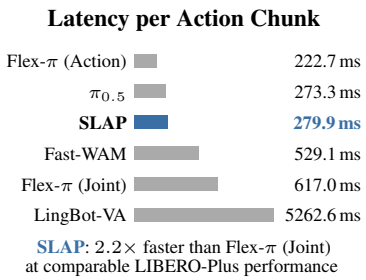
Most importantly, the robustness improvement is consistent across DINO, SigLIP, and V-JEPA. Despite their different pretraining objectives, all three outperform pixel prediction under environmental perturbations, suggesting that the gain arises from moving future prediction away from pixel-level reconstruction rather than from a property unique to a particular visual encoder.

Intermediate Read-out Layer. We next study the depth at which the visual and action predictions are read out from the pretrained video DiT. The right side of Tab. 2 shows that using an intermediate layer is important: $K = 15$ achieves the strongest performance on both LIBERO and LIBERO-Plus, outperforming the final layer ($K = 30$) as well as a much earlier read-out ($K = 8$).

The trend suggests a trade-off in backbone depth. Very early layers have not yet performed sufficient visual processing and video–action interaction, while later layers become increasingly specialized toward the original video-generation objective.

The intermediate layers provide a useful operating point where rich scene structure has emerged without requiring the full generation-specific computation of the pretrained backbone. This suggests that the later generation-specialized layers are not necessary for effective control and can be removed to improve both performance and inference efficiency.

Inference Speed. We quantify the efficiency benefit of SLAP. We measure the wall-clock latency of one policy rollout chunk, from receiving an observation to producing the predicted action chunk. All methods are evaluated under the same input setting and GPU synchronization protocol, and we report the average latency over 10 runs after warmup.



SLAP requires 279.9 ms per action chunk, comparable to $\pi_{0.5}$ at 273.3 ms and close to the 222.7 ms action-only Flex-π variant. Importantly, SLAP achieves 80.8% on LIBERO-Plus, essentially matching the 80.9% performance of Flex-π Joint, while being $2.2\times$ faster. Compared with other WAMs, SLAP is approximately $19\times$ faster than LingBot-VA, and $1.9\times$ faster than Fast-WAM.

This efficiency comes from two complementary design choices: the visual branch predicts the future representation in a single forward pass, while the intermediate read-out avoids executing the full video DiT. Together, these results show that SLAP retains the robustness benefit of predictive world modeling while operating at a latency much closer to action-only and VLA policies.

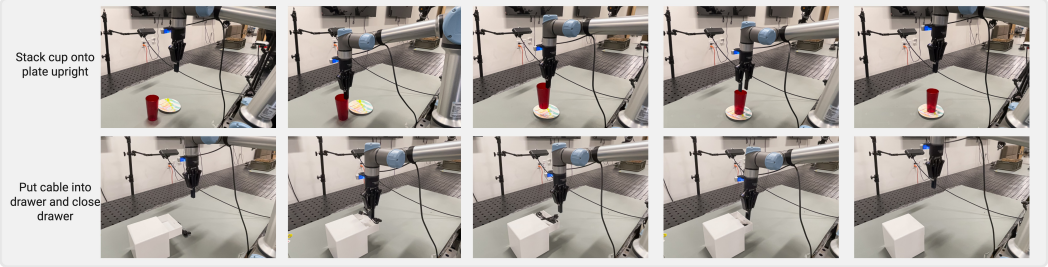


Figure 5: Illustrations of real robot tasks. Cup placement requires the robot to precisely grab the rim of the cup and place on a smooth plate surface without falling over. Cable Organization is a long horizon task that requires the robot to pick up deformable cables, place in the drawer, and properly close the drawer.

5 Conclusion

We present SLAP, a World-Action Model that replaces pixel-level future prediction with direct regression of semantic visual representations. This objective improves robustness to environmental perturbations while enabling single-pass visual prediction. Reading out from an intermediate video-transformer layer further improves control performance and reduces computation. Across LIBERO, LIBERO-Plus, and real-robot tasks, SLAP achieves strong performance without large-scale robot-data pretraining while operating at inference speeds close to standard VLA policies. Our results suggest that semantic predictive representations provide a more suitable post-training target than pixel reconstruction for robust and efficient robot control.

