Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Towards Long-Lived Robots: Continual Learning VLA Models via Reinforcement Fine-Tuning

Yuan Liu[1,2†], Haoran Li[1,3,4 ✉], Shuai Tian[1,3], Yuxing Qin[1,3], Yuhui Chen[1,3], Yupeng Zheng[1,3], Yongzhen Huang[2], Dongbin Zhao[1,3,4]

[1]SKL-MAIS, Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China
[2]School of Artificial Intelligence, Beijing Normal University, Beijing, China
[3]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[4]Beijing Academy of Artificial Intelligence, Beijing, China

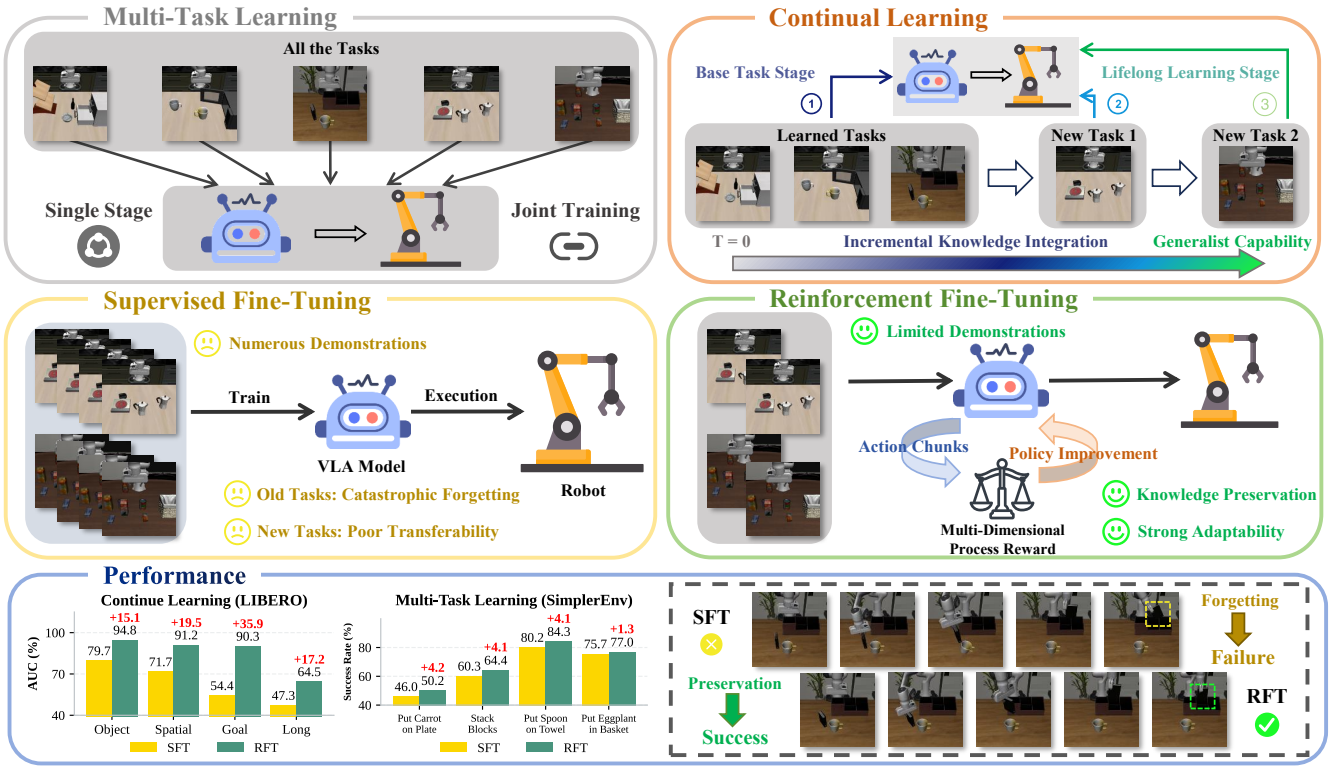

Fig. 1: **Overview of VLA post-training.** This phase involves single-stage multi-task adaptation and incremental continual learning. Addressing the substantial data dependence and susceptibility to catastrophic forgetting inherent in SFT, we introduce LifeLong-RFT, which combines on-policy RL with the multi-dimensional process reward mechanism.

***Abstract*—Pretrained on large-scale and diverse datasets, VLA models demonstrate strong generalization and adaptability as general-purpose robotic policies. However, Supervised Fine-Tuning (SFT), which serves as the primary mechanism for adapting VLAs to downstream domains, requires substantial amounts of task-specific data and is prone to catastrophic forgetting. To address these limitations, we propose LifeLong-RFT, a simple yet effective Reinforcement Fine-Tuning (RFT) strategy for VLA models independent of online environmental feedback and pre-trained reward models. By integrating chunking-level on-policy reinforcement learning with the proposed multi-dimensional process reward mechanism, LifeLong-RFT quantifies the heterogeneous contributions of intermediate action chunks across three dimensions to facilitate policy optimization. Specifically, (1) the Quantized Action Consistency Reward (QACR) ensures accurate action prediction within the discrete action space; (2) the Continuous Trajectory Alignment Reward (CTAR) aligns decoded continuous action chunks with reference trajectories to ensure precise control; (3) the Format Compliance Reward (FCR) guarantees the structural validity of outputs. Comprehensive experiments across SimplerEnv, LIBERO, and real-world tasks demonstrate that LifeLong-RFT exhibits strong performance in multi-task learning. Furthermore, for continual learning on the LIBERO benchmark, our method achieves a 22% gain in average success rate over SFT, while effectively adapting to new tasks using only 20% of the training data. Overall, our method provides a promising post-training paradigm for VLAs. The project page is available at https://yuan-liu-lifelong-rft.github.io.**

I. Introduction

Vision-Language-Action (VLA) models trained on large-scale datasets are progressively emerging as a pivotal approach

† Work done during an internship at CASIA. ✉ Corresponding author.

for achieving generalist robot policies [6, 7, 25, 13]. Despite these advances, adapting VLA models to new tasks via supervised fine-tuning (SFT) remains challenging, as illustrated in Fig. 1. First, SFT typically requires a substantial amount of task-specific data, limiting the ability of VLA models to rapidly adapt in low-data or few-shot settings. Second, SFT often leads to catastrophic forgetting, where learning new skills degrades previously acquired knowledge. These issues prevent the SFT from supporting the evolution of VLA into long-lived agents capable of continually acquiring new skills.

These two challenges are not independent [64]: improving data-efficient adaptation often exacerbates forgetting, while preserving prior knowledge restricts effective learning from limited new data. Achieving an effective balance between plasticity and stability is essential for robots to learn from limited data without erasing prior knowledge. In earlier work based on specialized models, this trade-off was widely viewed as intrinsic [50], motivating solutions based on task-specific adapters or handcrafted features [32, 43, 68, 51]. With the emergence of foundation models, representations learned from massive and diverse datasets exhibit substantially improved transferability, reshaping the plasticity–stability dilemma yet not eliminating it [54]. While such representations significantly reduce the data requirements for learning new tasks, directly applying SFT still results in severe catastrophic forgetting [47]. Consequently, continual learning techniques developed for specialized models are often reused to mitigate forgetting in foundation models [67, 69, 61]. However, these techniques struggle to scale to VLA settings involving both massive tasks and a high-capacity parameterization.

In contrast to SFT, which learns from the annotated datasets, recent advances in large language models suggest that on-policy reinforcement learning (RL), which updates the model using samples drawn from its current distribution, can exhibit stronger robustness to forgetting [60, 11, 31]. This observation raises an important question for robotics: *can on-policy reinforcement learning be leveraged to enable continual adaptation of VLA foundation models, supporting their evolution into long-lived agents?* A central challenge in answering this question lies in designing efficient, reliable, and scalable reward signals for reinforcement fine-tuning of VLA models.

