

A Survey on Reinforcement Learning of Vision-Language-Action Models for Robotic Manipulation

Haoyuan Deng*, Zhenyu Wu*, Haichao Liu*, Wenkai Guo, Yuquan Xue, Ziyu Shan, Chuanrui Zhang, Bofang Jia, Yuan Ling, Guanxing Lu, and Ziwei Wang[†]

Abstract—The vision of building generalist robotic systems capable of performing diverse manipulation tasks has been significantly advanced by Vision-Language-Action models (VLAs), which leverage large-scale pretraining to acquire general visuomotor priors via imitation learning. Current pre-trained VLAs still require fine-tuning to adapt to real-world deployment, where conventional imitation learning struggles with out-of-distribution (OOD) generalization due to the dependence on collected datasets with limited coverage of states and actions. Reinforcement learning (RL) leverages self-exploration and result-driven optimization to enhance OOD generalization in VLAs. This survey outlines how RL can bridge the gap between pre-training and real-world deployment, offering a comprehensive overview of the RL-VLA training paradigm. Our taxonomy is organized along four core dimensions that reflect the full learning-to-deployment lifecycle: RL-VLA architecture, training paradigms, real-world deployment, and benchmarking and evaluation. First, we introduce the key design principles of RL-VLA components, including action, reward, and transition modeling. Second, we review online, offline, and test-time RL paradigms, analyzing their effectiveness and challenges in improving VLA generalization. Third, we examine real-world deployment frameworks, from sim-to-real transfer to safe exploration, autonomous recovery, and human-in-the-loop alignment. Finally, we summarize benchmarking methods, highlight open challenges, and outline the path toward general robotic systems. Our project page can be found [here](#).

Index Terms—Vision-language-action models, reinforcement learning, out-of-distribution generalization, robotic manipulation.

I. INTRODUCTION

The dream of generalist robotic systems capable of performing diverse manipulation tasks in unstructured environments has long been a central goal of robotics and artificial intelligence research. Recent breakthroughs in scaling vision–language models (VLMs) [1], [2] and large-scale robot learning [3] yield VLAs [4]–[8] that leverage heterogeneous

Haoyuan Deng, Haichao Liu, Wenkai Guo, Yuquan Xue, Ziyu Shan, Chuanrui Zhang, Bofang Jia, Yuan Ling, and Ziwei Wang are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore (*e-mail*: {haoyuan.deng, haichao.liu, ziwei.wang}@ntu.edu.sg; {wenkai001, yuquan002, shan0148, chuanrui001, bofang001, LING0113}@e.ntu.edu.sg).

Zhenyu Wu is with the School of Intelligent Engineering and Automation, Beijing University of Posts and Telecommunications, China (*e-mail*: wuzhenyu@bupt.edu.cn).

Guanxing Lu is with the Shenzhen International Graduate School, Tsinghua University, China. (*e-mail*: lgx23@mails.tsinghua.edu.cn).

* Haoyuan Deng, Zhenyu Wu, and Haichao Liu contributed equally.

† Corresponding author: Ziwei Wang.

teleoperation datasets and multi-modal pretraining to acquire general visuomotor priors conditioned on language instructions. Current VLAs demonstrate remarkable zero-shot and few-shot generalization across object categories, task variations, and embodiments, marking a significant departure from task-specific policies that dominate traditional robot learning [9]. However, existing VLA systems are limited to imitation learning, purely repeating expert demonstrations from pre-training datasets. The limited action space and lack of failure recovery demonstrations [10] cause pre-trained VLAs to struggle in OOD scenarios during practical deployment. Furthermore, the purely imitative objective prevents the agent from exploring alternative or potentially superior strategies unseen in the original demonstrations.

To overcome these limitations, RL [11]–[15] has been employed in robot learning, formulating policy learning as a result-driven trial-and-error process, leveraging reward-guided optimization [16]–[18] to empower VLA with generalization ability in OOD scenarios. Building on its utility, RL-based post-training has also demonstrated strong performance gains and improved alignment for Large Language Models (LLMs), enhancing the accuracy and coherence of their step-by-step reasoning in complex domains like mathematics and coding [19], [20]. This success suggests a clear avenue: leveraging RL to effectively enhance the generalization capabilities of VLA models. Recent RL-VLA works [21]–[27] demonstrate a synergistic relationship, showing that VLA models carry rich multi-modal representations that significantly improve the sample efficiency of RL methods, while RL conversely enables VLAs to surpass suboptimal pretraining behaviors. Empirically, the RL-optimized VLA models show significant improvement on representative benchmarks [24], and the RL-trained policies demonstrate stronger generalization compared to those trained solely with Supervised Fine-Tuning (SFT) [28]. These representative RL-VLA works span several paradigms, including offline RL [21], [22], online RL [23]–[25], and test-time RL [26], [27]. A systematic analysis of this emerging literature is essential to summarize beneficial insights of the RL technique. However, despite the review articles covering the applications of RL in both LLMs [29]–[31] and robotics [32], [33], the academic community currently lacks a systematic review focused specifically on the application of RL to VLA models, which is essential to promote more rapid and steady development in this direction.

In this survey, we provide a comprehensive overview of RL-

VLA and clarify the pathway from pretrained VLA models to robust, deployable systems. We introduce a taxonomy (Fig. 1) that organizes both direct RL-VLA advancements and complementary robot-learning studies that inform future framework design. We begin with preliminaries of RL and VLA and outline the key challenges of their integration (**Section II**). We then analyze core RL-VLA design trade-offs: action representation, reward design, and transition modeling (**Section III**), followed by insights from online and offline RL paradigms with an emphasis on policy robustness and adaptation (**Section IV**). We review deployment frameworks from sim-to-real transfer to direct real-world RL (**Section V**), then summarize benchmarks, evaluation metrics, and existing RL-VLA methods (**Section VI**). Finally, we outline open challenges and promising future research directions (**Section VII**).

II. BACKGROUND

A. VLA Models

VLA represents a new paradigm in robotic learning that unifies visual perception, language understanding, and action generation within a single end-to-end framework [4], [34]. Unlike traditional robotic systems that rely on separate modules for perception, planning, and control, VLAs leverage large-scale pre-trained VLMs to directly predict robot actions conditioned on visual observations and language instructions.

The core architecture of VLAs [4]–[8], which integrates VLM backbones with action modules, typically consists of three key components: (1) a vision encoder that processes multi-view RGB images or depth information into visual tokens, (2) a language encoder that embeds task instructions into semantic representations, and (3) a policy decoder that maps the fused vision-language representations to continuous or discretized robot actions. Representative examples include RT-1 [35], which employs a Transformer-based architecture to predict discretized actions, and RT-2 [36], which utilizes PaLI-X to facilitate internet-scale knowledge transfer. Other models, such as OpenVLA [4], introduce an open-source VLA that builds on Llama 2 [37] combined with a visual encoder fusing features from DINOv2 [38] and SigLIP [39]. Building on this, OpenVLA-OFT [40] proposes an Optimized Fine-Tuning (OFT) recipe that integrates parallel decoding and continuous action representations to boost performance and inference efficiency in novel setups significantly. Furthermore, π_0 [34] utilizes a novel flow matching architecture built upon a pretrained VLM to inherit internet-scale semantic knowledge and demonstrate complex dexterous manipulation across diverse robot platforms. Moreover, $\pi_{0.5}$ [5] extends π_0 by using co-training on heterogeneous tasks to achieve broad generalization and enable complex, long-horizon manipulation in entirely novel real-world environments. By grounding language instructions in visual observations, VLAs support intuitive human robot interaction and draw on the semantic knowledge embedded in pre-trained VLMs. Through large-scale data collection, pre-training, and post-training, they exhibit strong generalization across a wide range of tasks, objects, and environments.

The dominant training paradigm for VLA models is IL, typically implemented as SFT or Behavioral Cloning (BC).

This approach leverages large-scale, heterogeneous teleoperation datasets to train the model. The objective is to learn a policy that maximizes the likelihood of expert actions given the corresponding multimodal observations. However, it is limited by the quality and coverage of the demonstration data. Therefore, despite their end-to-end framework and emergent capabilities, VLAs still face significant challenges in domain-shifted scenarios and suffer from high-quality data requirements. This collectively motivates the integration with RL techniques to further enhance their adaptability and robustness.

B. Reinforcement Learning for VLA

To illustrate the existing gap when extending RL methodologies to VLA models, we formalize the RL-VLA problem, detailing its state representations, action spaces, reward functions, and environment dynamics. We formally frame the robotic manipulation task as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$. The goal of RL is to learn a policy $\pi_\theta(a_t|s_t)$, parameterized by θ , that maximizes the expected discounted return $J(\pi)$:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^T \gamma^t r(s_t, a_t) \right] \quad (1)$$

where $\tau = (s_0, a_0, s_1, a_1, \dots)$ is a trajectory generated by the policy, and T is the task horizon. In the RL-VLA context [41], [42], this formulation is highly specialized. The **State** \mathcal{S} is multimodal and high-dimensional, typically defined as $s_t = (o_t^{\text{vis}}, o_t^{\text{prop}}, l_{\text{task}})$, comprising visual observations (e.g., RGB images, point clouds), proprioceptive information (e.g., joint angles, end-effector pose), and the language instruction. The **Action** \mathcal{A} $a_t \in \mathbb{R}^d$ is generated by the VLA’s decoder from its internal hidden state. Notably, VLAs often output *action chunks* $a_{t:t+k-1}$ using mechanisms like diffusion decoders or action tokenizers, rather than single-step actions [43]. The **Reward** $r(s_t, a_t)$ is crucial for optimization and often combines a sparse binary signal for task success with dense, process-based rewards (e.g., distance to target) to provide a richer learning signal [28]. Finally, the **Transition Model** $p(s_{t+1} | s_t, a_t)$ is either defined in simulation or implicitly determined by the physical interactions perceived by real-world robots [44].

RL algorithms developed for optimal decision-making fall into three main families. Value-based methods, such as Deep Q-Networks (DQN) [45], focus on estimating value functions to determine the expected cumulative reward from each state or state-action pair. In contrast, policy gradient methods, like Proximal Policy Optimization (PPO) [46], directly optimize the policy by computing gradients of the expected returns with respect to the policy parameters. Finally, actor-critic methods, such as Soft Actor-Critic (SAC) [47], combine these approaches by simultaneously learning a value function (critic) and a policy (actor). These methods can be *model-free*, learning policies directly from interaction, or *model-based*, which first learn the transition model p . Algorithms are further distinguished by being *on-policy* (learning from current policy data) or *off-policy* (learning from a replay buffer).

Despite significant advances in RL, integrating RL with VLA models presents distinct challenges that require further

investigation. Early work on RL-VLA has laid important foundations and demonstrated promising directions. They span several paradigms, including offline, online, and test-time RL. In the offline RL category, where policies learn from a fixed, pre-collected dataset, **ReinboT** [21] integrates RL principles by predicting dense returns to better leverage mixed-quality data, while **CO-RFT** [22] introduces Chunked RL, a novel framework that extends temporal difference (TD) learning to be compatible with the action chunking inherent in many VLAs. For online RL, which improves the policy by actively collecting new experience through trial-and-error, **VLA-RL** [23] leverages online improvement by proposing a trajectory-level RL formulation and using a VLM as a robotic process reward model to solve sparse reward challenges. Similarly, **SimpleVLA-RL** [24] provides an efficient RL framework with exploration-enhancing strategies that enable the policy to discover previously unseen patterns beyond the demonstration data. Finally, test-time approaches enhance policies at deployment: **V-GPS** [26] introduces Value-Guided Policy Steering to re-rank a policy's actions using an offline-learned value function without any weight updates, and **Hume** [27] implements a dual-system model that performs value-guided System-2 thinking by sampling and selecting the best action candidate at runtime. Despite the pioneering contributions and impressive progress in VLA-RL, substantial challenges remain before these systems can robustly operate in dynamic, open-ended physical environments, leaving abundant opportunities for further research.

III. ARCHITECTURE OF RL-VLA

While imitation learning-based pre-training has enabled VLA models to achieve strong performance across diverse manipulation tasks, their generalization is still limited by the restricted coverage of offline data, especially in OOD states. To address this limitation, recent work couples pre-trained VLA models with RL, turning open-loop inference into a closed-loop optimization process driven by online feedback. In this setting, the policy interacts with environments, collects trajectories and gradually adapts its behavior through reward-guided updates. Building on this, this section introduces the overall RL-VLA architecture, outlining how **Action**, **Reward**, and **Transition Modeling** are organized to jointly optimize perception, decision-making, and environment dynamics.

A. Action

The action model serves as the bridge between visual observation and physical interaction actions in RL-VLA. While pre-trained VLAs typically generate actions in an open-loop manner, RL further employs task-level supervision, which allows the policy to be adjusted through interaction and feedback, leading to higher generalization capabilities. Based on the prediction approach for the manipulation action space, RL can play distinct roles in refining action generation, from token-level supervision in autoregressive action models, to sequence-level optimization in generative action models, and hierarchical coordination in dual-system VLA that combine high-level reasoning with low-level control.

1) Autoregressive Models: Autoregressive VLAs follow the language modeling paradigm, formulate robotic manipulation as a sequential decision process in a discrete token space, where actions are generated step by step through next-token prediction. RL can directly utilize the token prediction probabilities output by autoregressive VLA to achieve more stable policy optimization through token-level supervision and reward-driven mechanisms. This paradigm has shown improved task adaptability and generalization in robotic settings, and has motivated a series of subsequent works [22]–[24], [28], [48] that explore autoregressive RL-VLA for both online fine-tuning and offline policy improvement. Specifically, **TGRPO** [48] rewrites the policy gradient objective as a token-level cross-entropy loss weighted by advantages, which enables stable RL fine-tuning of VLA action generation without changing the form of the action head. **CO-RFT** [22] further leverages the spatio-temporal dynamics of action probabilities to address the challenge of poor trajectory consistency in autoregressive VLAs' discrete action prediction.

Potential Challenges: Although autoregressive VLAs provide direct action prediction probabilities for RL training objectives, the discrete action tokens cause autoregressive VLAs to struggle with dexterous manipulation. Coarse token design causes VLAs to lose dexterous control, while fine-grained tokenization reduces the discrimination between action tokens, significantly increasing the difficulty of action prediction.

2) Generative Action VLAs: To address the challenge of poor temporal consistency in discrete action prediction, recent researchers have focused on directly generating action trajectories utilizing diffusion-based [133]–[135] or flow-matching [5], [34], [136] action heads. However, generative action heads cannot provide explicit action prediction probabilities, resulting in difficulties in obtaining the optimization objective for generative RL-VLA. Recent studies have focused on reparameterizing the output of generative action heads to approximate action prediction probabilities, enabling RL to supervise VLAs. π_{RL} [25] employs Flow-SDE or Flow-Noise interventions to denoise the process, generating approximate probabilities for action assignments, which aligns with existing RL policy updates. To further enhance the training stability of RL-VLA, several researchers have investigated the impact of samples during training. **FPO** [49] utilizes the change in each sample to replace the action probability, which reduces the gap between the flow-matching head and the RL update policy while enhancing convergence stability. **ARFM** [50] proposes a dynamic scaling factor adjustment policy to update the weight of each sample, which enhances sample utilization efficiency and achieves more stable RL-VLA training.

Potential Challenges: Because generative VLAs rely on approximate density or loss-based agents that are tuned only on high-reward regions, their updates are driven by locally sampled and imperfect signals rather than by global action distributions. Consequently, small mismatches between the agent and the pre-trained behavior can accumulate across multi-step generation and iterative updates, eventually distorting or collapsing parts of the original action distribution.

