Test-Time Training for Visual Foresight Vision-Language-Action Models

Sangwu Park [1] **Wonjoong Kim** [1] **Yeonjun In** [1] **Sein Kim** [1] **Hongseok Kang** [1] **Chanyoung Park** [1]

Abstract

Visual Foresight VLA (VF-VLA) has become a prominent architectural choice in the recent VLA due to its impressive performance. Nevertheless, the inherent design of VF-VLA makes it particularly vulnerable to out-of-distribution (OOD) shifts. Because the quality of action directly depends on the accuracy of the predicted future visual information, OOD conditions affect both stages at once. To address this vulnerability, we propose Test-Time Training Visual Foresight VLA (T^3VF), a test-time training approach motivated by the observation that the predicted future image and its subsequent observation form a natural supervision pair. To further address the practical challenges that arise from indiscriminate test-time updates, we introduce an adaptive update filtering mechanism. Empirically, T^3VF mitigates the OOD vulnerability of VF-VLA at a modest additional inference cost, without requiring any architectural modification or auxiliary modules[1].

1. Introduction

Vision-Language-Action (VLA) models have become a central paradigm for generalist robotic manipulation (Kim et al., 2025; Pertsch et al., 2025; Bjorck et al., 2025). Among them, a recent line of research adopts a two-stage formulation in which the model first predicts the future visual state that the robot is expected to reach and then generates actions conditioned on this prediction. Models that adopt this formulation are called Visual Foresight VLA (VF-VLA), and their impressive performance has made this design one of the prominent architectural choices in the recent VLA literature (Zhao et al., 2025; Zhang et al., 2025; Wang et al., 2025; Cen et al., 2025; Zhang et al.; Yang et al., 2025).

Despite the effectiveness of VF-VLA, its inherent design makes it particularly vulnerable to out-of-distribution (OOD) shifts. Because the quality of action generation directly depends on the accuracy of the predicted future visual information (Zhao et al., 2025), OOD conditions affect both the visual prediction stage and the action generation stage at once. As summarized in Fig. 1, recent VF-VLA such as WorldVLA (Cen et al., 2025) and Mantis (Yang et al., 2025) show substantial performance drops on LIBERO-Plus (Fei et al., 2025) compared to their in-distribution success rates on LIBERO (Liu et al., 2023). This support the dual-stage exposure of both pathways under OOD shifts.

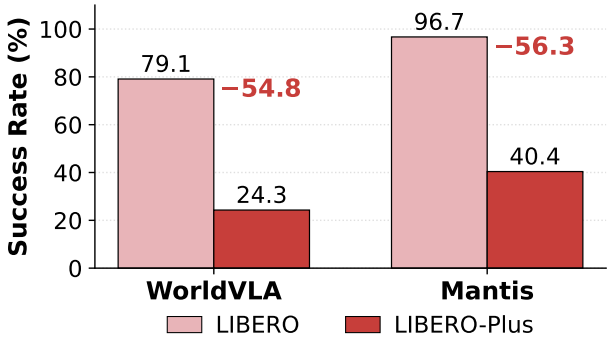

Figure 1. Performance comparison on LIBERO (In-Distribution) and LIBERO-Plus (Out-of-Distribution)

To address this vulnerability, we propose **T**est-**T**ime **T**raining **V**isual **F**oresight VLA (T^3VF), a test-time training approach motivated by a simple intuition that the predicted future image and its later observation form a natural self-supervision pair. In VF-VLA, at a given step the model predicts a future image from the current observation, and after executing the corresponding action, the actual image at that future step is observed. This observed image can be regarded as an oracle for the earlier prediction, allowing the test-time training to improve visual prediction. Accordingly, action generation also improves through the dependence.