Existing approaches to reinforcement fine-tuning VLA models rely primarily on two categories of reward signals. The first uses environment-provided ground-truth rewards [42, 36, 63, 48, 12], which are typically available only in simulation and depend on privileged information. Such methods face significant barriers in real-world deployment due to the sim-to-real gap and the difficulty of computing rewards without access to privileged state. The second category employs model-based reward estimation, such as predicting task success [62, 46, 15, 75], task progress [45, 17, 74, 34], or distance-based dense rewards [19, 24, 16, 25, 20]. However, inaccuracies and generalization errors in reward models make these approaches highly vulnerable to reward hacking [18]. Moreover, both categories require extensive interaction with an environment—whether simulators [36, 42], world models [80, 25], or real robots [15, 53]—resulting in prohibitively high training costs when scaling to large task sets. Importantly, existing methods predominantly optimize performance on the fine-tuning tasks themselves, while largely neglecting the continual learning properties required for long-lived VLA agents.

In this work, we propose a simple yet effective post-training paradigm for VLA models named **LifeLong-RFT**. By designing a multi-dimensional process reward mechanism, we enable chunking-level on-policy reinforcement fine-tuning without requiring interaction with the environment. Specifically, we decompose this mechanism into three dimensions to provide comprehensive rewards. First, we introduce the **Q**uantized **A**ction **C**onsistency **R**eward (QACR). Given that VLAs are built upon VLM backbones to generate discrete action tokens, QACR ensures precise prediction within the quantized action space by measuring the consistency between predicted and target tokens. Second, we design the **C**ontinuous **T**rajectory **A**lignment **R**eward (CTAR). While QACR ensures accuracy within the quantized action space, physical execution necessitates alignment with continuous trajectories. To this end, CTAR utilizes decoded action chunks to compute chunking-level rewards based on spatial deviations from reference trajectories, incentivizing the model to explore optimal motions. Third, we introduce the **F**ormat **C**ompliance **R**eward (FCR). Due to the generative diversity of VLA backbones, the model is prone to producing structurally invalid outputs (*e.g.*, mismatched action dimensions and inconsistent prediction horizons). To mitigate this instability, the FCR functions as a binary reward that promotes adherence to valid formats, ensuring action executability and enhancing inference efficiency.

Our main contributions are summarized as follows:

1) We propose **LifeLong-RFT**, a post-training strategy that integrates chunking-level on-policy reinforcement learning with the multi-dimensional process reward. This approach enables VLAs to continually master new tasks with limited demonstrations while preserving original capabilities.

2) The multi-dimensional process reward mechanism comprises the **Q**uantized **A**ction **C**onsistency **R**eward (QACR), the **C**ontinuous **T**rajectory **A**lignment **R**eward (CTAR), and the **F**ormat **C**ompliance **R**eward (FCR), quantifying the heterogeneous contributions of intermediate action chunks across three dimensions to facilitate policy optimization.

3) Comprehensive experiments across both simulated and real-world tasks demonstrate LifeLong-RFT's superior performance in multi-task learning. Notably, for continual learning on LIBERO, our method achieves a 22% improvement in average success rate over SFT, facilitating efficient adaptation to novel tasks with only 20% of the training data.

II. Related Work

A. Vision-Language-Action Models

Representing a paradigm shift, VLA models diverge from the traditional hierarchical architecture in favor of an end-to-end learning approach, directly mapping multimodal perceptual inputs to robot control actions [8, 82]. Generally, these models can be categorized into two streams based on their

action representations: Discrete Action Models [55, 33, 56, 29, 27] and Continuous Action Models [30, 7, 65, 23, 78]. Discrete Action Models typically utilize VLM backbones [1, 4] to generate discrete action tokens autoregressively for executing manipulation tasks. In contrast, Continuous Action Models explore the integration of diffusion policies [65, 14] or flow matching [6] with VLMs to directly output continuous actions, achieving dexterous control. Building on these architectures, current models typically leverage large-scale pretraining to acquire manipulation priors, followed by SFT to adapt to specific downstream tasks. Nevertheless, despite demonstrating promising performance, this SFT-based post-training paradigm remains limited by the need for substantial amounts of task-specific data and is prone to catastrophic forgetting.

B. Reinforcement Fine-tuning for VLA Models

To further enhance the robustness and self-refinement capabilities of VLAs, recent research increasingly explores reinforcement fine-tuning strategies. Current strategies primarily comprise three paradigms: simulation-based, real-world-based, and world model-driven approaches. Simulation-based methods [48, 73, 36, 45, 17] benefit from large-scale parallelization, significantly enhancing sample efficiency, and leverage privileged states to construct dense rewards. However, constrained by the sim-to-real gap, these approaches face challenges in deployment within the physical world. Real-world-based strategies [15, 22, 72, 76**?**] effectively enhance model generalization through online adaptation to physical environments. Nevertheless, such methods often involve prohibitive human costs and struggle with the acquisition of rewards. Notably, frontier research [25, 20, 37, 70, 28] employs world models for VLA reinforcement fine-tuning. By leveraging the capability of future state prediction, this approach provides dense reward signals for policy optimization. However, the inherent prediction errors of world models increase the susceptibility of VLAs to reward hacking. Overall, these methods necessitate extensive environmental interaction, limiting their scalability due to high training costs.

C. Continual Learning in Robotics

Continual learning in robotics aims to construct generalist policies capable of adapting to dynamic environmental changes [2, 3] while retaining existing capabilities. Several studies [49, 43, 32, 68, 38, 57, 71] address forgetting by allocating specific parameters to each new learning phase. Additionally, alternative approaches depend on task decomposition via clustering or multistage learning [67, 81, 51]. With the advent of VLAs, recent research focuses on enabling their continual adaptation. Along this line, MergeVLA [21] introduces a model-merging paradigm, aiming to achieve efficient skill expansion by resolving parameter conflicts during the fusion of multi-expert models. On the other hand, Stellar VLA [69] constructs a knowledge-driven continual imitation learning framework, effectively mitigating catastrophic forgetting. In contrast to the aforementioned methods, we integrate on-policy reinforcement learning with the proposed multi-dimensional process reward mechanism to effectively adapt to new tasks while retaining previously learned knowledge.

III. Problem Formulation and Preliminaries

VLA and Post-Training. The goal of VLA modeling is to learn a general-purpose robotic policy $\pi_\theta(\mathbf{a}|o, l)$, which maps observations o and natural language instructions l to robot actions $\mathbf{a}$. In practice, a VLA model is first *pretrained* on large-scale and diverse datasets to acquire rich semantic understanding and transferable representations. The pretrained parameters θ are then *post-trained* using task-specific data to adapt the action outputs $\mathbf{a}$ to the target robot embodiment and downstream tasks.

Continual Learning. SFT remains the primary approach for post-training VLA models. However, SFT primarily optimizes performance on the tasks present in the current training dataset, while largely overlooking the degradation of previously acquired capabilities. In real-world settings, a long-lived agent is expected to acquire new skills while retaining the skills learned earlier, a requirement commonly referred to as continual learning. Formally, this involves an agent learning from a sequence of tasks $\{\mathcal{T}_k\}_{k=1}^{\infty}$, where each task $\mathcal{T}_k$ is associated with N expert demonstrations $\{\tau_k^n\}_{n=1}^{N}$. Unlike a single adaptation stage, which assumes concurrent access to all task data, CL necessitates continuous knowledge acquisition under the constraint of limited access to historical data.