3) Dual-system Model: To further enhance the ability of VLAs to understand human instructions and perform long-

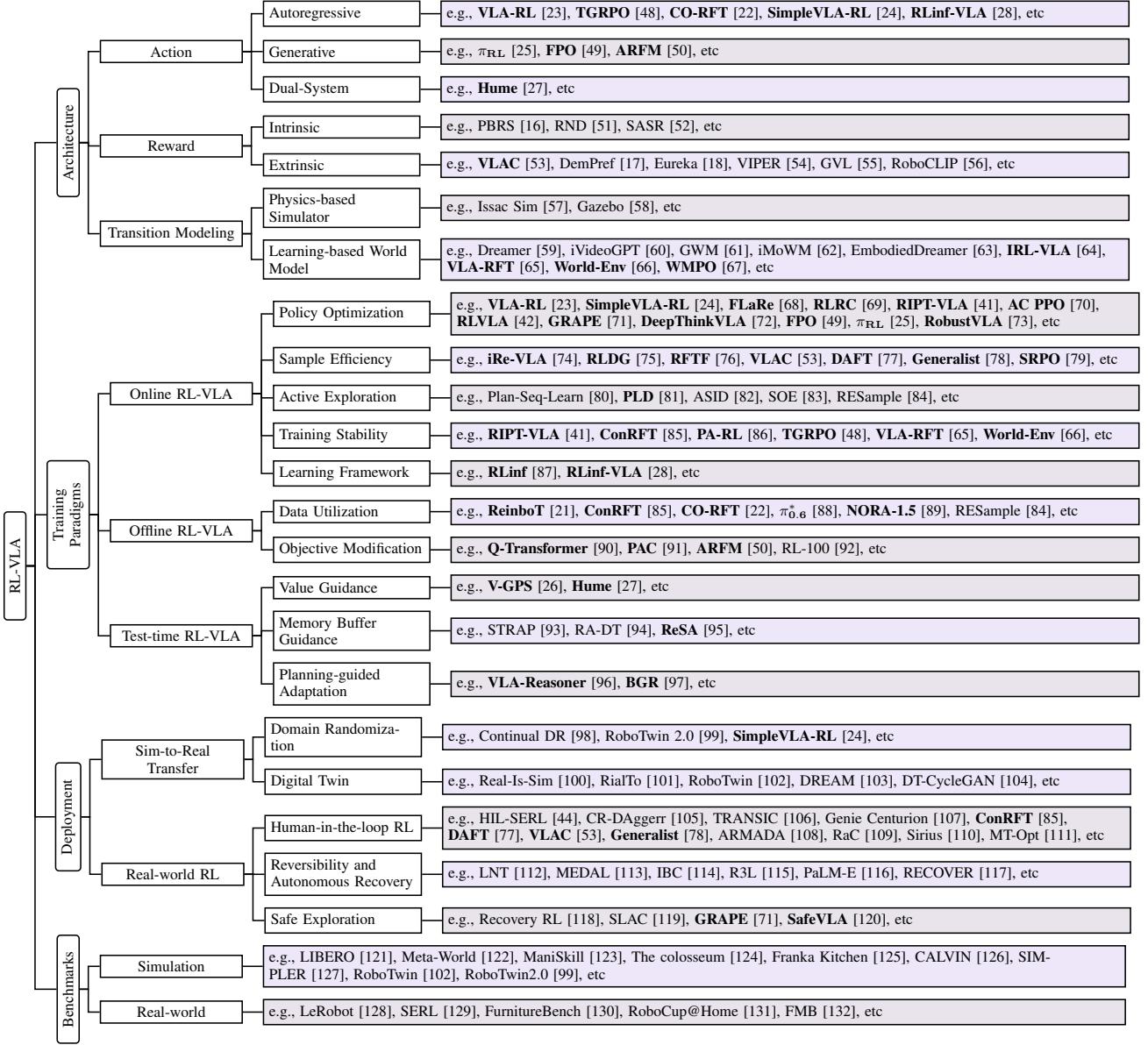


Fig. 1. Taxonomy of RL-VLA, covering the full RL optimization pipeline for VLA models. References in **bold** indicate works that directly advance RL-VLA development, while others represent complementary research in robot learning that informs future RL-VLA framework design.

horizon tasks, the dual-system VLM-VLA architecture has been proposed. Specifically, the high-level task planning VLMs understand human intent and generate step-by-step subtasks, while the low-level action control VLAs provide manipulation trajectories. However, Value misalignment between VLMs and VLAs leads to low dual-system performance. Recent researches employ RL to promote bidirectional value alignment between the two systems, ensuring that sub-tasks generated by VLMs are executable by VLAs. **Hume** [27] employs RL to train a high-level task planning system that selects optimal actions from multiple sampled actions, which significantly enhances the feasibility of low-level control.

Potential Challenges: A central challenge for dual-system VLAs is achieving reliable value alignment between the high-level VLM planner and the low-level VLA controller. Their heterogeneous representations and timescales often cause value estimates from language planning to diverge from

control-level returns, leading to unstable joint RL training and suboptimal coordination.

B. Reward

Reward in RL is the fundamental learning signal that quantifies task success and guides policy optimization, determining both the gradient variance and the convergence efficiency of the learned policy and shaping the overall learning dynamics. RL-VLA leverages the reward-driven feedback to overcome the limitations of imitation learning, enabling efficient generalization to OOD scenes. However, reward sparsity and delay cause VLA to struggle with policy optimization, which severely prevents the deployment of RL-VLA. To address the above challenges, recent approaches effectively guide policy optimization through constructing dense and informative reward signals, which can be broadly categorized into intrinsic rewards and extrinsic rewards based on their source.

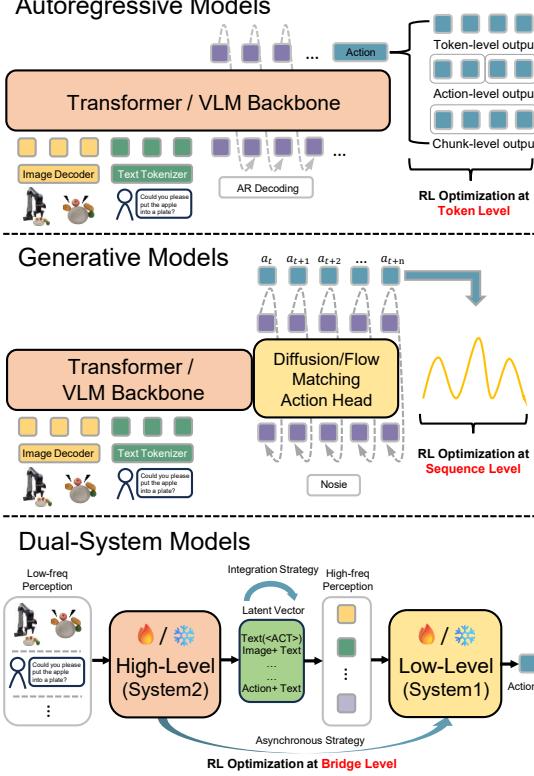


Fig. 2. **RL-VLA optimization of actions.** Autoregressive VLA optimizes actions at the token-level. Generative VLA optimizes along the action generation process at the sequence-level. Dual-system VLA optimizes at the bridge-level. RL decides which high-level action proposal to pass to the fast controller, so it complements the other two kinds of action policies.

1) **Intrinsic Rewards:** Intrinsic rewards are rule-based self-supervision signals derived from datasets or interactions between agents and the environment, encouraging agents to explore and construct their behavior. Through the provision of self-motivating feedback, intrinsic rewards enable RL-VLA to maintain stable learning and explore meaningful behaviors with sparse or delayed extrinsic rewards. Existing methods can be broadly categorized into potential-based reward shaping and exploration-driven rewards based on the reward motivation.

Potential-based Reward Shaping (PBRs): PBRs proposes to utilize auxiliary potential functions to modify the original reward signals into: $r'(s, a, s') = r(s, a, s') + \gamma\Phi(s') - \Phi(s)$, where Φ denotes the potential function [16], [137]. PBRs reshapes the reward pattern through potential differences, providing denser and more informative feedback without modifying the optimal policy, which significantly enhances training stability and convergence efficiency. The potential function in recent approaches can be either manually designed, using heuristic signals like distance-to-goal [138] and energy reduction, or learned from data such as approximated value functions [139] and latent progress estimators. The former provides interpretability and simplicity, whereas the latter offers adaptability but may introduce instability shaping.

Exploration-driven Rewards: Exploration-driven rewards encourage agents to explore novel or uncertain states by assigning additional intrinsic value to exploration action, which enhances the ability of agents to discover unknown

dynamics and avoids early convergence to suboptimal policies. Curiosity-driven approaches [140], [141] reward agents for visiting states with high prediction error, signaling novel environmental dynamics. **Random Network Distillation (RND)** [51], [142] scales this idea by measuring the agent’s familiarity with states through prediction error on a fixed random network. Count-based methods [52] similarly reward under-visited states to ensure systematic coverage of the state space.

Potential Challenges: Despite their autonomy, intrinsic rewards lack explicit alignment with task objectives. This can lead to improper and inconsistent behaviors such as reward hacking [143] and reward collapse in high-dimensional spaces [144], or policies that exploit easy sources of intrinsic reward without making task progress. Intrinsic rewards also rely on self-exploration and are inefficient in long-horizon manipulation tasks where most novel states are task-irrelevant.

2) **Extrinsic Rewards:** Extrinsic rewards involve utilizing external perceptual information, such as language instructions, visual observation, and human feedback, to guide agent behavior, which promotes alignment between the policy and the task objectives. Rather than relying on intrinsic rewards emerging from internal exploration dynamics, extrinsic rewards directly encode task objectives through external feedback or structured interpretations, leading to more grounded and interpretable policy optimization. Existing approaches can be broadly categorized based on their supervision source into human-aligned rewards and model-generated rewards.

Human-aligned Rewards: Human-aligned Reward represents human preferences, ensuring agents align with human values through policy updates to more effectively achieve human requirements. Reinforcement Learning from Human Feedback (RLHF) [145] trains a reward model on human preference comparisons between behavior pairs. **SEED** [146] applies RLHF to overcome reward sparsity through evaluative feedback. Beyond static preference datasets, interactive methods actively query humans during training. **DemPref** [17] iteratively queries preference labels on policy-generated trajectories, improving sample efficiency in the human feedback loop. **Sirius** [110] and **Transic** [106] enable humans to refine learned reward functions during training, providing human-in-the-loop reward shaping.

Model-generated Rewards: Model-generated reward primarily leverages pre-trained foundational models rather than human feedback, which aligns agent behavior with the commonsense from the foundation model and enables scalable supervision across diverse environments. **Reward Translator** [147] translates language instructions and interaction data into parameterized reward code, bridging natural language task specification to robot RL. **Eureka** [18] iteratively evolves reward code through LLM-generated proposals and environment feedback, often outperforming expert-designed rewards across diverse manipulation skills. Recent work [148] also demonstrates the potential to utilize LLMs/VLMs as proxies to learn effective reward functions without human intervention. **DVD** [149] learns multitask rewards by discriminating task similarity across human and robot videos, achieving zero-shot generalization. Video generative models provide an alternative approach: **VIPER** [54] learns a video prediction transformer

from expert demonstrations and uses the model’s likelihood as reward, while **TeViR** [150] employs text-to-video diffusion models to generate predicted image sequences and computes rewards by comparing them with actual observations. Other methods leverage VLMs for temporal reasoning and contrastive learning: **GVL** [55] formulates reward estimation as temporal ordering of video frames, **ReWiND** [151] augments task sequences with rewinding frames to improve robustness against failures, and **VLAC** [53] enhances interpretability through contrastive learning with negative samples. Query-based approaches such as **RoboCLIP** [56], **RL-VLM-F** [152], and **RG-VLM** [153] directly query VLMs to generate rewards from image observations and text task descriptions, demonstrating powerful generative ability to capture task progress. These methods share the core principle of rewarding distributional alignment: agents receive higher rewards when their behavior matches distributions learned from expert data or internet-scale videos.

Potential Challenges: Extrinsic rewards provide extrinsic task supervision but face persistent challenges in scalability, reliability, and alignment. They are prone to mis-specification for both human-aligned and model-generated rewards, domain shift, and perceptual noise, limiting their effectiveness in complex, real-world settings.

C. Transition Modeling

Transition modeling in RL aims to characterize environment dynamics conditioned on actions, enabling agents to infer the action sequences from the realistic physical consequences and corresponding rewards. As for RL-VLA systems, transition modeling further empowers VLAs to perform predictive rollouts and evaluate action sequences through simulators, addressing the limitations of conventional VLAs in reasoning over long-term dynamics and causal action–effect relationships. Existing transition modeling approaches can be divided into physics-based simulators and neural world models based on future prediction patterns.

1) *Physics-based Simulator*: Physics-based simulators explicitly replicate environment dynamics through accurate physical modeling, which defines the properties and interactions of each object in the environment to predict precise state transitions given specific actions. Existing research leverages object and scene assets, transferring real-world scene structures and object parameters into simulators. This alignment significantly enhances the transferability of RL-VLA policies from simulated to physical environments. By leveraging elaborate mechanics and dynamics engines, simulators such as **Isaac Sim** [57] and **Gazebo** [58] can precisely generate environment transitions conditioned on action sequences. **Potential Challenges:** Constructing high-fidelity simulators requires extensive human effort and accurate physical annotation, while the computational cost of physics-based rollouts remains high, limiting their scalability in data-hungry learning systems.

2) *Learning-based World Model*: Learning-based world models take a data-driven approach to transition modeling, learning to predict future states directly from large-scale manipulation demonstrations rather than relying on explicit

physical rules. World models encode the dynamics of the environment into latent representations or pixel-level observations and can generate plausible rollouts conditioned on current states and actions. Current world models fall into state-based methods, observation-based methods, and their integrated forms in VLA frameworks.

State-based Methods: State-based methods encode the environment into a compact latent state space to model transitions efficiently, which allows the model to predict long-horizon dynamics and rewards instead of reconstructing full visual observations. **PlaNet** [154] employs a Recurrent State-Space Model to construct a dynamic world model capable of predicting future latent states and rewards for action sequences. **Dreamer** [59] and **DreamerV2** [155] further improve the expressiveness of the latent state space, thereby enhancing the quality of long-horizon planning and the overall performance of model-based reinforcement learning (MBRL). **TransDreamer** [156] replaces the recurrent architecture with a transformer-based model to achieve more stable long-horizon prediction. However, these approaches treat image reconstruction as an auxiliary objective and therefore place limited emphasis on accurate visual observation modeling. As a result, they show limited generative capability in real-world settings and fail to fully exploit large-scale video data.

Observation-based Methods: Observation-based methods directly model pixel-level observation transitions, enabling the generation of realistic environment rollouts that preserve geometric and visual fidelity. This paradigm is better suited to representing real-world physics and enables rewards to be aligned with the visual predictions. **iVideoGPT** [60] leverages large autoregressive video prediction models [157] and fine-tunes a pretrained model for robotic scenarios. The pretrained visual world model, when combined with a learned reward model, can serve as a neural simulator for MBRL tasks. **GWM** [61] and **iMoWM** [62] incorporate multi-modal data to better represent the 3D geometric structure of the environment, thereby improving performance on MBRL tasks. These models demonstrate strong generalization ability across diverse tasks and enhance MBRL performance by improving both visual quality and reward prediction accuracy, since the reward is inherently coupled with visual fidelity in reflecting physical world understanding. However, relying solely on learning from real-world data distributions while ignoring strong physics priors reduces the reliability of these models, especially in complex or out-of-distribution scenarios. **EmbodiedDreamer** [63] addresses this limitation by introducing PhysAligner and VisAligner. PhysAligner incorporates physics-based simulator priors to provide physically consistent transition dynamics, while VisAligner uses video painting techniques to enhance the realism of generated observations. Although this approach improves physical accuracy, the involvement of a physics simulator reduces the computational efficiency that typically benefits learning-based world models.

