Trust Your Instincts: Confidence-Driven Test-Time RL for Vision-Language-Action Models

Siyao Chen[1], Jiakang Yuan[1], Jiaxin Wang[2], and Tao Chen[1,2,3⋆]

[1] College of Future Information Technology, Fudan University, Shanghai, China
siyaochen25@m.fudan.edu.cn, eetchen@fudan.edu.cn
[2] Shanghai Innovation Institute, Shanghai, China
[3] MoShen Intelligence, Shanghai, China

Abstract. Reinforcement learning (RL) has become indispensable for pushing Vision-Language-Action Models (VLA) beyond static imitation learning. However, existing RL methods typically necessitate external environmental feedback, relying on predefined success signals to guide policy updates. In this work, we demonstrate that VLA models possess strong internal evaluative capabilities: in discrete-action VLAs, trajectories with higher generation confidence are significantly more likely to succeed. Based on the observation, we introduce **T^{2}VLA** (**T**est-**t**ime **VLA**), an architecture-agnostic test-time RL framework that enables VLA models to achieve self-bootstrapping policy improvement. Instead of relying on external rewards, **T^{2}VLA** leverages the trajectory-level similarity to high-confidence expert demonstrations as an intrinsic reward signal. In addition, we propose a Confidence-Driven Dual Expert Bootstrapping mechanism. By dynamically balancing a Local Pseudo-Expert for aggressive exploration and a Global Expert Pool for training stability, **T^{2}VLA** prevents policy collapse while discovering further breakthroughs. Extensive experiments on the LIBERO and RoboTwin benchmarks show that **T^{2}VLA** consistently outperforms supervised baselines and approaches oracle RL performance with ground-truth rewards, achieving effective improvement without external reward feedback. Furthermore, **T^{2}VLA** can adapt to distinct VLA paradigms, including both OpenVLA-OFT and the π series.

Keywords: Vision-Language-Action Model · Test-Time Reinforcement Learning · Intrinsic Reward

1 Introduction

Driven by the rapid evolution of Vision-Language Models [10,17,24,32] (VLMs), Vision-Language-Action (VLA) models [3, 19, 48] have emerged as a transformative paradigm in embodied AI. Recently, Reinforcement Learning [13, 20, 27] (RL) has become a research hotspot for VLA, as it effectively addresses the prohibitive data collection costs and inherent generalization bottlenecks associated with traditional Supervised Fine-Tuning [4, 28, 39, 46] (SFT). Despite their

⋆ Corresponding author.

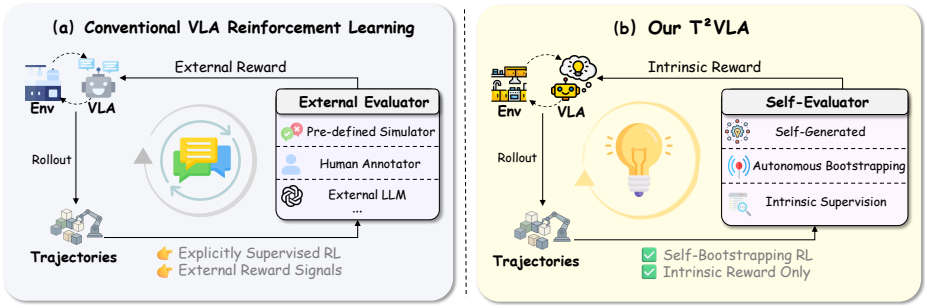

Fig. 1: Comparison of VLA Reinforcement Learning Paradigms. (a) Conventional VLA RL relies on explicitly supervised RL, where policy updates are driven by external reward signals from simulators, human annotators, or external LLMs. **(b) T^{2}VLA (Ours)** introduces a self-bootstrapping RL approach. By leveraging a model-internal signal, T^{2}VLA derives rewards to drive continuous policy adaptation without external reward supervision.

remarkable success in various embodied tasks, existing RL approaches heavily rely on external supervision signals (*e.g.*, environment-provided success flags), confining their efficacy to predefined scenarios.

To break free from these predefined scenarios and unlock true autonomous self-improvement, it is crucial to empower VLA models to learn directly from their own unannotated trajectories. This transition towards the "Era of Experience" [35] has recently been successfully validated in VLMs through test-time Reinforcement Learning [49] in mathematics and coding tasks, which enables models to self-optimize using purely unlabeled trajectories through majority voting. However, translating this success to VLA models exposes a critical gap: unlike math problems that yield a singular, discrete answer for easy verification, robotic manipulation requires generating continuous trajectories where multiple divergent action sequences can lead to the same goal, thereby eliminating a straightforward ground truth for self-reflection.

To bridge this gap, inspired by entropy-based confidence estimation in Large Language Models [11,31,36], we carefully investigate whether VLA models possess internal evaluative capabilities that can substitute for external verification. Our empirical analysis on discrete-action VLAs (Figure 3) reveals a strong positive correlation between internal generation confidence and task success rates (*i.e.*, trajectories with higher confidence are more likely to succeed). This observation indicates that, for token-based VLA policies, the length-normalized mean log-probability of generated trajectories can serve as a useful indicator of physical success, thereby enabling policy optimization without relying on external environmental rewards.

Building upon this empirical insight, we introduce **T^{2}VLA** (**T**est-**t**ime **VLA**), an architecture-agnostic framework for self-bootstrapping policy optimization, as illustrated in Figure 1. To eliminate reliance on extrinsic environmental rewards, we first propose a **Confidence-Driven Dual Expert Bootstrapping** mech-

anism, enabling the autonomous extraction of high-quality demonstrations directly from the model's own rollouts. Leveraging generative confidence, we elect a task-conditioned *Local Pseudo-Expert* to serve as an immediate optimization anchor, steering the policy toward the most promising behaviors discovered in the current iteration. Concurrently, we maintain a priority-based *Global Expert Pool* containing the top-K historical trajectories, which provides a stable behavioral baseline to prevent policy regression. Subsequently, we formulate a self-bootstrapping proxy reward based on behavioral alignment with these discovered experts. We employ Dynamic Time Warping (DTW) [33] to evaluate sequence similarity across continuous robotic actions with varying execution horizons. The final trajectory reward is computed via an adaptive weighting strategy that dynamically interpolates between local and global DTW similarities based on their relative confidence scores. Finally, the VLA policy is updated via Group Relative Policy Optimization (GRPO) [34], leveraging the group-normalized proxy rewards to drive continuous, self-guided learning.