However, this approach faces two practical challenges. First, a discrepancy between a prediction and its oracle does not always indicate a useful supervision signal. Such discrepancy can result from an inaccurate visual prediction or from an action-side error despite a correct prediction, which serves as noise during training. Second, the difficulty varies both across episodes and within an episode. Even with a proxy metric that can separate the two error sources discussed above, applying a fixed threshold cannot reflect this variation. Under such threshold, most steps in easier episodes or segments pass and allow excessive noise to enter training. In contrast, most steps in harder ones are skipped and fail to

[1]KAIST, Seoul, South Korea. Correspondence to: Chanyoung Park <cy.park@kaist.ac.kr>.

Presented at the ICML 2026 Workshop "Continual Adaptation at Scale: Towards Sustainable AI".

[1]Our source code is available at https://github.com/sangwu99/T3VF.git.

provide training signal.

To address the first challenge, we use the variance of the action at each step as the proxy metric. A low variance indicates that the model is internally consistent about the action it intends to execute. Such consistency makes a prediction error more plausibly attributable to the visual pathway and therefore a useful signal for updating the model. In contrast, high variance indicates that the source of the error cannot be reliably attributed. Since such an error may not provide a useful supervision signal, the corresponding step is skipped. To address the second challenge, we replace the fixed threshold with an adaptive variance buffer that maintains a running window of recent variances and permits an update only when its variance falls within the lower range of this buffer. Because this criterion is based on relative ranking rather than an absolute cutoff, it adapts naturally to the scale of variance in each episode as well as to its fluctuation between steps.

The main contributions of this work are as follows:

- **First identification of OOD vulnerability in VF-VLA.** We identify the amplified OOD vulnerability arising from the dual-stage exposure inherent to VF-VLA.
- **Test-time training with predicted-attained pairs.** We propose a test-time training approach leverages the correspondence between predicted images and their later observations, complemented by an adaptive update filtering.
- **Empirical effectiveness.** T^{3}VF improves the average success rate on LIBERO-Plus by 5% over the base VF-VLA, demonstrating that it mitigates the OOD vulnerability.

2. Related Works

Visual Foresight VLA refer to VLA models that predict a future image to be reached by the robot from the current observation and language instruction, and condition action generation on this prediction. The first group formulates future visual pixels and actions as autoregressive token sequences on a shared vocabulary (Zhao et al., 2025; Zhang et al., 2025; Wang et al., 2025; Cen et al., 2025). The second group predicts future visual information as compressed latent features or implicitly aligns latent representations with future visual states, avoiding the redundancy and computational overhead of autoregressive pixel-token generation. (Yang et al., 2025; Zhang et al.). Both groups share a common structure: action generation is conditioned on predicted visual information. Consequently, both the visual prediction stage and the action generation stage are exposed to OOD shifts (Pumacay et al.; Fei et al., 2025), leading to amplified vulnerability under such shifts. To the best of our knowledge, this vulnerability of VF-VLA remains unaddressed.

Another line of work adapts VLA models through test-time reinforcement learning. (Bai et al., 2025; Liu et al., 2026). They differ from ours in that they require a separate reward model, incur substantial overhead from online RL, and target VLA models in general rather than VF-VLA. For complete related work, please refer to Appendix A.

3. Method

3.1. Preliminaries

A VF-VLA consists of a VLM backbone P, an image head I_h, and an action head A_h. Given the language instruction l, the current observation o_t, and a placeholder q for any learnable query tokens,[2] backbone extracts following representations

$$(h_t^{\text{inst}},\, h_t^{\text{img}},\, h_t^{\text{act}}) \;=\; P(l,\, o_t,\, q). \tag{1}$$

This step computes h_t^{act} conditionally on h_t^{img} following the autoregressive nature of the VLM backbone, which discussed in Sec. 1. The image head predicts future observation $\hat{o}_{t+n}$ at a fixed gap of n steps ahead, and the action head generates the action $\hat{a}_t$,

$$\hat{o}_{t+n} \;=\; I_h\Big([h_t^{\text{inst}}, h_t^{\text{img}}],\, o_t\Big), \quad \hat{a}_t \;\sim\; A_h(h_t^{\text{act}}). \tag{2}$$

During training, the model is optimized through two objectives[2],

$$\mathcal{L}_{\text{train}} \;=\; \mathcal{L}_{\text{img}}(\hat{o}_{t+n},\, o_{t+n}) \;+\; \lambda\, \mathcal{L}_{\text{act}}(\hat{a}_t,\, a_t), \tag{3}$$

where o_{t+n} is the future ground-truth observation, a_t is the demonstration action, and λ balances the two terms.