VLA-designed Methods: Integrating world models into VLA frameworks bridges the gap between language-conditioned reasoning and physical environment understanding. The world model predicts future observation transitions, while the reward model evaluates and optimizes actions gen-

erated from VLAs through RL. **VLA-RFT** [65] generates multiple rollouts conditioned on VLA action sequences and employs the GRPO optimization framework [19] to update the VLA model using rewards predicted by the world model. **World-Env** [66] constructs an RL-VLA pipeline in which the VLA model generates action sequences, the world model predicts future observations, a vision-language model produces semantic reflections, and the LOOP optimization strategy [158] is applied for policy refinement. **WMPO** [67] proposes a world model-based policy optimization framework that enables reinforcement learning for VLA models by generating pixel-level imagined trajectories and optimizing policies via GRPO using rewards predicted by a learned reward model, all without interacting with the real environment.

Potential Challenges: Despite recent progress, world models still generalize poorly across diverse scenes, embodiments, and robot morphologies. Incorporating physics priors from human knowledge or high-fidelity simulators is crucial for improving reliability, yet balancing data-driven learning with physics-consistent dynamics remains a central challenge for building robust and transferable models.

IV. TRAINING PARADIGMS IN RL-VLA

RL training is a crucial step for enabling VLAs to generalize OOD from large-scale pre-trained data. Existing RL-VLA training paradigms can be categorized into three types based on the way agents obtain and utilize feedback from the environment: **Online RL-VLA**, which involves the direct interaction with the environment during training; **Offline RL-VLA**, which focuses on learning from static datasets without further environmental interaction; **Test-time RL-VLA**, where models adapt their behavior during deployment without altering their parameters. We also summarize the details of representative RL-VLA works in Table I.

A. Online RL-VLA

The Online RL-VLA paradigm enables interactive policy learning, where the agent continuously interacts with the environment to collect trajectories and update itself based on observed rewards and state transitions. The trial-and-error process of online RL-VLA empowers pre-trained VLAs with adaptive closed-loop control capability, effectively scaling VLAs to real-world OOD environments. Existing research on online RL-VLA has primarily advanced along 5 directions: policy optimization, sample efficiency, training stability, learning frameworks, and active exploration.

1) *Policy Optimization*: Policy optimization determines how a VLA updates its policy from environmental rewards, directly affecting stability and efficiency in online RL-VLA. Aggressive optimization can destabilize training in pre-trained action spaces, while overly conservative strategies demand excessive interaction and raise costs. Recent work mitigates this trade-off by adopting PPO variants that improve both learning efficiency and stability. On one hand, **FLaRe** [68] implements the PPO algorithm for post-training VLA models, serves as a foundational work in this area, followed by **RLRC** [69], which also utilizes PPO to fine-tune VLA models. More

recently, **RIPT-VLA** [41] combines Leave-One-Out (RLOO) advantage estimation with PPO for post-training, enabling efficient learning without shaped rewards or a value function. On the other hand, **VLA-RL** [23] fine-tunes autoregressive VLA models with the PPO algorithm while incorporating a Robotic Process Reward Model to provide dense rewards, thereby enhancing learning efficiency. **SimpleVLA-RL** [24] introduces GRPO to a more stabilized policy updating and earns significant performance improvements on the LIBERO benchmark. Crucially, an empirical study **RLVLA** [42] compares DPO, PPO, and GRPO algorithms for online RL fine-tuning of VLA models, providing compelling evidence that reinforcement learning fine-tuning substantially enhances generalization under OOD scenarios compared to standard supervised fine-tuning. **DeepThinkVLA** [72] introduces CoT with causal attention, and employs GRPO for policy optimization to causally align the full reasoning-action sequence with desired outcomes. For flow-matching-based VLA models, some works have also explored specialized policy optimization algorithms. **FPO** [49] proposes a Flow Policy Optimization algorithm that implements importance sampling in the flow-matching-based VLA models to improve policy optimization efficiency. π_{RL} [25] introduces two novel online RL algorithms for flow-matching based VLA models: Flow-Noise that models the denoising process as a discrete-time MDP, and Flow-SDE that integrates denoising with agent-environment interactions. Other works have also explored RL alignment. **GRAPE** [71] aligns VLAs with preferences by generating customized costs and optimizing policies on trajectory-wise data. **RobustVLA** [73] proposes a lightweight online RL post-training method to enhance the robustness and reliability of VLA models against perturbations through Jacobian regularization and smoothness regularization terms.

Potential Challenges: The diverse and dynamic demands of real-world tasks significantly expand the action space, which presents the challenge for policy optimization in online RL-VLA. Real-world environments demonstrate non-stationary dynamics and multimodal noise, rendering current policy optimization methods (typically designed for simulated or static benchmarks) struggling to maintain stable and reliable updates.

2) *Sample Efficiency*: Sample efficiency measures the ability of RL-VLA to learn effective policies under a limited budget, which is crucial for online RL-VLA, where interaction costs are expensive. Existing approaches focus on leveraging demonstration prior knowledge and designing more densely supervised signals. **RLDG** [75] combines human expert demonstrations with online RL fine-tuning to enhance sample efficiency. By distilling knowledge from a generalist policy trained on diverse datasets, RLDG accelerates learning in new tasks through targeted exploration and exploitation of prior knowledge. **iRe-VLA** [74] leverages a two-stage training process, where an initial phase of supervised fine-tuning warmup is followed by online RL, significantly reducing the number of interactions required to achieve proficient performance in complex manipulation tasks. **VLAC** [53] integrates actor-critic architecture within one single VLM model, enabling action generation along with dense progress delta and done signal prediction, which greatly improves sample efficiency during

TABLE I
A SUMMARY OF EXISTING RL-VLA WORKS.

- Action:** AR: Autoregressive, Diffusion: Diffusion, Flow: Flow-matching.
- Reward:** D: Dense Reward, S: Sparse Reward.
- Algorithm:** CQL: Conservative Q-Learning, AC: Actor-Critic, Cal-QL: Calibrated Q-Learning, BC: Behavior Cloning, DT: Decision Transformer, RTG: ReturnToGo, TD3: Twin Delayed Deep Deterministic Policy Gradient, ARFM: Adaptive Reinforced Flow Matching, PA-RL: Policy-Agnostic RL, SAC/FD: Soft Actor-Critic from Demonstrations, TPO: Trajectory-wise Preference Optimization, LOOP: Leave-One-Out Proximal Policy Optimization, GRPO: Group Relative Policy Optimization, DPO: Direct Preference Optimization, FPO: Flow Policy Optimization, SAC: Soft Actor-Critic.
- Type:** MB: Model-based, MF: Model-free.

Paradigm	Method	Date ↑	Environment Sim. Real	Base VLA Model	Action	Reward	Algorithm	Type
Offline RL	Q-Transformer [90]	202310	✓ ✘	Transformer	AR	S	CQL [159]	MF
	PAC [91]	202402	✓ ✓	Perceiver-Actor-Critic	AR	S	AC	MF
	ConRFT [85]	202504	✗ ✓	Octo-small [133]	Diffusion	S	Cal-QL + BC	MF
	ReinboT [21]	202505	✓ ✓	ReinboT	AR	D	DT + RTG	MF
	CO-RFT [22]	202508	✗ ✓	RoboVLMs [160]	AR	D	Cal-QL + TD3 [161]	MF
	ARFM [50]	202509	✓ ✓	π_0	Flow	D	ARFM	MF
	$\pi_{0.6}^*$ [88]	202511	✗ ✓	$\pi_{0.6}$	Flow	D	RECAP	MF
Online RL	NORA-1.5 [89]	202511	✓ ✓	NORA-1.5	AR / Flow	D	DPO	MB
	FLaRe [68]	202409	✓ ✓	SPOC [162]	AR	S	PPO	MF
	PA-RL [86]	202412	✓ ✓	OpenVLA	AR	S	PA-RL	MF
	RLDG [75]	202412	✗ ✓	OpenVLA / Octo	AR / Diffusion	S	RLPD [163]	MF
	iRe-VLA [74]	202501	✓ ✓	iRe-VLA		S	SACID [164] + SFT	MF
	GRAPE [71]	202502	✓ ✓	OpenVLA	AR	D	TPO	MF
	SafeVLA [120]	202503	✓ ✘	SPOC	AR	S	PPO	MF
	ConRFT [85]	202504	✗ ✓	Octo-small	Diffusion	S	Cal-QL + BC	MF
	RIPT-VLA [41]	202505	✓ ✘	QueST [165] / OpenVLA-OFT		AR	LOOP [158]	MF
	VLA-RL [23]	202505	✓ ✘	OpenVLA		D	PPO	MF
	RLVLA [42]	202505	✓ ✘	OpenVLA		S	PPO / GRPO / DPO [166]	MF
	RFTF [76]	202505	✓ ✘	GR-MG [167], Seer [168]		AR	PPO	MF
	TGRPO [48]	202506	✓ ✘	OpenVLA	AR	D	GRPO	MF
	RLRC [69]	202506	✓ ✘	OpenVLA	AR	S	PPO	MF
	SimpleVLA-RL [24]	202509	✓ ✓	OpenVLA-OFT	AR	S	GRPO	MF
	Dual-Actor FT [77]	202509	✓ ✓	Octo / SmolVLA	Diffusion	S	QL + BC	MF
	Generalist [78]	202509	✓ ✓	PaLI 3B [169]		D	REINFORCE [170]	MF
	VLAC [53]	202509	✗ ✓	VLAC	AR	D	PPO	MF
	AC PPO [70]	202509	✓ ✘	Octo-small	AR	S	PPO+BC	MF
	VLA-RFT [65]	202510	✓ ✘	VLA-Adapter [171]	Flow	D	GRPO	MF
	RLInf-VLA [28]	202510	✓ ✓	OpenVLA / OpenVLA-OFT	AR	S	PPO / GRPO	MF
	FPO [49]	202510	✓ ✘	π_0	Flow	S	FPO	MF
	ReSA [95]	202510	✓ ✘	OpenVLA	AR	D	PPO + SFT	MF
DeepThinkVLA [72]	π_{RL} [25]	202510	✓ ✘	$\pi_0 / \pi_{0.5}$	Flow	S	PPO / GRPO	MF
	PLD [81]	202510	✓ ✓	OpenVLA / π_0 / Octo	AR / Flow	S	Cal-QL + SAC	MF
	World-Env [66]	202510	✓ ✘	π_0 -Fast		S	GRPO	MF
	RobustVLA [73]	202511	✓ ✘	OpenVLA-OFT	AR	D	PPO	MF
	WMPO [67]	202511	✓ ✓	OpenVLA-OFT	AR	S	GRPO	MF
	SRPO [79]	202511	✓ ✓	OpenVLA* / π_0 / π -Fast	AR / Flow	D	SRPO	MF
	V-GPS [26]	202410	✓ ✓	Octo / RT-1 / OpenVLA	AR	D	Cal-QL	MF
	Hume [27]	202506	✓ ✓	Hume	Flow	S	Value Guidance	MF
Test-time RL	VLA-Reasoner [96]	202509	✓ ✓	OpenVLA / SpatialVLA [172] et al.	AR / Diffusion	D	MCTS	MB

online RL fine-tuning. **DAFT** [77] introduces human feedback to intervene in the exploration process and construct a language-intervention pair dataset, greatly accelerating policy learning in online RL-VLA. **Generalist** [78] proposes a multi-stage training pipeline that combines SFT and online RL self-improvement, where a well-shaped reward function is utilized to ensure unsupervised learning. **SRPO** [79] introduces a self-referential RL framework that leverages the policy's own successful trajectories as a self-reference to get progressive rewards, no need for reward labeling.

Potential Challenges: Despite existing efforts having achieved advances, sample efficiency in current online RL-VLA approaches remains limited in scalability and generalization. Most approaches focus on improving data utilization within specific tasks or environments, failing to efficiently transfer learned behaviors across diverse goals or domains, which prevents large-scale RL-VLA from leveraging shared experience to scale up learning in new environments.

3) *Active Exploration*: Active exploration aims to design efficient exploration policies that guide agents to rollout ac-

tion samples with higher performance gains, addressing the redundant costs introduced by random rollouts in conventional RL-VLA. Existing approaches can utilize semantic, latent-level, and information gap to guide exploration policies. **Plan-Seq-Learn** [80] uses an LLM to produce high-level task plans, turns them into motion-planning way-points, and trains a low-level vision-based RL policy to follow those way-points, thereby guiding exploration action towards task-relevant spaces. **SIME** [173] introduces modal-level exploration at the RL fine-tuning stage, so the robot can generate diverse, multi-modal interaction behaviors in the reasoning space beyond the pre-trained policy's typical output. **SOE** [83] learns a latent representation of task-relevant factors and constrains exploration to the manifold of valid actions, ensuring safety, diversity, and effectiveness. **ASID** [82] uses an active exploration policy to efficiently collect small amounts of informative real-world data, identifying unknown physical parameters of the environment, which creates a more accurate simulator for training a robust control policy. **RESample** [84] automatically generates challenging out-of-distribution data,

using exploratory sampling to create failure and recovery trajectories, forcing the model to learn how to recover from errors it would not see in standard offline datasets. **PLD** [81] employs a hybrid rollout scheme that biases residual interventions toward states frequently visited by the base policy, aligning collected trajectories with the generalist’s deployment distribution while capturing recovery behaviors.

Potential Challenges: Current methods generate exploration strategies from latent representations that are high-dimensional and often contaminated by irrelevant noise, reducing the quality of guidance and limiting exploration effectiveness. Moreover, safe active exploration remains difficult in real-world settings, as such guidance is neither fully explainable nor constrained, leading to potentially risky behaviors that may harm the environment or the robot.

4) *Training Stability*: Stable online RL-VLA training ensures policy update consistency, preventing poor generalization caused by oscillatory convergence during policy training. Existing research primarily achieves stable online training by scaling sample buffer size and reducing sample distribution variance. **RIPT-VLA** [41] leverages Dynamic Rollout Sampling, a rejection sampling mechanism that addresses the instability caused by high variance in rollout returns during online RL fine-tuning. **ConRFT** [85] introduces offline RL pre-training to stabilize the initial policy before online fine-tuning with HIL-SERL [44] framework. Similar to this, **PA-RL** [86] proposes a unified framework that directly optimizes action and online fine-tunes from a universal loss function, thereby decouples policy improvement from model parameter updates and enhances training stability. **TGRPO** [48] takes a step further by providing trajectory-level estimation, which is called Trajectory-wise Group Relative Policy Optimization, to reduce variance in policy updates and enhance stability during training. Another way to improve training stability is through world-model-based RL. **World-Env** [66] and **VLA-RFT** [65] both utilize learned world models as simulators to generate synthetic rollouts, thereby reducing the variance and instability associated with real-world interactions.

Potential Challenges: Existing approaches for stable online RL-VLA training remain limited to simple short-horizon manipulation tasks (e.g., picking up the block), which cannot be scaled to complex long-horizon tasks (e.g., making a sandwich). This is because long-horizon tasks require the VLA to maintain consistent interactions across the time sequence, where any single error can lead to failure. As task complexity and temporal depth improve, the difficulty of achieving training stability increases significantly.

