MOSAIC: MULTI-POLICY SAMPLING AND COLLABORATIVE TEST-TIME ADAPTATION FOR VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained vision-language-action (VLA) policies integrate visual observations and natural-language instructions to generate executable robot actions, providing a strong foundation for language-conditioned visuomotor control. However, large-scale pretraining and task-specific adaptation do not guarantee consistent deployment success. A single policy may produce different execution outcomes across stochastic action samples, while different policies may exhibit complementary successes and failures across initial states within the same task set. These variations create opportunities to discover successful behaviors during deployment. Yet a successful execution does not automatically become transferable to other policies or reliably reproducible by the policy that generated it. We therefore investigate how collaborative adaptation can turn successful deployment-time experience into transferable and reproducible policy behaviors.

We propose **MOSAIC**, a **M**ulti-p**O**licy **S**ampling **A**nd **I**nter-model **C**ollaboration framework for test-time adaptation of vision-language-action models that learns from successful trajectories generated by multiple policies and sampling seeds using environment-provided task-completion feedback. Starting from the same initial simulator state, participating policies perform rollouts with different sampling seeds and exploit successful experience through two complementary mechanisms. **Cross-policy action distillation** aligns other policies' action predictions with successful actions at the corresponding observations, facilitating behavioral transfer across policies. **Within-Policy Trajectory Consolidation** reuses a policy's own environment-validated successful trajectories and trains independently re-noised action samples toward the same successful target, encouraging behaviors that were discovered once to become more reproducible across stochastic generations. Both mechanisms update only a designated module near each policy's action output, requiring neither additional expert demonstrations nor additional trainable modules.

We adapt the policies on a subset of initial states for each task, then freeze the resulting parameters and evaluate on disjoint held-out initial states. On these held-out states, MOSAIC yields a positive numerical change in 103 out of 104 directional policy evaluations, with the average success rate increasing from 57.56% to 60.60%. These results provide evidence that successful deployment-time trajectories can be converted into policy updates that transfer beyond the initial states used for adaptation.

1 INTRODUCTION

The integration of Large Language Models (LLMs) with visual perception has catalyzed a paradigm shift in embodied intelligence, giving rise to generalist Vision-Language-Action (VLA) models Bai et al. (2025a); Bjorck et al. (2025); Brohan et al. (2023); Intelligence et al. (2025b); Kim et al. (2024); Jiang et al. (2025). By mapping visual observations and natural-language instructions to robot actions, pretrained VLA models provide a strong foundation for general robotic manipulation. However, their broad capabilities do not necessarily translate into reliable execution on specific downstream tasks. Different policies may succeed on different task instances, while a single policy may produce both successful and unsuccessful executions with different sampling seeds. These

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.

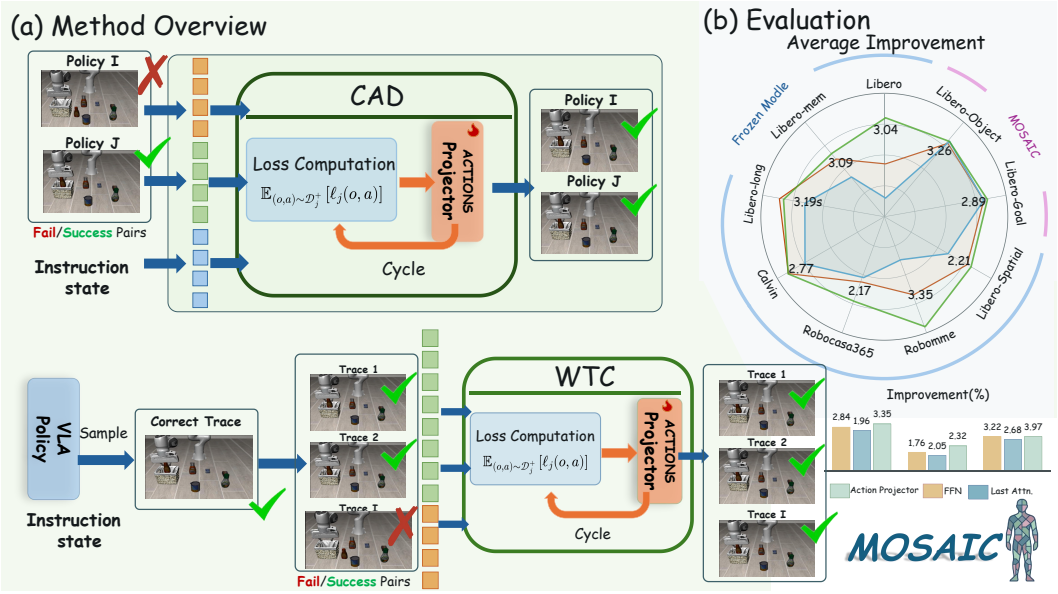

Figure 1: Overview of MOSAIC. Left: MOSAIC contains two complementary adaptation mechanisms: Cross-Policy Action Distillation (CAD), which transfers successful behavior across policies, and Within-Policy Trajectory Consolidation (WTC), which improves the reproducibility of behaviors within a policy across stochastic samples. Top right: Average performance changes after MOSAIC across the evaluated benchmarks. Bottom right: Comparison of alternative adaptation locations.

variations suggest that successful behaviors are already accessible within pretrained policies, but are neither consistently reproduced by individual policies nor readily shared across them. These observations motivate a central question: ***Can successful experience discovered during deployment be efficiently shared across different policies and sampling seeds at test time with minimal adaptation cost, and further consolidated into more robust behavior?***

To address this question, recent studies Li et al. (2026); Jeong & Yoon (2026) have increasingly explored how pretrained robot policies can exploit additional information available at deployment time. One line of work improves execution by allocating additional test-time computation, for example through multi-sample action generation, rollout-based planning, or auxiliary modules that refine action selection and manipulation accuracy. Although effective, these approaches can incur substantial computational and interaction overhead at inference time. A complementary line of work Park et al. (2026); Ji et al. (2026); Jang et al. (2026); Hao et al. (2026); Lin et al. (2026); Chen et al. (2026b) instead performs explicit policy adaptation during deployment, using self-supervised signals extracted from test-time interaction to update the policy online. However, when the policy encounters states beyond its existing task-completion distribution, its own deployment experience may fail to produce trajectories containing informative corrective signals, leaving insufficient positive supervision for further self-improvement. This limitation suggests a complementary opportunity: leveraging successful behaviors discovered by other policies or sampling seeds to provide supervision beyond the exploration boundary of any individual policy.

This motivates collaborative adaptation during deployment: using environment-validated successful experience to transfer behaviors across policies, and consolidate them within each policy. Building on this idea, we propose **MOSAIC**, a success-conditioned collaborative adaptation framework for pretrained generative robot policies. Across different policies, each receiving policy reconstructs its own training objective from shared successful experience, rather than directly matching the source policy's internal representations or intermediate generative predictions. Within each policy, multiple candidate behaviors are generated using different sampling seeds, and environment-validated successful executions are subsequently used to supervise the receiving policies.