Extensive experiments on multiple benchmarks including LIBERO [23] and RoboTwin 2.0 [9] demonstrate that **T^{2}VLA** can serve as an architecture-agnostic framework that yields substantial gains on various embodied tasks without relying on external rewards. Besides, **T^{2}VLA** is compatible with various VLA models such as OpenVLA-OFT [18, 20] and π_0 [3].

Our main contributions can be summarized as follows:

1. We propose **T^{2}VLA**, a self-bootstrapping test-time reinforcement learning framework that eliminates the requirement for external supervision signals. By leveraging the intrinsic correlation between VLA generation confidence and physical execution success, our approach enables self-bootstrapping policy optimization without environmental reward signals.
2. We introduce a confidence-driven dual-expert mechanism that synergizes a Local Pseudo-Expert for aggressive exploration with a Global Expert Pool for long-term stability. Coupled with a DTW-based adaptive reward, this provides a reliable soft anchor that prevents error compounding during unsupervised policy updates.
3. We demonstrate the architecture-agnostic efficacy of **T^{2}VLA** across diverse benchmarks. Relying solely on internal signals, our method achieves over a **20%** absolute gain on continuous-action and bimanual tasks, and elevates the average success rate of discrete-action models to **97.2%**.

2 Related Work

2.1 Vision-Language-Action Models

Vision-Language-Action (VLA) models unify perception, reasoning, and control by directly mapping multimodal visual observations and natural language instructions to physical actions [4, 48]. Current VLAs broadly follow two architectural paradigms based on their action representations. The first paradigm,

Discrete-Action VLAs (e.g., RT-1 [4], RT-2 [48], and OpenVLA [19]), leverages pre-trained vision-language models (VLMs) to formulate robotic control as an autoregressive token generation problem. The second paradigm, **Continuous-Action VLAs** (e.g., Octo [39], RDT [26], π_0 [3], $\pi_{0.5}$ [16], and GR00T [2]), synthesizes continuous control trajectories conditioned on multimodal embeddings via diffusion models or flow matching techniques. The standard training recipe for both classes of models is grounded in large-scale Supervised Fine-Tuning (SFT) or behavioral cloning [28].

2.2 Reinforcement Learning for Robotic Manipulation

Reinforcement Learning (RL) has been widely adopted to further optimize VLA policies beyond standard SFT. **Offline RL-VLA** approaches (e.g., GeRM [37], Q-Transformer [6], ReinboT [44], CO-RFT [15]) extract policies strictly from static datasets using pre-annotated rewards. To enable active environmental interaction, **Online RL-VLA** frameworks optimize policies via closed-loop feedback, extensively exploring policy gradient algorithms across diverse architectures (e.g., ThinkAct [14], RIPT-VLA [38], GRAPE [45], πRL [8], SimpleVLA-RL [20]). In these online methods, reward acquisition is primarily based on explicitly designed external supervision, such as sparse binary feedback from predefined simulators [8, 20] or dense proxy rewards engineered via auxiliary models [43]. Instead of requiring external reward signals, our proposed **T^{2}VLA** framework achieves self-bootstrapped RL by relying entirely on intrinsic rewards.

2.3 Test-Time Learning and Adaptation

Test-time optimization strategies for robotic manipulation broadly fall into two paradigms: learning and adaptation. **Test-Time Learning** [7, 40, 42] dynamically updates model representations or weights during inference. For example, VLS [25] and ADPro [21] optimize inference-time sampling via VLM-derived gradients and geometric constraints, while EVOLVE-VLA [1] continuously updates policy weights using learned progress estimators. Conversely, **Test-Time Adaptation** employs training-free mechanisms to guide pre-trained policies, typically through action re-ranking via external verifiers (e.g., V-GPS [29], VLA-Pilot [22]), MCTS-based planning (e.g., VLA-Reasoner [12], VLAPS [30]), or heuristic action filtering (e.g., TACO [41]). While these pipelines predominantly depend on auxiliary external evaluators, our **T^{2}VLA** framework performs test-time RL using intrinsic rewards derived from model-internal signals.

3 Method

In this section, we present **T^{2}VLA**, a self-bootstrapping test-time RL framework for Vision-Language-Action (VLA) models (Figure 2). Following the problem formulation and our empirical motivation (Section 3.1), we introduce a

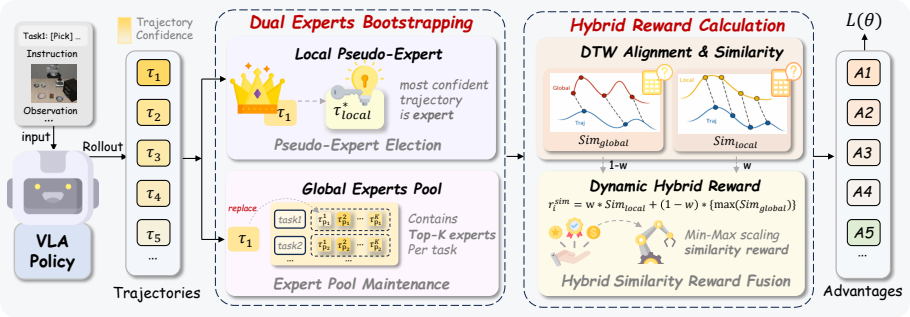

Fig. 2: Overview of the T^{2}VLA Framework. Our pipeline autonomously mines behavioral anchors by identifying a **Local Pseudo-Expert** from exploratory rollouts and maintaining a **Global Expert Pool**. These references are integrated via a **DTW-based Hybrid Similarity Reward** to compute advantages, enabling continuous policy optimization without external reward signals.

confidence-driven dual-expert mechanism to autonomously mine behavioral anchors directly from exploratory rollouts (Section 3.2). These anchors are subsequently translated into intrinsic signals via a DTW-based hybrid similarity reward (Section 3.3) to drive continuous policy optimization (Section 3.4).

3.1 Preliminaries

Problem Formulation. Given an embodied task, the robotic manipulation process can be formulated as a Markov Decision Process (MDP). At each timestep t, the VLA model observes a state $s_t = (o_t, l)$, comprising a visual observation o_t and a language instruction l, and generates an action a_t according to a parameterized policy $\pi_\theta(a_t|s_t)$. For discrete-action VLAs, the model outputs logits over discretized action tokens that are decoded into executable actions; for flow-based VLAs, actions are generated through iterative continuous refinement.