5) *Online RL-VLA Infrastructure*: Inspired by the promising performance of RL in fine-tuning LLMs/VLMs, the latest research has also investigated the infrastructure of online RL-VLA learning pipelines. **RLinF** [87], along with **RLinF-VLA** [28], propose a flexible infrastructure that supports efficient online RL fine-tuning of large-scale VLA models, supporting various policy optimization algorithms and model architectures. These frameworks improve training efficiency and enable the incorporation of additional learning signals such as human feedback and safety constraints. Learning frameworks from LLMs, including vLLM [174] and VeRL [175], have also

been adapted to VLAs to further enhance their capabilities.

Potential Challenges: Existing online RL-VLA infrastructures are often bound to specific architectures or optimization methods, limiting cross-framework adaptability. Differences in reward acquisition between autoregressive and generative VLAs further complicate unified support. While some LLM-based RL systems have been adapted to VLAs, transferring them remains difficult due to multimodal observations, real-time control, and physical constraints. Therefore, building a more general and flexible online RL-VLA infrastructure remains an open challenge.

B. Offline RL-VLA

Offline RL trains VLA models on static datasets without interacting with the environment, making it suitable for high-risk or resource-limited settings. Unlike IL-based VLAs that simply mimic demonstrations, offline RL-VLA optimizes long-term rewards from diverse past experiences, improving OOD generalization. This requires large datasets with full MDP tuples [176]–[178], yet many available datasets originate from IL or SFT pipelines and lack rich rewards, dynamics, and failure cases, limiting value estimation and policy generalization. Current offline RL-VLA research mainly advances along two directions: data utilization and objective modification.

1) *Data Utilization*: Data utilization focuses on the effective utilization of static datasets for policy improvement under the constraints of offline learning. Since new interactions cannot be collected, the effectiveness of offline RL-VLA largely depends on how the training algorithm leverages available trajectories to approximate optimal policies. Existing research mainly explores two complementary directions: customized representation to enhance the reward and conservative constraint to ensure stability.

Customized Representation: Customized representation approaches actively adapt offline datasets or associated reward signals to better align with policy optimization objectives. Through reshaping trajectories or generating task-specific costs, existing approaches enable VLA models to extract more informative training signals from a static dataset. **ReinboT** [21] enhances VLA performance by modifying the offline dataset to maximize cumulative rewards, achieving more robust decision-making than standard behavior cloning. $\pi_{0,6}^*$ [88] conditions VLA with a binarized value via a pretrained value function, utilizing both failure and success data. **NORA-1.5** [89] introduces offline Direct Preference Optimization to optimize VLA with model-generated rewards.

Conservative Constraint: Conservative constraint methods restrict policy updates to prevent deviation from the data distribution covered by the offline dataset. By limiting extrapolation to unseen states or actions, existing methods reduce distribution shifts, which leads to more reliable policy learning with improved utilization of static data. **ConRFT** [85] integrates behavior cloning with **Cal-QL** [179] to stable value estimation for learning from small datasets. Similarly, **Co-RFT** [22] utilizes the calibration mechanism of **Cal-QL** to constrain the policy training to be supported by training data, thereby mitigating distributional shift.

Potential Challenges: Challenges in offline RL-VLA often stem from dataset curation. Because VLA policies depend heavily on data quality and structure, unbalanced datasets worsen distributional shift and restrict learning. Without careful curation, offline datasets exhibit uneven task coverage, biased behavior distributions, and incomplete reward signals, leading to poor generalization in OOD environments.

2) *Objective Modification:* Objective modification methods adjust the RL objective to align learning signals with new architectures or to support dataset augmentation. Existing work primarily explores architecture-aware objective design and data-driven objective adaptation.

Architecture-aware Objective Design: As VLAs increasingly incorporate diverse architectures, it is crucial to design RL objectives tailored to diverse structures to unlock their full potential. Inspired by the success of deploying offline RL on transformer-based models like **Q-Transformer** and **PAC** [90], [91], current research focuses on optimizing models in various structures through offline RL. **ARFM** [50] introduces a flow-based training objective for offline RL on VLAs, controlling the RL influence via a self-adaptive balancing factor.

Data-driven Objective Adaptation: These methods leverage RL objectives to augment offline datasets, producing additional high-quality trajectories to improve following VLAs optimization, which enhances offline dataset diversity and coverage. **RL-100** [92] employs an offline RL objective to conservatively gate an online PPO agent, generating new, high-quality data, which is a technique potentially applicable for post-training VLAs. Another strategy involves model-based offline RL (MBRL), where a dynamics model learned from the static dataset is used to generate synthetic rollouts.

Potential Challenges: Although architecture-aware objectives expand the capabilities of different model structures, they add complexity and transfer poorly without a unified offline RL-VLA framework. Data-driven objective adaptation also risks distributional drift, as inaccurate generators can produce low-quality samples that degrade the training buffer and destabilize learning. These limitations underscore the need for more generalizable objectives and data curation strategies.

C. Test-time RL-VLA

Test-time RL-VLA training paradigm refers to VLAs adapting their behavior during deployment through lightweight updates or adapter modules, effectively addressing the expensive cost of full model fine-tuning in real-world deployment. This training paradigm empowers VLAs to rapidly adapt to novel states, improving both robustness and generalization without extensive training. Existing methods can be broadly categorized based on their adaptation mechanism into value guidance, memory buffer guidance, and structured planning.

Value Guidance: Value guidance approaches adapt the VLAs at test time by leveraging pretrained reward or value functions to directly influence action selection, allowing efficient adjustment to novel tasks without full policy updates. For example, **V-GPS** [26] leverages a pre-trained value function to re-rank action candidates from the base policy, ultimately selecting the one with the highest predicted value to adjust

the model behavior towards optimality. **Hume** [27] framework introduces a "value-guided thinking" process as part of a dual-system architecture. It generates multiple action candidates and employs a specialized value-query head to select the most promising one based on estimated state-action values.

Memory Buffer Guidance: To further improve the effectiveness of exploration at test time, recent works propose memory buffer guidance that retrieves relevant historical experiences during inference, improving exploration efficiency and knowledge reuse. **STRAP** [93] implements a compact and expressive pattern library that stores representative spatio-temporal patterns enriched with historical, structural, and semantic information, and retrieves sub-segments of trajectories based on similarity to the current input during inference. **RA-DT** [94] stores external memory of past experiences and retrieves only relevant sub-trajectories for in-context decision making. **ReSA** [95] identifies and selectively imitates high-quality successful trajectories from the replay buffer through intrinsic quality assessment, ensuring the agent remains aligned with the ultimate task goal.

Planning-guided Adaptation: Planning-guided adaptation methods improve test-time performance by explicitly reasoning possible future action sequences to select actions that are more likely to achieve task objectives, leveraging the base VLAs as initial proposals, and refining actions through simulated rollout or value-based evaluation. **VLA-Reasoner** [96] proposed a plug-in framework that enhances VLA models with planning capabilities at test-time. It utilizes an online Monte Carlo Tree Search (MCTS) that takes the base policy's initial action prediction as a starting point for exploration, effectively searching for a more optimal action by simulating future outcomes. An alternative use of the value function is for progress monitoring rather than proactive action selection. Bellman-Guided Retrials (**BGR**) [97] exemplifies this, employing a separately trained value function that estimates time-to-completion. At test-time, this function continuously monitors for inconsistencies in its own predictions, allowing it to detect when the robot deviates from a successful trajectory and trigger corrective actions.

Potential Challenges: Existing planning-guided adaptation methods require pre-inference of future action sequences, which introduces significant computational costs and limits real-time deployment. The additional need to evaluate large sets of action candidates further increases overhead, reducing responsiveness in dynamic environments.

V. REAL-WORLD DEPLOYMENT

Real-world deployment refers to running RL-VLA models on physical robots under real-world dynamics, enabling safe and autonomous operation in unstructured environments. Recent work leverages **Sim-to-Real Transfer** and **Real-world RL** based on the source of training interactions to address challenges in sample efficiency, safety, and hardware constraints.

A. Sim-to-Real Transfer

Sim-to-real transfer enables VLAs trained in simulation to be effectively generalized to physical robots, addressing the

distribution shift problem. Current approaches to fill the sim-to-real gap can be broadly categorized into domain randomization and digital twin, which achieve efficient transfer in both perception and environment dynamics.

Domain Randomization: Domain Randomization (DR) employs random simulation parameters to approach the perception diversity encountered in real-world deployments, expecting to reduce the gap between simulation and real-world. Specifically, DR [180] embraces uncertainty by randomizing a broad set of simulation parameters, such as lighting conditions, background texture, and actuator noise, during policy training [98] and data collection [99]. For instance, **SimpleVLA-RL** [24] demonstrates that applying DR across diverse task simulations allows policies to achieve zero-shot transfer to real robots without requiring additional fine-tuning.

Digital Twin: Digital Twins (DTs) create synchronized virtual replicas of physical systems, enabling safe and scalable policy training while reducing the sim-to-real gap. **Real-Is-Sim** [100] maintains a dynamic DT continuously corrected with real sensor streams, ensuring policies always operate within familiar simulator-domain states. **RialTo** [101] builds on-the-fly simulations from minimal real data and employs inverse-distillation RL to robustify manipulation policies. **RoboTwin** [102] uses a generative framework, leveraging 3D generative models and LLMs, to convert single 2D images into diverse, interactive DTs, serving as a benchmark for dual-arm manipulation. Moreover, **DT-CycleGAN** [104] combines digital twin with CycleGAN [181] to minimize visual and action consistency gaps between simulated and real robots to enable effective zero-shot sim-to-real transfer for visual grasping. Finally, **DREAM** [103] presents a real-to-sim-to-real framework using differentiable Gaussian Splat to create high-fidelity DTs, enabling the simultaneous identification of object mass and force-aware grasping policies training.

Potential Challenges: Despite notable progress, transferred policies still underperform their simulated counterparts. For example, SimpleVLA-RL [24] shows a substantial sim-to-real gap, with much lower success rates on physical robots than in simulation. This indicates that simulation alone is insufficient for reliable real-world VLA deployment.

B. Real-world RL

Real-world RL aims to train manipulation policies directly on physical robots, enabling them to acquire skills that operate reliably under real sensor feedback and physical dynamics. Compared with simulation, real-world RL provides more realistic learning signals but also introduces significant challenges due to limited rollout efficiency and safety risks. Existing approaches have proposed human-in-the-loop RL, reversibility, autonomous recovery, and safe exploration to address the challenges of sample efficiency, environment reset, and safety.

1) *Human-in-the-loop RL*: Human-in-the-loop (HiL) RL integrates human expertise into the policy learning process to accelerate real-world RL. Instead of purely autonomous exploration, HiL approaches leverage human interventions to correct robot actions and schedule learning tasks for faster policy convergence. Empirical studies demonstrate that incorporating

human expertise through human corrective interventions [44], [77], human recovery assistance, or human curriculum design [23], [111] helps stabilize learning, reduce unsafe exploration, and accelerate convergence, bridging the gap between brittle autonomous learning and the structured adaptability required in physical environments.

Human Corrective Intervention: Human corrective intervention uses real-time feedback to guide robots during learning, enabling faster skill acquisition and safer exploration through targeted corrections that help recover from errors and refine complex behaviors. **HIL-SERL** [44] introduces human-in-the-loop reinforcement learning, where systems leverage human corrective feedback to rapidly acquire precise and dexterous manipulation skills. **CR-Dagger** [105] introduces a compliant, force-sensitive interface for smooth human corrections and learns a residual policy using force feedback to enhance contact-rich manipulation. **TRANSIC** [106] is proposed as a sim-to-real framework that learns from online human corrections for adaptive policy transfer, and **Genie Centurion** [107] scales corrective intervention across multiple robots by detecting task failures via a VLM and requesting human assistance when necessary. Most recently, **ConRFT** [85] pioneered the integration of human-in-the-loop interventions into RL for VLA in real-world robotic manipulation by combining offline and online human-corrected RL fine-tuning. **DAFT** [77] extends corrective intervention into RL for VLA, converting natural-language feedback into semantically grounded corrective actions. **VLAC** [53] further explores human-guided exploration, where multi-robots learn key behaviors in real environments under human supervision, accelerating policy adaptation and stability.

Human Recovery Assistance: Refers to manual intervention required to reset robots or environments after failures when autonomous recovery is unreliable in real-world RL. Early robotic RL studies [182], [183] relied heavily on frequent manual resets, severely limiting the scalability of long-term, contact-rich manipulation tasks where accurate environment resets are important. To alleviate this, subsequent works explored semi-automated recovery pipelines that blend human-in-the-loop with scripted resets or motion primitives, enabling autonomous recovery under human oversight while retaining safety guarantees [184]. **ARMADA** [108] and **RaC** [109] further integrate learning-based recovery modules, allowing the robot to detect failure states and request human-guided recovery while self-recovery is infeasible. Most recently, in real-world RL-VLA, **Generalist** [78] minimizes human intervention by restricting assistance to performing resets only when the robot enters irreversible states or fails to complete a task within a prolonged period. Similarly, **VLAC** [53] engages human to observe where the VLA policy frequently fails and manually reset the robot and objects, enabling targeted reinitialization for continued RL exploration.

Human Curriculum Task Design: This approach applies curriculum learning principles, structuring tasks from simple to complex to facilitate stable and efficient policy acquisition. In real-world RL, human supervisors design curricula by selecting sub-tasks or adjusting difficulty boundaries to balance safety and learning efficiency [185]. Recent ad-

vances move toward semi-automated curricula, where large language models assist humans in decomposing complex tasks. **CurricuLLM** [186] leverages large language models to automatically decompose complex robot skills into hierarchical sub-tasks, aligning task progression with human-specified difficulty levels. Meanwhile, **Sirius** [110] introduces a human-in-the-loop autonomy framework in which human operators dynamically design and gate deployment curricula across real-world tasks, determining which skills can be safely attempted without intervention. At fleet scale, **MT-Opt** [111] operationalizes curriculum task design by prioritizing low-performing skills and controlling deployment thresholds based on performance metrics. **VLA-RL** [23] incorporates human curriculum design principles into RL for VLA post-training in simulation. However, curriculum task design in real-world RL-VLA remains largely unexplored, especially regarding how human instructors can structure multi-modal objectives and deployment thresholds under physical constraints.

Potential Challenges: Existing human-in-the-loop RL methods still depend heavily on human intervention samples to ensure safe and stable learning, resulting in high labor costs and poor scalability [187]. This reliance limits continuous training and hinders large-scale deployment in real-world settings, highlighting the need for more autonomous, self-sustaining learning mechanisms.

2) *Reversibility and Autonomous Recovery*: Reversibility and autonomous recovery enable robots to self-handle failure states and continue learning without external intervention, reducing manual resets and labor costs in real-world RL. With autonomous recovery to feasible states after task failures, robots maintain continuous interaction with the environment, which improves sample efficiency and long-term adaptability. Existing approaches can be categorized into Reset-free Learning, Functional Reversibility, and Semantic-aware Recovery based on recovery mechanisms.

Reset-free Learning: Reset-free learning aims to avoid external resets by encouraging agents to remain within recoverable regions of the state space. A natural approach introduces an auxiliary *reset policy* that drives the agent back to some initial states after failure, enabling continued training without human assistance. **LNT** [112] trains goal-conditioned reset policies to restore agents to the initial state distribution. **VaPRL** [188] incorporates curriculum learning to handle increasingly challenging tasks and **LSR** [189] promotes skill diversity through a discriminator-driven learning scheme, while **MEDAL** [113] employs demonstrations to guide both task and reset policies in a unified framework. **IBC** [114] extends this idea by learning reset goals directly from demonstrations without dense supervision. Beyond explicit reset policies, **R3L** [115] adopts multi-start training strategies, allowing the agent to revisit diverse initial conditions and thus increase robustness against exploration failures. **MTRF** [190] views reset-free RL as a multi-task learning problem, where tasks are designed such that their terminal states serve as valid initial states for other tasks.