Together, these two mechanisms allow the source of supervision to vary dynamically across executions, without requiring a fixed teacher–student hierarchy. To enable lightweight adaptation, we

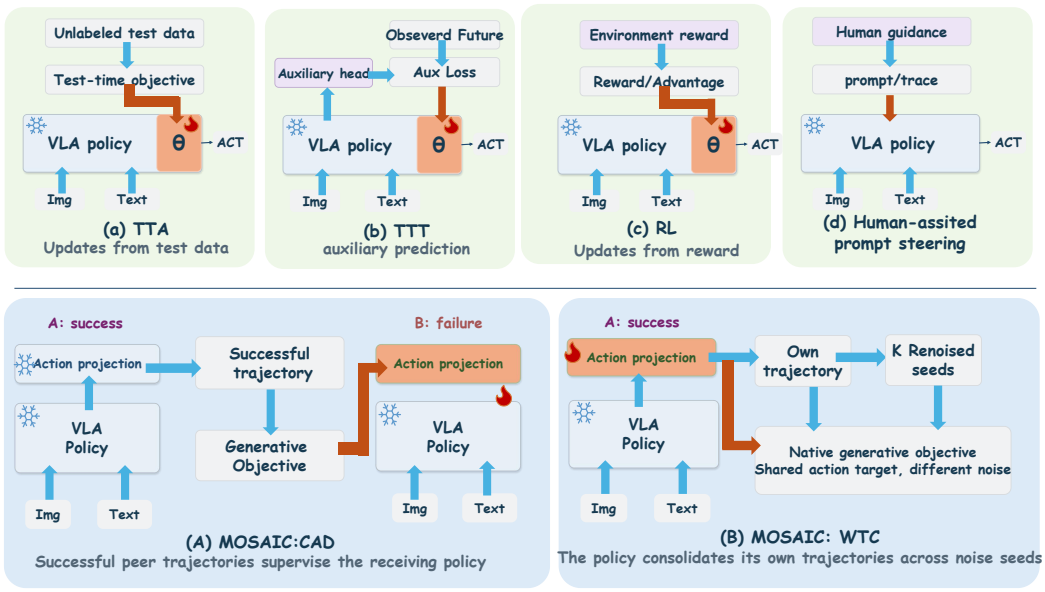

Figure 2: Overview of MOSAIC and existing deployment-time adaptation paradigms. **Top:** Existing approaches improve deployed policies using unlabeled test data, auxiliary future prediction, environment rewards, or human-provided guidance. **Bottom:** MOSAIC exploits successful behaviors already discovered during deployment through two complementary mechanisms.

update only a designated terminal module of the action-generation module, keeping the remaining parameters frozen. Our experiments show that, compared with alternative update strategies, adapting only the final module achieves a better trade-off between adaptation cost and performance, requiring less optimization overhead and yielding stronger results.Across four manipulation benchmarks and 52 policy pairs, CAD improves 103 of 104 policy-side evaluations with an average gain of 3.03 percentage points, while WTC independently improves average task success by 2.91 points.

Our main contributions are summarized as follows:

- We propose **MOSAIC**, a lightweight collaborative test-time adaptation framework for pretrained generative robot policies. Through multi-policy sampling and success-conditioned cross-policy learning, **MOSAIC** enables successful behaviors discovered by different policies and sampling seeds to be shared during deployment. Iterative sampling and adaptation further improve the reproducibility of successful behaviors, leading to more reliable task completion.
- We identify the terminal action module as an effective and efficient adaptation subspace for generative robot policies. By updating only this lightweight action-generation module while freezing the remaining parameters, **MOSAIC** achieves stable adaptation with minimal optimization overhead.

2 Related Work

2.1 Test-Time Adaptation

Test-time adaptation (TTA) improves pretrained models using information available during deployment, particularly under distribution shift Liang et al. (2025); Wang et al. (2025). Prior work explores self-supervised learning Sun et al. (2020), contrastive learning with pseudo-label refinement Chen et al. (2022), classifier adaptation Iwasawa & Matsuo (2021), prompt tuning Yoon et al. (2024), and forward-only optimization Niu et al. (2024). TTA has also been extended to LLMs Hu et al. (2025) and LLM agents Chen et al. (2026a).

In embodied settings, deployment-time adaptation includes both context-based mechanisms and feedback-driven policy updates. Behavioral Exploration adapts exploration from interaction his-

Table 1: VLA policies used on each benchmark for evaluating cross-policy action distillation.

Benchmark	Policies	Pairs
LIBEROLiu et al. (2023)	LingBotWu et al. (2026), DM0.5Dexmal Team (2026), GalaxeaLiu et al. (2026), $\pi_{0.5}$Intelligence et al. (2025b), StarVLAYe et al. (2026)	10
CALVINMees et al. (2022)	$\pi_{0.5}$, VLA-Adapter-ProWang et al. (2026), AVAXiao et al. (2026), UniVLABu et al. (2025), FLOWER variantsReuss et al. (2025)	21
RoboCasa365 Nasiriany et al. (2026)	$\pi_{0.5}$, SmolVLAShukor et al. (2025), Xiaomi-Robotics-1 Team et al. (2026), Qwen-GR00TYe et al. (2026)	6
RoboMMEDai et al. (2026)	$\pi_{0.5}$, SmolVLAShukor et al. (2025), MME-VLADai et al. (2026), MemoryVLAShi et al. (2026), OpenVLA-OFTKim et al. (2024), SimpleMemVLAYin et al. (2026)	15
Total		**52**

tories Wagenmaker et al. (2025), LocoFormer leverages long context to adapt locomotion Liu et al. (2025), and WorMI retrieves domain-specific world models at test time Yoo et al. (2025). FeedTTA adapts episodic feedback for online RL Kim et al. (2025), while EVOLVE-VLA uses a learned progress estimator to guide VLA adaptation Bai et al. (2025b).

MOSAIC instead focuses on transferring successful behaviors across heterogeneous policies and stochastic samples. Environment-validated successes serve as shared action supervision, while each policy optimizes its own native generative objective and updates only a lightweight terminal action module.

2.2 Deployment-Time Improvement for Manipulation Policies

Deployment-time improvement for robotic manipulation policies has been explored through several directions, including reinforcement-learning post-training, human-assisted prompt steering, and test-time training.

Reinforcement learning provides one established route for improving VLA policies after pretraining. Representative examples include $\pi^*_{0.6}$ Intelligence et al. (2025a), which studies RL-based self-improvement from heterogeneous experience and human corrections, and GR-RL Li et al. (2025), which transforms a generalist VLA into a dexterous specialist through multi-stage reinforcement learning.

Another line of work uses the conditioning interface of VLA models as a channel for human assistance during deployment. Yell At Your Robot Shi et al. (2024) incorporates real-time language corrections during execution, while MolmoAct Lee et al. (2025) and TraceVLA Zheng et al. (2025) expose editable trajectory representations as structured signals for spatial and temporal guidance. $\pi_{0.7}$ Intelligence et al. (2026) further broadens this interface with metadata, language guidance, and image-goal context, demonstrating that externally specified context can improve trajectory quality and generalization to unseen tasks.

Test-time training has also emerged as a promising direction for manipulation policies. Early work such as PAD Hansen et al. (2020) showed that self-supervised signals can help small-scale policies adapt to environmental changes. WorldAgen Wan et al. (2026) learns action prediction and world modeling, and adapts its world-model branch using transitions collected during deployment. More recent approaches, including Zero-WAM Zhou et al. (2026), TTT-VLA Zhang et al. (2026), RoboTTT Jiang et al. (2026), and WAM-TTT Feng et al. (2026), extend this idea to larger robotic policies. However, these methods rely on architectural components or adaptation mechanisms that must be explicitly introduced and pretrained in order to support test-time updates.

In contrast, our approach does not require introducing an additional adaptation module during pretraining. **MOSAIC** extracts supervision directly from deployment-time interaction, and reuses existing policy components as lightweight adaptation targets, providing a simpler and more general deployment-time adaptation mechanism.