On-Policy Reinforcement Learning. While SFT can efficiently improve performance on the currently targeted tasks, it often leads to rapid degradation of previously acquired capabilities, a phenomenon commonly known as catastrophic forgetting. In contrast, recent findings [60, 11, 31] in LLMs suggest that on-policy reinforcement learning exhibits stronger resistance to forgetting. Unlike SFT, which relies on fixed annotated datasets, on-policy reinforcement learning updates the policy using self-generated answers and optimizes the expected return over these answers.

IV. Method

To support the evolution of VLAs into long-lived agents capable of continually acquiring new skills, we propose LifeLong-RFT, a reinforcement fine-tuning strategy illustrated in Fig. 2. This strategy integrates chunking-level on-policy reinforcement learning with the proposed multi-dimensional process reward mechanism, which quantifies the heterogeneous contributions of intermediate action chunks across three dimensions without requiring environment interaction.

A. Chunking-Level On-Policy Reinforcement Learning

Most existing on-policy RL approaches [42, 36, 63, 12] for VLA post-training optimize model parameters by collecting full trajectories and relying on environment-provided rewards. While such methods can achieve strong performance, they require extensive interaction with the environment during training, leading to high training costs and limiting scalability to large-scale and multi-task settings. To eliminate the need for environment interaction, we adopt a simple alternative: instead

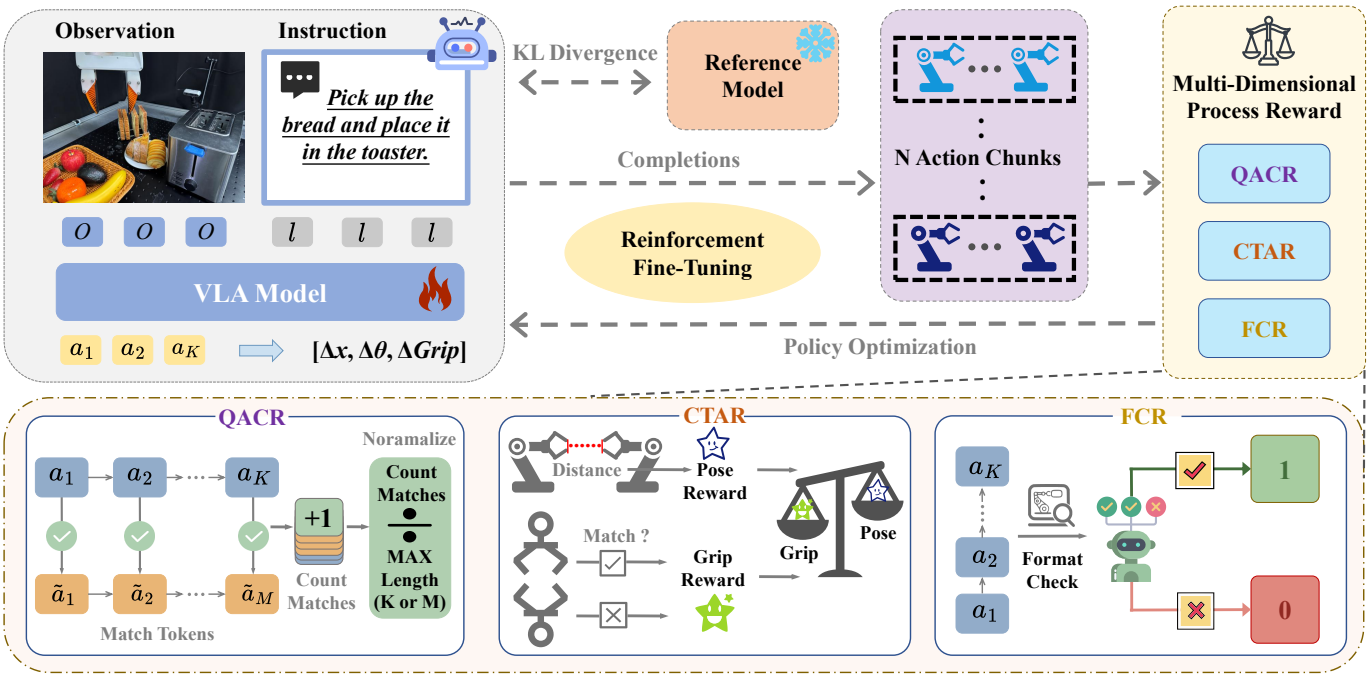

Fig. 2: **Overview of the proposed LifeLong-RFT.** This strategy integrates the chunking-level on-policy reinforcement learning algorithm with the multi-dimensional process reward mechanism to facilitate policy optimization.

of evaluating actions along complete trajectories, we evaluate each action chunk sampled by the VLA model independently, thereby removing the dependency on environment interaction. In this work, we employ Group Relative Policy Optimization (GRPO) [59]. In contrast to conventional algorithms such as PPO [58], which rely on an explicit critic network, GRPO estimates advantages via group-wise comparisons of sampled outputs, thereby considerably reducing computational overhead. Specifically, for each observation o and instruction l, a group of G action outputs $\{\mathbf{a}_i\}_{i=1}^{G}$ is first sampled from the old policy $\pi_{\theta_{\text{old}}}(\mathbf{a}|o,l)$. Then, corresponding rewards $\{r_i\}_{i=1}^{G}$ are computed via task-specific reward functions. Based on the mean and standard deviation of intra-group rewards, the relative advantage A_i for each output is computed as follows:

$$A_i = \frac{r_i - \text{mean}(\{r_1, \ldots, r_G\})}{\text{std}(\{r_1, \ldots, r_G\})}. \tag{1}$$

Given the advantage A_i, the policy parameters θ are optimized by maximizing the following objective:

$$\begin{aligned} J_{\text{GRPO}}(\theta) = \mathbb{E}_{(o,l)\sim\mathcal{B},\{\mathbf{a}_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|o,l)} \\ \frac{1}{G}\sum_{i=1}^{G}\{\min[\frac{\pi_\theta(\mathbf{a}_i|o,l)}{\pi_{\theta_{\text{old}}}(\mathbf{a}_i|o,l)}A_i, \\ \text{clip}\left(\frac{\pi_\theta(\mathbf{a}_i|o,l)}{\pi_{\theta_{\text{old}}}(\mathbf{a}_i|o,l)}, 1-\epsilon, 1+\epsilon\right)A_i] \\ -\gamma D_{KL}\left[\pi_\theta||\pi_{\text{ref}}\right]\}, \end{aligned} \tag{2}$$

where $\mathcal{B}$ denotes the dataset of expert demonstrations, each comprising an observation o and a language instruction l. To stabilize the training process, clip constrains the policy probability ratio, $\frac{\pi_\theta(\mathbf{a}_i|o,l)}{\pi_{\theta_{\text{old}}}(\mathbf{a}_i|o,l)}$, within $[1-\epsilon, 1+\epsilon]$. Furthermore, γ modulates the strength of the KL divergence regularization term $D_{KL}\left[\pi_\theta||\pi_{\text{ref}}\right]$, effectively preventing the new policy π_θ from deviating excessively from the reference policy π_{ref}. Building upon this formulation, the construction of an efficient and verifiable reward r_i becomes the key to optimization.

B. Multi-Dimensional Process Reward

To effectively guide the on-policy reinforcement learning process without requiring environment interaction, we design the multi-dimensional process reward mechanism. This mechanism decomposes the assessment of action chunks into three complementary dimensions, bridging discrete token generation and continuous robotic control. In this section, we detail the designs of the three dimension-specific rewards.

1) Quantized Action Consistency Reward: Built upon VLM backbones, contemporary VLAs [26, 29, 55] interpret language instructions and multi-modal observations to generate action tokens. This paradigm necessitates designing a specialized reward function to assess the consistency between generated tokens and the ground truth, facilitating accurate prediction within the quantized action space. For this purpose, we propose the Quantized Action Consistency Reward (QACR) function, as shown in Algorithm 1.