3.2. Test-Time Training with Predicted-Attained Image Correspondence

Intuition. At step t, the model predicts $\hat{o}_{t+n}$ and executes the corresponding action $\hat{a}_t$. After n steps, the actual observation o_{t+n} is attained from the environment. The predicted-attained pair $(\hat{o}_{t+n},\, o_{t+n})$ thus provides supervision for the image head in the same form as Eq. 3, enabling test-time training without any additional data collection.

Building on this intuition, we introduce our method T^{3}VF (Test-Time Training for Visual Foresight VLA). The overall framework and detail algorithms are presented in Fig. 2 and Algorithm 1, respectively. At test time, we accumulate predicted-attained pairs along the trajectory of an episode into a set $\mathcal{B}$. Once $|\mathcal{B}|$ reaches a predefined batch size B, we apply a single update step that minimizes

$$\mathcal{L}_{\text{TTT}} \;=\; \frac{1}{B} \sum_{(\hat{o}_{t+n},\, o_{t+n}) \in \mathcal{B}} \mathcal{L}_{\text{img}}(\hat{o}_{t+n},\, o_{t+n}), \tag{4}$$

where $\mathcal{L}_{\text{img}}$ is the same image loss used during training. The update is applied only to q,[3] while the rest of the model parameters are kept frozen throughout test time. Since this update proceeds in parallel with execution and does not add auxiliary modules, the additional overhead is reduced.

[2]The specific forms of placeholder q, $\mathcal{L}_{\text{img}}$ and $\mathcal{L}_{\text{act}}$ depend on the design choices of each VF-VLA implementation.

[3]Which subset of parameters to update is depends on each VF-VLA implementation. We choose q in our setting because it is the smallest module that participates in the image prediction pathway.

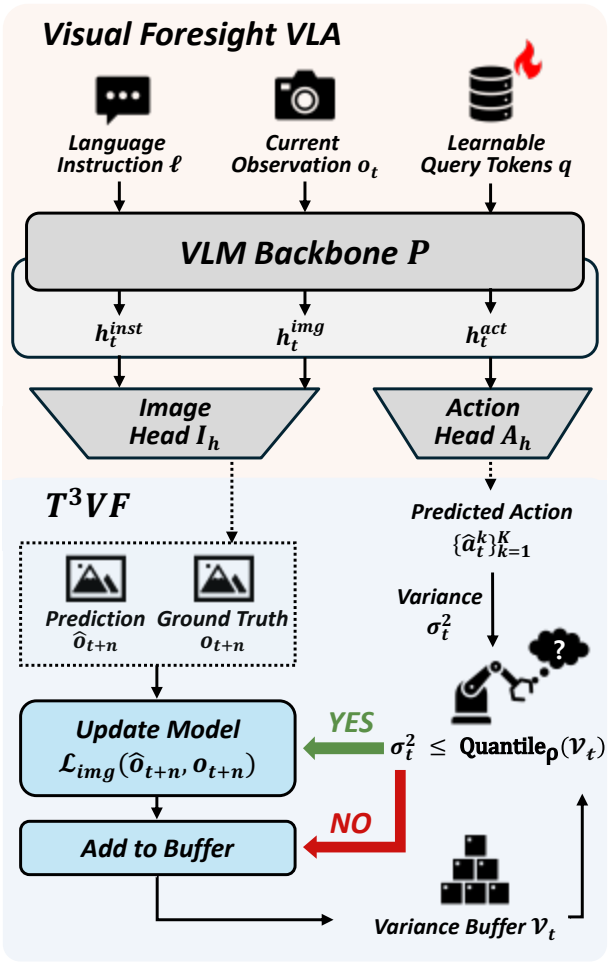

Figure 2. Overall framework of T^{3}VF. Built on standard VF-VLA (upper), T^{3}VF (lower) generates K action samples at each step and computes their variance σ_t^2. When σ_t^2 falls below the ρ percentile of the variance buffer $\mathcal{V}_t$, the predicted-attained pair $(\hat{o}_{t+n}, o_{t+n})$ is used for test-time training. Regardless, σ_t^2 is added to $\mathcal{V}_t$.