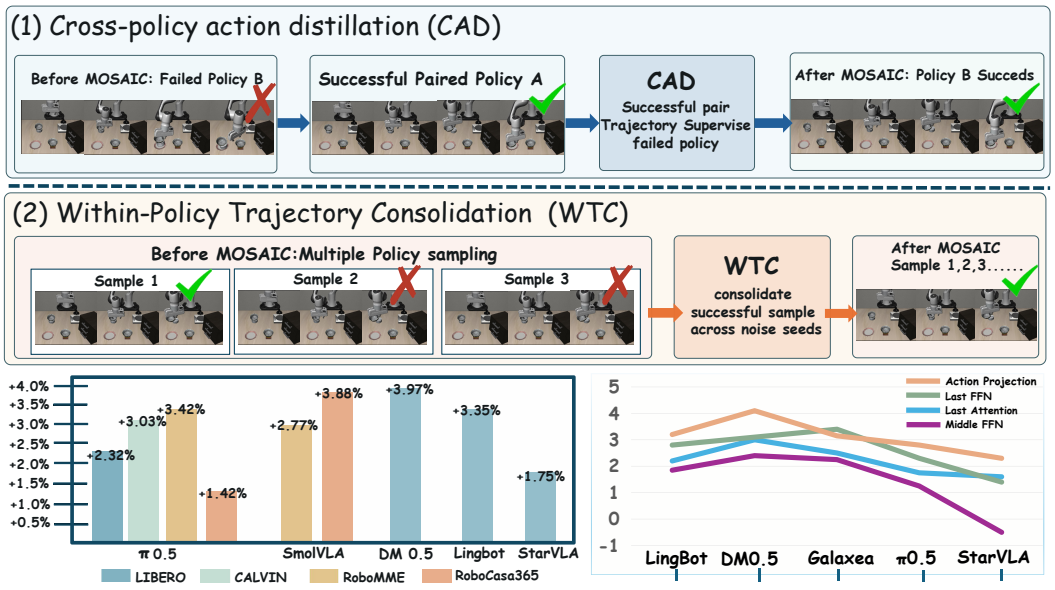

Figure 3: **Overview of MOSAIC and its empirical behavior. Top**: CAD transfers successful trajectories across policies, while WTC consolidates successful behaviors across stochastic samples. **Bottom left:** Model-wise gains from cross-policy adaptation across benchmarks. Adaptation-location comparison on LIBERO, where the action projection shows the most consistent improvements across policies.

3 METHOD: MOSAIC

Effective collaborative adaptation requires policies to address two complementary limitations: different policies may possess non-overlapping successful behaviors, while the stochastic action generation of an individual policy may make its existing behaviors difficult to reproduce consistently. To improve both behavioral coverage and sampling stability at test time, **MOSAIC** consists of two components: (1) **Cross-Policy Action Distillation (CAD)**, which transfers successful trajectories from a successful policy to policies that fail on the same task instance; and (2) **Within-Policy Trajectory Consolidation (WTC)**, which perturbs actions from a policy's rollout and improves the stability and reproducibility of the policy's behavior.

3.1 CROSS-POLICY ACTION DISTILLATION

Cross-Policy Action Distillation (CAD) exploits complementary success signals across different policies by using a successful trajectory from one policy as action supervision to improve another policy that fails on the same task instance.

Supervised Trajectory Identification. We evaluate multiple policies on the same task instance from an identical initial state s_0. Each policy π_{θ_i} generates a rollout, after which the environment returns a task-completion signal. Based on the execution outcome, we divide the policies into successful and failed sets. For a successful source policy $i \in \mathcal{P}^+(s_0)$ and a failed receiving policy $j \in \mathcal{P}^-(s_0)$, we replay the successful behavior of π_{θ_i} and use the resulting successful observation–action trajectory as supervision for π_{θ_j}.

$$\mathcal{D}_{i^+ \to j^-} = \{(o_t^+, a_t^+)\}_{t=0}^{T}, \quad \mathcal{P}^+(s_0) = \{i \mid \mathcal{E}(\tau_i) = 1\}, \quad \mathcal{P}^-(s_0) = \{j \mid \mathcal{E}(\tau_j) = 0\}. \tag{5}$$

Policy-Specific Action Distillation. Given a successful action a_t^+ at observation o_t^+, we adapt the receiving policy using its native generative training objective. Specifically, instead of introducing a shared cross-model distillation loss, each receiving policy reconstructs the diffusion- or flow-matching target already used by its own action generator. This design allows CAD to operate across heterogeneous generative policies without requiring their internal representations or intermediate generative states to be directly aligned. We use the native objective primarily as a

Table 2: **Directional pairwise gains from cross-policy adaptation.** Each entry (i, j) reports the absolute improvement in task success rate of the row policy when collaboratively adapted with the column policy. Diagonal entries are omitted because they do not involve cross-policy adaptation. F-ABC, F-ABCD, and F-D denote the corresponding FLOWER variants.

(a) LIBERO

	LingBot	DM0.5	Galaxea	$\pi_{0.5}$	StarVLA
LingBot	–	5.5	0.9	4.4	2.6
DM0.5	3.8	–	3.9	5.6	2.6
Galaxea	3.4	4.5	–	0.3	5.0
$\pi_{0.5}$	0.3	5.6	1.2	–	2.2
StarVLA	2.2	1.5	0.5	2.8	–

(b) CALVIN

	$\pi_{0.5}$	Adapter	AVA	UniVLA	F-ABC	F-ABCD	F-D
$\pi_{0.5}$	–	4.5	2.8	3.3	3.9	3.0	0.7
Adapter	0.5	–	1.0	3.5	2.7	5.1	2.9
AVA	3.0	4.8	–	5.6	0.8	0.1	2.3
UniVLA	3.3	3.0	2.5	–	5.7	2.9	4.8
F-ABC	3.6	5.8	0.8	5.5	–	2.7	1.7
F-ABCD	2.8	4.4	2.4	5.0	6.0	–	5.7
F-D	1.6	4.7	3.8	2.2	4.9	4.5	–

(c) RoboMME

	$\pi_{0.5}$	MME-VLA	MemoryVLA	OFT	SmolVLA	SimpleMem
$\pi_{0.5}$	–	0.6	1.3	4.3	0.1	0.8
MME-VLA	1.7	–	3.8	4.0	4.0	1.3
MemoryVLA	5.6	0.2	–	3.4	4.5	3.2
OFT	3.1	4.8	0.6	–	1.7	4.5
SmolVLA	5.6	3.0	5.5	0.1	–	2.7
SimpleMem	3.6	4.0	0.3	2.3	2.6	–

(d) RoboCasa365

	$\pi_{0.5}$	SmolVLA	Xiaomi	Qwen-GR00T
$\pi_{0.5}$	–	5.3	0.1	4.3
SmolVLA	2.3	–	1.9	4.1
Xiaomi	4.8	4.2	–	0.0
Qwen-GR00T	1.5	2.2	1.8	–

compatibility-preserving adaptation interface; the present experiments do not attempt to establish that it is universally superior to simpler action-space regression objectives.

$$\phi_j \leftarrow \phi_j - \eta \nabla_{\phi_j} \mathbb{E}_{(o,a)\sim\mathcal{D}_j^+} \left[\ell_j(o, a)\right]. \tag{6}$$

3.2 Within-Policy Trajectory Consolidation

Within-Policy Trajectory Consolidation (WTC) reduces stochastic variation within an individual policy by encouraging different sampling trajectories to reproduce a behavior that the policy has already generated. Unlike CAD, WTC does not require cross-policy supervision and can be applied to each rollout independently of its task-completion outcome.

Trajectory Re-noising and Resampling. Given a rollout $\tau_i = (o_t, a_t) * t = 0^T$ generated by policy $\pi * \theta_i$, we treat its original actions as reconstruction targets. For each action a_t, we inject noise and independently construct K stochastic training samples using different random seeds:

$$a_{t,k}^{\lambda} = \alpha_\lambda a_t + \sigma_\lambda \epsilon_k, \qquad \epsilon_k \sim \mathcal{N}(0, I), \quad k = 1, \ldots, K. \tag{7}$$

Within-Policy Consistency Consolidation. The original rollout serves as a shared target across different stochastic samples. Each sample is optimized using the policy's native generative objective, so that different noise realizations are trained toward the same underlying action trajectory. As in CAD, we update only the action projection layer ϕ_i while keeping the remaining parameters frozen:

$$\mathcal{L}_{\mathrm{WTC}}^i = \mathbb{E}_{(o_t,a_t)\sim\tau_i} \left[\frac{1}{K} \sum_{k=1}^{K} \ell_i \left(o_t, a_t; \epsilon_k\right) \right], \qquad \phi_i \leftarrow \phi_i - \eta \nabla_{\phi_i} \mathcal{L}_{\mathrm{WTC}}^i. \tag{8}$$

4 Experiments

4.1 Experimental Setup

Benchmarks and Policies. We evaluate CAD across four robotic manipulation benchmarks—LIBERO, CALVIN, RoboCasa365, and RoboMME—using a diverse collection of publicly released VLA checkpoints adapted to the corresponding benchmark settings. In total, we construct 52 cross-policy combinations, corresponding to 104 directional policy-side evaluations. The complete

Table 3: Cross-policy adaptation results across manipulation benchmarks.

