

FUTURE ANCHORED VERIFICATION AND ONLINE RECOVERY FOR WORLD ACTION MODELS

Zhibin Qin Zhenxiong Tan Xinchao Wang[†]
National University of Singapore
e1352224@u.nus.edu, xinchao@nus.edu.sg

ABSTRACT

World action models (WAMs) have emerged as a promising paradigm for robotic manipulation. They act by first predicting how a task should be performed and then decoding the actions from that future. However, the remaining actions are invalid once execution drifts from the prediction. Simply replanning from the already out of distribution state rarely restores what the task still requires; existing execution monitors decide when to stop, but not what to restore. We observe that the answer is already in hand: the future the WAM predicted before acting depicts exactly the states it intended to pass through. We introduce FAVOR (Future Anchored Verification and Online Recovery), a lightweight framework that keeps these predicted frames as anchors and uses them for verification and recovery. An Anchor Verifier compares each observation with its anchor, together with the executed actions, to flag deviations that break the task. Anchor-Guided Recovery uses a vision-language model to turn the flagged anchor into a short corrective instruction. Under strengthened instruction guidance, the WAM executes this instruction to return to the intended future. It then resumes the task. FAVOR raises the task success of the base WAM from 97.85% to 98.10% on LIBERO and from 72.60% to 72.98% on LIBERO-Plus without modifying the policy.

1 INTRODUCTION

World action models (WAMs) have emerged as a promising paradigm for robotic manipulation. They work in a forward and backward way: they first predict the visual future a task should pass through and then decode the actions that realize the future [12, 27, 32, 62, 69, 74, 75, 78, 82, 86, 87]. Unlike policies that map current observations and task instruction directly to actions [4, 8, 26], every action is thus grounded in an explicit future of where the scene should go.

That future is only a plan. Imprecise actions or external disturbances can push the scene off the predicted path, invalidating the remaining actions [33, 64]. Simply replanning from the deviated state, however, offers little help: WAMs are trained on successful demonstrations [29, 54, 55], so neither the future predicted from such an unfamiliar state nor the actions decoded from it can be trusted. Execution monitors for WAMs help by flagging when the remaining actions should no longer be trusted [1, 43, 64, 72, 79], but say nothing about what to restore. Recovery methods address this gap, but obtain their target from outside the policy [11, 50]. This raises a natural question: can a WAM tell when execution has left its plan, and find its way back on its own?

Our observation is that the target of recovery is already in the policy’s hands. The future the WAM predicted depicts the states it intended to pass through, and the actions were generated under its guidance, so the actions remain valid exactly as long as the scene follows it. When a deviation occurs, this prediction still shows the condition that was lost, such as a mug held in the gripper, and that state lies in the region where the policy was trained. A deviation therefore invalidates the remaining actions, but not the states they were meant to establish. This suggests a simple principle: use the policy’s own predicted future as an anchor.

[†]Corresponding author.

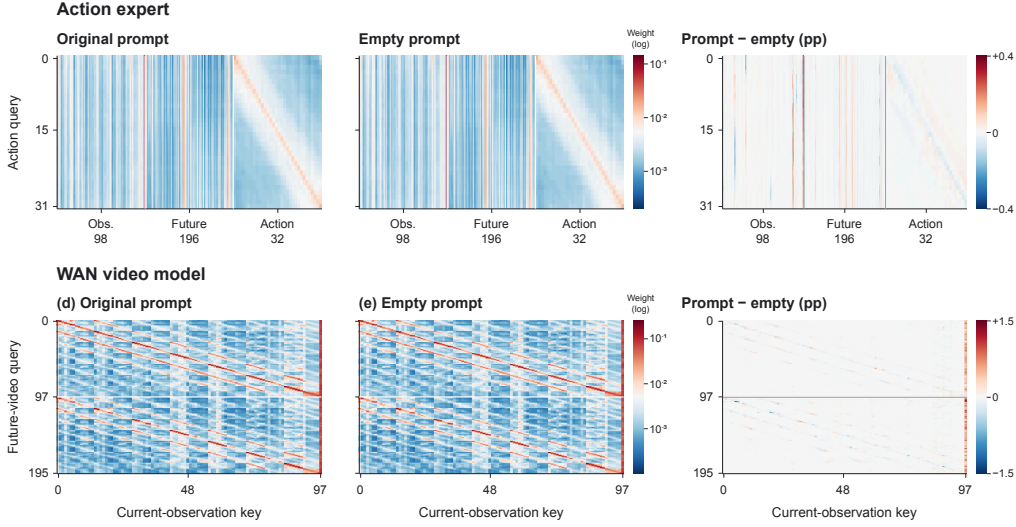


Figure 1: Attention of the frozen WAM under different instructions, with the observation and the random seed fixed. Top: attention of the action tokens. Bottom: attention of the future-video queries over the keys in the video model, under the original prompt, the empty prompt, and their difference. The two conditions produce nearly identical attention maps.

Following this principle, we propose **FAVOR (Future Anchored Verification and Online Recovery)**, which keeps the predicted frames as anchors and uses them both to verify execution and to guide recovery. Two challenges stand in the way. **(1)** Not every mismatch is a deviation: the anchor is an image produced by a generative model and may contain visual artifacts, so pixel-level differences do not necessarily imply a semantic mismatch. **(2)** The challenge is to convey the anchor to the policy and elicit the intended correction: the anchor is an image, whereas a WAM accepts only an observation and an instruction. Moreover, it leans on visual context and largely ignores the instruction: with the observation fixed, the original and the empty instruction yield nearly identical attention maps (Figure 1). The anchor must therefore be conveyed through a corrective instruction, and the influence of that instruction must be strengthened to elicit the intended correction.

For challenge (1), we introduce the *Anchor Verifier*, which compares each observation with its anchor together with the executed actions and tracks the mismatch along the chunk. It thus responds to deviations that break the task rather than to any visual difference.

For challenge (2), we introduce *Anchor-Guided Recovery* to return the scene to a state the policy knows how to continue from. It hands the alarmed anchor, the current observation, and the task to a vision-language model, which issues one short instruction naming the condition to restore rather than the pixels to reproduce. The frozen WAM then executes it under strengthened instruction guidance to reach the intended future before deviation.

We evaluate FAVOR on LIBERO [38] and LIBERO-Plus [16]. FAVOR raises the task success of the base WAM from 97.85% to 98.10% on LIBERO and from 72.60% to 72.98% on LIBERO-Plus. The Anchor Verifier separates failed from successful episodes with an AUROC of 0.992, above distance-based and learned detectors [17, 72].

In summary, our main contributions are as follows:

- A future-anchored view of WAM execution: the future predicted before acting serves not only as a reference for verifying execution but also as the target that recovery should restore.
- The Anchor Verifier, which grounds execution verification in the WAM’s own predicted future and executed actions, identifying when the scene no longer supports continuation.
- The Anchor-Guided Recovery, which translates the alarmed anchor into a corrective instruction and strengthens instruction guidance, enabling a WAM to use its own predicted future as a recovery goal and return to a state from which the original task can continue.