First, we perform a format check on the model generations to verify their compliance with the predefined specifications of the action tokenizer Fast+ [55] (*i.e.*, the action chunk size and action dimension). Only validated generations proceed to the subsequent consistency assessment stage, while those failing the verification receive a reward of zero. Second, we compute the consistency reward by position-wise matching between the predicted action token sequence $\mathbf{a} = \{a_u\}_{u=1}^{U}$ and its ground-

Algorithm 1 Pseudo code of the QACR function

Input: Predicted action token sequence $\mathbf{a} = \{a_u\}_{u=1}^{U}$; Ground-truth action token sequence $\tilde{\mathbf{a}} = \{\tilde{a}_v\}_{v=1}^{V}$

is_valid $\leftarrow$ FORMATCHECK($\mathbf{a}$)
if is_valid = False **then**
  QACR $\leftarrow 0$ ▷ Invalid format yields zero reward
else
  $\text{QACR} \leftarrow \dfrac{\sum_{\ell=1}^{\min(U,V)} \mathbb{I}(a_\ell = \tilde{a}_\ell)}{\max(U,V)}$
end if
return QACR

Output: The QACR score $\in [0, 1]$

Algorithm 2 Pseudo code of the CTAR function

Input: Predicted action token sequence $\mathbf{a} = \{a_u\}_{u=1}^{U}$; Ground-truth action token sequence $\tilde{\mathbf{a}} = \{\tilde{a}_v\}_{v=1}^{V}$

is_valid $\leftarrow$ FORMATCHECK($\mathbf{a}$)
if is_valid = False **then**
  CTAR $\leftarrow 0$ ▷ Invalid format yields zero reward
else
  $\mathbf{y} \triangleq (\mathbf{y}^{\text{pose}}, \mathbf{y}^{\text{grip}}) \leftarrow \text{DECODE}(\mathbf{a})$
  $\tilde{\mathbf{y}} \triangleq (\tilde{\mathbf{y}}^{\text{pose}}, \tilde{\mathbf{y}}^{\text{grip}}) \leftarrow \text{DECODE}(\tilde{\mathbf{a}})$
  $H \leftarrow \text{Length}(\tilde{\mathbf{y}})$
  $R_{\text{sum}} \leftarrow 0$
  for $t = 1$ **to** H **do**
    $d_t \leftarrow \dfrac{1}{\dim(\mathbf{y}_t^{\text{pose}})} \|\mathbf{y}_t^{\text{pose}} - \tilde{\mathbf{y}}_t^{\text{pose}}\|_1$
    $r_t^{\text{pose}} \leftarrow \exp(-\alpha \cdot d_t)$
    $r_t^{\text{grip}} \leftarrow \mathbb{I}(\mathbf{y}_t^{\text{grip}} = \tilde{\mathbf{y}}_t^{\text{grip}})$
    $r_t \leftarrow \beta \cdot r_t^{\text{pose}} + (1-\beta) \cdot r_t^{\text{grip}}$
    $R_{\text{sum}} \leftarrow R_{\text{sum}} + r_t$
  end for
  CTAR $\leftarrow R_{\text{sum}}/H$
end if
return CTAR

Output: The CTAR score $\in [0, 1]$

truth counterpart $\tilde{\mathbf{a}} = \{\tilde{a}_v\}_{v=1}^{V}$, which is formally defined as:

$$\text{QACR} = \begin{cases} \dfrac{\sum_{\ell=1}^{\min(U,V)} \mathbb{I}(a_\ell = \tilde{a}_\ell)}{\max(U,V)}, & \text{if valid} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $\mathbb{I}(\cdot)$ is the indicator function that returns 1 when the predicted action token a_ℓ matches the ground-truth $\tilde{a}_\ell$, and 0 otherwise. Additionally, the term "valid" indicates that the predicted sequence satisfies the decoding requirements of the Fast+ tokenizer. Based on this formulation, QACR provides a robust assessment for sequence consistency.

2) Continuous Trajectory Alignment Reward: While QACR ensures accuracy within the quantized action space, physical execution necessitates alignment with continuous trajectories. To address this, we introduce the Continuous Trajectory Alignment Reward (CTAR). This mechanism assesses the spatial alignment between decoded continuous action chunks and reference trajectories, providing dense feedback to facilitate dexterous manipulation. The implementation of this reward function is outlined in Algorithm 2.

Consistent with QACR, we first conduct format verification on the predicted action token sequence $\mathbf{a} = \{a_u\}_{u=1}^{U}$. Only sequences that pass this verification proceed to the subsequent reward calculation, while invalid ones are directly assigned a zero reward. Subsequently, we utilize the Fast+ [55] tokenizer to decode the predicted action tokens into the continuous action chunk $\mathbf{y}$, comprising a sequence of H actions. For the action chunk $\mathbf{y}$, the action vector $\mathbf{y}_t$ at each time step consists of a pose component $\mathbf{y}_t^{\text{pose}}$ and a gripper component $\mathbf{y}_t^{\text{grip}}$. Here, $\mathbf{y}_t^{\text{pose}}$ represents the end-effector pose (or joint angles) of the robot at step t, while $\mathbf{y}_t^{\text{grip}}$ indicates the gripper's open-close state. Based on this, we decompose the calculation of CTAR into the following steps: **(1)** To encourage precise pose alignment, we formulate the pose reward r_t^{pose} as an exponentially decaying function of the error relative to the ground truth. Specifically, we compute the normalized L1 distance d_t between the predicted pose vector $\mathbf{y}_t^{\text{pose}}$ and the ground truth $\tilde{\mathbf{y}}_t^{\text{pose}}$. Based on this error, we apply an exponential decay function $\exp(-\alpha \cdot d_t)$ to convert it into a reward signal, where the hyperparameter α regulates the sensitivity to pose deviation. **(2)** To incentivize precise gripper actuation, we employ a binary reward r_t^{grip}. This reward is defined as an indicator function $\mathbb{I}(\cdot)$ that assigns a value of 1 when the predicted gripper state $\mathbf{y}_t^{\text{grip}}$ matches the ground truth $\tilde{\mathbf{y}}_t^{\text{grip}}$, and 0 otherwise. **(3)** Finally, the normalized CTAR is computed by averaging the weighted combination of pose and grip rewards over the action chunk size H, formally defined as:

$$\text{CTAR} = \begin{cases} \dfrac{1}{H} \displaystyle\sum_{t=1}^{H} \left(\beta \cdot r_t^{\text{pose}} + (1-\beta) \cdot r_t^{\text{grip}} \right), & \text{if valid} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where the hyperparameter $\beta \in [0, 1]$ modulates the relative importance of the pose reward r_t^{pose} and the gripper reward r_t^{grip} at each time step t. In conclusion, the CTAR function establishes a dense reward mechanism by quantifying the prediction discrepancies in both robot poses and gripper states.

3) Format Compliance Reward: While QACR and CTAR focus on optimizing prediction accuracy and control precision, their effectiveness is dependent on the structural validity of the generated outputs. Specifically, the predicted sequences must adhere to the specified action dimensions and action chunk size. To this end, we propose the Format Compliance Reward (FCR) to guide the model in generating structurally well-formed token sequences.

Concretely, we employ the Fast+ tokenizer to verify the compliance of the generated token sequence with the required output shape. Accordingly, we define the FCR as a binary reward function that returns 1 if the validation passes and 0

TABLE I: Multi-Task learning performance on SimplerEnv.