Benchmark	# Pairs	Frozen	MOSAIC	$\Delta \uparrow$	Improved
LIBERO	10	89.66	92.60	**+2.94**	20/20
CALVIN	21	71.96	75.31	**+3.35**	42/42
RoboCasa365	6	28.65	31.36	**+2.71**	11/12
RoboMME	15	27.58	30.36	**+2.77**	30/30
Overall	**52**	**57.56**	**60.60**	**+3.03**	**103/104**

model–benchmark assignments are summarized in Table 1. Our evaluation intentionally includes policies with different architectures and substantially different baseline success rates in order to examine whether successful trajectories can serve as transferable supervision across heterogeneous policy families.

Adaptation and Evaluation Protocol. For each task, adaptation and evaluation are performed on disjoint sets of initial states. Successful trajectories used to construct adaptation targets are collected only from the adaptation split, while all reported post-adaptation results are measured after freezing the adapted parameters and executing the resulting policy on held-out initial states. Therefore, the reported improvements do not measure memorization of the trajectories directly used for optimization, but instead evaluate whether the adapted behavior transfers to unseen initial conditions. For CAD, participating policies are executed from the same adaptation-state initialization, and a directional source–receiver pair is constructed when the source policy succeeds while the receiving policy fails. The successful source trajectory is then converted into the receiving policy's native generative training objective. Unless otherwise specified, only the action projection module is updated while the remaining policy parameters are frozen.

4.2 Main Results

CAD improves heterogeneous VLA policies. Across 52 policy pairs and 104 directional policy-side evaluations, CAD yields a positive numerical change in 103 out of 104 directions. The average success rate increases from 57.56% to 60.60%, corresponding to an average gain of +3.03 pp. The gains span policy families with substantially different architectures and baseline performance levels, suggesting that a successful trajectory generated by one policy can provide a useful adaptation signal for another policy operating on the same task.

These results support the feasibility of *cross-policy behavior transfer*: successful deployment-time experience discovered by one pretrained policy can be converted into supervision for another pretrained policy. However, the current comparison does not disentangle cross-policy supervision from other possible uses of successful deployment experience, such as fine-tuning a policy on its own successful stochastic rollouts. We therefore interpret the results as evidence that successful trajectories from peer policies can serve as useful adaptation signals, rather than as evidence that cross-policy supervision is universally superior to self-generated supervision.

WTC improves sampling reliability. WTC improves average task success across all four evaluated manipulation benchmarks, increasing the overall success rate from 56.00% to 58.91% (+2.91 pp). Unlike CAD, WTC introduces no supervision from another policy. Instead, it consolidates successful behaviors that the policy has already discovered by training multiple independently renoised samples toward the same successful action trajectory. These results indicate that part of the apparent stochastic failure of generative VLA policies arises not only from missing capabilities, but also from an inability to reliably reproduce behaviors that are already contained within the policy's action distribution.

Relation to Additional Test-Time Sampling. MOSAIC is conceptually different from using additional stochastic rollouts only as repeated test-time attempts. Additional sampling can increase the probability of observing a successful execution for a particular test instance, but the discovered success is discarded once the episode ends. In contrast, MOSAIC uses successful deployment-time trajectories as optimization targets and evaluates the resulting frozen policy on disjoint initial states. Our experiments therefore examine whether a success discovered during deployment can be converted into a persistent policy update that transfers beyond the trajectory on which it was dis-

Table 4: Performance of WTC across manipulation benchmarks. Frozen and WTC report the average task success rate (%) over all evaluated policies. Δ denotes the average absolute improvement in percentage points (pp). Best Gain reports the largest policy-level absolute improvement within each benchmark, together with the corresponding policy.

Benchmark	Frozen	WTC	$\Delta \uparrow$	Best Gain
LIBERO	89.66	**92.76**	**+3.10**	DM0.5 (+5.1)
CALVIN	71.96	**74.96**	**+3.00**	FLOWER-D (+4.2)
RoboCasa365	28.65	**32.30**	**+3.65**	$\pi_{0.5}$/SmolVLA (+5.1)
RoboMME	27.58	**29.73**	**+2.15**	MemoryVLA (+4.6)
Overall	**56.00**	**58.91**	**+2.91**	**+5.1**

covered. We do not claim that MOSAIC dominates repeated sampling under every fixed interaction budget; a controlled comparison between parameter consolidation and additional rollout allocation is an important direction for future work.

4.3 Where to Adapt?

We further investigate whether the effectiveness of test-time adaptation depends on the choice of trainable module. Specifically, we compare adapting the action projection layer with the last FFN, the last attention layer, and a middle-layer FFN. We then analyze their effects from both the benchmark and model perspectives.

Across benchmarks. At the benchmark level, adapting the action projection achieves the strongest overall average improvement across the four evaluated benchmarks (+3.03 pp). The individually best adaptation location nevertheless varies across benchmarks: the last FFN performs slightly better on LIBERO, while the last attention layer performs slightly better on CALVIN. In contrast, the action projection performs best on RoboCasa365 and RoboMME and achieves the highest aggregate improvement overall. We therefore do not interpret the action projection as universally optimal for every policy or benchmark; rather, it provides a strong and comparatively consistent lightweight adaptation target across heterogeneous settings.

Across Policy Architectures. We further examine model-level variation on LIBERO, which contains five heterogeneous VLA policies. As shown in Table **??**, the action projection achieves the strongest average improvement across these policies, although the optimal adaptation location can vary for individual models. For example, Galaxea benefits slightly more from adapting the last FFN, whereas deeper attention adaptation degrades StarVLA. These results reinforce our interpretation that the action projection is a robust default adaptation location rather than a universally optimal choice.

Why Does the Action Projection Work Better? The action projection directly connects the pretrained policy representation to the executable action space. Updating this module therefore allows the model to adjust action generation while largely preserving the vision-language representations acquired during pretraining. In contrast, updating FFN or attention modules modifies more general intermediate representations and carry a higher risk of impairing pre-acquired perception, language understanding, and action reasoning abilities.

Because the reported pairwise results are success-rate estimates and several individual directional gains are small, we interpret the smallest changes as numerical trends rather than evidence of statistically significant improvement in isolation. Our main empirical conclusion is based on the aggregate behavior across policy pairs and benchmark

5 Conclusion

We introduced **MOSAIC**, a collaborative test-time adaptation framework for converting successful deployment-time experience into persistent updates of pretrained generative VLA policies. **MOSAIC** addresses two complementary sources of deployment variability: CAD transfers environment-

Table 5: Ablation on different test-time adaptation locations. Values denote the average absolute change in task success rate (pp).

Benchmark	Action Proj.	Last FFN	Last Attn.	Middle FFN
LIBERO	**+2.94**	+2.46	+1.79	+2.17
CALVIN	**+3.35**	+2.67	+2.52	+2.35
RoboCasa365	**+2.71**	+1.71	+1.87	+1.93
RoboMME	**+2.77**	+2.08	+1.67	+2.21
Overall	**+3.03**	+2.35	+2.06	+2.23

Table 6: Model-wise comparison of different test-time adaptation locations on LIBERO. Each value reports the average absolute change in task success rate (pp) across all available collaboration partners.