Functional Reversibility: Functional reversibility emphasizes the robot’s ability to reverse its actions and restore the environment to a recoverable or task-continuable state

after disruptions. Some approaches learn recovery skills that handle common failure cases, such as object drops or grasp slippage so that progress toward task goals can resume after interruptions [191]. Sharma et al. proposed State Entropy Maximization [113] to encourage diverse yet reversible exploration through intrinsic regularization. **Recovery RL** [118] learns a recovery policy that intervenes to prevent the robot from entering unsafe or irreversible states. **PAINT** (Proactive Agent Interventions) [192] extends this approach by training a classifier to predict potential failures and trigger corrective actions or safe resets in advance. Beyond explicit reversibility labeling, Lynch et al. [191] learn recovery skills for manipulation failures like object drops or grasp slippage, enabling seamless resumption of task progress. Recent multimodal models further integrate semantic guidance into recovery—language-conditioned policies like **PaLM-E** [116] allow robots to generate corrective behaviors from high-level instructions.

Semantic-aware Recovery: Semantic-aware recovery emphasizes reasoning about manipulation temporal-spatial dynamics, enabling robots to interpret failure causes and plan appropriate recovery behaviors within the context of ongoing tasks. For instance, Matsuoka et al. [193] build a failure ontology with time-dependent utility to choose recovery actions during slippage or displacement. **DAS** [194] uses semantic scene graphs to interpret spatial and relational failure contexts for corrective planning. **RECOVER** [117] combines ontology, logic, and language models to detect failures online and produce interpretable recovery plans. Ahmad et al. [195] integrate vision-language models and behavior trees for real-time reasoning and autonomous correction.

Potential Challenges: Although autonomous failure recovery is feasible, reversibility and recovery remain difficult due to unstable long-horizon training, partial observability, and the inherent irreversibility of real-world interactions. These issues hinder reliable failure detection, causal reasoning, and recovery execution in complex environments [188].

3) *Safe Exploration*: Real-world RL must ensure that the agent’s experience-gathering process avoids unsafe interactions with the physical environment. Safe exploration constrains policy search to task-relevant and reversible regions of the state space, ensuring effective learning while avoiding catastrophic outcomes. Existing approaches can be broadly categorized into conservative safety critics, structured task decomposition, and real-time safety enforcement, which achieves a trade-off between safety assurance and learning efficiency.

Conservative Safety Critics: Conservative safety critics provide a principled mechanism for evaluating the risk of action proposals during real-world exploration, which typically trains an auxiliary critic to estimate the likelihood of safety constraint violations. **Recovery RL** [118] introduces learned recovery zones: regions in state space from which the robot can safely return to nominal operation. In addition, by pre-training a task-agnostic latent action space in low-fidelity simulation, **SLAC** [119] establishes a safe and temporally structured behavioral prior that constrains real-world exploration.

Structured Task Decomposition: Structured task decomposition breaks complex robotic training tasks into simpler sub-tasks for manageable safety inspection. For instance,

GRAPE [71] leverages a vision-language model to decompose complex manipulation tasks into interpretable stages and automatically derive spatiotemporal safety constraints using semantic keypoints. Similarly, [196] employs a critic mechanism to probabilistically ascertain the satisfaction of safety constraints throughout the training process.

Real-time Safety Enforcement: Real-time safety enforcement ensures safe RL exploration by applying control-theoretic safety constraints directly at the execution level. For instance, impedance controllers [44] with reference limiting can bound end-effector forces and velocities in real time, preventing unsafe contact even when the RL policy proposes aggressive actions. Further, leveraging the Constrained Markov Decision Process (CMDP) paradigm, **SafeVLA** [120], an Integrated Safety Approach (ISA), optimizes VLAs from a min-max perspective against elicited safety risks, thereby achieving a safety-performance trade-off.

Potential Challenges: A persistent challenge is the integration of high-level semantic reasoning with low-level safety guarantees. Current frameworks struggle to connect abstract, semantic-based rules (e.g., handle fragile objects with care) with the concrete, low-level control policies that must enforce physical constraints (e.g., specific torque or velocity limits). This disconnect is particularly critical under a distribution shift. When the agent encounters novel states, it becomes exceptionally difficult to ensure that both semantic goals and physical safety are simultaneously satisfied.

VI. BENCHMARKING AND EVALUATION

Benchmarking and evaluation are critical for assessing progress in RL-VLA. Due to the embodied and multimodal nature of these systems, standardized evaluation must jointly consider algorithmic performance, interaction fidelity, and real-world deployability. Existing efforts can be broadly divided into **Simulation-based** and **Real-world Benchmarks**, complemented by a set of **Evaluation Metrics** that quantify efficiency, safety, and autonomy. This section reviews representative benchmarks and datasets across both domains and summarizes common metrics that support reproducible and scalable evaluation of RL-VLA systems.

A. Simulation Datasets and Benchmarks

The optimization of robotic foundation models through RL necessitates substantial interaction with the environment. To support this process, a wide range of simulation benchmarks are employed to evaluate the effectiveness of RL algorithms. These benchmarks commonly encompass synthetic scenarios, physics-based simulations, object manipulation tasks, task execution processes, and agent-environment interactions.

Unimanual Manipulation Benchmarks: The overview of major manipulation benchmarks is in Table II. Benchmarks are critical for scalable RL training in VLAs, where parallel environments enable fast rollout collection and reduced wall-clock learning time, a Gym-like API ensures standardized and reproducible evaluations, and GPU-accelerated rendering provides high-fidelity observations and realistic dynamics at scale. **LIBERO** [121] and **Meta-World** [122] are two

commonly used suites that support RL training on multi-task and long-horizon control. **ManiSkill** [123], [197] is a contact-rich, physically accurate simulation suite paired with expert demonstrations to scale RL data. **BEHAVIOR** [198] and **RoboVerse** [199] both provide a large scale of diverse tasks for scalable RL training and challenging tests. Focusing on household scenarios, **RoboCasa** [200] provides realistic scenes to enable robots to learn robust and transferable policies. **The Colosseum** [124] is built on RLBench [201], sweeping various perturbation axes (e.g., appearance, lighting), while **Franka Kitchen** [125] uses a MuJoCo [202] kitchen to study long-horizon manipulation by composing atomic goals such as turning oven burners and opening cabinets. **CALVIN** [126] provides a standard evaluation that trains on three environments and tests on the remaining one that supports RL training and tests cross-scene generalization. **SIMPLER** [127] provides simulated environments expressly calibrated to mirror common robot setups and narrow gaps without full digital twins, and paired sim-and-real studies report strong correlation between SIMPLER performance and real-robot success.

Bimanual Manipulation Benchmarks: **RoboTwin** [102] is a bimanual manipulation benchmark that pairs real teleoperation demonstrations with generative digital-twin scenes from single images, providing aligned sim/real task definitions and metrics. Furthermore, **RoboTwin2.0** [99] scales the platform with an automated expert data pipeline, a curated object set of multiple instances, and a dual-arm benchmark spanning multiple robot embodiments. Structured domain randomization (e.g., clutter, lighting, background) and a multimodal LLM strengthen the robustness of RL training.

B. Real-world Datasets and Benchmarks

The advancement of VLA algorithms depends significantly on the availability of high-quality real-world datasets. Over time, researchers have assembled diverse datasets that span multiple sensor modalities, tasks, and environmental conditions, forming a critical resource for driving progress in real-world robotics. These datasets capture multimodal interactions between robots and the environment, including sensory inputs such as vision, proprioception, and tactile feedback, along with corresponding actions and contextual information.

General-purpose RL Suite: **LeRobot** [128] provides a general-purpose, open-source foundation that unifies dataset organization, data-acquisition tooling, and integrated training–evaluation pipelines for RL, packaging deployable policy artifacts to lower the cost of on-hardware experimentation. **SERL** [129] is a real-robot RL suite that integrates a strong off-policy vision-based learner with practical components such as reward specification, automated resets, and safe control, enabling standardized tasks like PCB insertion and cable routing. It demonstrates that robust policies can be trained efficiently on hardware.

Domain-specific Benchmarks: These benchmarks target long-horizon control, generalization, and sim-to-real fidelity under reproducible conditions for specific tasks. **RoboTwin2.0** [99] supports standardized on-robot evaluation across multiple bimanual embodiments and a broad task set, and releases

TABLE II
OVERVIEW OF MAJOR MANIPULATION BENCHMARKS FOR RL-VLA METHODS.

Dataset	Dataset Size	Embodiments	Robot Morphology	EEF	Parallel Rollout	Gym-compatible	GPU-acceleration	Modality
LIBERO [121]	130 tasks	Franka	Single arm	Default gripper	✓	✓	✓	RGB-D; Language
Meta-World [122]	50 tasks	Sawyer	Single arm	Default gripper	✓	✓	✓	RGB; Language
ManiSkill2 [123]	20 tasks	Franka	Single arm	Default gripper	✓	✓	✓	RGB-D; Point cloud
ManiSkill3 [197]	100+ tasks	Franka	Single arm	Default gripper; Allegro Hand	✓	✓	✓	RGB-D; Point cloud
BEHAVIOR [198]	1k tasks	Galaxea	Humanoid	Default gripper	✓	✓	✓	RGB-D; Point cloud; Language
RoboVerse [199]	1k+ tasks	Franka; Fetch; etc.	Single arm; Wheeled	Default grippers	✓	✓	✓	RGB-D; Point cloud; Language
RoboCasa [200]	100 tasks	Franka	Wheeled	Default gripper	✓	✓	✓	RGB-D; Language
The Colosseum [124]	20 tasks	Franka	Single arm	Default gripper	✗	✗	✓	RGB-D; Point cloud; Language
Franka Kitchen [125]	6 tasks	Franka	Single arm	Default gripper	✗	✓	✗	RGB
CALVIN [126]	34 tasks	Franka	Single arm	Default gripper	✗	✗	✗	RGB-D; Language
SIMPLER [127]	8 tasks	Google robot; WidowX	Single arm	Default gripper	✓	✓	✓	RGB-D; Language
RoboTwin [102]	14 tasks	Aloha	Dual arms	Default gripper	✗	✗	✗	RGB-D; Language
RoboTwin2.0 [99]	50 tasks	Franka; Piper; UR5; etc.	Dual arms	Default grippers	✓	✓	✓	RGB-D; Language

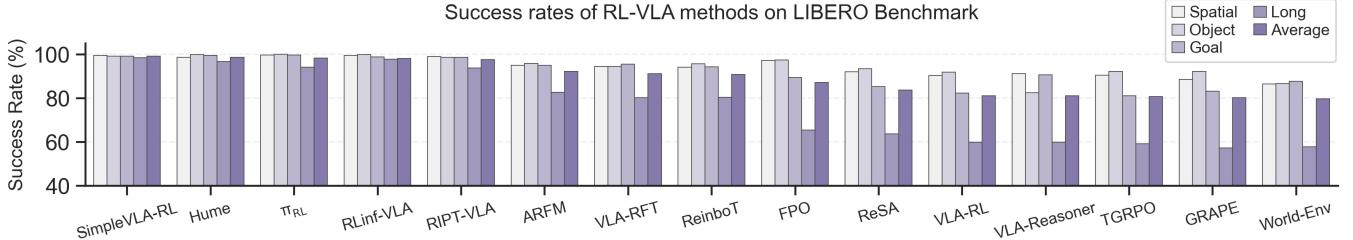


Fig. 3. Success rates of RL-VLA methods on LIBERO Benchmark. Methods are in reverse order according to average success rates.

a generator, curated multi-instance object library, and protocols that facilitate reproducible real-world studies. **FurnitureBench** [130] standardizes furniture assembly with a repeatable hardware setup, 3D-printable parts, and clear protocols spanning both isolated skills (grasping, insertion, screwing) and complete assemblies, while **RoboCup@Home** [131] frames an evolving competition for domestic service robots that jointly tracks progress in physical skills and higher-level cognitive abilities. **FMB** [132] centers on reproducible, 3D-printed objects requiring staged skills like grasping, fixture-assisted reorientation, and precise insertion. These skills are collected on a Franka Panda and released with multiple expert trajectories, multi-view RGB-D, and CAD assets, to assess generalization across unseen geometries and placements via a standardized long-horizon evaluation.

C. Evaluation Metrics

Assessing the performance of RL-VLA systems requires metrics that capture both traditional RL objectives and the unique multimodal and embodied challenges of real-world robotics. Unlike conventional benchmarks that focus solely on cumulative reward, RL-VLA evaluation must consider interpretability, physical efficiency, and human-robot interaction dynamics. This section summarizes several commonly adopted metrics and their representative usage across recent works.

(1) Success Rate: It measures the proportion of episodes in which the agent achieves the target goal, typically defined by task completion or semantic consistency with the instruction. It provides a clear indicator of overall policy competence and is used across both simulated and real-world settings.

(2) Average Episodic Return: This metric evaluates the expected cumulative reward per episode, reflecting both efficiency and stability of learning. It remains a standard objective in most reinforcement learning formulations, including vision-language-conditioned control [23].

(3) Safety Cost: Introduced by **SafeVLA** [120], this metric measures the degree of risk or constraint violation during training and deployment. It quantifies unsafe actions, collisions, or state transitions that could damage the robot or environment, serving as a key indicator of policy safety and reliability in real-world operation.

(3) Cycle Time: Introduced by **RLDG** [75] and **CoRFT** [22], cycle time quantifies the temporal efficiency of real-world learning cycles—capturing how quickly a system completes data collection, policy update, and deployment. It provides an important measure of real-world training throughput and system-level scalability.

(4) Episode Length: **ConRFT** [85] employs episode length as a proxy for task robustness, indicating whether the agent can sustain coherent action sequences without early failure or termination. Shorter average episode lengths often signal policy instability or unsafe exploration.

(5) Intervention Rate: Introduced in **ConRFT** [85], this metric measures how frequently human supervisors intervene during real-world training or deployment. A lower intervention rate implies greater autonomy and safer exploration—key desiderata for embodied RL-VLA in physical settings.

Potential Challenges: Current evaluation protocols in real-world RL often focus on task-level metrics such as success rate and episodic return, while overlooking system-level indicators that reflect real deployment performance. A comprehensive framework should also incorporate factors such as inference latency [92] and runtime stability, which are more critical for evaluating how efficiently and safely a policy operates in embodied environments.

VII. OPEN CHALLENGES AND FUTURE DIRECTIONS

Despite the impressive progress of RL-VLA, significant challenges remain before such systems can robustly operate in dynamic, open-ended physical environments.

Scaling to Long-horizon Tasks: Current RL-VLA systems struggle with long-horizon tasks because RL only supervises final actions, lacking guidance over intermediate reasoning processes. Promising solutions include chain-of-thought-like supervision and memory-retrieval mechanisms [93], [94], [203], [204], which couple structured reasoning with sequence modeling to help agents recall prior experiences and maintain temporal consistency over extended trajectories.

Model-based RL for VLA: Current RL-VLA systems remain inefficient due to sparse, delayed rewards and limited sample efficiency, relying heavily on massive simulated rollouts for policy updates. Promising solutions lie in model-based RL, where predictive world models learn environment dynamics to generate informative rewards and synthetic states for faster and more scalable training [65]–[67].

Efficient and Scalable Real-robot Training: Training RL-VLA on physical robots remains inefficient and costly due to limited parallelization and heavy reliance on human supervision for safe rollouts and resets [44], [53]. Promising solutions include reason agents for automatic failure handling, reactive agents for safe exploration, and multi-robot shared training with real-to-sim simulator rollouts to improve sample efficiency while reducing human intervention [78].