AI USE STATEMENT

In this work, we used generative AI tools for [design or provide feedback on research methodology or experiments; implement methods]. We have not used generative AI tools for [help develop theoretical models or conceptual frameworks; formulate mathematical claims; provide critical ingredients for proving mathematical claims; assist in the writing of proofs; propose or refine hypotheses; assist with translation; support qualitative and thematic data analysis; interpret results], and [Generate synthetic data sets; clean and reformat dataset] are not applicable to this work. Additionally, we used generative AI tools for [create or modify scientific figures or images; create or edit software code; draft parts of a research paper; summarize or analyse existing literature; edit a research paper to improve readability; propose a title or keywords for a research paper.] We have reviewed all AI-assisted work. We checked LLM-generated paper content so it closely follows what we propose in the paper and does not make factual mistakes. We verified and tested the code generated by LLM. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To make the results reproducible, we reveal the key design choices including backbone model selection, hyperparameter selection, and training details in Section 4.1. We will release our code after internal review process.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew J. Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *CoRR*, abs/2506.09985, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. *CoRR*, abs/2512.13030, 2025.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *CoRR*, abs/2310.10639, 2023.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *CoRR*, abs/2410.24164, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong T. Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: robotics transformer for real-world control at scale. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu (eds.), *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. Worldvla: Towards autoregressive action world model. *CoRR*, abs/2506.21539, 2025.

Chilam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *CoRR*, abs/2410.06158, 2024.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *CoRR*, abs/2506.18088, 2025.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (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.

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

Zeyuan He, Bowen Yang, Zhirui Fang, Keru Zhou, Lei Jiang, Jingjing Qian, Fan Mo, Junchi Yan, Philip Torr, Xiu Li, Li Jiang, and Jialin Yu. ELAN4D: embodiment-centric 4d supervision for vision-language-action models via plug-and-play adaptation. *CoRR*, abs/2605.30484, 2026.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (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.

Zixuan Huang, Xiang Li, Zhaoyang Lv, and James M. Rehg. How much 3d do video foundation models encode? *CoRR*, abs/2512.19949, 2025.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. NORA: A small open-sourced generalist vision language action model for embodied tasks. *CoRR*, abs/2504.19854, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *CoRR*, abs/2504.16054, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, Proceedings of Machine Learning Research.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *CoRR*, abs/2502.19645, 2025.

Hongyu Li, Lingfeng Sun, Yafei Hu, Duy Ta, Jennifer Barry, George Konidaris, and Jiahui Fu. Novaflow: Zero-shot manipulation via actionable flow from generated videos. *CoRR*, abs/2510.08568, 2025.

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

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. 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.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: benchmarking knowledge transfer for lifelong robot learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (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.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, 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.

Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond vlas. *CoRR*, abs/2512.15692, 2025.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: efficient action tokenization for vision-language-action models. *CoRR*, abs/2501.09747, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3. *CoRR*, abs/2508.10104, 2025.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: enhancing vision-language-action model with latent world model. *CoRR*, abs/2602.10098, 2026.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *CoRR*, abs/2505.17016, 2025.

Michael Tschannen, Alexey A. Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier J. Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *CoRR*, abs/2502.14786, 2025.

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Xiaofeng Meng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu

Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *CoRR*, abs/2503.20314, 2025a.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. *CoRR*, abs/2506.19850, 2025b.

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

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

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 "Jim" Fan, and Joel Jang. World action models are zero-shot policies. *CoRR*, abs/2602.15922, 2026.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *CoRR*, abs/2603.16666, 2026.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong T. Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, Proceedings of Machine Learning Research, pp. 2165–2183. PMLR, 2023.

APPENDIX

A LIBERO-PLUS PER-PERTURBATION BREAKDOWN FOR ABLATIONS

In Table 3, 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. Taken together, these results suggest that semantic latents supervision provides the best overall balance, improving robustness to nuisance visual changes while preserving strong task performance.

In Table 4, 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	90.2	55.5	81.2	97.1	65.4	95.3	78.4	80.8
SigLIP	87.3	57.0	81.5	97.2	66.3	90.5	79.5	79.9
V-JEPA	89.7	60.8	77.8	93.4	79.5	92.8	80.1	81.8
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 3: Per-perturbation LIBERO-Plus breakdown for different supervision representations.

Layer	Camera	Robot	Language	Light	Background	Noise	Layout	Total
30	59.3	66.6	79.8	93.3	79.8	80.5	77.0	75.8
20	83.1	56.3	76.3	94.5	66.7	91.0	80.4	78.3
15	90.2	55.5	81.2	97.1	65.4	95.3	78.4	80.8
8	82.4	52.3	68.2	90.6	40.7	82.2	73.4	70.7

Table 4: Per-perturbation LIBERO-Plus breakdown for the read-out layer ablation.