In standard reinforcement learning for VLAs, the model interacts with the environment to collect a rollout trajectory $\tau = (s_1, a_1, \ldots, s_T, a_T)$ of length T. Policy optimization proceeds by sampling a batch of rollouts $\{\tau_i\}_{i=1}^N$, assigning each trajectory a scalar reward r_i, estimating advantages, and updating θ via a policy-gradient objective. The standard formulation maximizes the expected external return:

$$\max_\theta \ \mathbb{E}_{\tau\sim\pi_\theta}\left[R_{\text{env}}(\tau)\right], \tag{1}$$

where $R_{\text{env}}(\tau)$ denotes the external environment reward. Consequently, existing RL-based VLA optimization methods inherently rely on explicit external verification signals to guide policy updates.

Empirical Motivation and Reformulation. To eliminate the reliance on external environmental reward $R_{\text{env}}(\tau)$, we investigate whether VLA models

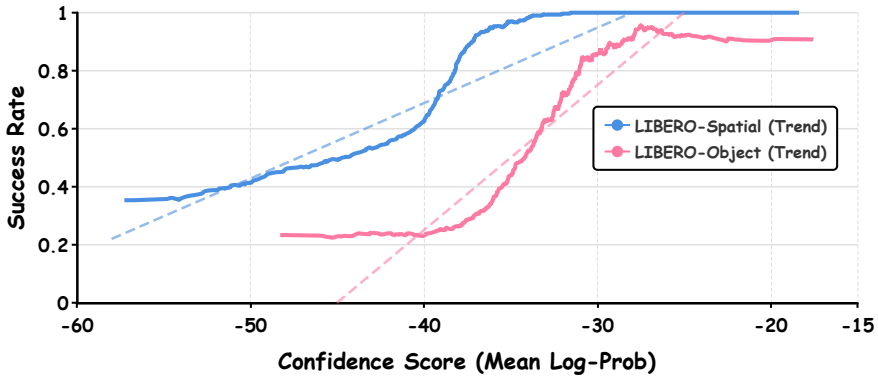

Fig. 3: Correlation between VLA generation confidence and task success rate. Analysis of 2,000 OpenVLA-OFT rollouts on LIBERO suites reveals that higher mean log-probabilities consistently correlate with execution success, validating internal confidence as a reliable foundation for autonomous self-bootstrapping.

can inherently evaluate their physical execution. As shown in Figure 3, empirical analysis of discrete-action VLAs reveals a strong positive correlation between trajectory-level generation confidence (defined as the mean action log-probability) and execution success. Trajectories assigned higher likelihood by the model consistently achieve higher task success rates. This observation indicates that the model's intrinsic confidence provides a robust signal regarding behavioral quality, suggesting its potential as a foundation for self-bootstrapping.

In light of this observation, we propose **T^{2}VLA**, which optimizes the VLA policy as a **self-bootstrapping reinforcement learning process** during test time. Instead of relying on external environment rewards, we replace the return with an intrinsic reward $R_{\text{self}}(\tau)$ computed solely from signals produced by the model itself. Consequently, the optimization objective transitions to:

$$\max_{\theta} \mathbb{E}_{\tau \sim \pi_\theta} \left[R_{\text{self}}(\tau) \right]. \tag{2}$$

This paradigm shift introduces a primary challenge: how to formulate an effective trajectory-level intrinsic reward $R_{\text{self}}(\tau)$ without external verification signals. While empirical evidence links confidence with task success, adopting the raw generation confidence directly as a standalone reward scalar proves inadequate for stable policy optimization. This prompts a critical question: could these high-confidence trajectories themselves serve as reliable anchors to provide effective guidance signals? To answer this, we decouple the overall challenge into two fundamental sub-problems: (i) How to reliably extract and maintain high-quality reference trajectories from exploratory rollouts. (ii) How to formulate a robust trajectory reward design based on these established references.

3.2 Confidence-Driven Dual Expert Bootstrapping

To extract high-quality reference trajectories without external verification, we propose a **Confidence-Driven Dual Expert Bootstrapping** mechanism. This module is designed to autonomously mine and maintain behavioral anchors from the model's exploratory rollouts by leveraging internal confidence signals. Specifically, it consists of two primary components: the election of a task-conditioned *Local Pseudo-Expert* from the current batch, and the maintenance of a dynamic *Global Expert Pool* to preserve historical best practices.

Length-Normalized Confidence Estimation. During exploration, the active policy π_θ generates a batch of N trajectories $\mathcal{D} = \{\tau_1, \ldots, \tau_N\}$. For discrete-action VLAs, directly summing action log-probabilities inherently penalizes long-horizon trajectories and may favor degenerate, short-horizon behaviors. We therefore compute a length-normalized confidence score. For a trajectory τ_i with an effective execution horizon T_i, determined by environment termination and strictly excluding padding tokens, its confidence score is defined as the mean action log-probability:

$$c_i^{\text{disc}} = \frac{1}{T_i} \sum_{t=1}^{T_i} \log \pi_\theta(a_{i,t} \mid s_{i,t}). \tag{3}$$

This normalization enables confidence scores to be compared across trajectories with different execution horizons.

For continuous flow-based VLAs, confidence is estimated from the Gaussian likelihood of transitions along the denoising process. Let L_i denote the effective rollout length, measured by the number of valid action-chunk predictions before environment termination. At each prediction step j, we compute

$$\ell_{i,j}^{\text{flow}} = \sum_{h,d} \log \mathcal{N}\left(x_{i,j,h,d}^{k+1}; \mu_\theta(x_{i,j}^k, k, s_{i,j})_{h,d}, \sigma_\theta(x_{i,j}^k, k, s_{i,j})_{h,d}\right), \tag{4}$$

where $x_{i,j}^k$ is the intermediate denoising state at the selected denoising step k, while h and d index the action-chunk horizon and action dimension, respectively. The trajectory-level confidence is then computed as

$$c_i^{\text{flow}} = \frac{1}{L_i} \sum_{j=1}^{L_i} \ell_{i,j}^{\text{flow}}. \tag{5}$$

This likelihood characterizes denoising consistency and provides an effective ranking signal for pseudo-expert election. The elected trajectories subsequently serve as behavioral references for computing intrinsic rewards.

For simplicity, we use c_i to denote the corresponding trajectory-level confidence score, either c_i^{disc} or c_i^{flow}, in the following expert election process.