2 RELATED WORK

2.1 WORLD ACTION MODELS AND FUTURE PREDICTION

WAMs integrate future prediction into robot control [56]. Joint video–action models couple anticipated visual changes with robot actions in a single generative process [5, 27, 32, 35, 57, 74, 75, 86, 87], whereas cascaded interfaces pass visual future states as explicit conditions to inverse dynamics models (IDMs) [3, 12, 23, 28, 36, 62, 69]. The role of prediction depends on the inference interface: Fast-WAM separates the benefit of video co-training from that of explicit future generation at test time [78]. Related efforts use video models as simulators or reward sources [15, 20, 73] and diagnose whether generated futures follow the commanded actions [6, 33, 42]. We focus on WAMs whose predicted future states condition action generation and thus accompany the resulting actions.

In these interfaces, and in work on aligning visual features and instructions with action decoding [21, 44, 53], the predicted future is an input to action generation and plays no further role once the actions are produced. FFDC is the exception: it retains the prediction and compares it with incoming observations to judge whether the remaining actions can still be trusted, shortening execution or replanning when they cannot [64]. Dense correspondence from long-range flow or all-pairs point tracking [65, 68] could also relate the two frames, but it measures motion rather than task relevance. HarnessWAM also intervenes during WAM execution, but monitors subgoal completion rather than the prediction [18]. Here, as in adaptive-horizon methods [48, 63, 64], the signal only decides whether to continue, and replanning starts from the deviated state.

2.2 EXECUTION VERIFICATION AND RECOVERY

Execution monitoring for learned policies draws on internal features, action consistency, and semantic progress [1, 13, 14, 17, 34, 43, 47, 72, 79, 84]. Diagnostics for world models further expose inconsistency between imagination and execution, from rollouts that violate physical and causal consistency [7] to actions that diverge from a correctly imagined future [33]. A detector, however, says only that something went wrong, not what state to return to.

Existing recovery methods obtain their target from outside the acting policy. Planner-level loops revise behavior from environmental feedback and constraint violations [19, 24, 85], and test-time language correction takes the instruction from a human [39, 40, 58]. Policy-level recovery for VLAs is learned from failure-and-recovery data [11, 31, 45, 46] or inferred by an auxiliary predictive component, either a dynamics model [50] or imagined states toward which recovery is directed [59, 83]. Each obtains its objective at the cost of training or a second model, and none derives it from the plan the policy was executing. FAVOR takes the objective from that plan: the future the WAM predicted before acting is what recovery restores. Since a WAM accepts only an observation and an instruction, the VLM only translates this future into an instruction rather than estimating a target of its own. This translation step draws on VLM work in task-directed visual perception [76] and spatial reasoning [71, 80]. However, formulating an appropriate corrective instruction does not ensure that the WAM will act on it.

In the WAM considered here, an IDM decodes actions from the predicted future, so the influence of a corrective instruction on execution depends largely on how it shapes future generation. Yet the WAM used here exhibits limited sensitivity to instruction changes, producing largely unchanged outputs across different instructions (Figure 1). On the policy side, prior work makes corrections effective by training on them [39, 40, 58] or by aligning visual features and instructions with action decoding [21, 44, 53]. On the generation side, classifier-free guidance strengthens a condition at inference time [22], and controllable generation and editing steer diffusion models with visual conditions [10, 30, 52, 60, 66, 67, 70, 81] or instructions [77]. FAVOR strengthens the instruction on the generation side at inference time: guidance makes it shape the generated future, and the IDM turns that future into recovery actions, without retraining.

3 PRELIMINARIES

World action models. A WAM first predicts the visual states the task should pass through and then inverts them into actions [12, 62, 78, 82]; we build on the IDM variant of Fast-WAM [78]. Formally, let o_t denote the observation at step t , and ℓ the task instruction. At each planning step t_c

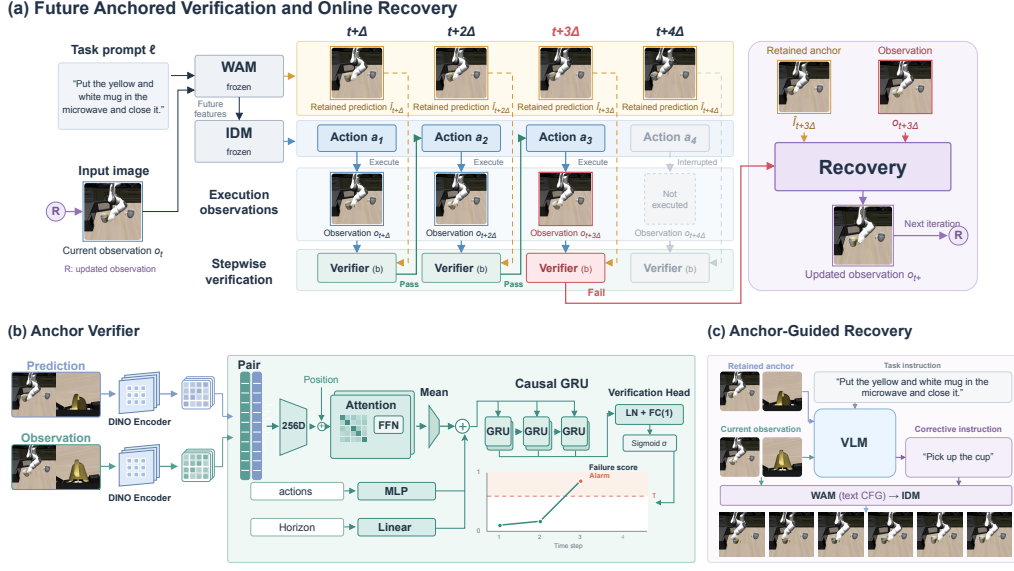


Figure 2: Overview of FAVOR. **(a)** The frozen WAM predicts future frames and the IDM decodes an action chunk; the predicted frames are retained as anchors. Every Δ steps, the verifier compares the observation with its anchor: a pass continues execution, and the failure interrupts the remaining actions and triggers recovery. **(b)** The Anchor Verifier encodes the prediction–observation pair with DINO, fuses it with action and horizon embeddings, and outputs a failure score through a causal GRU; an alarm is raised when the score exceeds τ . **(c)** Anchor-Guided Recovery: a VLM reads the alarmed anchor, the current observation, and the task, and issues a corrective instruction.

the policy produces a chunk of H actions $\mathbf{A}_c = (a_{c,0}, \dots, a_{c,H-1})$, with $a_{c,k}$ applied at step $t_c + k$. The WAM factorizes the chunk through the future $\mathbf{Z}_c$, a sequence of visual states over the chunk:

$$p(\mathbf{A}_c | o_{t_c}, \ell) = \int \underbrace{p(\mathbf{Z}_c | o_{t_c}, \ell)}_{\text{future prediction}} \underbrace{p(\mathbf{A}_c | o_{t_c}, \mathbf{Z}_c, \ell)}_{\text{inverse dynamics}} d\mathbf{Z}_c. \quad (1)$$

Future prediction. The video model is a diffusion transformer [51, 61] trained by flow matching [37]. Let z_t be the latent of I_t and $\mathbf{Z}_c = z_{t_c+\Delta:t_c+H}$ the latents of the chunk’s future frames. Flow matching learns a velocity field that transports Gaussian noise $\mathbf{Z}^{(1)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to the future $\mathbf{Z}^{(0)} = \mathbf{Z}_c$ through the ordinary differential equation

$$\frac{d\mathbf{Z}^{(s)}}{ds} = \mathbf{v}_\theta(\mathbf{Z}^{(s)}, s | z_{t_c}, \ell), \quad s \in [0, 1], \quad (2)$$

conditioned on the current latent and the instruction. Rectified flow [41] takes the straight path $\mathbf{Z}^{(s)} = (1-s)\mathbf{Z}_c + s\mathbf{Z}^{(1)}$, so the target velocity is simply $\mathbf{Z}^{(1)} - \mathbf{Z}_c$:

$$\mathbf{v}_\theta(\mathbf{Z}^{(s)}, s | z_{t_c}, \ell) \approx \mathbf{Z}^{(1)} - \mathbf{Z}_c. \quad (3)$$