Method	Training Strategy	WidowX (Visual Matching)					Google Robot (Visual Matching)			
		Put Carrot on Plate	Stack Blocks	Put Spoon on Towel	Put Eggplant in Basket	Avg	Pick Coke Can	Move Near	Open/Close Drawer	Avg
Continuous Action Models										
Octo-Base [65]	SFT	8.3	0.0	12.5	43.1	16.0	17.0	4.2	22.7	16.8
RoboVLM [41]	SFT	25.0	12.5	29.2	58.3	31.3	77.3	61.7	43.5	63.4
GR00T N1.5 [52]	SFT	–	–	–	–	–	69.3	68.7	35.8	52.4
π_0 [6]	SFT	**58.8**	21.3	63.3	79.2	55.7	72.7	65.3	38.3	58.7
ThinkAct [23]	SFT + RFT	37.5	8.7	58.3	70.8	43.8	92.0	72.4	50.0	71.5
NORA-1.5 [25]	SFT	–	–	–	–	–	92.8	78.7	62.2	77.9
NORA-1.5 [25] (DPO)	SFT+RFT	–	–	–	–	–	94.0	**88.0**	**66.4**	**82.8**
Discrete Action Models										
TraceVLA [79]	SFT	–	–	–	–	–	28.0	53.7	57.0	42.0
RT-1-X [8]	SFT	4.2	0.0	0.0	0.0	1.1	56.7	31.7	59.7	53.4
OpenVLA [29]	SFT	0.0	0.0	0.0	4.1	1.0	16.3	46.2	35.6	27.7
SpatialVLA [56]	SFT	25.0	29.2	16.7	**100.0**	42.7	86.0	77.9	57.4	73.7
π_0-FAST [55]	SFT	22.0	**83.0**	29.0	48.0	45.5	75.3	67.5	42.6	61.9
NORA-1.5-FAST [25]	SFT	–	–	–	–	–	88.6	86.4	41.2	72.1
NORA-Long [26] (Baseline)	SFT	46.0	60.3	80.2	75.7	65.5	86.0	82.3	56.0	74.7
NORA-Long [26]	**RFT (Ours)**	50.2	64.4	**84.3**	77.0	**69.0**	**94.0**	84.7	58.5	79.1
Δ	–	+4.2	+4.1	+4.1	+1.3	**+3.5**	+8.0	+2.4	+2.5	**+4.4**

otherwise. The specific formulation is defined as:

$$\text{FCR} = \begin{cases} 1, & \text{if valid} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where the condition "valid" indicates that the model output adheres to the predefined output format, enabling the Fast+ tokenizer to decode it into a continuous action chunk. By explicitly incentivizing the model to acquire structurally valid output patterns, this reward establishes the necessary prerequisites for effective trajectory exploration.

Finally, we synthesize QACR, CTAR, and FCR to formulate the multi-dimensional process reward as follows:

$$r = \omega \cdot \text{QACR} + (1 - \omega) \cdot \text{CTAR} + \lambda \cdot \text{FCR}, \tag{6}$$

where $\omega \in [0, 1]$ governs the trade-off between discrete action consistency and continuous trajectory alignment, and λ scales the significance of structural format compliance.

V. EXPERIMENTS

In this section, we investigate the performance of LifeLong-RFT through comprehensive experiments in both simulation and real-world settings. We first present the implementation details of the method, then detail the experimental configurations and results for both multi-task and continual learning.

A. Implementation Details

In our experiments, we adopt NORA-Long [26] as the base VLA model, which utilizes the Fast+ [55] tokenizer for action representation. During the reinforcement fine-tuning phase, the model undergoes full-parameter optimization. Specifically, we set the rollout group size for GRPO to 8 and employ the AdamW [44] optimizer with a learning rate of 1×10^{-6}. For the CTAR configuration, the hyperparameters α and β are set to 5 and 0.8, respectively. Finally, the multi-dimensional process reward is formulated as a weighted combination of rewards across three dimensions, with weighting coefficients $\omega = 0.7$ and $\lambda = 0.1$. All experiments are conducted on 8 NVIDIA H20 GPUs.

B. Multi-Task Learning Experiments

1) Experimental Setup:

a) Training Settings: To evaluate multi-task learning in simulation, we utilize SimplerEnv [39] and LIBERO [40]. For SimplerEnv, we train the model on BridgeData V2 [66] for WidowX and Fractal [8] for Google Robot. For LIBERO, we fine-tune the model for each task suite (*i.e.*, Object, Spatial, Goal, and Long), utilizing all 10 tasks with third-person and wrist inputs. Moreover, we conduct experiments within real-world environments on the Franka robot, as shown in Fig. 3. Concretely, we jointly train the model using 40 demonstrations each for the first three tasks, and 50 for the last.

b) Evaluation Protocols: In SimplerEnv, we evaluate the model's performance on the WidowX and Google Robot platforms under the Visual Matching setting. To ensure robust evaluation, each task is repeated over 24 trials under diverse initial object poses and environmental configurations. For LIBERO, we evaluate the model on each task suite with 500 trials. Additionally, in real-world experiments, we perform 20 trials per task. Across all the above experiments, we report the average success rate (SR) as the evaluation metric.

2) Performance on Simulation: In Table I, LifeLong-RFT consistently improves the performance of the SFT baseline across diverse evaluation scenarios, achieving average success rate improvements of 3.5% on WidowX and 4.4% on the Google Robot. Moreover, results in Table II show that our method surpasses all competing continuous and discrete action models, achieving a superior average success rate of 95.6%.

3) Performance on Real-World: Beyond simulation, we also conducted real-world experiments. Table III demonstrates

Fig. 3: **Overview of real-world experimental tasks:** Pick & Place (Banana, Bread), Pull Drawer, and Hang Chinese Knot.

TABLE II: Multi-Task learning performance on LIBERO.

Method	Training Strategy	LIBERO				Avg
		Object	Spatial	Goal	Long	
Continuous Action Models						
Octo-Base [65]	SFT	85.7	78.9	84.6	51.1	75.1
GR00T N1 [5]	SFT	97.6	94.4	93.0	90.6	93.9
π_0 [6]	SFT	98.8	96.8	95.8	85.2	94.2
OpenVLA-OFT [30]	SFT	98.1	96.9	95.5	**91.1**	95.4
ThinkAct [23]	SFT + RFT	91.4	88.3	87.1	70.9	84.4
VLA-RFT [37]	SFT + RFT	94.4	94.4	95.4	80.2	91.1
NORA-1.5 [25]	SFT	96.4	97.3	94.5	89.6	94.5
NORA-1.5 [25] (DPO)	SFT + RFT	96.0	98.0	95.4	90.5	95.0
Discrete Action Models						
TraceVLA [79]	SFT	85.2	84.6	75.1	54.1	74.8
OpenVLA [29]	SFT	88.4	84.7	79.2	53.7	76.5
SpatialVLA [56]	SFT	89.9	88.2	78.6	55.5	78.1
CoT-VLA [77]	SFT	91.6	87.5	87.6	69.0	83.9
WorldVLA [9]	SFT	96.2	87.6	83.4	60.0	79.1
π_0-Fast [55]	SFT	96.8	96.4	88.6	60.2	85.5
MolmoAct-7B-D [33]	SFT	95.4	87.0	87.6	77.2	86.6
TGRPO [17]	SFT + RFT	92.2	90.4	81.0	59.2	80.7
NORA-Long [26] (Baseline)	SFT	97.5	96.4	91.0	82.4	91.8
NORA-Long [26]	**RFT (Ours)**	**99.2**	**98.2**	**95.8**	89.0	**95.6**
Δ	–	+1.7	+1.8	+4.8	+6.6	**+3.8**

TABLE III: Multi-Task learning performance on real-world.