Policy	Action Proj.	Last FFN	Last Attn.	Middle FFN
LingBot	**+3.35**	+2.84	+1.96	+2.41
DM0.5	**+3.97**	+3.22	+2.68	+3.10
Galaxea	+3.30	**+3.52**	+2.43	+2.81
$\pi_{0.5}$	**+2.32**	+1.76	+2.05	+1.33
StarVLA	**+1.75**	+0.94	-0.18	+1.21
Average	**+2.94**	+2.46	+1.79	+2.17

validated successful trajectories across heterogeneous policies, while WTC reduces within-policy stochastic variability by consolidating trajectories across different noise realizations. Across four robotic manipulation benchmarks and 52 policy pairs, the experiments show a broadly positive trend from cross-policy adaptation, while within-policy consolidation also improves average task success. Beyond the aggregate performance gains, our results suggest that existing VLA policies contain a lightweight adaptation interface near their action output. Updating only the action projection achieves the strongest overall average improvement among the evaluated adaptation locations, while avoiding modification of the pretrained perception and language representations. Taken together, these findings support the broader view that successful behaviors discovered during deployment can serve as useful supervision for improving future policy executions, although further controlled comparisons are needed to isolate the respective contributions of cross-policy collaboration, self-generated successful experience, and additional test-time interaction.

6 LIMITATIONS

Our work has several limitations. First, although adapting the action projection performs consistently well, the underlying mechanism remains insufficiently understood. Future work could study how different VLM layers contribute to action generation and how to select more effective information for adaptation. Second, due to computational constraints, our experiments mainly rely on publicly released VLA checkpoints. Differences in pretraining and post-training settings may introduce uncontrolled factors, motivating more controlled evaluations with uniformly trained models. Finally, our framework relies on task-completion feedback and repeated rollouts during deployment. While this operational requirement is readily available in simulation, it can be costly in real-world robotics, motivating future work on weaker feedback, learned critics, and more sample-efficient adaptation. The current evaluation reports aggregate success-rate changes but does not provide confidence intervals for every directional policy pair. This limits the strength with which very small pair-level changes can be interpreted individually. Future evaluations should report repeated adaptation runs or paired uncertainty estimates over held-out initial states.

AI USE STATEMENT

We used large language models to assist with English language editing, drafting and revising selected manuscript passages, and literature retrieval and summarization. AI assistance was also used to prepare LaTeX tables and calculate summary statistics from experimental results provided by the authors. The authors take full responsibility for the accuracy, originality, and scientific integrity of the final manuscript, including all AI-assisted text, data, and analyses.

REFERENCES

Shuanghao Bai, Wenxuan Song, Jiayi Chen, Yuheng Ji, Zhide Zhong, Jin Yang, Han Zhao, Wanqi Zhou, Wei Zhao, Zhe Li, et al. Towards a unified understanding of robot manipulation: A comprehensive survey. *arXiv preprint arXiv:2510.10903*, 2025a.

Zechen Bai, Chen Gao, and Mike Zheng Shou. Evolve-vla: Test-time training from environment feedback for vision-language-action models. *arXiv preprint arXiv:2512.14666*, 2025b.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 295–305. IEEE, 2022.

Xiao Chen, Jiazhen Huang, Zhiming Liu, Qinting Jiang, Fanding Huang, Jingyan Jiang, and Zhi Wang. Test-time distillation for continual model adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7593–7604, 2026a.

Yanzhe Chen, Zhijun Cao, and Mike Zheng Shou. Where success breaks: Failure-boundary learning for robust vision-language-action models. *arXiv preprint arXiv:2609.06114*, 2026b.

Yinpei Dai, Hongze Fu, Jayjun Lee, Yuejiang Liu, Haoran Zhang, Jianing Yang, Chelsea Finn, Nima Fazeli, and Joyce Chai. Robomme: Benchmarking and understanding memory for robotic generalist policies. *arXiv preprint arXiv:2603.04639*, 2026.

Dexmal Team. DM0.5: An open-world foundation model for general-purpose embodied intelligence, July 2026. URL `https://www.dexmal.com/blog/dm0.5/index_en.html`.

Yusen Feng, Bingchen Han, Jiangran Lyu, Kai Liu, Yixin Zheng, Yuxuan Wan, Weiheng Liu, Sun Han, Ruiqin Li, Yulong Zhang, et al. Wam-ttt: Steering world-action models by watching human play at test time. *arXiv preprint arXiv:2607.06988*, 2026.

Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. *arXiv preprint arXiv:2007.04309*, 2020.

Haoran Hao, Shahram Najam Syed, Jeffrey Ichnowski, and Jeff Schneider. Far: Failure-aware retry for test-time recovery and continual policy improvement. *arXiv preprint arXiv:2607.01111*, 2026.

Jinwu Hu, Zhitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-time learning for large language models. *arXiv preprint arXiv:2505.20633*, 2025.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi^*_{0.6}$: a vla that learns from experience, 2025a. URL https://arxiv.org/abs/2511.14759.

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, 2025b. URL https://arxiv.org/abs/2504.16054.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities, 2026. URL https://arxiv.org/abs/2604.15483.

Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.

Sanghwan Jang, Minjin Jeon, Minsoo Kim, Seongjin Choi, Dongha Kim, and Hwanjo Yu. Ra-vla: Retrieval-augmented vla for test-time adaptation. *arXiv preprint arXiv:2608.25585*, 2026.

Yuhwan Jeong and Kuk-Jin Yoon. Reuse before you retrieve: Diagnosing headroom and complementarity for test-time augmentation of embodied multimodal policies. *arXiv preprint arXiv:2608.17484*, 2026.

Hongjin Ji, Guoyang Xia, Luoyang Sun, Fangxiang Feng, and Lei Ren. Vane: Reliable test-time training for vision-language-action models via future visual representation prediction. *arXiv preprint arXiv:2608.09448*, 2026.

Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, Xiao Liu, Shuiqi Cheng, Jiyang Gao, Huazhe Xu, and Hang Zhao. Galaxea open-world dataset and g0 dual-system vla model. *arXiv preprint arXiv:2509.00576*, 2025.

Yunfan Jiang, Yevgen Chebotar, Ruijie Zheng, Fengyuan Hu, Yunhao Ge, Jimmy Wu, Tianyuan Dai, Scott Reed, Li Fei-Fei, Yuke Zhu, et al. Robottt: Context scaling for robot policies. *arXiv preprint arXiv:2607.15275*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Sungjune Kim, Gyeongrok Oh, Heeju Ko, Daehyun Ji, Dongwook Lee, Byung-Jun Lee, Sujin Jang, and Sangpil Kim. Test-time adaptation for online vision-language navigation with feedback-based reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

Andrew Zou Li, Gokul Swamy, Yonatan Bisk, and Andrea Bajcsy. Elastic: Efficiently learning to adaptively scale test-time compute for generative control policies. *arXiv preprint arXiv:2606.31132*, 2026.

Yunfei Li, Xiao Ma, Jiafeng Xu, Yu Cui, Zhongren Cui, Zhigang Han, Liqun Huang, Tao Kong, Yuxiao Liu, Hao Niu, et al. Gr-rl: Going dexterous and precise for long-horizon robotic manipulation. *arXiv preprint arXiv:2512.01801*, 2025.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.

Tzu-Hsiang Lin, Srinivas Shakkottai, Dileep Kalathil, and PR Kumar. Reguide: From test-time guidance to self-improving diffusion policies. *arXiv preprint arXiv:2606.28939*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Min Liu, Deepak Pathak, and Ananye Agarwal. Locoformer: Generalist locomotion via long-context adaptation. *arXiv preprint arXiv:2509.23745*, 2025.