Reliable and Reproducible RL-VLA: RL-VLA systems often exhibit unstable optimization and poor reproducibility due to the high sensitivity of multimodal RL to design choices, hyperparameters, and stochastic environment dynamics [22], [205]. Improving reliability requires consistent training pipelines, controlled evaluation environments, and standardized reporting of algorithmic settings to ensure fair comparison and reproducible results across robotic platforms.

Safe and Risk-aware RL-VLA: Ensuring safety in real-world RL-VLA remains challenging due to irreversible risks from imperfect perception, delayed control, and limited supervision during exploration [118]. Promising solutions combine predictive risk modeling, constraint-based policy optimization, and language-conditioned safety reasoning to achieve safe and reliable deployment of embodied agents [120], [206].

VIII. CONCLUSION

This survey presented a comprehensive overview of RL-VLA for robotic manipulation, highlighting its effectiveness in overcoming the fundamental limitations of IL when tackling OOD situations. We first established a unified problem formulation and a detailed taxonomy, then systematically analyzed the core components of the RL-VLA architecture, including trade-offs in action representation, reward design, and transition modeling. We distilled key insights and algorithmic trends across online, offline, and test-time RL training paradigms, with a specific focus on leveraging data to enhance generalization. Moreover, our examination of real-world deployment reveals a clear progression from sim-to-real transfer to the core challenges of real-world RL, underscoring the crucial need to ensure safe exploration, autonomous recovery, and effective human-in-the-loop alignment. By organizing the landscape of benchmarking datasets, simulation environments, and evaluation metrics, we provided a structured

understanding of the pathway from pretrained VLA models to robust, deployable systems. While significant progress has been made, RL-VLA still faces critical challenges. Future RL-VLA research must overcome critical challenges in scalability and sample inefficiency. This will involve a pivot towards model-based RL to improve efficiency and the development of memory-augmented frameworks to enable the robust, long-term reasoning required for multi-stage tasks. Crucially, these advancements must be coupled with strong safety constraints to ensure reliable operation outside of simulation. Ultimately, successfully navigating these challenges will be the key to unlocking the promise of VLA models: autonomous, general-purpose manipulation in the real world.

REFERENCES

- [1] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello *et al.*, “Paligemma: A versatile 3b vlm for transfer,” *arXiv preprint arXiv:2407.07726*, 2024.
- [2] J. Xu, Z. Guo, H. Hu, Y. Chu, X. Wang, J. He, Y. Wang, X. Shi, T. He, X. Zhu *et al.*, “Qwen3-omni technical report,” *arXiv preprint arXiv:2509.17765*, 2025.
- [3] A. O’Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, “Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0,” in *ICRA*, 2024, pp. 6892–6903.
- [4] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, “Open-vla: An open-source vision-language-action model,” *arXiv preprint arXiv:2406.09246*, 2024.
- [5] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai *et al.*, “ $\pi_{0.5}$: a vision-language-action model with open-world generalization,” *arXiv preprint arXiv:2504.16054*, 2025.
- [6] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang *et al.*, “Gr0otn1: An open foundation model for generalist humanoid robots,” *arXiv preprint arXiv:2503.14734*, 2025.
- [7] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, “Rdt-1b: a diffusion foundation model for bimanual manipulation,” in *ICLR*, 2025.
- [8] C.-Y. Hung, Q. Sun, P. Hong, A. Zadeh, C. Li, U. Tan, N. Majumder, S. Poria *et al.*, “Nora: A small open-sourced generalist vision language action model for embodied tasks,” *arXiv preprint arXiv:2504.19854*, 2025.
- [9] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, “Recent advances in robot learning from demonstration,” *Annual review of control, robotics, and autonomous systems*, vol. 3, no. 1, pp. 297–330, 2020.
- [10] V. Saxena, M. Bronars, N. R. Arachchige, K. Wang, W. C. Shin, S. Nasiriany, A. Mandlekar, and D. Xu, “What matters in learning from large-scale datasets for robot manipulation,” in *ICLR*, 2025.
- [11] A. Z. Ren, J. Lidard, L. L. Ankile, A. Simeonov, P. Agrawal, A. Majumdar, B. Burchfiel, H. Dai, and M. Simchowitz, “Diffusion policy policy optimization,” *arXiv preprint arXiv:2409.00588*, 2024.
- [12] H. Ma, T. Chen, K. Wang, N. Li, and B. Dai, “Soft diffusion actor-critic: Efficient online reinforcement learning for diffusion policy,” *arXiv e-prints*, pp. arXiv-2502, 2025.
- [13] S. Park, Q. Li, and S. Levine, “Flow q-learning,” in *ICML*, 2025.
- [14] I. Kostrikov, A. Nair, and S. Levine, “Offline reinforcement learning with implicit q-learning,” *arXiv preprint arXiv:2110.06169*, 2021.
- [15] Q. Li, Z. Zhou, and S. Levine, “Reinforcement learning with action chunking,” *arXiv preprint arXiv:2507.07969*, 2025.
- [16] A. Y. Ng, D. Harada, and S. Russell, “Policy invariance under reward transformations: Theory and application to reward shaping,” in *ICML*, 1999.
- [17] E. Biyik, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, “Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences,” *IJRR*, vol. 41, no. 1, pp. 45–67, 2022.

- [18] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, “Eureka: Human-level reward design via coding large language models,” *IROS*, 2024.
- [19] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu *et al.*, “Deepseekmath: Pushing the limits of mathematical reasoning in open language models,” *arXiv preprint arXiv:2402.03300*, 2024.
- [20] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi *et al.*, “Deepseek-r1 incentivizes reasoning in llms through reinforcement learning,” *Nature*, vol. 645, no. 8081, pp. 633–638, 2025.
- [21] H. Zhang, Z. Zhuang, H. Zhao, P. Ding, H. Lu, and D. Wang, “Reinbot: Amplifying robot visual-language manipulation with reinforcement learning,” *arXiv preprint arXiv:2505.07395*, 2025.
- [22] D. Huang, Z. Fang, T. Zhang, Y. Li, L. Zhao, and C. Xia, “Co-rft: Efficient fine-tuning of vision-language-action models through chunked offline reinforcement learning,” *arXiv preprint arXiv:2508.02219*, 2025.
- [23] G. Lu, W. Guo, C. Zhang, Y. Zhou, H. Jiang, Z. Gao, Y. Tang, and Z. Wang, “Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning,” *arXiv preprint arXiv:2505.18719*, 2025.
- [24] H. Li, Y. Zuo, J. Yu, Y. Zhang, Z. Yang, K. Zhang, X. Zhu, Y. Zhang, T. Chen, G. Cui *et al.*, “Simplevla-rl: Scaling vla training via reinforcement learning,” *arXiv preprint arXiv:2509.09674*, 2025.
- [25] K. Chen, Z. Liu, T. Zhang, Z. Guo, S. Xu, H. Lin, H. Zang, Q. Zhang, Z. Yu, G. Fan *et al.*, “ π_{RL} : Online rl fine-tuning for flow-based vision-language-action models,” *arXiv preprint arXiv:2510.25889*, 2025.
- [26] M. Nakamoto, O. Mees, A. Kumar, and S. Levine, “Steering your generalists: Improving robotic foundation models via value guidance,” in *CoRL*. PMLR, 2020.
- [27] H. Song, D. Qu, Y. Yao, Q. Chen, Q. Lv, Y. Tang, M. Shi, G. Ren, M. Yao, B. Zhao *et al.*, “Hume: Introducing system-2 thinking in visual-language-action model,” *arXiv preprint arXiv:2505.21432*, 2025.
- [28] H. Zang, M. Wei, S. Xu, Y. Wu, Z. Guo, Y. Wang, H. Lin, L. Shi, Y. Xie, Z. Xu *et al.*, “Rlinf-vla: A unified and efficient framework for vla+ rl training,” *arXiv preprint arXiv:2510.06710*, 2025.
- [29] J. Sun, C. Zheng, E. Xie, Z. Liu, R. Chu, J. Qiu, J. Xu, M. Ding, H. Li, M. Geng *et al.*, “A survey of reasoning with foundation models: Concepts, methodologies, and outlook,” *ACM Computing Surveys*, vol. 57, no. 11, pp. 1–43, 2025.
- [30] K. Zhang, Y. Zuo, B. He, Y. Sun, R. Liu, C. Jiang, Y. Fan, K. Tian, G. Jia, P. Li *et al.*, “A survey of reinforcement learning for large reasoning models,” *arXiv preprint arXiv:2509.08827*, 2025.
- [31] Y. Cao, H. Zhao, Y. Cheng, T. Shu, Y. Chen, G. Liu, G. Liang, J. Zhao, J. Yan, and Y. Li, “Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods,” *TNNLS*, 2024.
- [32] B. Singh, R. Kumar, and V. P. Singh, “Reinforcement learning in robotic applications: a comprehensive survey,” *Artificial Intelligence Review*, vol. 55, no. 2, pp. 945–990, 2022.
- [33] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *IJRR*, vol. 32, no. 11, pp. 1238–1274, 2013.
- [34] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, “ $\pi\theta$: A vision-language-action flow model for general robot control,” *arXiv preprint arXiv:2410.24164*, 2024.
- [35] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. S. Ryoo, G. Salazar, P. R. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, “Rt-1: Robotics transformer for real-world control at scale,” in *RSS*, 2023.
- [36] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” in *CoRL*. PMLR, 2023, pp. 2165–2183.
- [37] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
- [38] M. Oquab, T. Dariseti, T. Moutakanni, H. Vo, M. Szafraniec, V. Khaldov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, “Dinov2: Learning robust visual features without supervision,” *TMLR*, pp. 1–31, 2024.
- [39] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, “Sigmoid loss for language image pre-training,” in *ICCV*, 2023, pp. 11 975–11 986.
- [40] M. J. Kim, C. Finn, and P. Liang, “Fine-tuning vision-language-action models: Optimizing speed and success,” *arXiv preprint arXiv:2502.19645*, 2025.
- [41] S. Tan, K. Dou, Y. Zhao, and P. Krähenbühl, “Interactive post-training for vision-language-action models,” *arXiv preprint arXiv:2505.17016*, 2025.
- [42] J. Liu, F. Gao, B. Wei, X. Chen, Q. Liao, Y. Wu, C. Yu, and Y. Wang, “What can rl bring to vla generalization? an empirical study,” *arXiv preprint arXiv:2505.19789*, 2025.
- [43] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, “Learning fine-grained bimanual manipulation with low-cost hardware,” *arXiv preprint arXiv:2304.13705*, 2023.
- [44] J. Luo, C. Xu, J. Wu, and S. Levine, “Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning,” *Science Robotics*, vol. 10, no. 105, p. eads5033, 2025.
- [45] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, “Human-level control through deep reinforcement learning,” *nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.
- [47] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *ICML*. Pmlr, 2018, pp. 1861–1870.
- [48] Z. Chen, R. Niu, H. Kong, and Q. Wang, “Tgrpo: Fine-tuning vision-language-action model via trajectory-wise group relative policy optimization,” *arXiv preprint arXiv:2506.08440*, 2025.
- [49] M. Lyu, Y. Sun, E. Lin, H. Li, R. Chen, F. Zhao, and Y. Zeng, “Reinforcement fine-tuning of flow-matching policies for vision-language-action models,” *arXiv preprint arXiv:2510.09976*, 2025.
- [50] H. Zhang, S. Zhang, J. Jin, Q. Zeng, Y. Qiao, H. Lu, and D. Wang, “Balancing signal and variance: Adaptive offline rl post-training for vla flow models,” *arXiv preprint arXiv:2509.04063*, 2025.
- [51] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, “Exploration by random network distillation,” *ICLR*, 2019.
- [52] H. Ma, Z. Luo, T. V. Vo, K. Sima, and T.-Y. Leong, “Highly efficient self-adaptive reward shaping for reinforcement learning,” in *ICLR*, 2025.
- [53] S. Zhai, Q. Zhang, T. Zhang, F. Huang, H. Zhang, M. Zhou, S. Zhang, L. Liu, S. Lin, and J. Pang, “A vision-language-action-critic model for robotic real-world reinforcement learning,” *arXiv preprint arXiv:2509.15937*, 2025.
- [54] A. Escontrela, A. Adeniji, W. Yan, A. Jain, X. B. Peng, K. Goldberg, Y. Lee, D. Hafner, and P. Abbeel, “Video prediction models as rewards for reinforcement learning,” *NeurIPS*, vol. 36, pp. 68 760–68 783, 2023.
- [55] Y. J. Ma, J. Hejna, C. Fu, D. Shah, J. Liang, Z. Xu, S. Kirmani, P. Xu, D. Driess, T. Xiao *et al.*, “Vision language models are in-context value learners,” in *ICLR*, 2025.
- [56] S. Sontakke, J. Zhang, S. Arnold, K. Pertsch, E. Biyik, D. Sadigh, C. Finn, and L. Itti, “Roboclip: One demonstration is enough to learn robot policies,” *NeurIPS*, vol. 36, pp. 55 681–55 693, 2023.
- [57] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox, “Gpu-accelerated robotic simulation for distributed reinforcement learning,” in *CoRL*. PMLR, 2018, pp. 270–282.
- [58] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *IROS*, vol. 3, 2004, pp. 2149–2154.
- [59] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, “Dream to control: Learning behaviors by latent imagination,” *arXiv preprint arXiv:1912.01603*, 2019.
- [60] J. Wu, S. Yin, N. Feng, X. He, D. Li, J. Hao, and M. Long, “ivideoopt: Interactive videogopts are scalable world models,” *NeurIPS*, vol. 37, pp. 68 082–68 119, 2024.
- [61] G. Lu, B. Jia, P. Li, Y. Chen, Z. Wang, Y. Tang, and S. Huang, “Gwm: Towards scalable gaussian world models for robotic manipulation,” *arXiv preprint arXiv:2508.17600*, 2025.
- [62] C. Zhang, Z. Wu, G. Lu, Y. Tang, and Z. Wang, “imowm: Taming interactive multi-modal world model for robotic manipulation,” *arXiv preprint arXiv:2510.09036*, 2025.
- [63] B. Wang, X. Meng, X. Wang, Z. Zhu, A. Ye, Y. Wang, Z. Yang, C. Ni, G. Huang, and X. Wang, “Embodiedreamer: Advancing real2sim2real