At inference, the future is obtained by integrating Eq. (2) backward from noise to a clean latent,

$$\mathbf{Z}^{(s_{i-1})} = \mathbf{Z}^{(s_i)} - (s_i - s_{i-1}) \mathbf{v}_\theta(\mathbf{Z}^{(s_i)}, s_i | z_{t_c}, \ell), \quad i = T, \dots, 1, \quad (4)$$

which yields the predicted future $\hat{\mathbf{Z}}_c = \mathbf{Z}^{(s_0)}$. Decoding $\hat{\mathbf{Z}}_c$ produces H/Δ future frames $(\hat{I}_{c,1}, \dots, \hat{I}_{c,H/\Delta})$, spaced Δ control steps apart. Each frame $\hat{I}_{c,j}$ depicts the state expected at step $t_c + j\Delta$, with $\hat{I}_{c,0} = I_{t_c}$ denoting the current observation. These predicted frames serve as anchors for execution verification and recovery in Section 4.

Inverse dynamics. Given the current latent and the predicted future, the IDM infers the actions needed to realize that future [78]. The action tokens are denoised with the same flow-matching procedure, conditioned on the current latent, the predicted future, and the task instruction,

$$\mathbf{A}_c \sim p(\cdot | z_{t_c}, \hat{\mathbf{Z}}_c, \ell). \quad (5)$$

Predicted frames and generated actions follow a strict temporal correspondence: each group of Δ consecutive actions is meant to realize the transition between two consecutive predicted frames,

$$\hat{I}_{c,j-1} \xrightarrow{\mathbf{a}_{c,j}} \hat{I}_{c,j}, \quad \mathbf{a}_{c,j} = (a_{c,(j-1)\Delta}, \dots, a_{c,j\Delta-1}), \quad j = 1, \dots, H/\Delta, \quad (6)$$

with $\hat{I}_{c,0} = I_{t_c}$ for the first action. The action $\mathbf{a}_{c,j}$ is executed from step $t_c + (j-1)\Delta$ through step $t_c + j\Delta - 1$, so that $\hat{I}_{c,j}$ describes the state these actions are meant to reach at step $t_c + j\Delta$. The observation $I_{t_c+j\Delta}$ can then be compared with $\hat{I}_{c,j}$ to check whether execution has reached it.

4 METHOD

Overview. As observed in Section 1, the future predicted before acting depicts the states the chunk was meant to reach, and these states lie where the policy is reliable. FAVOR therefore keeps the predicted frames as *anchors* and uses them twice (Figure 2). An Anchor Verifier checks whether the current observation has deviated too far from its anchor, and Anchor-Guided Recovery uses that anchor to guide the robot back on course.

4.1 ANCHOR VERIFIER

Whether an observation still matches its anchor $\hat{I}_{c,j}$ need be judged semantically and over time. A fixed image distance [64] captures neither, so we learn the verifier from where the mismatch lies, how much motion the actions $\mathbf{a}_{c,j}$ command, and how it evolves along the chunk.

Verifier architecture. As shown in Figure 2(b), at anchor j of chunk c a frozen DINOv2 encoder [49] tokenizes the observation $I_{t_c+j\Delta}$ and the anchor $\hat{I}_{c,j}$ into $O_{c,j}$ and $\hat{O}_{c,j}$. Together with the executed actions $\mathbf{a}_{c,j}$ and the anchor horizon $p_j = j\Delta/H$, they form the input

$$X_{c,j} = [O_{c,j}, \hat{O}_{c,j}, \mathbf{a}_{c,j}, p_j]. \quad (7)$$

Observation and anchor tokens are paired position by position and fused by a spatial Transformer into a visual summary, to which embeddings of the actions and of the position are added, giving the anchor feature $x_{c,j}$. A GRU [9] carries these features across the anchors of the chunk and a linear head g produces the logit $u_{c,j}$ and the deviation score $s_{c,j}$

$$u_{c,j} = g(\text{GRU}(x_{c,1}, \dots, x_{c,j})), \quad s_{c,j} = \sigma(u_{c,j}) \in [0, 1]. \quad (8)$$

An alarm is raised when $s_{c,j}$ exceeds a threshold τ fixed before evaluation.

Training data. To train the Anchor Verifier, we construct two datasets with different levels of supervision. The first is a *locally labeled* set with a deviation label at every anchor. From the same simulator state, we execute a clean and a perturbed version of the planned actions and label each anchor by whether the perturbed execution has departed from the clean one. We use two perturbations, late Gaussian noise added to the arm commands in the latter half of the chunk and tail scaling that multiplies these commands by a fixed factor. The second is an *episode-labeled* set that carries only a terminal success or failure label per episode; it contains natural rollouts of the policy together with the demonstrations from its training set, which all count as successes.

Training strategy. We initialize the verifier using local supervision and then adapt it to natural rollouts using terminal outcomes. Since a failed rollout may contain many anchors that were correctly reached, we use multiple-instance learning to supervise episode-level predictions. Let $u_{i,c,j}$ denote the verifier logit at anchor (c, j) of episode i , $\mathcal{V}_i$ the set of its valid anchors, and $m_i = \max_{(c,j) \in \mathcal{V}_i} u_{i,c,j}$ the episode logit. For the failed and successful episode sets $\mathcal{F}$ and $\mathcal{S}$ in each training batch, we minimize

$$L_F = \frac{1}{|\mathcal{F}|} \sum_{i \in \mathcal{F}} \text{sp}(-m_i), \quad (9)$$

$$L_S = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \left[\frac{1}{2} \text{sp}(m_i) + \frac{1}{2|\mathcal{V}_i|} \sum_{(c,j) \in \mathcal{V}_i} \text{sp}(u_{i,c,j}) \right], \quad (10)$$

$$L_{\text{ep}} = \frac{1}{2}(L_F + L_S), \quad \text{sp}(x) = \log(1 + e^x). \quad (11)$$



Figure 3: Effect of the guidance scale on recovery predictions. All branches start from the same trigger state and the corrective instruction “pick up the cup”.

The failure term encourages at least one high score anchor without assigning positive targets throughout a failed episode. The success term suppresses both the strongest and the average score of successful episodes, so that alarms stay rare while execution is on track.

4.2 ANCHOR-GUIDED RECOVERY

At the alarm, the controller discards the unexecuted actions of the chunk and keeps the alarmed anchor I^{anc} . The WAM is conditioned on a current observation and an instruction, so the anchor needs to reach it through language. Specifically, we use a VLM $\mathcal{M}$ to generate a corrective instruction for transitioning from the current observation I_{t^*} to the anchored state I^{anc} .

$$\ell_{\text{corr}} = \mathcal{M}(I^{\text{anc}}, I_{t^*}, \ell). \quad (12)$$

The anchor specifies the nominal state that execution was expected to reach. Expressing this target as a short instruction makes the correction depend on the corresponding object-level state change, such as grasping an object, rather than on exact visual agreement with the predicted frame. If the VLM judges that no correction is needed, it returns the original task, $\ell_{\text{corr}} = \ell$. The WAM then generates futures and actions from the current observation and ℓ_{corr} , exactly as in normal execution.