Task Split	π_0 [6]	OpenVLA [29]	NORA-Long [25]		
	SFT	SFT	SFT	**RFT (Ours)**	**Δ**
Pick Banana	90.0	75.0	85.0	90.0	+5.0
Pick Bread	75.0	70.0	75.0	85.0	+10.0
Pull Drawer	95.0	85.0	95.0	100.0	+5.0
Hang Chinese Knot	65.0	55.0	60.0	75.0	+15.0
Overall	81.3	71.3	78.8	**87.5**	**+8.7**

that our method consistently outperforms all competing methods across four tasks. Specifically, when employing NORA-Long as the backbone, LifeLong-RFT achieves an average success rate improvement of 8.7% over the SFT baseline. Notably, for the dexterous task "Hang Chinese Knot", this method outperforms the SFT baseline by 15%.

C. Continual Learning Experiments

1) Experimental Setup:

a) Training Settings: We utilize LIBERO [40] to conduct experiments within simulated environments. Following LOTUS [67], the training process consists of a base task stage and a lifelong learning stage. For each task suite, we conduct the base task stage training using its first six tasks, each comprising 50 demonstrations. Subsequently, the lifelong learning stage focuses on incremental learning for the remaining four tasks. In this stage, each new task consists of only 10 demonstrations, while 5 demonstrations per previously learned task are retained for Experience Replay (ER) [10, 35]. Overall, a complete experimental cycle comprises one base learning step and four sequential lifelong learning steps. Additionally, for real-world experiments, we sequentially train the model on four tasks as illustrated in Fig. 3, utilizing 20 demonstrations for each new task and retaining 5 for each previous task.

b) Evaluation Protocols: We utilize three metrics [40] to assess the model's continual learning capabilities: Forward Transfer (FWT), Negative Backward Transfer (NBT), and Area Under the Success Rate Curve (AUC). All three metrics are derived from the task success rate. Specifically, a higher FWT indicates improved adaptation to new tasks; a lower NBT implies effective mitigation of catastrophic forgetting of previously learned tasks; and a higher AUC reflects better average success rates across all evaluated tasks. Given that the model sequentially learns over K tasks $\{\mathcal{T}_k\}_{k=1}^{K}$, let $s_{k,j}$ denote the agent's success rate on task j after learning the first k tasks. These three metrics are defined as follows: $\text{FWT} = \sum_{k\in[K]} \frac{s_{k,k}}{K}$, $\text{NBT} = \sum_{k\in[K]} \frac{\text{NBT}_k}{K}$, $\text{NBT}_k = \frac{1}{K-k}\sum_{q=k+1}^{K}(s_{k,k} - s_{q,k})$, and $\text{AUC} = \sum_{k\in[K]} \frac{\text{AUC}_k}{K}$, $\text{AUC}_k = \frac{1}{K-k+1}(s_{k,k}+\sum_{q=k+1}^{K} s_{q,k})$. In our experiments, we evaluate policies on all learned tasks, conducting 50 episodes for LIBERO and 20 episodes for real-world experiments.

2) Performance on Simulation: First, we evaluate LifeLong-RFT on LIBERO to validate its continual learning capabilities. We compare against models [81, 67, 71] trained with behavioral cloning (BC) loss. Additionally, we assess large-scale VLAs [6, 29, 30, 26] optimized by SFT. As shown in Table IV, our method consistently outperforms other methods across all task suites. Notably, on LIBERO-Goal, LifeLong-RFT demonstrates significant superiority, achieving a substantial gain of 35.9 in AUC over the SFT baseline.

3) Performance on Real-World: Furthermore, we assess real-world continual learning performance. As presented in Table V, our method shows a substantial improvement of 23.7 in FWT over the SFT baseline, significantly outperforming the other two models. Furthermore, the approach yields an

TABLE IV: Continual learning performance on LIBERO.

Task Split	Metrics	BUDS [81]	LOTUS [67]	SPECI [71]	π_0 [6]	OpenVLA [29]	OpenVLA-OFT [30]	NORA-Long [26]		
		BC	BC	BC	SFT	SFT	SFT	SFT	**RFT (Ours)**	**Δ**
LIBERO-Object	FWT (↑)	52.0	74.0	83.0	73.0	59.4	89.8	84.8	96.0	+11.2
	NBT (↓)	21.0	11.0	10.0	16.2	17.9	3.1	6.8	1.5	-5.3
	AUC (↑)	47.0	65.0	78.0	59.3	45.1	87.4	79.7	**94.8**	**+15.1**
LIBERO-Spatial	FWT (↑)	–	–	67.0	74.4	64.2	88.6	82.8	94.0	+11.2
	NBT (↓)	–	–	6.0	23.7	17.6	9.4	14.0	3.7	-10.3
	AUC (↑)	–	–	66.0	55.5	50.8	81.7	71.7	**91.2**	**+19.5**
LIBERO-Goal	FWT (↑)	50.0	61.0	74.0	74.6	58.6	90.2	72.8	92.4	+19.6
	NBT (↓)	39.0	30.0	20.0	23.9	5.8	13.8	25.2	3.1	-22.1
	AUC (↑)	42.0	56.0	65.0	56.3	53.5	79.2	54.4	**90.3**	**+35.9**
LIBERO-Long	FWT (↑)	–	–	58.0	53.8	32.0	64.0	61.0	74.2	+13.2
	NBT (↓)	–	–	21.0	14.2	14.1	31.4	17.3	12.8	-4.5
	AUC (↑)	–	–	46.0	42.5	20.8	38.7	47.3	**64.5**	**+17.2**

TABLE V: Continual learning performance on real-world.

Task Split	Metrics	π_0 [6]	OpenVLA [29]	NORA-Long [26]		
		SFT	SFT	SFT	**RFT (Ours)**	**Δ**
Real-World	FWT (↑)	58.8	46.3	56.3	80.0	+23.7
	NBT (↓)	16.3	17.8	18.3	6.1	-12.2
	AUC (↑)	47.9	35.1	44.2	**75.9**	**+31.7**

TABLE VI: Ablation of the multi-dimensional process reward.

Settings	Object		Spatial		Goal		Long		Avg	
	SR	**Δ**	**SR**	**Δ**	**SR**	**Δ**	**SR**	**Δ**	**SR**	**Δ**
w/o QACR	97.0	-2.2	96.4	-1.8	92.2	-3.6	85.6	-3.4	92.8	-2.8
w/o CTAR	8.0	**-91.2**	6.2	**-92.0**	2.4	**-93.4**	2.0	**-87.0**	4.7	**-90.9**
w/o FCR	98.0	-1.2	96.2	-2.0	93.2	-2.6	84.6	-4.4	93.0	-2.6
RFT (Ours)	**99.2**	-	**98.2**	-	**95.8**	-	**89.0**	-	**95.6**	-

NBT of only 6.1, demonstrating a strong capability to preserve performance on learned tasks. Overall, the model fine-tuned with LifeLong-RFT achieves an average success rate of 75.9% across the learning cycle, exhibiting robust continual learning.

D. Ablation Studies

1) Effectiveness of Multi-Dimensional Process Reward: To verify the effectiveness of multi-dimensional process reward, we conduct multi-task learning experiments on LIBERO. The first row of Table VI indicates that removing QACR leads to a 2.8% average performance drop across the four task suites, confirming its necessity for accurate quantized action prediction. The second row further underscores the critical role of CTAR, where its exclusion leads to a 90.9% performance drop, resulting in the model being nearly incapable of task completion. Additionally, the third row shows that FCR is critical for guaranteeing the structural validity of the output. Particularly in LIBERO-Long, the absence of FCR leads to a 4.4% degradation. Overall, each reward component exhibits consistent effectiveness, enhancing overall performance.