Yicheng Liu, Zibin Dong, Baijun Ye, Tianyuan Yuan, Tao Jiang, Anqi Yang, Shicheng Cao, Haonan Liu, Yue Sun, Zihan Guo, et al. G0. 5: One autoregressive stream for robot reasoning and action. *arXiv preprint arXiv:2608.11739*, 2026.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

Soroush Nasiriany, Sep Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations*, volume 2026, pp. 98643–98667, 2026.

Shuaicheng Niu, Chunyan Miao, Guohao Chen, Pengcheng Wu, and Peilin Zhao. Test-time model adaptation with only forward passes. *arXiv preprint arXiv:2404.01650*, 2024.

Sangwu Park, Wonjoong Kim, Yeonjun In, Sein Kim, Hongseok Kang, and Chanyoung Park. Test-time training for visual foresight vision-language-action models. *arXiv preprint arXiv:2605.08215*, 2026.

Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies. *arXiv preprint arXiv:2509.04996*, 2025.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 18567–18602, 2026.

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. *arXiv preprint arXiv:2403.12910*, 2024.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Xiaomi Robotics Team, Jun Guo, Piaopiao Jin, Jason Li, Peiyan Li, Yingyan Li, Futeng Liu, Wanli Peng, Optimus Qin, Yifei Su, et al. Xiaomi-robotics-1: Scaling vision-language-action models with over 100k hours of real-world trajectories. *arXiv preprint arXiv:2607.15330*, 2026.

Andrew Wagenmaker, Zhiyuan Zhou, and Sergey Levine. Behavioral exploration: Learning to explore via in-context adaptation. In *ICML*, 2025.

Chi Wan, Kangrui Wang, Yuan Si, Pingyue Zhang, and Manling Li. Worldagen: Unified state-action prediction with test-time world model training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18584–18592, 2026.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 40, pp. 18638–18646, 2026.

Zixin Wang, Yadan Luo, Liang Zheng, Zhuoxiao Chen, Sen Wang, and Zi Huang. In search of lost online test-time adaptation: A survey. *International Journal of Computer Vision*, 133(3): 1106–1139, 2025.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026.

Lei Xiao, Jifeng Li, Juntao Gao, Feiyang Ye, Yan Jin, Jingjing Qian, Jing Zhang, Yong Wu, and Xiaoyuan Yu. Ava-vla: improving vision-language-action models with active visual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13453–13463, 2026.

Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. Starvla-

$\backslash$

alpha : $Reducingcomplexityinvision-language-actionsystems$.*arXiv preprint arXiv:2604.11757*, 2026.

Cheng Yin, Wang Xu, Junpeng Yang, Sikyuen Tam, Hanyu Liu, Yuan Yao, Xiangrui Zeng, Junbo Cui, Yequan Wang, Zhouping Yin, et al. Simplememvla: A simple but effective native-video memory for vision-language-action models. *arXiv preprint arXiv:2609.05533*, 2026.

Minjong Yoo, Jinwoo Jang, Sihyung Yoon, and Honguk Woo. World model implanting for test-time adaptation of embodied agents. *arXiv preprint arXiv:2509.03956*, 2025.

Hee Suk Yoon, Eunseop Yoon, Joshua Tian Jin Tee, Mark Hasegawa-Johnson, Yingzhen Li, and Chang Yoo. C-tpt: Calibrated test-time prompt tuning for vision-language models via text feature dispersion. In *International Conference on Learning Representations*, volume 2024, pp. 8636–8660, 2024.

Wenbo Zhang, Jianxiong Li, Shuai Yang, Sijin Chen, Jiajun Liu, Lingqiao Liu, and Xiao Ma. Ttt-vla: Test-time latent prompt optimization for vision-language-action models. *arXiv preprint arXiv:2606.03127*, 2026.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *International Conference on Learning Representations*, volume 2025, pp. 54277–54296, 2025.

Jiaming Zhou, Qihang Zhang, Gangwei Xu, Cunxin Fan, Yujie Zhao, Ruilin Wang, Yiming Luo, Shuai Yang, Xing Zhu, Yujun Shen, et al. Zero-wam: In-context world-action modeling from human videos for open-ended task generalization. *arXiv preprint arXiv:2608.26103*, 2026.

A Evaluated Models and Benchmarks

This section provides brief descriptions of the VLA policies and robotic manipulation benchmarks used in our experiments.

A.1 Evaluated VLA Policies

- **LingBot-VLA**Wu et al. (2026): A pragmatic VLA foundation model pretrained on large-scale real-world robotic data, designed to generalize across tasks and robot embodiments while maintaining efficient adaptation.
- **DM0.5**Dexmal Team (2026): An open-world VLA model designed for general-purpose embodied intelligence, supporting open-ended instructions, long-horizon manipulation, dynamic disturbances, and multi-embodiment robot control.
- **G0.5 (Galaxea)**Liu et al. (2026): A unified autoregressive VLA model that generates reasoning and action tokens within a single transformer stream, enabling joint robotic reasoning and action generation.
- $\pi_{0.5}$Intelligence et al. (2025b): A generalist VLA model that combines heterogeneous robot data, semantic prediction, and multimodal co-training to enable broad open-world manipulation generalization.
- **StarVLA-α**Ye et al. (2026): A deliberately simplified VLA baseline that combines a strong vision-language backbone with a lightweight action prediction head, reducing architectural complexity while retaining strong manipulation performance.
- **VLA-Adapter-Pro**Wang et al. (2026): An enhanced implementation of VLA-Adapter, which bridges vision-language representations and robot actions through a lightweight policy module for efficient VLA adaptation.
- **AVA-VLA**Xiao et al. (2026): A VLA model equipped with active visual attention that leverages temporal context to dynamically emphasize task-relevant visual information during sequential decision making.
- **UniVLA**Bu et al. (2025): A cross-embodiment VLA framework that learns task-centric latent actions from heterogeneous video data and transfers them to different robotic embodiments through action decoding.
- **FLOWER**Reuss et al. (2025): An efficient flow-based VLA architecture that reallocates model capacity toward action generation and provides lightweight generalist policies for robotic manipulation. We evaluate multiple FLOWER variants on CALVIN.
- **SmolVLA**Shukor et al. (2025): A compact and efficient VLA model designed for low-cost training and deployment while retaining competitive manipulation performance.
- **Xiaomi-Robotics-1**Team et al. (2026): A large-scale robot foundation model pretrained on over 100K hours of real-world manipulation trajectories, designed for generalizable mobile manipulation and efficient downstream adaptation.
- **Qwen-GR00T**Ye et al. (2026): A dual-system VLA architecture that combines a Qwen vision-language backbone for high-level reasoning with a flow-matching action module for continuous robot control.
- **MME-VLA**Dai et al. (2026): A family of memory-augmented VLA policies built on the $\pi_{0.5}$ backbone, incorporating different memory representations and integration mechanisms for history-dependent robotic manipulation.
- **MemoryVLA**Shi et al. (2026): A memory-augmented VLA framework that maintains perceptual and cognitive memory of previous observations to support temporally dependent and long-horizon manipulation.
- **OpenVLA-OFT**Kim et al. (2024): An optimized variant of OpenVLA that uses continuous action prediction, parallel decoding, and action chunking for efficient robotic control and task-specific adaptation.
- **SimpleMemVLA**Yin et al. (2026): A memory-augmented VLA model that preserves visual history as timestamped video context and uses the pretrained backbone's self-attention as its native memory mechanism.

A.2 Evaluation Benchmarks

- **LIBERO**Liu et al. (2023): A language-conditioned robotic manipulation benchmark designed to evaluate knowledge transfer and lifelong learning. It contains four commonly used task suites—LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long—covering different forms of spatial, object-centric, goal-conditioned, and long-horizon manipulation.
- **CALVIN**Mees et al. (2022): A simulated benchmark for long-horizon language-conditioned robotic manipulation, requiring policies to compose multiple manipulation skills according to natural-language instructions.
- **RoboCasa365**Nasiriany et al. (2026): A large-scale simulation benchmark for generalist household robots, containing diverse manipulation tasks and kitchen environments designed to evaluate multi-task learning and generalization.
- **RoboMME**Dai et al. (2026): A benchmark for evaluating memory-augmented robotic policies on history-dependent manipulation. It covers temporal, spatial, object, and procedural memory through a set of long-horizon manipulation tasks.