Recovery starts from a state outside the policy’s training distribution under an instruction it never saw, and Figure 1 shows that the instruction has little influence in the WAM. We apply classifier-free guidance (CFG) [22] during recovery to strengthen the effect of ℓ_{corr} on the predicted future:

$$\hat{\mathbf{v}}_{\theta}(\mathbf{Z}^{(s)}, s \mid z_{t^*}, \ell_{\text{corr}}; \gamma) = \mathbf{v}_{\theta}(\cdot \mid z_{t^*}, \emptyset) + \gamma [\mathbf{v}_{\theta}(\cdot \mid z_{t^*}, \ell_{\text{corr}}) - \mathbf{v}_{\theta}(\cdot \mid z_{t^*}, \emptyset)], \quad (13)$$

where $\emptyset$ denotes the empty instruction and γ the guidance scale (Figure 3). The IDM then decodes the guided future into recovery actions.

At the end of the window, the VLM compares the current observation with the anchor once more: if it judges that the anchored condition is restored, control returns to the policy.

Table 1: Task success (% , $\uparrow$) on LIBERO and LIBERO-Plus, and the LIBERO failure rate (% , $\downarrow$) with its relative change from the base WAM. Bold marks the best entry in each column; the last row gives the gain of FAVOR over the base WAM in percentage points.

Method	LIBERO							LIBERO-Plus
	Success (% , $\uparrow$)					Failure (% , $\downarrow$)		Success ($\uparrow$)
	Spatial	Object	Goal	Long	Avg.	Rate	Rel.	Avg.
Fast-WAM-IDM	98.20	98.90	98.30	96.00	97.85	2.15	–	72.60
+ Verifier, direct replanning	98.30	99.20	98.20	96.00	97.92	2.08	–3.3%	72.60
+ Verifier, correction w/o anchor	98.30	99.30	98.40	96.10	98.02	1.98	–7.9%	72.53
FAVOR (Anchor-Guided Recovery)	98.30	99.40	98.40	96.30	98.10	1.90	–11.6%	72.98
Gain over base (pp)	+0.10	+0.50	+0.10	+0.30	+0.25	–0.25		+0.37

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Benchmarks and evaluation protocols. We evaluate FAVOR on LIBERO [38] and LIBERO-Plus [16]. LIBERO comprises four suites of ten tasks each: Spatial, Object, Goal, and Long (LIBERO-10). LIBERO-Plus uses the LIBERO tasks while varying the environment, task configuration, and observation distribution beyond the standard setting. We run 100 rollouts per task on LIBERO and a total of 1698 rollouts on LIBERO-Plus.

Implementation details. We instantiate FAVOR on the IDM variant of Fast-WAM [78]. The Anchor Verifier is trained on one NVIDIA A5000 GPU (24 GB). We use Qwen3-VL-2B-Instruct [2] as the VLM in Anchor-Guided Recovery. All evaluations run on four A5000 GPUs.

Evaluation metrics. The primary metric is native task success, determined by the environment’s own success condition and computed over the complete evaluation cohort, including episodes in which no intervention occurs. It thus credits successful recoveries and penalizes interventions that disrupt otherwise successful execution. The Anchor Verifier and Anchor-Guided Recovery are additionally evaluated in isolation in Section 5.3.

5.2 MAIN RESULTS

Results on LIBERO and LIBERO-Plus. Table 1 reports native task success. FAVOR raises the average success of the base WAM from 97.85% to 98.10% on LIBERO, cutting the failure rate by 11.6% relative, and from 72.60% to 72.98% on LIBERO-Plus. Both components are needed. From the same alarms, direct replanning under the original instruction removes only 3.3% of the base failures on LIBERO and none on LIBERO-Plus. Language correction without the anchor removes 7.9% on LIBERO but slightly hurts on LIBERO-Plus (72.53%); supplying the anchor accounts for the rest. The gain is largest on Object and Long. Object tasks share one scene and differ only in the target object, so the policy relies more on the instruction and a corrective instruction moves it more; on Long, an early deviation has the most time to compound.

Qualitative results. Figure 4 shows two cases that fail under native continuation and are rescued by FAVOR. In both, the Anchor Verifier detects the deviated state as soon as the observation stops matching its anchor: a bottle that the plan expects in the gripper but is not, and a mug that has left its intended path. Anchor-Guided Recovery reads the anchor to identify what was lost and issues a short instruction that restores it, “pick up the orange juice bottle” and “take the white mug”. The scene returns to the anchored state, the original plan applies again, and both tasks complete. Native continuation, by contrast, keeps executing the remaining actions as if nothing had happened, and both tasks fail once the budget runs out.

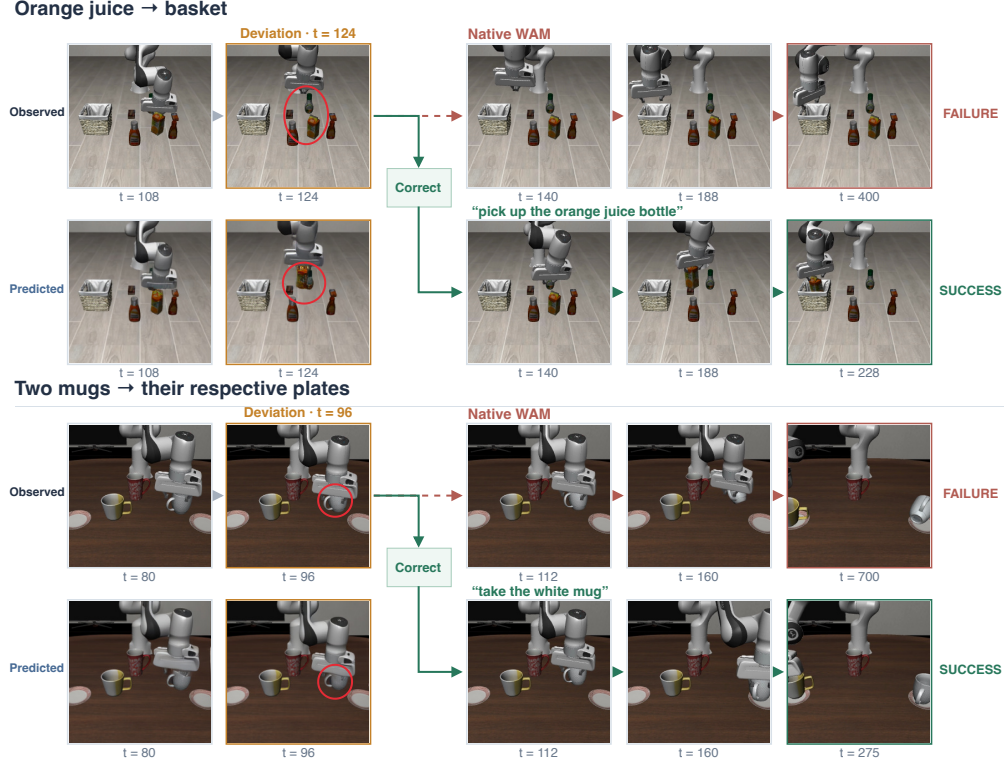


Figure 4: Qualitative examples of Anchor-Guided Recovery on LIBERO. In each case, the top row shows the observed execution and the bottom row the WAM’s retained prediction up to the alarm (orange frame); red circles mark where the observation departs from its anchor. Right of the alarm, native continuation under the original instruction fails (top), while the corrective instruction issued by the VLM (green) restores the anchored condition and the task succeeds (bottom). Both branches start from the same simulator state at the alarm.