B ROBOTWIN RESULTS

Table 5 reports results on RoboTwin under both clean and perturbed evaluation settings. SLAP achieves 90.01% success in the clean setting and 90.04% under perturbations, resulting in an overall success rate of 90.0%. Notably, its performance remains essentially unchanged under environmental perturbations, indicating that the learned policy retains strong robustness across different perturbations.

SLAP substantially outperforms π_0 and $\pi_{0.5}$ while remaining competitive with recent WAMs such as Fast-WAM and LingBot-VA. We emphasize that these results are obtained under a more constrained training budget. Our RoboTwin experiments use only 8 A6000 GPUs, limiting the available batch size relative to larger-scale training setups such as Fast-WAM. Despite this reduced training

Model	Robot PT	Clean	Pert.	Overall
π_0 Black et al. (2024)	Yes	65.92	58.40	62.2
$\pi_{0.5}$ Intelligence et al. (2025)	Yes	82.74	76.76	79.8
LingBot-VA Li et al. (2026)	Yes	92.90	91.50	92.2
Fast-WAM Yuan et al. (2026)	No	91.88	91.78	91.8
SLAP (WAN2.2)	No	90.01	90.04	90.0

Table 5: Comparison on RoboTwin. **Robot PT** indicates whether the model uses large-scale robot-data pretraining prior to RoboTwin fine-tuning.

throughput, SLAP reaches within 1.8 points of Fast-WAM in overall success rate while requiring no large-scale robot data pretraining.

C Scaling Test for SLAP

We evaluate how SLAP scales with the amount of task-specific training data. Specifically, we train separate models using 10, 20, 30, and 50 demonstrations per task from LIBERO and evaluate each model on both LIBERO and LIBERO-Plus. As shown in Fig. 6, performance improves consistently as the number of demonstrations increases. Notably, with only 10 demonstrations per task, SLAP already achieves a success rate of 69.2%, performance comparable to strong VLA baselines, including OpenVLA-OFT and π_0-FAST, highlighting its data efficiency and robustness. Due to computational constraints, we restrict this analysis to LIBERO rather than conducting a comprehensive scaling study on larger robotic datasets. Nevertheless, these results provide preliminary evidence that SLAP continues to benefit from additional training data.

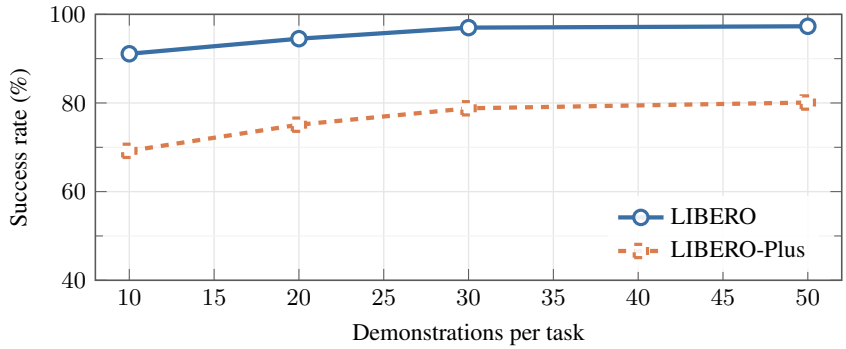


Figure 6: Scaling behavior of SLAP as the number of training demonstrations per task increases. We report the average success rate on LIBERO and LIBERO-Plus.

D Visual Results for SLAP on Real Robot Experiments

We provide extra visual results for SLAP evaluating on our real robot experiments as shown in Fig. 8. We demonstrate all four tasks we designed for experiments and all four environmental perturbations used to test model's robustness. We also show a comparison between $\pi_{0.5}$ and SLAP on one of the real robot tasks, as shown in Fig. 7.

This visual demonstrates SLAP successful retry behavior after a miss during rollouts. On the other hand, $\pi_{0.5}$ often fails to recover during retry. We attribute this as one of the important factors for SLAP's better performance in real robot experiments, especially when environmental perturbations are applied. We provide additional qualitative results for SLAP evaluating on LIBERO-Plus. Fig. 9 shows successful rollouts in the first five rolls and failure cases in the bottom four rolls. We include more rollouts from real robot experiments and simulations as GIFs in this anonymous website: `https://anonymous.4open.science/w/ICLR-SLAP-B237/`.