Table 7: Complete pair-level results for cross-policy adaptation. Each policy column reports Frozen → MOSAIC with the absolute improvement in percentage points in parentheses.

Policy Pair	Policy A	Policy B
LIBERO		
LingBot / DM0.5	92.7 → **98.2** (+5.5)	89.5 → **93.3** (+3.8)
LingBot / Galaxea	92.7 → **93.6** (+0.9)	89.0 → **92.4** (+3.4)
LingBot / $\pi_{0.5}$	92.7 → **97.1** (+4.4)	89.6 → **89.9** (+0.3)
LingBot / StarVLA	92.7 → **95.3** (+2.6)	87.5 → **89.7** (+2.2)
DM0.5 / Galaxea	89.5 → **93.4** (+3.9)	89.0 → **93.5** (+4.5)
DM0.5 / $\pi_{0.5}$	89.5 → **95.1** (+5.6)	89.6 → **95.2** (+5.6)
DM0.5 / StarVLA	89.5 → **92.1** (+2.6)	87.5 → **89.0** (+1.5)
Galaxea / $\pi_{0.5}$	89.0 → **89.3** (+0.3)	89.6 → **90.8** (+1.2)
Galaxea / StarVLA	89.0 → **94.0** (+5.0)	87.5 → **88.0** (+0.5)
$\pi_{0.5}$ / StarVLA	89.6 → **91.8** (+2.2)	87.5 → **90.3** (+2.8)
CALVIN		
$\pi_{0.5}$ / Adapter-Pro	61.9 → **66.4** (+4.5)	80.3 → **80.8** (+0.5)
$\pi_{0.5}$ / AVA	61.9 → **64.7** (+2.8)	82.7 → **85.7** (+3.0)
$\pi_{0.5}$ / FLOWER-ABCD	61.9 → **64.9** (+3.0)	82.9 → **85.7** (+2.8)
Adapter-Pro / FLOWER-ABC	80.3 → **83.0** (+2.7)	75.1 → **80.9** (+5.8)
FLOWER-ABC / FLOWER-ABCD	75.1 → **77.8** (+2.7)	82.9 → **88.9** (+6.0)
FLOWER-ABC / FLOWER-D	75.1 → **76.8** (+1.7)	68.5 → **73.4** (+4.9)
$\pi_{0.5}$ / UniVLA	61.9 → **65.2** (+3.3)	52.3 → **55.6** (+3.3)
$\pi_{0.5}$ / FLOWER-ABC	61.9 → **65.8** (+3.9)	75.1 → **78.7** (+3.6)
$\pi_{0.5}$ / FLOWER-D	61.9 → **62.6** (+0.7)	68.5 → **70.1** (+1.6)
Adapter-Pro / AVA	80.3 → **81.3** (+1.0)	82.7 → **87.5** (+4.8)
Adapter-Pro / UniVLA	80.3 → **83.8** (+3.5)	52.3 → **55.3** (+3.0)
Adapter-Pro / FLOWER-ABCD	80.3 → **85.4** (+5.1)	82.9 → **87.3** (+4.4)
Adapter-Pro / FLOWER-D	80.3 → **83.2** (+2.9)	68.5 → **73.2** (+4.7)
AVA / UniVLA	82.7 → **88.3** (+5.6)	52.3 → **54.8** (+2.5)
AVA / FLOWER-ABC	82.7 → **83.5** (+0.8)	75.1 → **75.9** (+0.8)
AVA / FLOWER-ABCD	82.7 → **82.8** (+0.1)	82.9 → **85.3** (+2.4)
AVA / FLOWER-D	82.7 → **85.0** (+2.3)	68.5 → **72.3** (+3.8)
UniVLA / FLOWER-ABC	52.3 → **58.0** (+5.7)	75.1 → **80.6** (+5.5)
UniVLA / FLOWER-ABCD	52.3 → **55.2** (+2.9)	82.9 → **87.9** (+5.0)
UniVLA / FLOWER-D	52.3 → **57.1** (+4.8)	68.5 → **70.7** (+2.2)
FLOWER-ABCD / FLOWER-D	82.9 → **88.6** (+5.7)	68.5 → **73.0** (+4.5)
RoboCasa365		

Policy Pair	Policy A	Policy B
$\pi_{0.5}$ / SmolVLA	$16.6 \rightarrow \mathbf{21.9}$ $(+5.3)$	$9.1 \rightarrow \mathbf{11.4}$ $(+2.3)$
$\pi_{0.5}$ / Xiaomi-Robotics-1	$16.6 \rightarrow \mathbf{16.7}$ $(+0.1)$	$51.7 \rightarrow \mathbf{56.5}$ $(+4.8)$
$\pi_{0.5}$ / Qwen-GR00T	$16.6 \rightarrow \mathbf{20.9}$ $(+4.3)$	$37.2 \rightarrow \mathbf{38.7}$ $(+1.5)$
SmolVLA / Xiaomi-Robotics-1	$9.1 \rightarrow \mathbf{11.0}$ $(+1.9)$	$51.7 \rightarrow \mathbf{55.9}$ $(+4.2)$
SmolVLA / Qwen-GR00T	$9.1 \rightarrow \mathbf{13.2}$ $(+4.1)$	$37.2 \rightarrow \mathbf{39.4}$ $(+2.2)$
Xiaomi-Robotics-1 / Qwen-GR00T	$51.7 \rightarrow \mathbf{51.7}$ $(+0.0)$	$37.2 \rightarrow \mathbf{39.0}$ $(+1.8)$
RoboMME		
$\pi_{0.5}$ / SmolVLA	$17.2 \rightarrow \mathbf{17.3}$ $(+0.1)$	$4.1 \rightarrow \mathbf{9.7}$ $(+5.6)$
$\pi_{0.5}$ / MME-VLA	$17.2 \rightarrow \mathbf{17.8}$ $(+0.6)$	$44.7 \rightarrow \mathbf{46.4}$ $(+1.7)$
$\pi_{0.5}$ / MemoryVLA	$17.2 \rightarrow \mathbf{18.5}$ $(+1.3)$	$8.4 \rightarrow \mathbf{14.0}$ $(+5.6)$
$\pi_{0.5}$ / OpenVLA-OFT	$17.2 \rightarrow \mathbf{21.5}$ $(+4.3)$	$9.3 \rightarrow \mathbf{12.4}$ $(+3.1)$
$\pi_{0.5}$ / SimpleMemVLA	$17.2 \rightarrow \mathbf{18.0}$ $(+0.8)$	$81.8 \rightarrow \mathbf{85.4}$ $(+3.6)$
MME-VLA / MemoryVLA	$44.7 \rightarrow \mathbf{48.5}$ $(+3.8)$	$8.4 \rightarrow \mathbf{8.6}$ $(+0.2)$
MME-VLA / OpenVLA-OFT	$44.7 \rightarrow \mathbf{48.7}$ $(+4.0)$	$9.3 \rightarrow \mathbf{14.1}$ $(+4.8)$
MME-VLA / SmolVLA	$44.7 \rightarrow \mathbf{48.7}$ $(+4.0)$	$4.1 \rightarrow \mathbf{7.1}$ $(+3.0)$
MME-VLA / SimpleMemVLA	$44.7 \rightarrow \mathbf{46.0}$ $(+1.3)$	$81.8 \rightarrow \mathbf{85.8}$ $(+4.0)$
MemoryVLA / OpenVLA-OFT	$8.4 \rightarrow \mathbf{11.8}$ $(+3.4)$	$9.3 \rightarrow \mathbf{9.9}$ $(+0.6)$
MemoryVLA / SmolVLA	$8.4 \rightarrow \mathbf{12.9}$ $(+4.5)$	$4.1 \rightarrow \mathbf{9.6}$ $(+5.5)$
MemoryVLA / SimpleMemVLA	$8.4 \rightarrow \mathbf{11.6}$ $(+3.2)$	$81.8 \rightarrow \mathbf{82.1}$ $(+0.3)$
OpenVLA-OFT / SmolVLA	$9.3 \rightarrow \mathbf{11.0}$ $(+1.7)$	$4.1 \rightarrow \mathbf{4.2}$ $(+0.1)$
OpenVLA-OFT / SimpleMemVLA	$9.3 \rightarrow \mathbf{13.8}$ $(+4.5)$	$81.8 \rightarrow \mathbf{84.1}$ $(+2.3)$
SmolVLA / SimpleMemVLA	$4.1 \rightarrow \mathbf{6.8}$ $(+2.7)$	$81.8 \rightarrow \mathbf{84.4}$ $(+2.6)$