5.3 ABLATION STUDIES

Anchor Verifier. We compare the Anchor Verifier with DINO adaptations of SAFE-MLP and SAFE-LSTM [17], and FAIL-Detect logpZO [72], as well as DINO feature distance based on DINOv2 [49] and PSNR [25]. All six methods score the same saved base-WAM test episodes, whose initial states are disjoint from those used to train, validate, and calibrate the learned detectors.

Each episode is scored by its maximum anomaly score over the anchors of complete chunks. The baselines split into learned detectors that do not see the anchor and fixed comparisons with it. SAFE-MLP and SAFE-LSTM [17] learn a failure classifier from terminal labels on the same DINO features of the observation, per anchor and over the anchor sequence respectively. FAIL-Detect logpZO [72] fits a flow model to the observation features of successful episodes and scores an episode by how unlikely its features are under that model. DINO distance averages Euclidean differences between corresponding observed and anchor tokens, and PSNR uses the mean squared RGB error between the observation and the anchor across both cameras, negated so that higher scores indicate greater discrepancy; neither is trained. Table 2 reports episode-level AUROC and AUPRC with task failure as the positive class, and failure coverage at each detector’s own threshold, calibrated on successful episodes at a 5% alarm rate and fixed before testing.

The Anchor Verifier leads on all three metrics. The fixed comparisons rank episodes well (AUROC 0.94–0.96) but have low AUPRC and coverage, so a raw distance to the anchor cannot tell a lost grasp from a harmless mismatch. The learned detectors without the anchor do no better, and SAFE-LSTM collapses. Learning the comparison against the anchor combines the two and raises AUPRC by 0.12 and coverage by 24 points. Figure 5 shows why. Under every baseline, the scores of successful episodes spread over the same range as those of failures. Under the Anchor Verifier, the

Table 2: Detector comparison on LIBERO.

Detector	AUROC $\uparrow$	AUPRC $\uparrow$	Coverage $\uparrow$
SAFE-MLP (DINO) [17]	0.935	0.664	67.6%
SAFE-LSTM (DINO) [17]	0.712	0.152	38.2%
FAIL-Detect logpZO (DINO) [72]	0.951	0.512	73.5%
DINO feature distance [49]	0.944	0.401	73.5%
PSNR [25]	0.956	0.472	67.6%
Anchor Verifier (ours)	0.992	0.781	97.1%

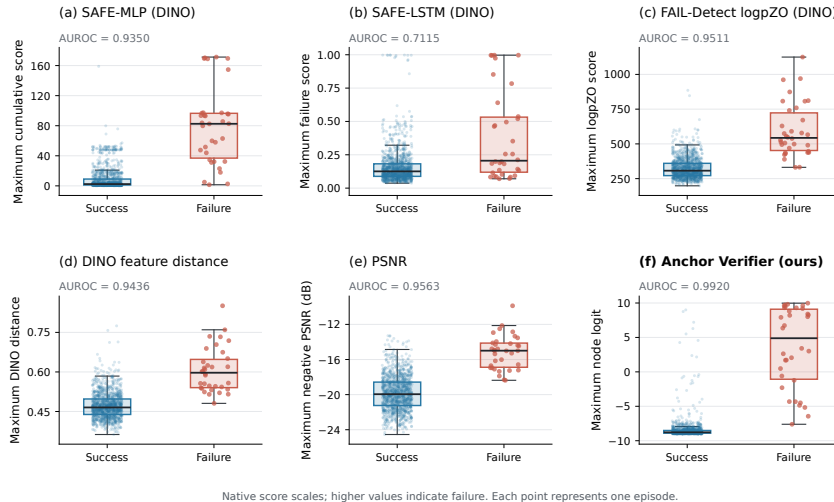


Figure 5: Distribution of the maximum episode score for successful and failed LIBERO episodes under each detector, on its native scale (higher values indicate failure); each point is one episode.

successes collapse into a tight band at the bottom of the scale and the failures rise well above it, so one threshold separates them, yielding high coverage at a low alarm rate.

Anchor-Guided Recovery. We ablate the guidance scale of Anchor-Guided Recovery: Figure 3 fixes the state and the corrective instruction “pick up the cup” and varies only γ in Eq. (13). At $\gamma = 1$, which reduces to plain conditional sampling, the predicted future is nearly identical to the continuation under the original task: the correction is ignored. At $\gamma = 3$ the prediction begins to depart from that continuation, but it does not yet realize the instruction, as the arm leaves the original path without reaching a grasp; the correction is registered but not followed. At $\gamma = 5$ the predicted future shows the arm grasping the cup, and the IDM turns this future into the corresponding recovery actions. The guidance scale thus sets how far the corrective instruction overrides the policy’s habitual continuation, and it must be large enough for the instruction to prevail.

6 CONCLUSION

World action models plan by predicting the future, yet once execution drifts from that plan they have no way to notice or to recover. FAVOR adds this capability to a frozen WAM by reusing its own prediction. The Anchor Verifier compares observations with their anchors to detect deviations that break the task, and Anchor-Guided Recovery turns the alarmed anchor into a corrective instruction that steers the policy back to a state it can continue from. On LIBERO and LIBERO-Plus it reduces failures and improves end-to-end success without modifying the policy.

REFERENCES

- [1] Christopher Agia, Rohan Sinha, Jingyun Yang, Ziang Cao, Rika Antonova, Marco Pavone, and Jeannette Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. In *Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pp. 689–723, 2025.
- [2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibao Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.
- [3] 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. *arXiv preprint arXiv:2310.10639*, 2023.
- [4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert 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, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0 : A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025. doi: 10.15607/RSS.2025.XXI.010.
- [5] 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. *arXiv preprint arXiv:2506.21539*, 2025.
- [6] Sixiang Chen, Jiaming Liu, Jixian Wu, Yichen Guo, Tinghao Wang, Siyuan Qian, Hao Chen, Jiajun Cao, Jian Tang, and Shanghang Zhang. Do robotic world models really follow actions? diagnosing and aligning action-conditioned generation for policy learning. *arXiv preprint arXiv:2608.24885*, 2026.
- [7] Weiliang Chen, Haowen Sun, Jun Gao, Jiawei Chi, Hanyang Wang, Qiyu Dai, Yihao Li, Hao Li, Jingnan Gao, Yi-Hsin Hung, Xingzhuo Guo, Shangchen Miao, Zhiyuan Shi, Xiang Li, Fengrui Tian, Weihua Du, Ziqi Huang, Shenyuan Gao, Siqiao Huang, Mingyu Liu, Yifei Li, Shizun Wang, Xi Wang, Tianqi Zhang, Xue Luo, Xiyin Ren, Jinshan Ren, Xiaoyang Shen, Xiaobo Hu, Zhiyang Dou, Mingyu Ding, Yichao Yan, Xinchao Wang, Yizhou Wang, Shilong Liu, Wenzhao Zheng, Yueqi Duan, Yuan Gong, Ziwei Liu, Ming-Yu Liu, Jialong Wu, Jiangran Lyu, and Fangfu Liu. HarnessEval-W: Agentifying the evaluation of visual worlds. *arXiv preprint arXiv:2608.16859*, 2026.
- [8] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023. doi: 10.15607/RSS.2023.XIX.026.
- [9] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014. doi: 10.3115/v1/D14-1179.
- [10] Xuehao Cui, Guangyang Wu, Zhenghao Gan, Guangtao Zhai, and Xiaohong Liu. Face2QR: A unified framework for aesthetic, face-preserving, and scannable QR code generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 29142–29165, 2024.
- [11] Yinpei Dai, Jayjun Lee, Nima Fazeli, and Joyce Chai. RACER: Rich Language-Guided Failure Recovery Policies for Imitation Learning. *arXiv preprint arXiv:2409.14674*, 2024.