3.3. Adaptive Update Filtering

However, applying test-time training of Sec. 3.2 requires addressing the following two practical challenges. First, **the source of a large prediction error cannot be identified from its magnitude alone.** A predicted-attained pair can exhibit a large discrepancy from a genuinely inaccurate visual prediction or from action-side error despite a correct prediction. Therefore indiscriminate test-time training risks canceling the gains from the former with the harm from the latter. Second, **a fixed threshold over proxy does not generalize across or within episodes.** Even with a reliable proxy for distinguishing the two cases, a fixed threshold is inappropriate to the test-time setting, because the difficulty varies both within and across episodes. Within an episode, easier segments allow too many update and may inject noise into the parameters, while harder segments allow almost none and suppress learning. The same effect manifests across episodes respectively, where overall easy episodes oversaturate updates and overall hard episodes receive almost no supervision.

We address the first challenge by using the variance of the action at each step as a proxy metric. Concretely, at step t we draw K samples $\{\hat{a}_t^{(k)}\}_{k=1}^K$ and define

$$\bar{a}_t = \frac{1}{K}\sum_{k=1}^{K}\hat{a}_t^{(k)}, \qquad \sigma_t^2 = \frac{1}{K}\sum_{k=1}^{K}\left\|\hat{a}_t^{(k)} - \bar{a}_t\right\|_2^2, \quad (5)$$

where σ_t^2 is the squared L2 deviation averaged over the K action samples. A low σ_t^2 indicates that the model is internally consistent with the action it intends to execute, so that any large prediction error at step t is more plausibly attributable to the visual pathway. Therefore, it is a useful signal for updating q. In addition, this skip mechanism can be conducted at the moment $\hat{a}_t$ is produced, regardless of whether the paired o_{t+n} is yet available, and the K samples can be obtained from a single forward of the backbone followed by parallel decoding through the action head. Therefore, the additional cost is relatively less than calculating predicted-attained pairs through $\mathcal{L}_{\text{img}}$.

We address the second challenge by replacing a fixed threshold with a running window of recent variance values. Let $\mathcal{V}_t = \{\sigma_{t'}^2 : t' \in \mathcal{W}_t\}$ be the buffer of variances collected over the most recent $|\mathcal{V}|$ steps, where $\mathcal{W}_t$ denotes the corresponding step indices. We include step t in predicted-attained set $\mathcal{B}$ of Eq. 4 if and only if

$$\sigma_t^2 \leq \text{Quantile}_\rho(\mathcal{V}_t), \quad (6)$$

where $\rho \in (0, 1)$ is the percentile threshold. Because this criterion is based on relative ranking within the recent window rather than on an absolute cutoff, it naturally adapts to the scale of variance in each episode and keeps the overall frequency of accepted steps stable across episodes of differing difficulty. Combined with the test-time training procedure of Sec. 3.2, this filtering completes T^{3}VF by turning indiscriminately occurring predicted-attained pairs into reliable supervision for the visual prediction pathway.

4. Experiments

4.1. Setup

We evaluate T^{3}VF using Mantis (Yang et al., 2025) as the baseline, a representative VF-VLA. We evaluate on LIBERO-Plus (Fei et al., 2025) following its standard evaluation protocol on the seven perturbation dimensions and report average success rates. We consider two settings: "w/ Perturbed Train" uses a model fine-tuned on LIBERO-Plus, partially adapted to the perturbations; "w/o Perturbed Train" uses the official LIBERO checkpoint, which is fully OOD at evaluation. Within each setting, we compare the base model with and without T^{3}VF. For complete experimental setup, please refer to Appendix B.