2) Efficiency of New Task Adaptation: To assess the adaptation efficiency of LifeLong-RFT in acquiring new tasks, we conduct experiments across the four task suites of the LIBERO benchmark. Specifically, we first train the model on the initial six tasks of each suite via multi-task learning. Subsequently,

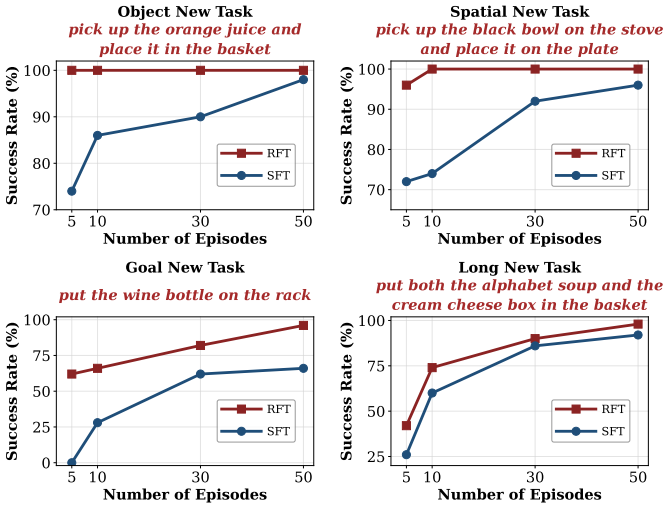

Fig. 4: Adaptation efficiency on representative new tasks.

we select a representative task from the remaining four unseen tasks in each suite to evaluate the model's efficiency for new task adaptation. In Fig. 4, LifeLong-RFT shows superior adaptation efficiency compared to SFT. On the "Pick Orange Juice" task, it achieves 100% success using merely 5 demonstrations, whereas SFT reaches 98% even when trained with 50. Similarly, for the "Pick Bowl" task, LifeLong-RFT matches the performance of the SFT baseline trained on 50 demonstrations using only 5, and improves to 100% success with just 10. In addition to few-shot scenarios, our approach sustains its advantage with the full set of demonstrations: it surpasses SFT by 30% on the "Put Wine Bottle" task and exhibits superior performance on the long-horizon task "Put Alphabet Soup and Cream Cheese".

VI. Conclusion and Future Outlook

In this work, we introduce LifeLong-RFT, a reinforcement fine-tuning strategy to overcome the extensive data requirements and catastrophic forgetting associated with SFT. Unlike existing methods, our method integrates the chunking-level on-policy RL with the multi-dimensional process reward mechanism to achieve efficient new task adaptation while preserving

pre-existing knowledge. Specifically, this mechanism employs Quantized Action Consistency Reward, Continuous Trajectory Alignment Reward, and Format Compliance Reward to quantify the heterogeneous contributions of action chunks across three dimensions, independent of environmental feedback and pre-trained reward models. Comprehensive experiments indicate that LifeLong-RFT consistently surpasses SFT-based methods in both multi-task and continual learning.

Limitations & Future Work. This work primarily focuses on discrete action models, yet their performance falls short of the levels achieved by continuous action models. Future research extending the LifeLong-RFT training strategy to continuous action models will significantly accelerate the transition of VLAs from laboratory research to industrial applications.

VII. ACKNOWLEDGMENTS

This work was supported by the New Generation Artificial Intelligence-National Science and Technology Major Project 2025ZD0122902; in part by the National Natural Science Foundation of China (NSFC) under Grants 62136008 and 62293545; the Beijing Major Science and Technology Project under Contract no. Z251100008125023; the Suzhou Innovation and Entrepreneurship Leading Talents Programme – Innovation Leading Talent in Universities and Research Institutes under Grant ZXL2025310; and the Beijing Academy of Artificial Intelligence (BAAI).

REFERENCES

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[2] Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*, 2023.

[3] Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A tutorial on meta-reinforcement learning. *Foundations and Trends in Machine Learning*, 18(2-3):224–384, 2025.

[4] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

[5] 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.

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[7] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert 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. In *9th Annual Conference on Robot Learning*, 2025.

[8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[9] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

[10] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

[11] Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting. *arXiv preprint arXiv:2510.18874*, 2025.

[12] Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Quanlu Zhang, Zhaofei Yu, Guoliang Fan, et al. πrl: Online rl fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv:2510.25889*, 2025.

[13] Yaran Chen, Wenbo Cui, Yuanwen Chen, Mining Tan, Xinyao Zhang, Jinrui Liu, Haoran Li, Dongbin Zhao, and He Wang. Robogpt: An llm-based long-term decision-making embodied agent for instruction following tasks. *IEEE Transactions on Cognitive and Developmental Systems*, 17(5):1163–1174, 2025.

[14] Yuhui Chen, Haoran Li, and Dongbin Zhao. Boosting continuous control with consistency policy. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, page 335–344, Richland, SC, 2024.

[15] Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. ConRFT: A reinforced fine-tuning method for vla models via consistency policy. In *Robotics: Science and Systems, RSS*, 2025.

[16] Yuhui Chen, Haoran Li, Zhennan Jiang, Haowei Wen, and Dongbin Zhao. Tevir: Text-to-video reward with diffusion models for efficient reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics:*

Systems, 56(2):893–905, 2026.

[17] Zengjue Chen, Runliang Niu, He Kong, Qi Wang, Qianli Xing, and Zipei Fan. Tgrpo: Fine-tuning vision-language-action model via trajectory-wise group relative policy optimization. *arXiv preprint arXiv:2506.08440*, 2025.

[18] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[19] Alejandro Escontrela, Ademi Adeniji, Wilson Yan, Ajay Jain, Xue Bin Peng, Ken Goldberg, Youngwoon Lee, Danijar Hafner, and Pieter Abbeel. Video prediction models as rewards for reinforcement learning. *Advances in Neural Information Processing Systems*, 36:68760–68783, 2023.

[20] Senyu Fei, Siyin Wang, Li Ji, Ao Li, Shiduo Zhang, Liming Liu, Jinlong Hou, Jingjing Gong, Xianzhong Zhao, and Xipeng Qiu. Srpo: Self-referential policy optimization for vision-language-action models. *arXiv preprint arXiv:2511.15605*, 2025.

[21] Yuxia Fu, Zhizhen Zhang, Yuqi Zhang, Zijian Wang, Zi Huang, and Yadan Luo. Mergevla: Cross-skill model merging toward a generalist vision-language-action agent. *arXiv preprint arXiv:2511.18810*, 2025.

[22] Yanjiang Guo, Jianke Zhang, Xiaoyu Chen, Xiang Ji, Yen-Jen Wang, Yucheng Hu, and Jianyu Chen. Improving vision-language-action model with online reinforcement learning. *arXiv preprint arXiv:2501.16664*, 2025.

[23] Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *arXiv preprint arXiv:2507.16815*, 2025.

[24] Tao Huang, Guangqi Jiang, Yanjie Ze, and Huazhe Xu. Diffusion reward: Learning rewards via conditional video diffusion. In *European Conference on Computer Vision*, pages 478–495. Springer, 2024.

[25] Chia-Yu Hung, Navonil Majumder, Haoyuan Deng, Liu Renhang, Yankang Ang, Amir Zadeh, Chuan Li, Dorien Herremans, Ziwei Wang, and Soujanya Poria. Nora-1.5: A vision-language-action model trained using world model-and action-based preference rewards. *arXiv preprint arXiv:2511.14659*, 2025.

[26] Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U Tan, Navonil Majumder, Soujanya Poria, et al. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*, 2025.

[27] Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. *arXiv preprint arXiv:2502.21257*, 2025.

[28] Zhennan Jiang, Kai Liu, Yuxin Qin, Shuai Tian, Yupeng Zheng, Mingcai Zhou, Chao Yu, Haoran Li, and Dongbin Zhao. World4rl: Diffusion world models for policy refinement with reinforcement learning for robotic manipulation. *arXiv preprint arXiv:2509.19080*, 2025.