- transfer for policy training via embodied world modeling,” *arXiv preprint arXiv:2507.05198*, 2025.
- [64] A. Jiang, Y. Gao, Y. Wang, Z. Sun, S. Wang, Y. Heng, H. Sun, S. Tang, L. Zhu, J. Chai *et al.*, “Irl-vla: Training an vision-language-action policy via reward world model,” *arXiv preprint arXiv:2508.06571*, 2025.
- [65] H. Li, P. Ding, R. Suo, Y. Wang, Z. Ge, D. Zang, K. Yu, M. Sun, H. Zhang, D. Wang *et al.*, “Vla-rft: Vision-language-action reinforcement fine-tuning with verified rewards in world simulators,” *arXiv preprint arXiv:2510.00406*, 2025.
- [66] J. Xiao, Y. Yang, X. Chang, R. Chen, F. Xiong, M. Xu, W.-S. Zheng, and Q. Zhang, “World-env: Leveraging world model as a virtual environment for vla post-training,” *arXiv preprint arXiv:2509.24948*, 2025.
- [67] Z. Fangqi, Y. Zhengyang, H. Zicong, S. Quanxin, M. Xiao, and G. Song, “Wmpo: World model-based policy optimization for vision-language-action models,” *ArXiv preprint arXiv:2511.09515*, 2025.
- [68] J. Hu, R. Hendrix, A. Farhadi, A. Kembhavi, R. Martín-Martín, P. Stone, K.-H. Zeng, and K. Ehsani, “Flare: Achieving masterful and adaptive robot policies with large-scale reinforcement learning finetuning,” in *ICRA*. IEEE, 2025, pp. 3617–3624.
- [69] Y. Chen and X. Li, “Rlrc: Reinforcement learning-based recovery for compressed vision-language-action models,” *arXiv preprint arXiv:2506.17639*, 2025.
- [70] S.-C. Wang, T.-Y. Xiang, X.-H. Zhou, M.-J. Gui, X.-L. Xie, S.-Q. Liu, S.-Y. Wang, A.-Q. Jin, and Z.-G. Hou, “Vla model post-training via action-chunked ppo and self behavior cloning,” *arXiv preprint arXiv:2509.25718*, 2025.
- [71] Z. Zhang, K. Zheng, Z. Chen, J. Jang, Y. Li, S. Han, C. Wang, M. Ding, D. Fox, and H. Yao, “Grape: Generalizing robot policy via preference alignment,” *arXiv preprint arXiv:2411.19309*, 2024.
- [72] C. Yin, Y. Lin, W. Xu, S. Tam, X. Zeng, Z. Liu, and Z. Yin, “Deepthinkvla: Enhancing reasoning capability of vision-language-action models,” *arXiv preprint arXiv:2511.15669*, 2025.
- [73] H. Zhang, S. Zhang, J. Jin, Q. Zeng, R. Li, and D. Wang, “Robustvla: Robustness-aware reinforcement post-training for vision-language-action models,” *arXiv preprint arXiv:2511.01331*, 2025.
- [74] Y. Guo, J. Zhang, X. Chen, X. Ji, Y.-J. Wang, Y. Hu, and J. Chen, “Improving vision-language-action model with online reinforcement learning,” *arXiv preprint arXiv:2501.16664*, 2025.
- [75] C. Xu, Q. Li, J. Luo, and S. Levine, “Rldg: Robotic generalist policy distillation via reinforcement learning,” *arXiv preprint arXiv:2412.09858*, 2024.
- [76] J. Shu, Z. Lin, and Y. Wang, “Rftf: Reinforcement fine-tuning for embodied agents with temporal feedback,” *arXiv preprint arXiv:2505.19767*, 2025.
- [77] P. Jin, Q. Wang, G. Sun, Z. Cai, P. He, and Y. You, “Dual-actor fine-tuning of vla models: A talk-and-tweak human-in-the-loop approach,” *arXiv preprint arXiv:2509.13774*, 2025.
- [78] S. K. S. Ghasemipour, A. Wahid, J. Tompson, P. Sanketi, and I. Mordatch, “Self-improving embodied foundation models,” *arXiv preprint arXiv:2509.15155*, 2025.
- [79] S. Fei, S. Wang, L. Ji, A. Li, S. Zhang, L. Liu, J. Hou, J. Gong, X. Zhao, and X. Qiu, “Srpo: Self-referential policy optimization for vision-language-action models,” *arXiv preprint arXiv:2511.15605*, 2025.
- [80] M. Dalal, T. Chiruvolu, D. Chaplot, and R. Salakhutdinov, “Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks,” *arXiv preprint arXiv:2405.01534*, 2024.
- [81] W. Xiao, H. Lin, A. Peng, H. Xue, T. He, Y. Xie, F. Hu, J. Wu, Z. Luo, L. Fan *et al.*, “Self-improving vision-language-action models with data generation via residual rl,” *arXiv preprint arXiv:2511.00091*, 2025.
- [82] M. Memmel, A. Wagenmaker, C. Zhu, P. Yin, D. Fox, and A. Gupta, “Asid: Active exploration for system identification in robotic manipulation,” *ICLR*, 2024.
- [83] Y. Jin, J. Lv, H. Xue, W. Chen, C. Wen, and C. Lu, “Soe: Sample-efficient robot policy self-improvement via on-manifold exploration,” *arXiv preprint arXiv:2509.19292*, 2025.
- [84] Y. Xue, G. Lu, Z. Wu, C. Zhang, B. Jia, Z. Gu, Y. Tang, and Z. Wang, “Resample: A robust data augmentation framework via exploratory sampling for robotic manipulation,” *arXiv preprint arXiv:2510.17640*, 2025.
- [85] Y. Chen, S. Tian, S. Liu, Y. Zhou, H. Li, and D. Zhao, “Conrft: A reinforced fine-tuning method for vla models via consistency policy,” *arXiv preprint arXiv:2502.05450*, 2025.
- [86] M. S. Mark, T. Gao, G. G. Sampaio, M. K. Srirama, A. Sharma, C. Finn, and A. Kumar, “Policy agnostic rl: Offline rl and online rl fine-tuning of any class and backbone,” *arXiv preprint arXiv:2412.06685*, 2024.
- [87] C. Yu, Y. Wang, Z. Guo, H. Lin, S. Xu, H. Zang, Q. Zhang, Y. Wu, C. Zhu, J. Hu *et al.*, “Rlinf: Flexible and efficient large-scale reinforcement learning via macro-to-micro flow transformation,” *arXiv preprint arXiv:2509.15965*, 2025.
- [88] P. Intelligence, A. Amin, R. Aniceto, A. Balakrishna, K. Black, K. Conley, G. Connors, J. Darpinian, K. Dhabalia, J. DiCarlo, D. Driess, M. Equi, A. Esmail, Y. Fang, C. Finn, C. Glossop, T. Godden, I. Goryachev, L. Groom, H. Hancock, K. Hausman, G. Hussein, B. Ichter, S. Jakubczak, R. Jen, T. Jones, B. Katz, L. Ke, C. Kuchi, M. Lamb, D. LeBlanc, S. Levine, A. Li-Bell, Y. Lu, V. Mano, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, C. Sharma, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, W. Stoeckle, A. Swerdlow, J. Tanner, M. Torne, Q. Vuong, A. Walling, H. Wang, B. Williams, S. Yoo, L. Yu, U. Zhilinsky, and Z. Zhou, “ $\pi_{0,6}^*$: a vla that learns from experience,” *arXiv preprint arXiv:2511.14759*, 2025.
- [89] C.-Y. Hung, N. Majumder, H. Deng, L. Renhang, Y. Ang, A. Zadeh, C. Li, D. Herremans, Z. Wang, and S. Poria, “Nora-1.5: A vision-language-action model trained using world model- and action-based preference rewards,” *arXiv preprint arXiv:2511.14659*, 2025.
- [90] Y. Chebotar, Q. Vuong, K. Hausman, F. Xia, Y. Lu, A. Irpan, A. Kumar, T. Yu, A. Herzog, K. Pertsch *et al.*, “Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions,” in *CoRL*. PMLR, 2023, pp. 3909–3928.
- [91] J. T. Springenberg, A. Abdolmaleki, J. Zhang, O. Groth, M. Bloesch, T. Lampe, P. Brakel, S. Bechtle, S. Kapituowski, R. Hafner *et al.*, “Offline actor-critic reinforcement learning scales to large models,” *arXiv preprint arXiv:2402.05546*, 2024.
- [92] K. Lei, H. Li, D. Yu, Z. Wei, L. Guo, Z. Jiang, Z. Wang, S. Liang, and H. Xu, “Rl-100: Performant robotic manipulation with real-world reinforcement learning,” *arXiv preprint arXiv:2510.14830*, 2025.
- [93] M. Memmel, J. Berg, B. Chen, A. Gupta, and J. Francis, “Strap: Robot sub-trajectory retrieval for augmented policy learning,” *ICLR*, 2025.
- [94] T. Schmied, F. Paischer, V. Patil, M. Hofmarcher, R. Pascanu, and S. Hochreiter, “Retrieval-augmented decision transformer: External memory for in-context rl,” *CoLLAs*, 2025.
- [95] B. Li, D. Wu, Z. Yan, X. Liu, Z. Zeng, L. Li, and H. Zha, “Reflection-based task adaptation for self-improving vla,” *arXiv preprint arXiv:2510.12710*, 2025.
- [96] W. Guo, G. Lu, H. Deng, Z. Wu, Y. Tang, and Z. Wang, “Vla-reasoner: Empowering vision-language-action models with reasoning via online monte carlo tree search,” *arXiv preprint arXiv:2509.22643*, 2025.
- [97] M. Du, A. Khazatsky, T. Gerstenberg, and C. Finn, “To err is robotic: Rapid value-based trial-and-error during deployment,” *arXiv preprint arXiv:2406.15917*, 2024.
- [98] J. Josifovski, S. Auddy, M. Malmir, J. Piater, A. Knoll, and N. Navarro-Guerrero, “Continual domain randomization,” in *IROS*, 2024, pp. 4965–4972.
- [99] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Z. Li, Q. Liang, X. Lin, Y. Ge, Z. Gu *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.
- [100] J. Abou-Chakra, L. Sun, K. Rana, B. May, K. Schmeckpeper, N. Suerderhauf, M. Vittoria Minniti, and L. Herlant, “Real-is-Sim: Bridging the Sim-to-Real Gap with a Dynamic Digital Twin,” *arXiv e-prints*, p. arXiv:2504.03597, Apr. 2025.
- [101] N. Dean, W. Yu, G. Turk, and S. Ha, “Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation,” *RSS*, 2024.
- [102] Y. Mu, T. Chen, Z. Chen, S. Peng, Z. Lan, Z. Gao, Z. Liang, Q. Yu, Y. Zou, M. Xu *et al.*, “Robotwin: Dual-arm robot benchmark with generative digital twins,” in *CVPR*, 2025, pp. 27649–27660.
- [103] H. Lou, M. Zhang, H. Geng, H. Zhou, S. He, Z. Gao, S. Zhao, J. Mao, P. Abbeel, J. Malik *et al.*, “Dream: Differentiable real-to-sim-to-real engine for learning robotic manipulation,” in *3rd RSS Workshop on Dexterous Manipulation: Learning and Control with Diverse Data*.
- [104] D. Liu, Y. Chen, and Z. Wu, “Digital twin (dt)-cyclegan: Enabling zero-shot sim-to-real transfer of visual grasping models,” *RA-L*, vol. 8, no. 5, pp. 2421–2428, 2023.
- [105] X. Xu, Y. Hou, Z. Liu, and S. Song, “Compliant residual dagger: Improving real-world contact-rich manipulation with human corrections,” *arXiv preprint arXiv:2506.16685*, 2025.
- [106] Y. Jiang, C. Wang, R. Zhang, J. Wu, and L. Fei-Fei, “Transic: Sim-to-real policy transfer by learning from online correction,” *CoRL*, 2025.

- [107] W. Wang, J. Song, C. Liu, J. Ma, S. Feng, J. Wang, Y. Jiang, K. Chen, S. Zhan, Y. Wang *et al.*, “Genie centurion: Accelerating scalable real-world robot training with human rewind-and-refine guidance,” *arXiv preprint arXiv:2505.18793*, 2025.
- [108] W. Yu, J. Lv, Z. Ying, Y. Jin, C. Wen, and C. Lu, “Armada: Autonomous online failure detection and human shared control empower scalable real-world deployment and adaptation,” *arXiv preprint arXiv:2510.02298*, 2025.
- [109] Z. Hu, R. Wu, N. Enock, J. Li, R. Kadakia, Z. Erickson, and A. Kumar, “Rac: Robot learning for long-horizon tasks by scaling recovery and correction,” *arXiv preprint arXiv:2509.07953*, 2025.
- [110] H. Liu, S. Nasiriany, L. Zhang, Z. Bao, and Y. Zhu, “Robot learning on the job: Human-in-the-loop autonomy and learning during deployment,” *IJRR*, vol. 44, no. 10-11, pp. 1727–1742, 2025.
- [111] D. Kalashnikov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman, “Mt-opt: Continuous multi-task robotic reinforcement learning at scale,” *arXiv preprint arXiv:2104.08212*, 2021.
- [112] B. Eysenbach, S. Gu, J. Ibarz, and S. Levine, “Leave no trace: Learning to reset for safe and autonomous reinforcement learning,” *arXiv preprint arXiv:1711.06782*, 2017.
- [113] A. Sharma, R. Ahmad, and C. Finn, “A state-distribution matching approach to non-episodic reinforcement learning,” *arXiv preprint arXiv:2205.05212*, 2022.
- [114] J. Kim, D. Cho, and H. J. Kim, “free autonomous reinforcement learning via implicit and bidirectional curriculum,” in *ICML*, 2023, pp. 16441–16457.
- [115] H. Zhu, J. Yu, A. Gupta, D. Shah, K. Hartikainen, A. Singh, V. Kumar, and S. Levine, “The ingredients of real-world robotic reinforcement learning,” *arXiv preprint arXiv:2004.12570*, 2020.
- [116] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, “Palm-e: an embodied multimodal language model,” in *ICML*, 2023.
- [117] C. Cornelio and M. Diab, “Recover: A neuro-symbolic framework for failure detection and recovery,” in *IROS*, 2024, pp. 12 435–12 442.
- [118] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg, “Recovery rl: Safe reinforcement learning with learned recovery zones,” *RA-L*, vol. 6, no. 3, pp. 4915–4922, 2021.
- [119] J. Hu, P. Stone, and R. Martín-Martín, “Slac: Simulation-pretrained latent action space for whole-body real-world rl,” *arXiv preprint arXiv:2506.04147*, 2025.
- [120] B. Zhang, Y. Zhang, J. Ji, Y. Lei, J. Dai, Y. Chen, and Y. Yang, “SafeVLA: Towards safety alignment of vision-language-action model via constrained learning,” *arXiv preprint arXiv:2503.03480*, 2025.
- [121] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, “Libero: Benchmarking knowledge transfer for lifelong robot learning,” *NeurIPS*, vol. 36, pp. 44 776–44 791, 2023.
- [122] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, “Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning,” in *CoRL*. PMLR, 2020, pp. 1094–1100.
- [123] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, “Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations,” *arXiv preprint arXiv:2107.14483*, 2021.
- [124] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, and D. Fox, “The colosseum: A benchmark for evaluating generalization for robotic manipulation,” *arXiv preprint arXiv:2402.08191*, 2024.
- [125] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, “Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning,” *arXiv preprint arXiv:1910.11956*, 2019.
- [126] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, “Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks,” *RA-L*, vol. 7, no. 3, pp. 7327–7334, 2022.
- [127] X. Li, K. Hsu, J. Gu, O. Mees, K. Pertsch, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani *et al.*, “Evaluating real-world robot manipulation policies in simulation,” in *CoRL*. PMLR, 2025, pp. 3705–3728.
- [128] R. Cadene, S. Alibert, A. Soare, Q. Galloudec, A. Zouitine, S. Palma, P. Kooijmans, M. Aractingi, M. Shukor, D. Aubakirova, M. Russi, F. Capuano, C. Pascal, J. Choghari, J. Moss, and T. Wolf, “Lerobot: State-of-the-art machine learning for real-world robotics in pytorch,” <https://github.com/huggingface/lerobot>, 2024.
- [129] J. Luo, Z. Hu, C. Xu, Y. L. Tan, J. Berg, A. Sharma, S. Schaal, C. Finn, A. Gupta, and S. Levine, “Serl: A software suite for sample-efficient robotic reinforcement learning,” in *ICRA*, 2024, pp. 16 961–16 969.
- [130] M. Heo, Y. Lee, D. Lee, and J. J. Lim, “Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation,” *IJRR*, p. 02783649241304789, 2023.
- [131] T. van der Zant and L. Iocchi, “Robocup@ home: Adaptive benchmarking of robot bodies and minds,” in *ICSR*. Springer, 2011, pp. 214–225.
- [132] J. Luo, C. Xu, F. Liu, L. Tan, Z. Lin, J. Wu, P. Abbeel, and S. Levine, “Fmb: a functional manipulation benchmark for generalizable robotic learning,” *IJRR*, vol. 44, no. 4, pp. 592–606, 2025.
- [133] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu *et al.*, “Octo: An open-source generalist robot policy,” *arXiv preprint arXiv:2405.12213*, 2024.
- [134] J. Wen, M. Zhu, Y. Zhu, Z. Tang, J. Li, Z. Zhou, C. Li, X. Liu, Y. Peng, C. Shen *et al.*, “Diffusion-vla: Generalizable and interpretable robot foundation model via self-generated reasoning,” *arXiv preprint arXiv:2412.03293*, 2024.
- [135] Z. Liang, Y. Li, T. Yang, C. Wu, S. Mao, L. Pei, X. Yang, J. Pang, Y. Mu, and P. Luo, “Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies,” *arXiv preprint arXiv:2508.20072*, 2025.
- [136] S. Deng, M. Yan, S. Wei, H. Ma, Y. Yang, J. Chen, Z. Zhang, T. Yang, X. Zhang, W. Zhang *et al.*, “Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data,” *arXiv preprint arXiv:2505.03233*, 2025.
- [137] E. Wiewiora, “Potential-based shaping and q-value initialization are equivalent,” *JAIR*, p. 205–208, Sep. 2003.
- [138] A. Trott, S. Zheng, C. Xiong, and R. Socher, “Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards,” *NeurIPS*, vol. 32, 2019.
- [139] A. P. Badia, P. Sprechmann, A. Vitvitskyi, D. Guo, B. Piot, S. Kapturowski, O. Tielemans, M. Arjovsky, A. Pritzel, A. Bolt *et al.*, “Never give up: Learning directed exploration strategies,” *arXiv preprint arXiv:2002.06038*, 2020.
- [140] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, “Curiosity-driven exploration by self-supervised prediction,” in *ICML*. PMLR, 2017, pp. 2778–2787.
- [141] C. Schwarke, V. Klemm, M. Van der Boon, M. Bjelonic, and M. Hutter, “Curiosity-driven learning of joint locomotion and manipulation tasks,” in *CoRL*, vol. 229, 2023, pp. 2594–2610.
- [142] K. Yang, J. Tao, J. Lyu, and X. Li, “Exploration and anti-exploration with distributional random network distillation,” *arXiv preprint arXiv:2401.09750*, 2024.
- [143] J. Skalse, N. Howe, D. Krasheninnikov, and D. Krueger, “Defining and characterizing reward gaming,” *NeurIPS*, vol. 35, pp. 9460–9471, 2022.
- [144] Y. Ma, D. Jayaraman, and O. Bastani, “Conservative offline distributional reinforcement learning,” *NeurIPS*, vol. 34, pp. 19 235–19 247, 2021.
- [145] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, “Training language models to follow instructions with human feedback,” *NeurIPS*, vol. 35, pp. 27 730–27 744, 2022.
- [146] A. Hiranaka, M. Hwang, S. Lee, C. Wang, L. Fei-Fei, J. Wu, and R. Zhang, “Primitive skill-based robot learning from human evaluative feedback,” in *IROS*, 2023, pp. 7817–7824.
- [147] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplík *et al.*, “Language to rewards for robotic skill synthesis,” *arXiv preprint arXiv:2306.08647*, 2023.
- [148] Y. Zeng, Y. Mu, and L. Shao, “Learning reward for robot skills using large language models via self-alignment,” *arXiv preprint arXiv:2405.07162*, 2024.
- [149] A. S. Chen, S. Nair, and C. Finn, “Learning generalizable robotic reward functions from “in-the-wild” human videos,” *RSS*, 2021.
- [150] Y. Chen, H. Li, Z. Jiang, H. Wen, and D. Zhao, “Tevir: Text-to-video reward with diffusion models for efficient reinforcement learning,” *arXiv preprint arXiv:2505.19769*, 2025.
- [151] J. Zhang, Y. Luo, A. Anwar, S. A. Sontakke, J. J. Lim, J. Thomason, E. Biyik, and J. Zhang, “Rewind: Language-guided rewards teach robot policies without new demonstrations,” *arXiv preprint arXiv:2505.10911*, 2025.
- [152] Y. Wang, Z. Sun, J. Zhang, Z. Xian, E. Biyik, D. Held, and Z. Erickson, “Rl-vlm-f: Reinforcement learning from vision language foundation model feedback,” *arXiv preprint arXiv:2402.03681*, 2024.
- [153] Y. Lee, T. M. Luu, D. Lee, and C. D. Yoo, “Reward generation via large vision-language model in offline reinforcement learning,” *ICASSP*, 2025.