Task-Conditioned Local Expert Election. Since generative confidence strongly correlates with execution success (Section 3.1), the confidence score c_i provides a reliable basis for identifying trajectories that are more likely to achieve

the task. By leveraging c_i to anchor the most confident executions from the exploratory rollouts, we can extract behavioral references to drive iterative policy improvement. To steer suboptimal exploratory behaviors toward these successful modes, we formally designate the most confident trajectory as a pseudo-expert. However, during the RL training loop, a sampled batch typically encompasses trajectories across diverse environments and manipulation tasks. To ensure a fair and task-specific evaluation, we first group the trajectories based on their corresponding language instruction l. Within each task-specific subset $\mathcal{D}_l \subseteq \mathcal{D}$, the *Local Pseudo-Expert* is explicitly elected as:

$$\tau^*_{local,l} = \arg\max_{\tau_i \in \mathcal{D}_l} c_i \tag{6}$$

Thus, this local expert encapsulates the most reliable behavioral mode discovered for the specific task during the current iteration, serving as a dynamic, on-policy target for immediate alignment.

Dynamic Global Expert Pool. Relying exclusively on the local batch expert can occasionally introduce optimization instability, particularly if an exploratory batch happens to yield overall suboptimal rollouts. To maintain training stability and preserve previously discovered high-quality behaviors, we concurrently maintain a task-conditioned dynamic *Global Expert Pool*, denoted as $\mathcal{P}_l$. To ensure the high quality of this historical knowledge base, we restrict the pool's entry strictly to the elected local batch experts $\tau^*_{local,l}$, filtering out the remaining suboptimal exploratory trajectories.

At each training iteration, the newly elected local expert $\tau^*_{local,l}$ is integrated into $\mathcal{P}_l$. The pool acts as a priority-based memory buffer, retaining only the top-K historical experts (with capacity $K = 5$ in our implementation) sorted by their confidence scores c:

$$\mathcal{P}_l \leftarrow \text{Top-}K\left(\mathcal{P}_l \cup \{\tau^*_{local,l}\}\right) \text{ based on } c \tag{7}$$

By bounding the capacity, this mechanism evicts stale behaviors from outdated policy, ensuring the pool remains aligned with the evolving policy distribution.

3.3 DTW-Based Hybrid Similarity Reward

To address the second sub-problem of rewarding exploratory rollouts, we design a **DTW-based Hybrid Similarity Reward**. This mechanism is designed to translate expert guidance into informative signals by aligning trajectories and balancing expertise sources. Specifically, it is composed of a DTW-based trajectory alignment to handle temporal variations, and an adaptive weighting scheme to balance current batch consensus with historical best practices.

Trajectory Alignment via DTW. The stochastic generative nature of VLAs frequently yields action sequences with varying temporal horizons. Standard Euclidean distance enforces strict one-to-one temporal synchronization, causing massive distance penalties under minor temporal shifts even when spatial paths are identical. As visualized in Figure 5, such rigid point-wise matching

fails to capture true structural similarity. To decouple spatial geometry from temporal progression, we employ Dynamic Time Warping (DTW) [33]. By non-linearly warping the time axis to enable many-to-one state mappings, DTW aligns trajectories based on their spatial morphology rather than strict temporal indices, providing a robust similarity measure for heterogeneous rollouts.

Let $\tau_A = (a_{A,1}, \ldots, a_{A,M})$ and $\tau_B = (a_{B,1}, \ldots, a_{B,T})$ denote two action sequences of continuous end-effector poses $a \in \mathbb{R}^d$. To prevent dimensions with large physical magnitudes from dominating the metric, all action dimensions are independently min-max normalized to $[0, 1]$. We construct a cost matrix $D \in \mathbb{R}^{M \times T}$ via dynamic programming:

$$d_{i,j} = \|a_{A,i} - a_{B,j}\|_2 + \min(d_{i-1,j}, d_{i,j-1}, d_{i-1,j-1}) \tag{8}$$

The cumulative distance $D_{\mathrm{DTW}}(\tau_A, \tau_B) = d_{M,T}$ is then normalized by the maximum temporal length to ensure length-invariance and yield a bounded similarity score in $(0, 1]$:

$$\mathrm{Sim}_{\mathrm{DTW}}(\tau_A, \tau_B) = \frac{1}{1 + \frac{D_{\mathrm{DTW}}(\tau_A, \tau_B)}{\max(M,T)}} \tag{9}$$

Dynamic Hybrid Reward Design. To adaptively determine the trust placed in the local batch expert, we compute the alignment of each exploratory trajectory τ_i (corresponding to task l) against both its task-specific local expert $\tau^*_{local,l}$ and the global pool $\mathcal{P}_l$. The dynamic hybrid similarity reward r_i^{sim} incorporates the maximum similarity score among all historical experts $\tau_p \in \mathcal{P}_l$:

$$r_i^{sim} = w \cdot \mathrm{Sim}_{\mathrm{DTW}}(\tau_i, \tau^*_{local,l}) + (1 - w) \cdot \max_{\tau_p \in \mathcal{P}_l} \mathrm{Sim}_{\mathrm{DTW}}(\tau_i, \tau_p) \tag{10}$$

The interpolation weight $w \in [0, 1]$ balances current batch consensus and historical best practices by normalizing the local expert's confidence score $c^*_{local,l}$:

$$w = \mathrm{clip}\left(\frac{c^*_{local,l} - c_{min,l}}{c_{max,l} - c_{min,l} + \epsilon}, 0, 1\right) \tag{11}$$

where $c_{max,l}$ and $c_{min,l}$ represent the maximum and minimum confidence scores currently maintained within the task-specific pool $\mathcal{P}_l$, and ϵ ensures numerical stability. This relative measure ensures that highly confident local experts ($w \to 1$) prioritize immediate on-policy discoveries, while uncertain batches ($w \to 0$) smoothly fall back to historical experts to prevent policy degradation. A detailed analysis validating the necessity of this continuous Min-Max scaling over rigid or hard-gating alternatives is provided in Sec. 4.4.

KL Penalty. To prevent over-optimization and maintain the valid behavioral distribution learned during pretraining, we incorporate a Kullback-Leibler (KL) divergence penalty. The final proxy reward r_i for trajectory τ_i is:

$$r_i = r_i^{sim} - \beta \sum_{t=1}^{T_i} \mathbb{D}_{\mathrm{KL}}\left(\pi_\theta(\cdot|s_{i,t}) \| \pi_{\mathrm{ref}}(\cdot|s_{i,t})\right) \tag{12}$$

where the initial SFT model π_{ref} acts as a behavioral anchor, and β controls the penalty strength.