- [12] 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 *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 9156–9172, 2023. doi: 10.52202/075280-0403.
- [13] Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando de Freitas, and Serkan Cabi. Vision-language models as success detectors. *arXiv preprint arXiv:2303.07280*, 2023.
- [14] Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. In *International Conference on Learning Representations (ICLR)*, 2025.
- [15] Alejandro Escontrela, Ademi Adeniji, Wilson Yan, Ajay Jain, Xue Bin Peng, Ken Goldberg, Youngwoon Lee, Danijar Hafner, and Pieter Abbeel. Video prediction models as rewards for reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 68760–68783, 2023. doi: 10.52202/075280-3009.
- [16] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.
- [17] Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. SAFE: Multitask failure detection for vision-language-action models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, Main Conference, pp. 40041–40076, 2025. doi: 10.52202/085713-1337.
- [18] Zhaopeng Gu, Bingke Zhu, Tianxi Lin, Guibo Zhu, Yingying Chen, Kai Wang, Tingyu Yuan, Chaoyang Zhao, Zhaowen Li, Peng Su, and Jinqiao Wang. HarnessWAM: Bridging Prediction and Deliberation in World Action Models. *arXiv preprint arXiv:2608.09516*, 2026.
- [19] Yanjiang Guo, Yen-Jen Wang, Lihan Zha, and Jianyu Chen. DoReMi: Grounding language model by detecting and recovering from plan-execution misalignment. *arXiv preprint arXiv:2307.00329*, 2023.
- [20] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations (ICLR)*, 2026.
- [21] Junjie He, Junfeng Li, Zhide Zhong, Haodong Yan, Ruixin Li, Yangyang Zheng, Jiaguan Zhu, Tianran Zhang, Yuqiao Du, Wen Chen, Shunbo Zhou, and Haoang Li. SG-WAM: Text-Grounded and Spatial-aware Semantic Guidance for World-Action Models. *arXiv preprint arXiv:2608.08839*, 2026.
- [22] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [23] 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 *International Conference on Machine Learning (ICML)*, 2025.
- [24] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and brian ichter. Inner Monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning (CoRL)*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1769–1782, 2023.
- [25] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 44(13):800–801, 2008. doi: 10.1049/el:20080522.

- [26] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag 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 *Conference on Robot Learning (CoRL)*, 2024.
- [27] Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.
- [28] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. In *International Conference on Learning Representations (ICLR)*, 2024.
- [29] Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. DART: Noise injection for robust imitation learning. In *Conference on Robot Learning (CoRL)*, volume 78 of *Proceedings of Machine Learning Research*, pp. 143–156, 2017.
- [30] Bonan Li, Yinhan Hu, Songhua Liu, and Xinchao Wang. Control and realism: Best of both worlds in layout-to-image without training. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34618–34629, 2025.
- [31] Chenxuan Li, Jiaming Liu, Guanqun Wang, Xiaoqi Li, Sixiang Chen, Liang Heng, Chuyan Xiong, Jiaxin Ge, Renrui Zhang, Kaichen Zhou, and Shanghang Zhang. A self-correcting vision-language-action model for fast and slow system manipulation. *arXiv preprint arXiv:2405.17418*, 2024.
- [32] Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.
- [33] Qi Li, Xingyi Yang, and Xinchao Wang. BadWAM: When world-action models dream right but act wrong. *arXiv preprint arXiv:2607.15207*, 2026.
- [34] Qi Li, Bo Yin, Weiqi Huang, Ruhao Liu, Bojun Zou, Runpeng Yu, Jingwen Ye, Weihao Yu, and Xinchao Wang. Vision-language-action safety: Threats, challenges, evaluations, and mechanisms. *arXiv preprint arXiv:2604.23775*, 2026.
- [35] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Robotics: Science and Systems (RSS)*, 2025.
- [36] Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025.
- [37] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.
- [38] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 44776–44791, 2023. doi: 10.52202/075280-1939.
- [39] Huihan Liu, Alice Chen, Yuke Zhu, Adith Swaminathan, Andrey Kolobov, and Ching-An Cheng. Interactive robot learning from verbal correction. *arXiv preprint arXiv:2310.17555*, 2023.
- [40] Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. In *Robotics: Science and Systems (RSS)*, 2023.

- [41] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.
- [42] Yuejiang Liu, Fan Feng, Lingjing Kong, Weifeng Lu, Jinzhou Tang, Kun Zhang, Kevin Murphy, Chelsea Finn, and Yilun Du. World Action Verifier: Self-improving world models via forward-inverse asymmetry. *arXiv preprint arXiv:2604.01985*, 2026.
- [43] Yushan Liu, Peibo Sun, Xintao Chao, Zhenyang Yang, Yifan Xie, Lingfeng Zhang, Shoujie Li, Chenyu Tang, Fang Chen, Xiao-Ping Zhang, and Wenbo Ding. CheckVLA: Execution-time verification with action-conditioned world model for long-horizon mobile manipulation. *arXiv preprint arXiv:2607.26789*, 2026.
- [44] Yushan Liu, Peibo Sun, Shoujie Li, Yifan Xie, Lingfeng Zhang, Xintao Chao, Shiyuan Dong, Fang Chen, Xiao-Ping Zhang, and Wenbo Ding. OA-WAM: Object-addressable world action model for robust robot manipulation. *arXiv preprint arXiv:2605.06481*, 2026.
- [45] Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing robot experiences for failure explanation and correction. In *Conference on Robot Learning (CoRL)*, 2023.
- [46] Weifeng Lu, Minghao Ye, Zewei Ye, Ruihan Tao, Shuo Yang, and Bo Zhao. RoboFAC: A comprehensive framework for robotic failure analysis and correction. *arXiv preprint arXiv:2505.12224*, 2025.
- [47] Yecheng Jason Ma, Joey Hejna, Ayzaan Wahid, Chuyuan Fu, Dhruv Shah, Jacky Liang, Zhuo Xu, Sean Kirmani, Peng Xu, Danny Driess, Ted Xiao, Jonathan Tompson, Osbert Bastani, Dinesh Jayaraman, Wenhao Yu, Tingnan Zhang, Dorsa Sadigh, and Fei Xia. Vision language models are in-context value learners. In *International Conference on Learning Representations (ICLR)*, 2025.
- [48] Junnan Nie, Jiayi Li, Chenghao Liu, Junyi Lao, Jiachen Zhang, Tianle Zhang, Liang Lin, and Songfang Huang. PACE: Phase-aware chunk execution for robot policies with action chunking. *arXiv preprint arXiv:2606.00537*, 2026.
- [49] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khilodov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.
- [50] Yi Pan, Miao Pan, Qi Lu, Jiaming Huang, Man Zhang, Siteng Huang, Xin Li, Jie Zhang, Yongliang Shen, Xuhong Zhang, and Wenqi Zhang. VLA-Corrector: Lightweight detect-and-correct inference for adaptive action horizon. *arXiv preprint arXiv:2607.01804*, 2026.
- [51] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [52] Zhibin Qin, Zhenxiong Tan, Zeqing Wang, Songhua Liu, and Xinchao Wang. SpotEdit: Selective region editing in diffusion transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18683–18692, 2026.
- [53] Lu Qiu, Yizhuo Li, Yi Chen, Yuying Ge, Yixiao Ge, and Xihui Liu. Making Foresight Actionable: Repurposing Representation Alignment in World Action Models. *arXiv preprint arXiv:2606.12217*, 2026.
- [54] Alfredo Reichlin, Giovanni Luca Marchetti, Hang Yin, Ali Ghadirzadeh, and Danica Kragic. Back to the manifold: Recovering from out-of-distribution states. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8660–8666, 2022. doi: 10.1109/IROS47612.2022.9981315.