- [154] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, “Learning latent dynamics for planning from pixels,” in *ICML*. PMLR, 2019, pp. 2555–2565.
- [155] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, “Mastering atari with discrete world models,” *arXiv preprint arXiv:2010.02193*, 2020.
- [156] C. Chen, Y.-F. Wu, J. Yoon, and S. Ahn, “Transdreamer: Reinforcement learning with transformer world models,” *arXiv preprint arXiv:2202.09481*, 2022.
- [157] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, “Videogpt: Video generation using vq-vae and transformers,” *arXiv preprint arXiv:2104.10157*, 2021.
- [158] K. Chen, M. Cusumano-Towner, B. Huval, A. Petrenko, J. Hamburger, V. Koltun, and P. Krähenbühl, “Reinforcement learning for long-horizon interactive llm agents,” *arXiv preprint arXiv:2502.01600*, 2025.
- [159] A. Kumar, A. Zhou, G. Tucker, and S. Levine, “Conservative q-learning for offline reinforcement learning,” *NeurIPS*, vol. 33, pp. 1179–1191, 2020.
- [160] X. Li, P. Li, M. Liu, D. Wang, J. Liu, B. Kang, X. Ma, T. Kong, H. Zhang, and H. Liu, “Towards generalist robot policies: What matters in building vision-language-action models,” *arXiv e-prints*, pp. arXiv-2412, 2024.
- [161] S. Fujimoto, H. Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in *ICML*. PMLR, 2018, pp. 1587–1596.
- [162] K. Ehsani, T. Gupta, R. Hendrix, J. Salvador, L. Weihs, K.-H. Zeng, K. P. Singh, Y. Kim, W. Han, A. Herrasti *et al.*, “Spoc: Imitating shortest paths in simulation enables effective navigation and manipulation in the real world,” in *CVPR*, 2024, pp. 16238–16250.
- [163] P. J. Ball, L. Smith, I. Kostrikov, and S. Levine, “Efficient online reinforcement learning with offline data,” in *ICML*. PMLR, 2023, pp. 1577–1594.
- [164] D. Wang, R. Walters, and R. Platt, “ $so(2)$ -equivariant reinforcement learning,” *arXiv preprint arXiv:2203.04439*, 2022.
- [165] A. Mete, H. Xue, A. Wilcox, Y. Chen, and A. Garg, “Quest: Self-supervised skill abstractions for learning continuous control,” *NeurIPS*, vol. 37, pp. 4062–4089, 2024.
- [166] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, “Direct preference optimization: Your language model is secretly a reward model,” *NeurIPS*, vol. 36, pp. 53728–53741, 2023.
- [167] P. Li, H. Wu, Y. Huang, C. Cheang, L. Wang, and T. Kong, “Gr-mg: Leveraging partially-annotated data via multi-modal goal-conditioned policy,” *RA-L*, 2025.
- [168] Y. Tian, S. Yang, J. Zeng, P. Wang, D. Lin, H. Dong, and J. Pang, “Predictive inverse dynamics models are scalable learners for robotic manipulation,” *arXiv preprint arXiv:2412.15109*, 2024.
- [169] X. Chen, X. Wang, S. Changpinyo, A. J. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer *et al.*, “Pali: A jointly-scaled multilingual language-image model,” *arXiv preprint arXiv:2209.06794*, 2022.
- [170] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.
- [171] Y. Wang, P. Ding, L. Li, C. Cui, Z. Ge, X. Tong, W. Song, H. Zhao, W. Zhao, P. Hou *et al.*, “Vla-adapter: An effective paradigm for tiny-scale vision-language-action model,” *arXiv preprint arXiv:2509.09372*, 2025.
- [172] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang *et al.*, “Spatialvia: Exploring spatial representations for visual-language-action model,” *RSS*, 2025.
- [173] Y. Jin, J. Lv, W. Yu, H. Fang, Y.-L. Li, and C. Lu, “Sime: Enhancing policy self-improvement with modal-level exploration,” *arXiv preprint arXiv:2505.01396*, 2025.
- [174] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, “Efficient memory management for large language model serving with pagedattention,” in *SOSP*, 2023.
- [175] G. Sheng, C. Zhang, Z. Ye, X. Wu, W. Zhang, R. Zhang, Y. Peng, H. Lin, and C. Wu, “Hybridflow: A flexible and efficient rlhf framework,” *arXiv preprint arXiv:2409.19256*, 2024.
- [176] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, “D4rl: Datasets for deep data-driven reinforcement learning,” *arXiv preprint arXiv:2004.07219*, 2020.
- [177] G. Zhou, L. Ke, S. Srinivasa, A. Gupta, A. Rajeswaran, and V. Kumar, “Real world offline reinforcement learning with realistic data source,” *arXiv preprint arXiv:2210.06479*, 2022.
- [178] M. Hussing, J. A. Mendez, A. Singrodia, C. Kent, and E. Eaton, “Robotic manipulation datasets for offline compositional reinforcement learning,” *arXiv preprint arXiv:2307.07091*, 2023.
- [179] M. Nakamoto, S. Zhai, A. Singh, M. Sobol Mark, Y. Ma, C. Finn, A. Kumar, and S. Levine, “Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning,” *NeurIPS*, vol. 36, pp. 62244–62269, 2023.
- [180] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *ICROS*, 2017, pp. 23–30.
- [181] C. Chu, A. Zhmoginov, and M. Sandler, “Cyclegan, a master of steganography,” *arXiv preprint arXiv:1712.02950*, 2017.
- [182] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, “Learning dexterous in-hand manipulation,” *IJRR*, vol. 39, no. 1, pp. 3–20, 2020.
- [183] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, “Scalable deep reinforcement learning for vision-based robotic manipulation,” in *CoRL*. PMLR, 2018, pp. 651–673.
- [184] J. Kim, J. Hyeon Park, D. Cho, and H. J. Kim, “Automating reinforcement learning with example-based resets,” *RA-L*, vol. 7, no. 3, pp. 6606–6613, 2022.
- [185] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, “Curriculum learning for reinforcement learning domains: A framework and survey,” *JMLR*, vol. 21, no. 181, pp. 1–50, 2020.
- [186] K. Ryu, Q. Liao, Z. Li, P. Delgosha, K. Sreenath, and N. Mehr, “Curricullm: Automatic task curricula design for learning complex robot skills using large language models,” in *ICRA*. IEEE, 2025, pp. 4470–4477.
- [187] A. Muraleedharan *et al.*, “Selective progress-aware querying for human-in-the-loop reinforcement learning,” *arXiv preprint arXiv:2509.20541*, 2025.
- [188] A. Sharma, A. Gupta, S. Levine, K. Hausman, and C. Finn, “Autonomous reinforcement learning via subgoal curricula,” *NeurIPS*, vol. 34, pp. 18474–18486, 2021.
- [189] K. Xu, S. Verma, C. Finn, and S. Levine, “Continual learning of control primitives: Skill discovery via reset-games,” *NeurIPS*, vol. 33, pp. 4999–5010, 2020.
- [190] A. Gupta, J. Yu, T. Z. Zhao, V. Kumar, A. Rovinsky, K. Xu, T. Devlin, and S. Levine, “Reset-free reinforcement learning via multi-task learning: Learning dexterous manipulation behaviors without human intervention,” in *ICRA*. IEEE, 2021, pp. 6664–6671.
- [191] C. Lynch and P. Sermanet, “Language conditioned imitation learning over unstructured data,” *arXiv preprint arXiv:2005.07648*, 2020.
- [192] A. Xie, F. Tajwar, A. Sharma, and C. Finn, “When to ask for help: Proactive interventions in autonomous reinforcement learning,” *NIPS*, vol. 35, pp. 16918–16930, 2022.
- [193] S. Matsuoka and T. Sawaragi, “Recovery planning of industrial robots based on semantic information of failures and time-dependent utility,” *Advanced Engineering Informatics*, vol. 51, p. 101507, 2022.
- [194] D. Das and S. Chernova, “Semantic-based explainable ai: Leveraging semantic scene graphs and pairwise ranking to explain robot failures,” in *IROS*, 2021, pp. 3034–3041.
- [195] F. Ahmad, H. Ismail, J. Styrud, M. Stenmark, and V. Krueger, “A unified framework for real-time failure handling in robotics using vision-language models, reactive planner and behavior trees,” *arXiv preprint arXiv:2503.15202*, 2025.
- [196] H. Bharadhwaj, A. Kumar, N. Rhinehart, S. Levine, F. Shkurti, and A. Garg, “Conservative safety critics for exploration,” in *ICLR*, 2021.
- [197] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin, Y. Liu, T.-k. Chan *et al.*, “Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai,” *arXiv preprint arXiv:2410.00425*, 2024.
- [198] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, W. Ai, B. Martinez, H. Yin, M. Lingelbach, M. Hwang, A. Hiranaka, S. Garlanka, A. Aydin, S. Lee, J. Sun, M. Anvari, M. Sharma, D. Bansal, S. Hunter, K.-Y. Kim, A. Lou, C. R. Matthews, I. Villa-Renteria, J. H. Tang, C. Tang, F. Xia, Y. Li, S. Savarese, H. Gweon, C. K. Liu, J. Wu, and L. Fei-Fei, “Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation,” *arXiv preprint arXiv:2403.09227*, 2024.
- [199] H. Geng, F. Wang, S. Wei, Y. Li, B. Wang, B. An, C. T. Cheng, H. Lou, P. Li, Y.-J. Wang *et al.*, “Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning,” *arXiv preprint arXiv:2504.18904*, 2025.
- [200] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, “Robocasa: Large-scale simulation of

- everyday tasks for generalist robots,” *arXiv preprint arXiv:2406.02523*, 2024.
- [201] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, “Rlbench: The robot learning benchmark & learning environment,” *RA-L*, vol. 5, no. 2, pp. 3019–3026, 2020.
 - [202] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *IROS*, 2012, pp. 5026–5033.
 - [203] M. Shen, G. Zeng, Z. Qi, Z.-W. Hong, Z. Chen, W. Lu, G. Worrell, S. Das, D. Cox, and C. Gan, “Satori: Reinforcement learning with chain-of-action-thought enhances llm reasoning via autoregressive search,” *arXiv preprint arXiv:2502.02508*, 2025.
 - [204] R. Li, W. Guo, Z. Wu, C. Wang, H. Deng, Z. Weng, Y.-P. Tan, and Z. Wang, “Map-vla: Memory-augmented prompting for vision-language-action model in robotic manipulation,” *arXiv preprint arXiv:2511.09516*, 2025.
 - [205] N. A. Lynnerup, L. Nolling, R. Hasle, and J. Hallam, “A survey on reproducibility by evaluating deep reinforcement learning algorithms on real-world robots,” in *CoRL*. PMLR, 2020, pp. 466–489.
 - [206] H. Deng, W. Guo, Q. Wang, Z. Wu, and Z. Wang, “Safebimanual: Diffusion-based trajectory optimization for safe bimanual manipulation,” in *CoRL*, 2025.