3.4 Policy Optimization via GRPO

Given the proxy rewards $\{r_1, \ldots, r_N\}$ for the N sampled trajectories, GRPO computes a baseline-free advantage A_i via group normalization:

$$A_i = \frac{r_i - \mu(r)}{\sigma(r) + \epsilon} \tag{13}$$

where $\mu(r)$ and $\sigma(r)$ denote the batch mean and standard deviation. The policy is then updated by maximizing the clipped surrogate objective:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T_i} \min \left(\rho_{i,t}(\theta) A_i, \text{clip}(\rho_{i,t}(\theta), 1 - \epsilon_{clip}, 1 + \epsilon_{clip}) A_i \right) \tag{14}$$

where $\rho_{i,t}(\theta) = \frac{\pi_\theta(a_{i,t}|s_{i,t})}{\pi_{\text{old}}(a_{i,t}|s_{i,t})}$ and ϵ_{clip} bounds the policy update. By iteratively maximizing $\mathcal{L}(\theta)$, the policy π_θ is continuously aligned with the extracted expert guidance, completing the self-bootstrapping loop.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate our framework on two robotic manipulation benchmarks: (1) **LIBERO** [23]: We evaluate the policy's robustness and long-horizon planning capacity across four diverse task suites (Spatial, Object, Goal, and Long). (2) **RoboTwin 2.0** [9]: We assess bimanual coordination across five tasks with varying execution horizons (100-650 steps).

Baseline Models. To demonstrate the architecture-agnostic applicability of **T^{2}VLA**, we instantiate our framework across two generative paradigms: (1) **Discrete-Action VLAs**: We adopt OpenVLA-OFT [18]. We use the modified architecture and SFT weights provided by SimpleVLA-RL [20], which optimizes the decoding head for RL compatibility. (2) **Continuous-Action VLAs**: We evaluate flow-matching policies π_0 [3] and $\pi_{0.5}$ [16] by adopting π_{RL}'s [8] SFT weights and MDP to compute confidence from denoising log-likelihoods.

Implementation Details. We optimize **T^{2}VLA** using GRPO [34] with AdamW (peak LR 5×10^{-6}; cosine schedule for OpenVLA-OFT, constant for $\pi_0/\pi_{0.5}$). We sample $N = 8$ trajectories per instruction at temperature 1.6 (OpenVLA-OFT) or 1.0 ($\pi_0/\pi_{0.5}$). Policy updates apply a universal initial KL penalty $\beta = 0.02$. The PPO clip ratio is $[0.2, 0.28]$ for OpenVLA-OFT, and 0.2 for $\pi_0/\pi_{0.5}$ (with gradient clipping at 2.0 and 1.0, respectively). Execution horizons are capped at 500 steps (OpenVLA-OFT) and 480 steps ($\pi_0/\pi_{0.5}$, max 1024 tokens), with $\pi_{0.5}$ using 4 denoising steps.

4.2 Main Results

In this section, we present the main evaluation results on the LIBERO and RoboTwin benchmarks. To verify the effectiveness of our method, we compare

Table 1: Main Results on the LIBERO Benchmark. Success rates (%) are reported. Δ indicates the absolute improvement over the baseline.

Method	Reward	Spatial	Object	Goal	Long	Avg.
Reference: Prior SFT, RL, & Test-Time Training Baselines						
Octo [39]	None (SFT)	78.9	84.6	85.7	51.1	75.1
OpenVLA [19]	None (SFT)	84.7	79.2	88.4	53.7	76.5
UniVLA [5]	None (SFT)	96.5	96.8	95.6	92.0	95.2
VLA-RL [27]	Env. Success	90.2	94.3	91.8	82.2	89.6
SimpleVLA-RL [20]	Env. Success	99.4	99.1	99.2	98.5	99.1
EVOLVE-VLA [1]	Learned Critic	95.4	97.4	95.8	94.4	95.8
Our Framework (Self-Rewarding)						
OpenVLA-OFT [18, 20]	None (SFT)	91.6	95.3	90.6	86.5	91.0
Ours (OpenVLA-OFT)	**Self-Reward**	**97.7**	**99.6**	**96.1**	**95.3**	**97.2**
	Improvement (Δ)	*+6.1*	*+4.3*	*+5.5*	*+8.8*	*+6.2*
π_0 (Base) [3, 8]	None (SFT)	65.3	64.4	49.8	51.2	57.7
Ours (π_0)	**Self-Reward**	**86.3**	**91.0**	**82.0**	**68.0**	**81.9**
	Improvement (Δ)	*+21.0*	*+26.6*	*+32.2*	*+16.8*	*+24.2*
$\pi_{0.5}$ (Base) [8, 16]	None (SFT)	84.6	95.4	84.6	43.9	77.1
Ours ($\pi_{0.5}$)	**Self-Reward**	**94.9**	**98.4**	**91.8**	**55.1**	**85.1**
	Improvement (Δ)	*+10.3*	*+3.0*	*+7.2*	*+11.2*	*+8.0*

T^{2}VLA against a diverse spectrum of baselines, including prior SFT methods (*i.e.*, UniVLA [5]), explicitly supervised RL methods (*i.e.*, VLA-RL [27] and SimpleVLA-RL [20]), and test-time training (TTT) approaches (*i.e.*, EVOLVE-VLA [1]). Further, to demonstrate the advantage of our self-bootstrapping mechanism, we directly compare the performance of our method against its respective base policies under standard data regimes: OpenVLA-OFT is fine-tuned with full demonstrations (Traj-all), while both π_0 and $\pi_{0.5}$ utilize few-shot initialization. Table 1 details our absolute performance gains on LIBERO, categorized by reliance on external rewards. Additionally, Table 2 reports the framework's effectiveness in bimanual control scenarios across varying execution horizons.

Results on the LIBERO Benchmark. Table 1 demonstrates that **T^{2}VLA** (based on OpenVLA-OFT) delivers consistent gains across all four task suites, improving upon the OpenVLA-OFT baseline by +6.2% on average and outperforming the state-of-the-art SFT model, UniVLA [5] (97.2% vs. 95.2%). Notably, despite operating entirely without external reward models or environmental supervision, **T^{2}VLA** surpasses the concurrent EVOLVE-VLA [1] (95.8%), which relies on a separate foundation critic. Moreover, it substantially reduces the performance gap to the oracle-guided SimpleVLA-RL [20] (99.1%), which has access to true environmental feedback. These results further demonstrate that intrinsic

Table 2: Results on RoboTwin 2.0. Success rates (%) using OpenVLA-OFT. Tasks are categorized by execution horizon length.

Task Category & Name	SFT Baseline	Ours	Improvement (Δ)
Short Horizon (100-130 Steps)			
Lift Pot	10.1	**39.8**	*+29.7*
Beat Hammer	28.1	**68.0**	*+39.9*
Medium Horizon (150-230 Steps)			
Place Empty Cup	77.3	**84.8**	*+7.5*
Long & Extra Long Horizon (280-650 Steps)			
Handover Block	33.1	**44.1**	*+11.0*
Stack Bowls	40.6	**59.0**	*+18.4*
Average	37.8	**59.1**	***+21.3***

generative log-probabilities can provide a reliable bootstrapping signal that can boost the performance on different embodied tasks.