4.2. Main Results

Table 1 shows that applying T^{3}VF consistently improves the overall average success rate in both settings. This pattern is

Table 1. Main results on LIBERO-Plus. Success Rate (%) under two settings.

Setting	Model	Robot	Language	Noise	Layout	Background	Camera	Light	Avg
w/ Perturbed Train	Mantis	29.0	47.8	47.4	42.3	60.3	50.5	67.8	49.3
	Mantis + T^{3}VF	31.8	49.2	48.2	44.9	63.0	55.3	72.4	52.1
	Δ	**+1.8**	**+1.4**	**+0.8**	**+2.6**	**+2.7**	**+4.8**	**+4.6**	**+2.8**
w/o Perturbed Train	Mantis	15.7	41.8	45.9	45.1	28.9	39.2	62.5	39.8
	Mantis + T^{3}VF	16.5	42.6	44.8	45.4	28.7	41.5	62.3	40.3
	Δ	**+0.8**	**+0.8**	**-1.1**	**+0.3**	**-0.2**	**+2.3**	**-0.2**	**+0.5**

in line with the motivation discussed in Sec. 1. T^{3}VF targets the visual prediction pathway with supervision extracted from predicted-attained image pairs, mitigating the dual-stage vulnerability of VF-VLA, and the resulting improvement in visual prediction propagates to action generation through their dependence. The relatively smaller improvement in the w/o Perturbed Train setting may reflect that the base model has not been adapted to the perturbation factors and therefore absorbs less supervision signal. However, T^{3}VF still yields a positive improvement.

4.3. Ablation Study

To isolate the contribution of each component of T^{3}VF, we conduct an ablation study on Robot perturbation[4] in the w/ Perturbed Train setting. Starting from the base model, we incrementally add (i) the test-time training of Sec. 3.2 without any filtering, (ii) the action variance filter of Sec. 3.3 with a fixed threshold, and (iii) the adaptive variance buffer of Sec. 3.3, which together form T^{3}VF. Table 2 reports the success rate at each step.

Table 2. Ablation of T^{3}VF components.

TTT	Var. Filter	Adap. Buffer	Success Rate
✗	✗	✗	29.0
✓	✗	✗	29.8
✓	✓	✗	28.6
✓	✓	✓	**31.8**

Adding the test-time training procedure on the base model improves the success rate, confirming that predicted-attained image pairs carry useful supervision even when applied indiscriminately. However, the fixed-threshold filter slightly degrades the success rate, indicating that an absolute cutoff cannot consistently separate informative from noisy steps, as discussed in Sec. 3.3. Replacing the fixed threshold with the adaptive variance buffer yields the largest improvement. This supports that our relative ranking criterion makes the proxy reliable.

4.4. Efficiency Analysis

Fig. 3 reports the average time per-episode on Robot perturbation setup, comparing the base model with two test-time training variants. The first variant is an indiscriminate test-time training that updates at every step without filtering, and the second is the full T^{3}VF.

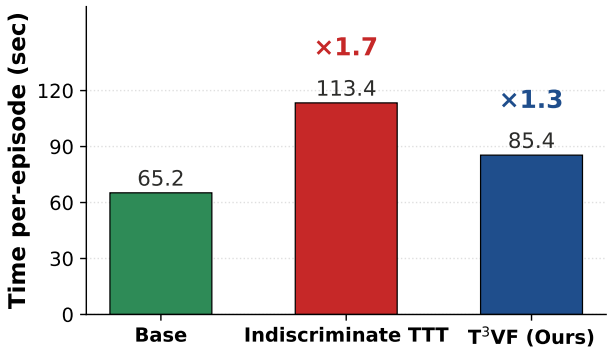

Figure 3. Efficiency Comparison across the settings.