- [55] Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, 2011.
- [56] Qihong Shen, Shihua Zhang, Yue Liao, Qi Li, Zhenxiong Tan, Shizun Wang, Shuicheng Yan, and Xinchao Wang. World action models: A survey. *arXiv preprint arXiv:2606.20781*, 2026.
- [57] Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. VideoVLA: Video Generators Can Be Generalizable Robot Manipulators. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.
- [58] Lucy Xiaoyang Shi, Zheyuan Hu, Tony Z. Zhao, Archit Sharma, Karl Pertsch, Jianlan Luo, Sergey Levine, and Chelsea Finn. Yell at your robot: Improving on-the-fly from language corrections. In *Robotics: Science and Systems (RSS)*, 2024.
- [59] Suyeon Shin, Juwon Kim, Hyeonbin Park, Hyunseo Kim, Hyundo Lee, Hyung-Sin Kim, and Byoung-Tak Zhang. Back to the Familiar Future: Failure Recovery for VLA Policies via Pre-Imagined Milestone Selection. *arXiv preprint arXiv:2606.09258*, 2026.
- [60] Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. OminiControl: Minimal and universal control for diffusion transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14940–14950, 2025.
- [61] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwei 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, Mengyang Feng, 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, Wenten 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. *arXiv preprint arXiv:2503.20314*, 2025.
- [62] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *International Conference on Learning Representations (ICLR)*, 2025.
- [63] Haoxuan Wang, Gengyu Zhang, Yan Yan, Ramana Rao Kompella, and Gaowen Liu. VLA knows its limits: Adaptive execution horizons for robot policies. *arXiv preprint arXiv:2602.21445*, 2026.
- [64] Rui Wang, Yue Zhang, Canyang Chen, Jiehong Lin, Zhongrui Wang, and Xiaojuan Qi. When to trust imagination: Adaptive action execution for world action models. *arXiv preprint arXiv:2605.06222*, 2026.
- [65] Guangyang Wu, Xiaohong Liu, Kunming Luo, Xi Liu, Qingqing Zheng, Shuaicheng Liu, Xinyang Jiang, Guangtao Zhai, and Wenyi Wang. AccFlow: Backward accumulation for long-range optical flow. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12119–12128, 2023.
- [66] Guangyang Wu, Xiaohong Liu, Jun Jia, Xuehao Cui, and Guangtao Zhai. Text2QR: Harmonizing aesthetic customization and scanning robustness for text-guided QR code generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8456–8465, 2024.
- [67] Guangyang Wu, Huayu Zheng, Siqi Luo, Guangtao Zhai, and Xiaohong Liu. AnimateQR: Bridging aesthetics and functionality in dynamic QR code generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025. doi: 10.52202/085713-0122.

- [68] Guangyang Wu, Youran Ding, Xinyu Che, Benyuan Sun, Yi Yang, and Xiaohong Liu. Matching every pair to track every point: PairFormer for all-pairs tracking and video trajectory fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35187–35196, 2026.
- [69] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations (ICLR)*, 2024.
- [70] Yinwei Wu, Xianpan Zhou, Bing Ma, Xuefeng Su, Kai Ma, and Xinchao Wang. IFAdapter: Instance feature control for grounded text-to-image generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15949–15959, 2025.
- [71] Lexiang Xiong, Qi Li, Jingwen Ye, and Xinchao Wang. Anatomy of a lie: A multi-stage diagnostic framework for tracing hallucinations in vision-language models. *arXiv preprint arXiv:2603.15557*, 2026.
- [72] Chen Xu, Tony Khuong Nguyen, Emma Dixon, Christopher Rodriguez, Patrick Miller, Robert Lee, Paarth Shah, Rares Andrei Ambrus, Haruki Nishimura, and Masha Itkina. Can we detect failures without failure data? uncertainty-aware runtime failure detection for imitation learning policies. In *Robotics: Science and Systems (RSS)*, 2025. doi: 10.15607/RSS.2025.XXI.073.
- [73] Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations (ICLR)*, 2024.
- [74] Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, Min Cao, Peng Li, Qiuping Deng, Wenjun Mei, Xiaofeng Wang, Xinze Chen, Xinyu Zhou, Yang Wang, Yifan Chang, Yifan Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. GigaWorld-Policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026.
- [75] 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. *arXiv preprint arXiv:2602.15922*, 2026.
- [76] Runpeng Yu, Xinyin Ma, and Xinchao Wang. Auto-controlled image perception in MLLMs via visual perception tokens. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21822–21831, 2025.
- [77] Ruonan Yu, Zhenxiong Tan, Zigeng Chen, Songhua Liu, and Xinchao Wang. ViFeEdit: A video-free tuner of your video diffusion transformer. *arXiv preprint arXiv:2603.15478*, 2026.
- [78] Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.
- [79] Haoran Zhang, Yifu Lu, Boyang Wang, Xuhui Kang, Yen-Ling Kuo, Zezhou Cheng, Mengdi Wang, and Odest Chadwicke Jenkins. Foresight: Failure detection for long-horizon robotic manipulation with action-conditioned world model latents. *arXiv preprint arXiv:2606.23085*, 2026.
- [80] Shihua Zhang, Qiuhong Shen, Shizun Wang, Tianbo Pan, and Xinchao Wang. Make geometry matter for spatial reasoning. *arXiv preprint arXiv:2603.26639*, 2026.
- [81] Shihua Zhang, Qiuhong Shen, and Xinchao Wang. Align images before you generate. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30521–30531, 2026.
- [82] Xiaoxiong Zhang, Xiong Zeng, and Wei Zhang. From world models to world action models: A concise tutorial for robotics. *arXiv preprint arXiv:2607.00836*, 2026.