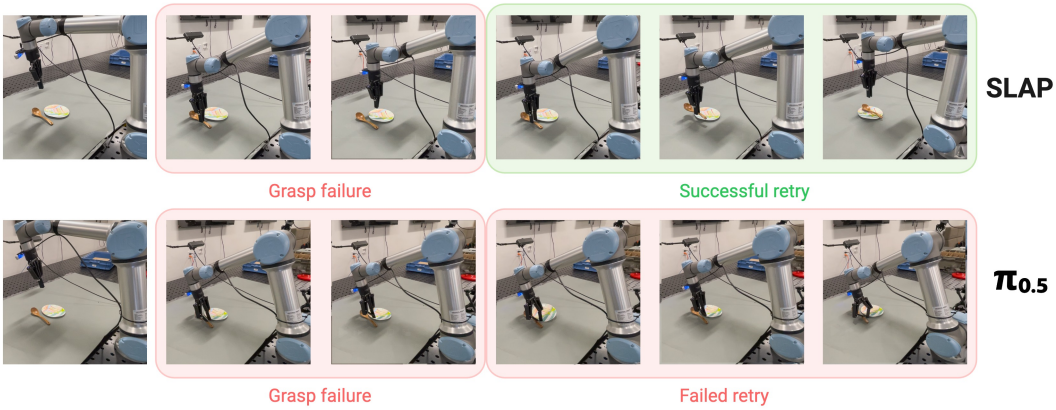


Figure 7: Visual comparison between $\pi 0.5$ and SLAP on the task "place the spoon on the plate" with the gripper starting at an unseen position during training. SLAP quickly recovers after a grasp failure of the spoon, while $\pi_{0.5}$ stuck during the retry.

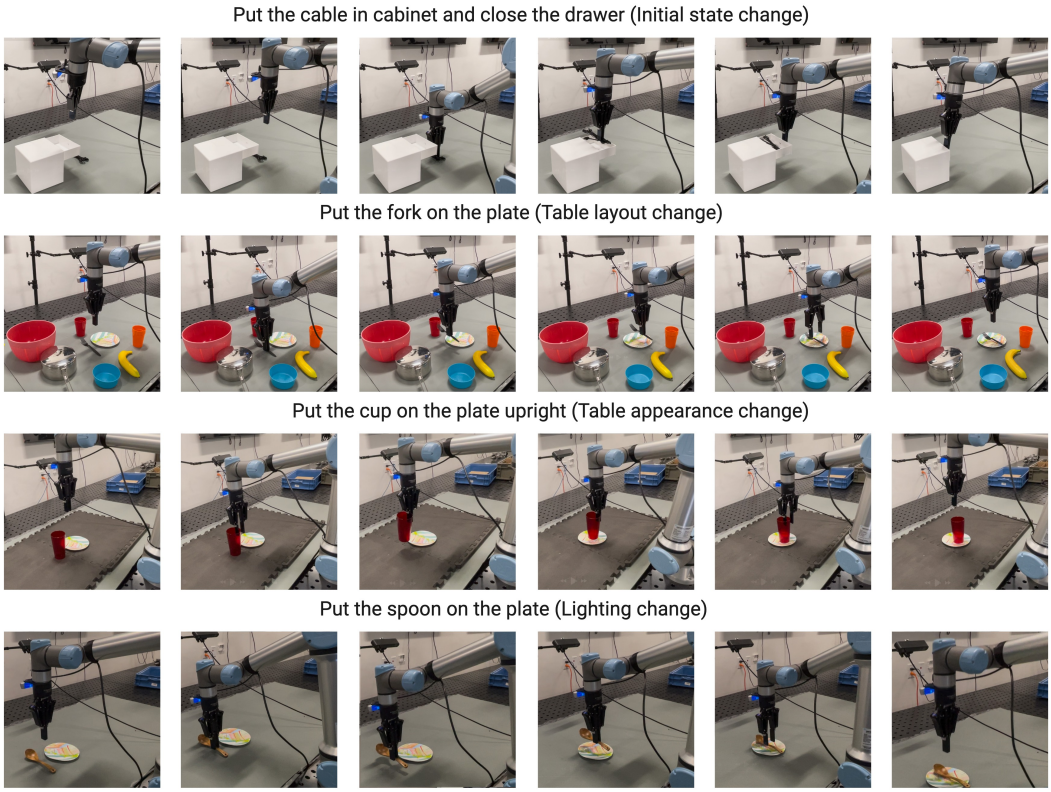


Figure 8: Visual results for SLAP on real robot experiments. We show all four tasks we designed and all four environmental perturbations we used for evaluation.