Indiscriminate test-time training increases the per-episode time to approximately 1.7 times the base time, reflecting the cost of computing $\mathcal{L}_{\text{img}}$ and parameter update at every step. T^{3}VF reduces this to approximately 1.3 times through adaptive update filter, which triggers updates only on a fraction of the steps. Combined with the results in Fig. 3 and the ablation in Table 2, T^{3}VF achieves performance improvement of test-time training while reducing the overhead relative to the indiscriminate variant.

5. Discussion

We position T^{3}VF as a lightweight approach to mitigate the OOD vulnerability of VF-VLA at test time, rather than a dominant solution. Although gains seem incremental relative to the additional inference cost, T^{3}VF does not require additional training pipelines and operates without auxiliary modules, external reward signals, and offline adaptation. In addition, Its overhead is further contained by using fast skip criterion and updating only the smallest module that participates in the visual prediction pathway. Within these constraints, T^{3}VF offers a practical option for adapting VF-VLA at test-time without architectural modification.

6. Conclusion

We presented T^{3}VF, a test-time training approach that addresses the amplified OOD vulnerability of VF-VLA by leveraging predicted-attained image pairs as a self-supervision signal, complemented by an adaptive update filter. T^{3}VF offers a practical option for partially mitigating OOD shifts in VF-VLA without architectural modification.

[4]Robot perturbation alters the initial state, affecting both the visual and action, and is therefore the most challenging OOD.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bai, Z., Gao, C., and Shou, M. Z. Evolve-vla: Test-time training from environment feedback for vision-language-action models. *arXiv preprint arXiv:2512.14666*, 2025.

Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Cen, J., Yu, C., Yuan, H., Jiang, Y., Huang, S., Guo, J., Li, X., Song, Y., Luo, H., Wang, F., et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Fei, S., Wang, S., Shi, J., Dai, Z., Cai, J., Qian, P., Ji, L., He, X., Zhang, S., Fei, Z., et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Kim, M. J., Finn, C., and Liang, P. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., and Stone, P. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Liu, C., Liu, Y., Wang, T., Zhuang, Q., Liang, J. C., Yang, W., Xu, R., Wang, Q., Liu, D., and Han, C. On-the-fly vla adaptation via test-time reinforcement learning. *arXiv preprint arXiv:2601.06748*, 2026.

Pertsch, K., Stachowicz, K., Ichter, B., Driess, D., Nair, S., Vuong, Q., Mees, O., Finn, C., and Levine, S. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Pumacay, W., Singh, I., Duan, J., Krishna, R., Thomason, J., and Fox, D. The colosseum: A benchmark for evaluating generalization for robotic manipulation. In *RSS 2024 Workshop: Data Generation for Robotics*.

Wang, Y., Li, X., Wang, W., Zhang, J., Li, Y., Chen, Y., Wang, X., and Zhang, Z. Unified vision-language-action model. *arXiv preprint arXiv:2506.19850*, 2025.

Yang, Y., Li, X., Chen, Y., Song, J., Wang, Y., Xiao, Z., Su, J., Qiaoben, Y., Liu, P., and Deng, Z. Mantis: A versatile vision-language-action model with disentangled visual foresight. *arXiv preprint arXiv:2511.16175*, 2025.

Zhang, J., Guo, Y., Hu, Y., Chen, X., Zhu, X., and Chen, J. Up-vla: A unified understanding and prediction model for embodied agent. In *International Conference on Machine Learning*, pp. 74911–74922. PMLR, 2025.

Zhang, W., Liu, H., Qi, Z., Wang, Y., Yu, X., Zhang, J., Dong, R., He, J., Wang, H., Zhang, Z., et al. Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zhao, Q., Lu, Y., Kim, M. J., Fu, Z., Zhang, Z., Wu, Y., Li, Z., Ma, Q., Han, S., Finn, C., et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1702–1713. IEEE Computer Society, 2025.