[29] 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.

[30] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[31] Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Qingfu Zhang, Hongbin Liu, et al. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training. *arXiv preprint arXiv:2507.05386*, 2025.

[32] Daehee Lee, Minjong Yoo, Woo Kyung Kim, Wonje Choi, and Honguk Woo. Incremental learning of retrievable skills for efficient continual task adaptation. *Advances in Neural Information Processing Systems*, 37: 17286–17312, 2024.

[33] 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.

[34] Tony Lee, Andrew Wagenmaker, Karl Pertsch, Percy Liang, Sergey Levine, and Chelsea Finn. Roboreward: General-purpose vision-language reward models for robotics. *arXiv preprint arXiv:2601.00675*, 2026.

[35] Haoran Li, Zhennan Jiang, Yuhui Chen, and Dongbin Zhao. Generalizing consistency policy to visual rl with prioritized proximal experience regularization. *Advances in Neural Information Processing Systems*, 37:109672–109700, 2024.

[36] Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, et al. Simplevla-rl: Scaling vla training via reinforcement learning. *arXiv preprint arXiv:2509.09674*, 2025.

[37] Hengtao Li, Pengxiang Ding, Runze Suo, Yihao Wang, Zirui Ge, Dongyuan Zang, Kexian Yu, Mingyang Sun, Hongyin Zhang, Donglin Wang, et al. Vla-rft: Vision-language-action reinforcement fine-tuning with verified rewards in world simulators. *arXiv preprint arXiv:2510.00406*, 2025.

[38] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International conference on machine learning*, pages 3925–3934. PMLR, 2019.

[39] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

[40] 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.

[41] Huaping Liu, Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, and Hanbo Zhang. Towards generalist robot policies: What matters in building vision-language-action models. 2025.

[42] Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What can rl bring to vla generalization? an empirical study. *arXiv preprint arXiv:2505.19789*, 2025.

[43] Zuxin Liu, Jesse Zhang, Kavosh Asadi, Yao Liu, Ding Zhao, Shoham Sabach, and Rasool Fakoor. Tail: Task-specific adapters for imitation learning with large pre-trained models. *arXiv preprint arXiv:2310.05905*, 2023.

[44] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[45] Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025.

[46] Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 10 (105):eads5033, 2025.

[47] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

[48] Mingyang Lyu, Yinqian Sun, Erliang Lin, Huangrui Li, Ruolin Chen, Feifei Zhao, and Yi Zeng. Reinforcement fine-tuning of flow-matching policies for vision-language-action models. *arXiv preprint arXiv:2510.09976*, 2025.

[49] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[50] M Anwar Ma'sum, Mahardhika Pratama, and Igor Skrjanc. Latest advancements towards catastrophic forgetting under data scarcity: A comprehensive survey on few-shot class incremental learning. *arXiv preprint arXiv:2502.08181*, 2025.

[51] Yuan Meng, Zhenshan Bing, Xiangtong Yao, Kejia Chen, Kai Huang, Yang Gao, Fuchun Sun, and Alois Knoll. Preserving and combining knowledge in robotic lifelong reinforcement learning. *Nature Machine Intelligence*, 7 (2):256–269, 2025.

[52] NVIDIA Isaac Robotics Team. Gr00t n1.5: An upgraded foundation model for humanoid robots. https://research.nvidia.com/labs/gear/gr00t-n1_5/, 2025.

[53] Mingjie Pan, Siyuan Feng, Qinglin Zhang, Xinchen Li, Jianheng Song, Chendi Qu, Yi Wang, Chuankang Li, Ziyu Xiong, Zhi Chen, et al. Sop: A scalable online post-training system for vision-language-action models. *arXiv preprint arXiv:2601.03044*, 2026.

[54] Keon-Hee Park, Kyungwoo Song, and Gyeong-Moon Park. Pre-trained vision and language transformers are few-shot incremental learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23881–23890, 2024.

[55] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[56] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

[57] Dushyant Rao, Francesco Visin, Andrei Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. *Advances in neural information processing systems*, 32, 2019.

[58] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[59] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[60] Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

[61] Mingchen Song, Xiang Deng, Guoqiang Zhong, Qi Lv, Jia Wan, Yinchuan Li, Jianye Hao, and Weili Guan. Few-shot vision-language action-incremental policy learning. *arXiv preprint arXiv:2504.15517*, 2025.

[62] Sumedh Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies. *Advances in Neural Information Processing Systems*, 36:55681–55693, 2023.

[63] Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025.

[64] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12183–12192, 2020.

[65] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An

open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[66] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[67] Weikang Wan, Yifeng Zhu, Rutav Shah, and Yuke Zhu. Lotus: Continual imitation learning for robot manipulation through unsupervised skill discovery. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 537–544. IEEE, 2024.

[68] Yixiao Wang, Yifei Zhang, Mingxiao Huo, Ran Tian, Xiang Zhang, Yichen Xie, Chenfeng Xu, Pengliang Ji, Wei Zhan, Mingyu Ding, et al. Sparse diffusion policy: A sparse, reusable, and flexible policy for robot learning. *arXiv preprint arXiv:2407.01531*, 2024.

[69] Yuxuan Wu, Guangming Wang, Zhiheng Yang, Maoqing Yao, Brian Sheil, and Hesheng Wang. Continually evolving skill knowledge in vision language action model. *arXiv preprint arXiv:2511.18085*, 2025.

[70] Junjin Xiao, Yandan Yang, Xinyuan Chang, Ronghan Chen, Feng Xiong, Mu Xu, Wei-Shi Zheng, and Qing Zhang. World-env: Leveraging world model as a virtual environment for vla post-training. *arXiv preprint arXiv:2509.24948*, 2025.

[71] Jingkai Xu and Xiangli Nie. Speci: Skill prompts based hierarchical continual imitation learning for robot manipulation. *arXiv preprint arXiv:2504.15561*, 2025.

[72] Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Mengke Zhang, and Hao Su. Policy decorator: Model-agnostic online refinement for large policy model. *arXiv preprint arXiv:2412.13630*, 2024.

[73] Hongzhi Zang, Mingjie Wei, Si Xu, Yongji Wu, Zhen Guo, Yuanqing Wang, Hao Lin, Liangzhi Shi, Yuqing Xie, Zhexuan Xu, et al. Rlinf-vla: A unified and efficient framework for vla+ rl training. *arXiv preprint arXiv:2510.06710*, 2025.

[74] Shaopeng Zhai, Qi Zhang, Tianyi Zhang, Fuxian Huang, Haoran Zhang, Ming Zhou, Shengzhe Zhang, Litao Liu, Sixu Lin, and Jiangmiao Pang. A vision-language-action-critic model for robotic real-world reinforcement learning. *arXiv preprint arXiv:2509.15937*, 2025.

[75] Jiahui Zhang, Ze Huang, Chun Gu, Zipei Ma, and Li Zhang. Reinforcing action policies by prophesying. *arXiv preprint arXiv:2511.20633*, 2025.

[76] Jiahui Zhang, Yusen Luo, Abrar Anwar, Sumedh Anand Sontakke, Joseph J Lim, Jesse Thomason, Erdem Biyik, and Jesse Zhang. Rewind: Language-guided rewards teach robot policies without new demonstrations. *arXiv preprint arXiv:2505.10911*, 2025.

[77] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713, 2025.

[78] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

[79] 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. *arXiv preprint arXiv:2412.10345*, 2024.

[80] Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. WMPO: world model-based policy optimization for vision-language-action models. *arXiv preprint arXiv:2511.09515*, 2025.

[81] Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2):4126–4133, 2022.

[82] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.