B Complete Cross-Policy Adaptation Results

We provide the complete pair-level results of Cross-Policy Action Distillation (CAD) in Table 7. For each policy pair, we report the task success rate before and after adaptation for both participating policies. Across 52 policy pairs and 104 policy-side evaluations, CAD improves 103 cases, increasing the average success rate from 57.56% to 60.60%, corresponding to an average gain of 3.03 percentage points.

The improvements are broadly distributed across benchmarks and policy families, with average gains of 2.94, 3.35, 2.71, and 2.77 percentage points on LIBERO, CALVIN, RoboCasa365, and RoboMME, respectively. These detailed results further show that the benefits of collaborative adaptation are not concentrated in a small number of policy pairs, but consistently appear across heterogeneous VLA architectures and different baseline performance levels.

C Complete Within-Policy Consolidation Results

Table 8 reports the complete model-wise results of Within-Policy Trajectory Consolidation (WTC). Across all 22 evaluated policies, WTC consistently improves task success, increasing the overall average success rate from 56.00% to 58.91% (+2.91 pp). The average gains are +3.10 pp on LIBERO, +3.00 pp on CALVIN, +3.65 pp on RoboCasa365, and +2.15 pp on RoboMME. These results show that WTC improves sampling reliability across heterogeneous policy architectures and across benchmarks with substantially different baseline performance levels.

WTC yields positive gains for all 22 evaluated policies. The improvement is not restricted to policies with low baseline performance: both strong policies, such as DM0.5 on LIBERO, and weaker policies, such as SmolVLA on RoboCasa365, benefit from trajectory consolidation. The largest gains reach +5.1 pp, while the overall average improvement is +2.91 pp.

Table 8: Complete model-wise results of Within-Policy Trajectory Consolidation (WTC). Frozen and WTC report task success rate (%), and Gain denotes the absolute improvement in percentage points (pp).

Policy	Frozen	WTC Gain	WTC
LIBERO			
LingBot	92.7	+0.6	93.3
DM0.5	89.5	+5.1	94.6
Galaxea	89.0	+5.0	94.0
$\pi_{0.5}$	89.6	+3.6	93.2
StarVLA	87.5	+1.2	88.7
CALVIN			
$\pi_{0.5}$	61.9	+3.3	65.2
VLA-Adapter-Pro	80.3	+4.1	84.4
AVA	82.7	+0.8	83.5
UniVLA	52.3	+2.6	54.9
FLOWER-ABC	75.1	+2.3	77.4
FLOWER-ABCD	82.9	+3.7	86.6
FLOWER-D	68.5	+4.2	72.7
RoboCasa365			
$\pi_{0.5}$	16.6	+5.1	21.7
SmolVLA	9.1	+5.1	14.2
Xiaomi-Robotics-1	51.7	+2.7	54.4
Qwen-GR00T	37.2	+1.7	38.9
RoboMME			
$\pi_{0.5}$	17.2	+0.8	18.0
MME-VLA	44.7	+1.5	46.2
MemoryVLA	8.4	+4.6	13.0
OpenVLA-OFT	9.3	+0.9	10.2
SmolVLA	4.1	+2.3	6.4
SimpleMemVLA	81.8	+2.8	84.6
Overall	**56.00**	**+2.91**	**58.91**

Table 9: Model-wise comparison of different test-time adaptation locations on LIBERO. Values denote the average absolute change in task success rate (pp) across all available collaboration partners.

Policy	Action Proj.	Last FFN	Last Attn.	Middle FFN
LingBot	**+3.35**	+2.84	+1.96	+2.41
DM0.5	**+3.97**	+3.22	+2.68	+3.10
Galaxea	+3.30	**+3.52**	+2.43	+2.81
$\pi_{0.5}$	**+2.32**	+1.76	+2.05	+1.33
StarVLA	**+1.75**	+0.94	-0.18	+1.21
Average	**+2.94**	+2.46	+1.79	+2.17

D Additional Adaptation-Location Ablations

To complement the aggregate analysis in the main paper, we report model-wise results for different adaptation locations on all four benchmarks. We compare the action projection layer with the last FFN, the last self-attention module, and a middle-layer FFN. Values denote the absolute change in task success rate relative to the frozen policy.

Across the four benchmarks, the action projection provides the most consistent adaptation target. It achieves the highest average gain on every benchmark and is the best-performing location for the

Table 10: Model-wise comparison of different test-time adaptation locations on CALVIN. Values denote the average absolute change in task success rate (pp).

Policy	Action Proj.	Last FFN	Last Attn.	Middle FFN
$\pi_{0.5}$	**+3.03**	+2.54	+1.86	+2.31
VLA-Adapter-Pro	+2.62	+2.11	**+2.81**	+1.74
AVA	**+2.77**	+1.93	+2.26	+2.04
FLOWER-ABC	**+3.35**	+2.88	+2.41	+2.67
FLOWER-ABCD	**+4.38**	+3.72	+3.95	+3.31
FLOWER-D	**+3.62**	+3.08	+2.57	+2.95
UniVLA	**+3.70**	+2.46	+1.81	+1.44
Average	**+3.35**	+2.67	+2.52	+2.35

Table 11: Model-wise comparison of different test-time adaptation locations on RoboCasa365. Values denote the average absolute change in task success rate (pp).

Policy	Action Proj.	Last FFN	Last Attn.	Middle FFN
$\pi_{0.5}$	**+3.23**	+2.46	+1.83	+2.68
SmolVLA	**+2.77**	+1.95	+2.34	+1.72
Xiaomi VLA	**+3.00**	+2.72	+2.18	+2.43
Qwen-GR00T	**+1.83**	-0.31	+1.12	+0.87
Average	**+2.71**	+1.71	+1.87	+1.93

majority of individual policies. A small number of policies benefit more from deeper adaptation—e.g., Galaxea on LIBERO, VLA-Adapter-Pro on CALVIN, and $\pi_{0.5}$ on RoboMME—but these exceptions depend on the policy architecture. In contrast, the action projection remains consistently effective across heterogeneous policy families. Deeper adaptation can also introduce negative transfer, as observed for StarVLA on LIBERO, Qwen-GR00T on RoboCasa365, and SimpleMemVLA on RoboMME.

Table 12: Model-wise comparison of different test-time adaptation locations on RoboMME. Values denote the average absolute change in task success rate (pp).

Policy	Action Proj.	Last FFN	Last Attn.	Middle FFN
$\pi_{0.5}$	+1.42	+0.73	+0.56	**+1.67**
MME-VLA	**+2.96**	+2.34	+2.08	+2.41
MemoryVLA	**+3.38**	+2.75	+2.21	+2.90
OFT	**+2.94**	+2.19	+2.54	+1.88
SmolVLA	**+3.38**	+2.62	+2.76	+2.33
SimpleMemVLA	**+2.56**	+1.84	-0.12	+2.05
Average	**+2.77**	+2.08	+1.67	+2.21