A. Complete Related Work

A.1. Visual Foresight VLA

Visual Foresight VLA refer to VLA models that predict a future image to be reached by the robot from the current observation and language instruction, and condition action generation on this prediction. These models can be organized into two groups depending on whether they predict the image itself or its latent features. The first group formulates the prediction of future visual information and actions within a unified autoregressive token space, in which image, text, and action are treated as sequences on a shared discrete vocabulary. In this formulation, future images and actions are generated autoregressively within a single sequence (Zhao et al., 2025; Zhang et al., 2025; Wang et al., 2025; Cen et al., 2025). The second group emerges as a refinement of the first, predicting future visual information as compressed latent features rather than discrete image tokens, or implicitly aligning latent representations with future visual states. This approach avoids the redundancy and computational overhead inherent to autoregressive pixel-token generation while still providing dense visual supervision (Yang et al., 2025; Zhang et al.). Despite these differences, both groups share a common structure in which action generation is conditioned on predicted visual information, and consequently both the visual prediction stage and the action generation stage are exposed to OOD shifts leading to amplified vulnerability under such shifts.

A.2. OOD Evaluation in VLA

Several recent studies evaluate the out-of-distribution robustness of VLA models by injecting controlled perturbations into existing manipulation benchmarks (Pumacay et al.; Fei et al., 2025). These studies consistently report that existing VLA models, despite achieving high success rates under in-distribution evaluation, are uniformly vulnerable once such perturbations are introduced.

A.3. Test-Time Training in VLA

Several recent studies attempt to adapt VLA models during deployment through test-time reinforcement learning (Bai et al., 2025; Liu et al., 2026). They commonly rely on a learned progress estimator that replaces oracle reward signals and update the policy online via reinforcement learning. These approaches differ from ours in that they require training a separate reward model and incur substantial computational overhead from online reinforcement learning, and are designed as generic frameworks for VLA models.

B. Complete Experimental Setup

The hyperparameters of T^{3}VF are set to $n = 4$ for the prediction gap, $B = 4$ for the test-time training batch size, $K = 5$ for the action rollout count used in variance estimation, $|\mathcal{V}| = 10$ for the variance buffer size, and $\rho = 0.3$ for the percentile threshold. The base Mantis model serves as the comparison baseline within each setting.

Algorithm 1 T^{3}VF: Test-Time Training Visual Foresight VLA

Input: Trained VF-VLA $\{P, I_h, A_h, q\}$, instruction l, gap n, batch size B, sample count K, buffer size $|\mathcal{V}|$, percentile ρ, max step T
$\mathcal{B} \leftarrow \emptyset$, $\mathcal{V} \leftarrow \emptyset$
for $t = 1$ **to** T **do**
  $(h_t^{\text{inst}}, h_t^{\text{img}}, h_t^{\text{act}}) \leftarrow P(l, o_t, q)$
  $\{\hat{a}_t^{(k)}\}_{k=1}^{K} \sim A_h(h_t^{\text{act}})$
  $\bar{a}_t \leftarrow \frac{1}{K}\sum_k \hat{a}_t^{(k)}, \quad \sigma_t^2 \leftarrow \frac{1}{K}\sum_k \|\hat{a}_t^{(k)} - \bar{a}_t\|_2^2$
  if $|\mathcal{V}| < |\mathcal{V}|_{\max}$ **then**
    $\mathcal{V}.\text{append}(\sigma_t^2)$
    Execute $\bar{a}_t$
    continue
  else
    $\mathcal{V}.\text{pop}()$, $\mathcal{V}.\text{append}(\sigma_t^2)$
  end if
  Execute $\bar{a}_t$
  if $\sigma_t^2 \le \text{Quantile}_\rho(\mathcal{V})$ **then**
    $\hat{o}_{t+n} \leftarrow I_h([h_t^{\text{inst}}, h_t^{\text{img}}], o_t)$
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{(\hat{o}_{t+n}, o_{t+n})\}$ once o_{t+n} available
  end if
  if $|\mathcal{B}| = B$ **then**
    $q \leftarrow q - \alpha \nabla_q \mathcal{L}_{\text{TTT}}$ $\quad$ {TTT update}
    $\mathcal{B} \leftarrow \emptyset$
  end if
  if task done **then break**
end for