In addition, to verify the broad applicability of **T^{2}VLA**, we extend our evaluation to continuous flow-matching models. As detailed in Table 1, our approach yields a substantial 24.2% absolute improvement in the average success rate of π_0 [3, 8]. Additionally, it boosts the performance of $\pi_{0.5}$ [8, 16] on the Goal suite from 84.6% to 91.8%. These results confirm that **T^{2}VLA** is an architecture-agnostic approach that can be flexibly combined with various VLA models.

Results on the Bimanual Scenario (RoboTwin 2.0). To validate the versatility of **T^{2}VLA** in adapting to different tasks, we further conduct experiments on the RoboTwin 2.0 benchmark. As shown in Table 2, **T^{2}VLA** consistently outperforms the SFT baseline across diverse execution horizons. Specifically, it achieves a remarkable performance leap on short-horizon tasks like *Beat Hammer* (*i.e.*, from 28.1% to 68.0%). For complex long-horizon tasks such as *Stack Bowls*, it can also achieve an 18.4% absolute improvement. This demonstrates that the DTW-based self-bootstrapping mechanism scales effectively to high-dimensional control scenarios without environment-provided rewards.

4.3 Ablation Studies

To investigate the impact of our core designs, we conduct ablation studies on the LIBERO-Long benchmark using OpenVLA-OFT [18, 20].

Effectiveness of Dual Expert Bootstrapping. We first investigate the role of the proposed dual-expert bootstrapping mechanism. As detailed in Table 3, relying exclusively on the local expert captures recent high-confidence behaviors (94.5%) but remains vulnerable to occasional suboptimal generations within a single batch. Conversely, utilizing only the global pool provides stable historical references (93.0%) but slows down continuous adaptation, as the optimization tends to rely heavily on older trajectories. Integrating both modules effectively balances these aspects, achieving the highest success rate (95.3%).

Table 3: Dual Expert Ablation. Success rates (%) on LIBERO-Long. The dual-expert design achieves the best performance.

Method	SR (%)
Local Expert Only	94.5
Global Expert Only	93.0
Dual Expert (Ours)	**95.3**

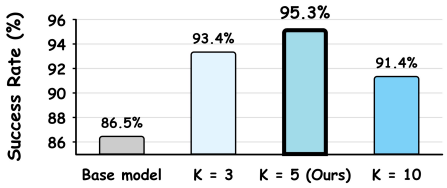

Fig. 4: Impact of Expert Pool Capacity (K). Evaluated on LIBERO-Long, where $K = 5$ yields the best performance.

Overall, these results indicate that the local and global experts are highly complementary: the former steers the policy toward newly discovered successful modes, while the latter serves as a stabilizing anchor to prevent policy degradation when current rollouts are poor. Their synergy yields a balanced, progressive reward signal that drives consistent performance improvements.

Impact of Expert Pool Capacity. We further analyze the sensitivity to the global expert pool capacity (K). As illustrated in Figure 4, all evaluated pool capacities consistently exceed the performance of the Base model (86.5%), validating the general effectiveness of our self-bootstrapping framework. Notably, performance peaks at a moderate capacity ($K = 5$, 95.3%), suggesting a trade-off between reference diversity and quality control. A restricted pool ($K = 3$, 93.4%) offers insufficient coverage of diverse successful behavioral modes, potentially limiting the richness of the supervision signal. On the other hand, an overly expansive pool ($K = 10$, 91.4%) retains stale trajectories from earlier, less-proficient training stages. The persistence of these outdated references can introduce suboptimal anchors into the reward calculation, which dilute the quality of the guidance signal and impede learning progress. Thus, $K = 5$ provides an effective balance, maintaining sufficient historical diversity while ensuring the pool remains aligned with high-quality behavioral standards.

4.4 Insight Analysis

Why Dynamic Time Warping (DTW) for Trajectory Similarity? We empirically justify selecting DTW over Euclidean distance for trajectory alignment. In continuous control, successful rollouts frequently exhibit temporal heterogeneities, such as pacing variations or phase shifts, compared to the reference expert. As visualized in Figure 5, projecting in-situ 3D trajectories onto their principal action axis reveals that Euclidean distance enforces rigid point-wise alignments. This causes residual errors to explode under minor temporal shifts, yielding an artificially depressed similarity score (0.6914). Conversely, DTW computes an optimal non-linear sequence alignment. By mapping corresponding geometric configurations across flexible time intervals, DTW neutralizes temporal misalignments and recovers a robust spatial congruence score (0.9460). This

Table 4: Comparison of Advantage Shaping Strategies. Evaluated on LIBERO-Long. † indicates severe policy collapse at early stages (< 100 epochs).

Expert Fusion Strategy	SR (%)	Expert Fusion Strategy	SR (%)
Static Weighting ($w = 0.5$)	88.3†	Asymmetric Z-Score Gate	91.4
Max Routing	87.5†	Sigmoid Gate	92.6
Adaptive Margin Fallback (AMF)	90.0	**Min-Max Normalization (Ours)**	**95.3**

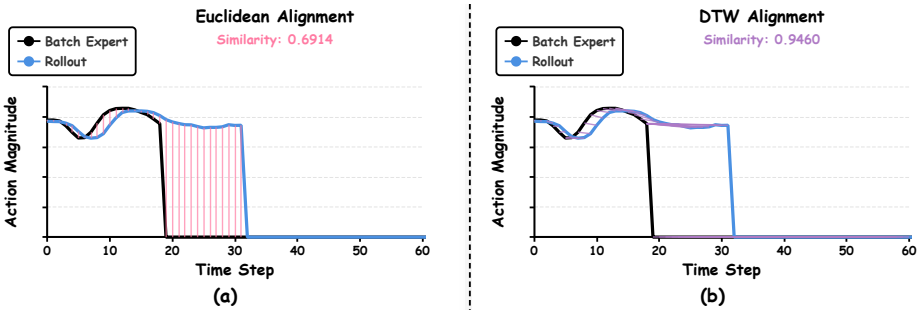

Fig. 5: Trajectory alignment comparison. In-situ training trajectories projected onto the principal action axis. (a) Rigid Euclidean matching yields high residual errors under temporal shifts. (b) DTW dynamically warps the time axis to map structurally similar states, recovering a robust spatial similarity measure.

geometric-aware matching prevents high-quality exploratory rollouts from being assigned spuriously low rewards.