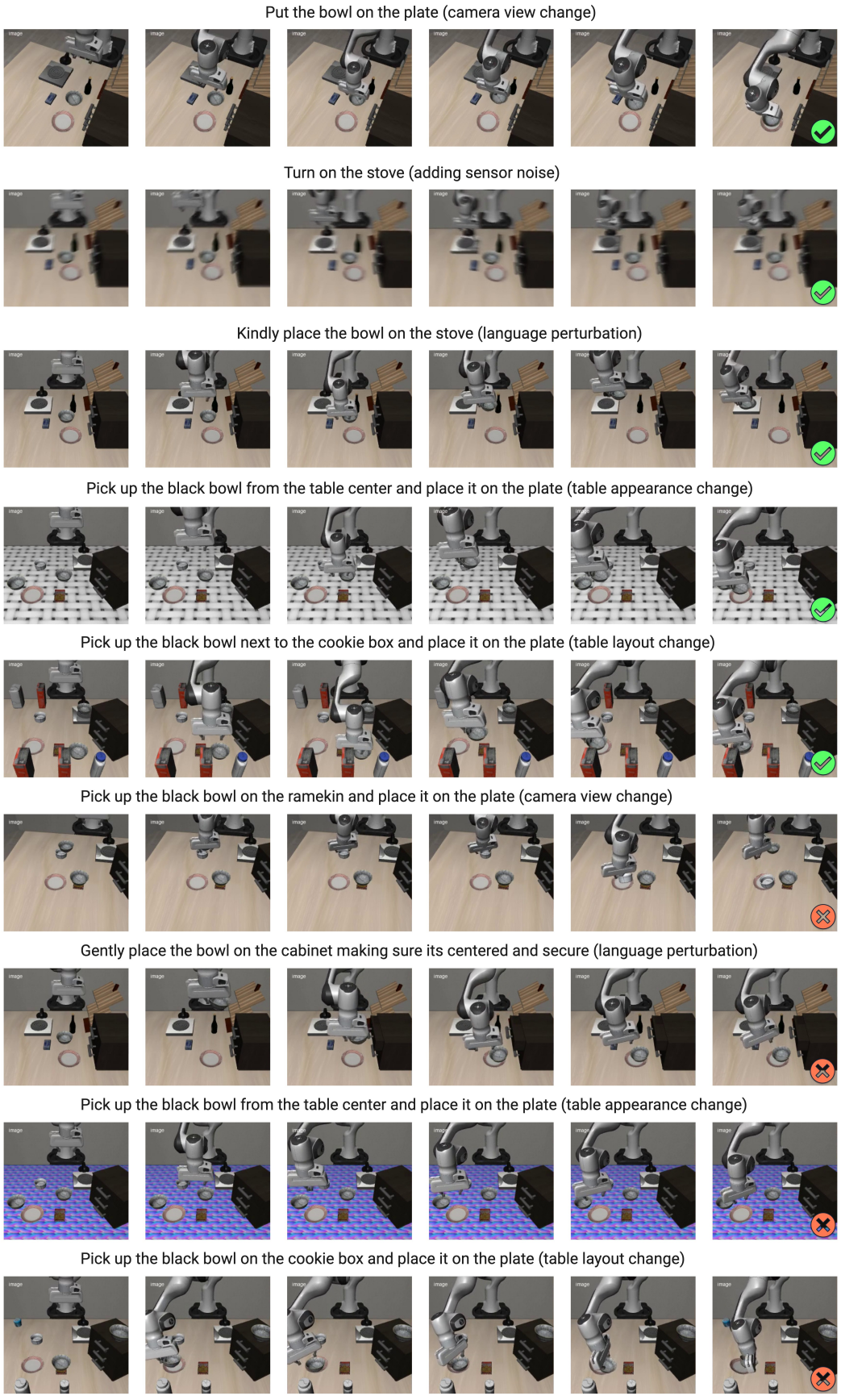


Figure 9: Visualizations of SLAP across nine tasks from LIBERO-Plus.