- [83] Yanyan Zhang, Disheng Liu, Kai Ye, Chaoda Song, Xinpeng Li, Mohsen Hariri, Vikash Singh, Yu Yin, and Vipin Chaudhary. Imagining Recovery: Inference-Time Counterfactual Realignment for Vision-Language-Action Models. *arXiv preprint arXiv:2608.14822*, 2026.
- [84] Guiyu Zhao, Longteng Guo, Junyou Zhu, Jun Fu, Yanghong Mei, Bin Cao, Jie Jiang, Xingjian He, and Jing Liu. VeriSpace: Spatially grounded action verification for vision-language-action models. *arXiv preprint arXiv:2606.10568*, 2026.
- [85] Enshen Zhou, Qi Su, Cheng Chi, Zhizheng Zhang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, and He Wang. Code-as-Monitor: Constraint-aware visual programming for reactive and proactive robotic failure detection. *arXiv preprint arXiv:2412.04455*, 2024.
- [86] Pengfei Zhou, Shengcong Chen, Di Chen, Jiaxu Wang, Rongjun Jin, Bingwen Zhu, Yike Pan, Songen Gu, Kuanning Wang, Shufeng Nan, Xingyu Qiu, Chenhao Qiu, Pu Yang, Yunuo Cai, Jianxiong Gao, Yifan Li, Yanwei Fu, Xiangyu Yue, Zhi Chen, and Jianlan Luo. τ_0 -WM: A unified video-action world model for robotic manipulation. *arXiv preprint arXiv:2606.01027*, 2026.
- [87] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Robotics: Science and Systems (RSS)*, 2025. doi: 10.15607/RSS.2025.XXI.015.

A COMPLETE CONTROLLER

Algorithm 1 gives the complete FAVOR controller used in all experiments.

Algorithm 1 FAVOR

Require: Frozen WAM-IDM policy π , VLM $\mathcal{M}$, verifier $\mathcal{V}$, task instruction ℓ ; chunk horizon R , verification interval Δ , correction horizon B , threshold τ

- 1: Observe o_0 ; $t \leftarrow 0$
- 2: $\ell_{\text{act}} \leftarrow \ell$
- 3: intervened $\leftarrow$ false
- 4: **while** task not successful **do**
- 5: Predict future states and generate an action chunk from (o_t, ℓ_{act})
- 6: **for** $k = 1, \dots, R$ **do**
- 7: Execute a_k and observe o_{t+k}
- 8: **if** not intervened and $k \bmod \Delta = 0$ **then**
- 9: Compare o_{t+k} with its corresponding predicted state using $\mathcal{V}$
- 10: **if** $s_k > \tau$ **then**
- 11: Select the corresponding predicted state as anchor I^{anc}
- 12: $\ell_{\text{corr}} \leftarrow \mathcal{M}(o_{t+k}, I^{\text{anc}}, \ell)$
- 13: **repeat**
- 14: Execute ℓ_{corr} with guidance scale γ for B steps
- 15: $\ell_{\text{corr}} \leftarrow \mathcal{M}(o_t, I^{\text{anc}}, \ell)$
- 16: **until** $\mathcal{M}$ reports that the anchor is restored, or the budget is exhausted
- 17: $\ell_{\text{act}} \leftarrow \ell$
- 18: intervened $\leftarrow$ true
- 19: **break** $\triangleright$ Discard the remaining action suffix
- 20: **end if**
- 21: **end for**
- 22: **if** task successful **then**
- 23: **return**
- 24: **end if**
- 25: **end for**
- 26: Update the current observation and replan
- 27: **end while**

B IMPLEMENTATION DETAILS

Default settings. Table 3 lists the settings that the main text refers to symbolically.

Table 3: Default settings of FAVOR in our experiments.

Symbol	Meaning	Value
H	Predicted actions per chunk	16
R	Executed actions per chunk before replanning	16
Δ	Actions between consecutive anchors	4
τ	Alarm threshold of the fixed operating point	0.5 (LIBERO), 0.9 (LIBERO-Plus)
B	Maximum length of the correction window (actions)	64
γ	Guidance scale during recovery	5
λ, η	Failure weight and auxiliary weight in L_{local}	2, 0.1
T	Action budget per suite	400 (Spatial/Object/Goal), 700 (Long)

Training data. We construct paired execution snippets and complete policy rollouts. For each paired snippet, we save a simulator state and use the frozen policy to generate a 16-step action sequence and its imagined visual trajectory. We restore the same state for clean and perturbed executions, sharing the original imagined reference. Perturbations either add Gaussian noise to the arm commands or scale their magnitudes over the last eight steps, while leaving gripper commands unchanged. We record both branches’ observations and physical states to derive local deviation labels from their object and end-effector displacements.

For complete rollouts, we execute the frozen policy until success or the action budget is exhausted. We record the task instruction, initial-state identifier and random seed, imagined frames and planned actions at each replanning point, executed actions, synchronized external and wrist-camera observations, robot and object states, and the terminal task outcome. Observations are aligned with imagined frames after actions 4, 8, 12, and 16 of each chunk; incomplete trailing chunks are excluded. Paired snippets provide local deviation labels, whereas complete rollouts provide terminal outcomes.

Episode-level training. The episode-level stage warm-starts from a checkpoint previously trained with paired data and localized labels and optimizes L_{ep} in Eq. (11); the episode maximum spans the valid anchors of all complete chunks, and padding contributes to no loss term. Each epoch visits every training success once, batching up to 7 successes with one cyclically sampled failure, so that all training failures are covered. We use AdamW with a learning rate of 10^{-5} , weight decay of 10^{-4} , and gradient norm clipping at 1.0 for 5 epochs.

Locally supervised refinement. The episode maximum may become high only after an error has become irreversible. To study temporal specificity, a training variant supplements terminal-outcome supervision with reviewed anchor labels:

$$L_{\text{local}} = \frac{\lambda L_F + L_S}{\lambda + 1} + \eta L_{\text{ann}}. \quad (14)$$

Here λ weights failed episodes and η weights the auxiliary local-label loss. This refinement changes training but not the inference architecture.

Specifically, $L_{\text{ann}} = 0.5L_+ + 0.5L_{\text{event}} + L_- + 0.2L_{\text{rank}} + 0.1L_{\text{hard}}$. The per-anchor losses L_+ and L_- are binary cross-entropy on reviewed positive and negative anchors, averaged within each annotated episode and then across episodes. For the event term L_{event} , events are contiguous runs of positive anchors within a chunk, pooled from logits u by $\text{pool}(u) = 0.5 \log \text{mean} \exp(2u)$. The ranking term averages $\text{sp}(0.5 - \text{pool}(u_+) + \text{pool}(u_-))$ against reviewed-negative bags from the same task, including the same episode. The hard-negative term averages the negative-target binary cross-entropy over the four highest reviewed-negative logits of each episode, or over all of them if fewer exist. Successful episodes are excluded from this term, and empty terms contribute zero.

C VLM PROMPTS FOR DIAGNOSIS AND RECOVERY INSTRUCTIONS

The deployed prompt substitutes the original task instruction below. Each image combines the external and wrist views; the predicted anchor precedes the current observation.

```
Task: {task_instruction}

Image 1 is the nominal state expected at this moment.
Image 2 is the current observed state.
Give one short robot command that moves the current state
toward the nominal state. If they already match, reply
"continue the task". Reply with the command only.
```

Qwen3-VL-2B-Instruct runs in bfloat16 with deterministic decoding, an image longest edge of 384 pixels, and at most 64 new tokens. Raw responses and input images are retained; no subsequent ground-truth outcome enters the prompt.

Empty-instruction control. For the language analysis in Section 4.2, the literal empty string bypasses the usual instruction template. Its cached text encoding retains one valid special token, and padding and the original attention interface remain unchanged; the text branch is therefore not removed. This control differs from setting the video guidance scale to zero, which affects video denoising while the action stage still receives the task text.