The Necessity of Smooth Expert Fusion. Dynamically balancing the fusion weight ($w \in [0, 1]$) between the local batch and the global pool is essential for training stability. Table 4 demonstrates that rigid assignments ($w = 0.5$) or hard switching mechanisms (Max Routing) cause early policy collapse (< 100 epochs) due to abrupt gradient shifts. While hard-gating approaches like Adaptive Margin Fallback (AMF) prevent this collapse, they lack fine-grained adaptation. Soft-gating strategies, such as the Temperature-scaled Sigmoid gate, offer smoother adjustments and successfully stabilize learning (92.6%). Ultimately, our proposed Min-Max Confidence Scaling yields the best performance (95.3%) by mapping non-stationary log-probabilities into a strictly bounded, continuous scale which ensures an adaptive and smooth integration of both experts.

Analysis of the Bootstrapping Threshold (1-Shot Data Scarcity). To investigate the boundary conditions of the self-bootstrapping framework and stress-test its capability to bootstrap from minimal supervision, we evaluate OpenVLA-OFT under an extreme 1-shot data scarcity setting across all LIBERO suites. As shown in Table 5, our method effectively improves the 1-shot SFT baseline on the Spatial, Object, and Goal suites, yielding approximately 20% absolute gains. However, performance degrades on the Long-horizon suite (from 17.3% to 11.0%). This indicates a critical "prior threshold" for autonomous bootstrap-

Table 5: Applicability under Challenging Settings. Success rates (%) under limited demonstrations and world-model-generated interactions. The degradation on the Long suite highlights the minimum prior required for successful self-bootstrapping.

Task	**Base**	**Ours**	Δ
1-Shot SFT			
LIBERO-Spatial	63.6	**84.0**	*+20.4*
LIBERO-Object	54.9	**74.6**	*+19.7*
LIBERO-Goal	59.6	**83.6**	*+24.0*
LIBERO-Long	**17.3**	11.0	*-6.3*
World-Model Rollouts			
OpenSora-Spatial	61.2	**63.3**	*+2.1*

ping: for complex tasks, an excessively weak initialization causes exploratory rollouts to deviate irrecoverably from the expert demonstration. Consequently, the similarity reward degenerates into uninformative noise, and without meaningful gradient guidance, the policy optimization collapses into random drift.

Applicability to World-Model-Generated Interactions. To evaluate the applicability of **T^{2}VLA** beyond direct interaction with standard simulators, we conduct policy optimization using action-conditioned observations synthesized by an OpenSora world model [47]. The resulting OpenVLA-OFT policy is evaluated in the original LIBERO-Spatial simulator. As shown in Table 5, our method improves the success rate from 61.2% to 63.3%. This demonstrates that **T^{2}VLA** remains applicable when its exploratory interactions are generated by a learned world model.

5 Conclusion

In this work, we explore the self-bootstrapping optimization of Vision-Language-Action (VLA) models without relying on external rewards and propose **T^{2}VLA**, an architecture-agnostic framework for autonomous policy evolution. By autonomously mining high-quality demonstrations through the proposed confidence-driven dual expert bootstrapping mechanism and constructing a dynamic DTW-based hybrid similarity reward, **T^{2}VLA** enables VLA models to continuously refine their execution using intrinsic signals. Our **T^{2}VLA** establishes a scalable, intrinsic-reward-driven paradigm, demonstrating that a model's internal signals can effectively bootstrap continuous and autonomous policy improvement.

Acknowledgements

This work is supported by National Key R&D Program of China (No. 2026YFE 0101200). It is also supported by Shanghai Natural Science Foundation (No. 23ZR 1402900). The computations in this research were performed using the CFFF platform of Fudan University.

References

1. Bai, Z., Gao, C., Shou, M.Z.: Evolve-vla: Test-time training from environment feedback for vision-language-action models. arXiv preprint arXiv:2512.14666 (2025)
2. 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)
3. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: π_0: A Vision-Language-Action Flow Model for General Robot Control. arXiv preprint arXiv:2410.24164 (2024)
4. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817 (2022)
5. Bu, Q., Yang, Y., Cai, J., Gao, S., Ren, G., Yao, M., Luo, P., Li, H.: Univla: Learning to act anywhere with task-centric latent actions. arXiv preprint arXiv:2505.06111 (2025)
6. Chebotar, Y., Vuong, Q., Hausman, K., Xia, F., Lu, Y., Irpan, A., Kumar, A., Yu, T., Herzog, A., Pertsch, K., et al.: Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions. In: Conference on Robot Learning. pp. 3909–3928. PMLR (2023)
7. Chen, D., Wang, D., Darrell, T., Ebrahimi, S.: Contrastive test-time adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 295–305 (2022)
8. Chen, K., Liu, Z., Zhang, T., Guo, Z., Xu, S., Lin, H., Zang, H., Li, X., Zhang, Q., Yu, Z., et al.: π_{RL}: Online RL Fine-tuning for Flow-based Vision-Language-Action Models. arXiv preprint arXiv:2510.25889 (2025)
9. Chen, T., Chen, Z., Chen, B., Cai, Z., Liu, Y., Li, Z., Liang, Q., Lin, X., Ge, Y., Gu, Z., et al.: Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. arXiv preprint arXiv:2506.18088 (2025)
10. Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: Palm-e: An embodied multimodal language model. arXiv preprint arXiv:2303.03378 (2023)
11. Geng, J., Cai, F., Wang, Y., Koeppl, H., Nakov, P., Gurevych, I.: A survey of confidence estimation and calibration in large language models. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). pp. 6577–6595 (2024)
12. Guo, W., Lu, G., Deng, H., Wu, Z., Tang, Y., Wang, Z.: Vla-reasoner: Empowering vision-language-action models with reasoning via online monte carlo tree search. arXiv preprint arXiv:2509.22643 (2025)
13. Guo, Y., Zhang, J., Chen, X., Ji, X., Wang, Y.J., Hu, Y., Chen, J.: Improving vision-language-action model with online reinforcement learning. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 15665–15672. IEEE (2025)
14. Huang, C.P., Wu, Y.H., Chen, M.H., Wang, Y.C.F., Yang, F.E.: Thinkact: Vision-language-action reasoning via reinforced visual latent planning. arXiv preprint arXiv:2507.16815 (2025)
15. Huang, D., Fang, Z., Zhang, T., Li, Y., Zhao, L., Xia, C.: Co-rft: Efficient fine-tuning of vision-language-action models through chunked offline reinforcement learning. arXiv preprint arXiv:2508.02219 (2025)

16. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., et al.: $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. arXiv preprint arXiv:2504.16054 (2025)
17. Karamcheti, S., Nair, S., Balakrishna, A., Liang, P., Kollar, T., Sadigh, D.: Prismatic vlms: Investigating the design space of visually-conditioned language models. In: Forty-first International Conference on Machine Learning (2024)
18. Kim, M.J., Finn, C., Liang, P.: Fine-tuning vision-language-action models: Optimizing speed and success. arXiv preprint arXiv:2502.19645 (2025)
19. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
20. Li, H., Zuo, Y., Yu, J., Zhang, Y., Yang, Z., Zhang, K., Zhu, X., Zhang, Y., Chen, T., Cui, G., et al.: Simplevla-rl: Scaling vla training via reinforcement learning. arXiv preprint arXiv:2509.09674 (2025)
21. Li, Z., Yang, R., Chen, R., Luo, Z., Chen, L.: Adpro: a test-time adaptive diffusion policy via manifold-constrained denoising and task-aware initialization for robotic manipulation. arXiv preprint arXiv:2508.06266 (2025)
22. Li, Z., Liu, J., Dong, Z., Teng, T., Rouxel, Q., Caldwell, D., Chen, F.: Towards deploying vla without fine-tuning: Plug-and-play inference-time vla policy steering via embodied evolutionary diffusion. arXiv preprint arXiv:2511.14178 (2025)
23. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems **36**, 44776–44791 (2023)
24. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
25. Liu, S., Singh, I.S., Xu, Y., Duan, J., Krishna, R.: Vls: Steering pretrained robot policies via vision-language models. arXiv preprint arXiv:2602.03973 (2026)
26. Liu, S., Wu, L., Li, B., Tan, H., Chen, H., Wang, Z., Xu, K., Su, H., Zhu, J.: Rdt-1b: a diffusion foundation model for bimanual manipulation. arXiv preprint arXiv:2410.07864 (2024)
27. Lu, G., Guo, W., Zhang, C., Zhou, Y., Jiang, H., Gao, Z., Tang, Y., Wang, Z.: Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. arXiv preprint arXiv:2505.18719 (2025)
28. Mandlekar, A., Xu, D., Wong, J., Nasiriany, S., Wang, C., Kulkarni, R., Fei-Fei, L., Savarese, S., Zhu, Y., Martín-Martín, R.: What matters in learning from offline human demonstrations for robot manipulation. arXiv preprint arXiv:2108.03298 (2021)
29. Nakamoto, M., Mees, O., Kumar, A., Levine, S.: Steering your generalists: Improving robotic foundation models via value guidance. arXiv preprint arXiv:2410.13816 (2024)
30. Neary, C., Younis, O.G., Kuramshin, A., Aslan, O., Berseth, G.: Improving pretrained vision-language-action policies with model-based search. arXiv preprint arXiv:2508.12211 (2025)
31. Nguyen, D., Payani, A., Mirzasoleiman, B.: Beyond semantic entropy: Boosting llm uncertainty quantification with pairwise semantic similarity. In: Findings of the Association for Computational Linguistics: ACL 2025. pp. 4530–4540 (2025)
32. Peng, T., Li, M., Yuan, J., Zhou, H., Xia, R., Zhang, R., Bai, L., Mao, S., Wang, B., Zhou, A., et al.: Chimera: Improving generalist model with domain-specific experts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3011–3022 (2025)

33. Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., Keogh, E.: Searching and mining trillions of time series subsequences under dynamic time warping. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 262–270 (2012)
34. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
35. Silver, D., Sutton, R.S.: Welcome to the era of experience. Google AI **1**, 11 (2025)
36. Song, H., Ji, R., Shi, N., Lai, F., Kontar, R.A.: Inv-entropy: A fully probabilistic framework for uncertainty quantification in language models. arXiv preprint arXiv:2506.09684 (2025)
37. Song, W., Zhao, H., Ding, P., Cui, C., Lyu, S., Fan, Y., Wang, D.: Germ: A generalist robotic model with mixture-of-experts for quadruped robot. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 11879–11886. IEEE (2024)
38. Tan, S., Dou, K., Zhao, Y., Krähenbühl, P.: Interactive post-training for vision-language-action models. arXiv preprint arXiv:2505.17016 (2025)
39. Team, O.M., Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Kreiman, T., Xu, C., et al.: Octo: An open-source generalist robot policy. arXiv preprint arXiv:2405.12213 (2024)
40. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. arXiv preprint arXiv:2006.10726 (2020)
41. Yang, S., Zhang, Y., He, H., Pan, L., Li, X., Bai, C., Li, X.: Steering vision-language-action models as anti-exploration: A test-time scaling approach. arXiv preprint arXiv:2512.02834 (2025)
42. Yuan, J., Zhang, B., Gong, K., Yue, X., Shi, B., Qiao, Y., Chen, T.: Reg-tta3d: Better regression makes better test-time adaptive 3d object detection. In: European conference on computer vision. pp. 197–213. Springer (2024)
43. Zang, H., Wei, M., Xu, S., Wu, Y., Guo, Z., Wang, Y., Lin, H., Shi, L., Xie, Y., Xu, Z., et al.: Rlinf-vla: A unified and efficient framework for vla+ rl training. arXiv preprint arXiv:2510.06710 (2025)
44. Zhang, H., Zhuang, Z., Zhao, H., Ding, P., Lu, H., Wang, D.: Reinbot: Amplifying robot visual-language manipulation with reinforcement learning. arXiv preprint arXiv:2505.07395 (2025)
45. Zhang, Z., Zheng, K., Chen, Z., Jang, J., Li, Y., Han, S., Wang, C., Ding, M., Fox, D., Yao, H.: Grape: Generalizing robot policy via preference alignment. arXiv preprint arXiv:2411.19309 (2024)
46. Zhao, T.Z., Kumar, V., Levine, S., Finn, C.: Learning fine-grained bimanual manipulation with low-cost hardware. arXiv preprint arXiv:2304.13705 (2023)
47. Zheng, Z., Peng, X., Yang, T., Shen, C., Li, S., Liu, H., Zhou, Y., Li, T., You, Y.: Open-sora: Democratizing efficient video production for all. arXiv preprint arXiv:2412.20404 (2024)
48. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: Conference on Robot Learning. pp. 2165–2183. PMLR (2023)
49. Zuo, Y., Zhang, K., Sheng, L., Qu, S., Cui, G., Zhu, X., Li, H., Zhang, Y., Long, X., Hua, E., et al.: Ttrl: Test-time reinforcement learning. arXiv preprint arXiv:2504.16084 (2025)