2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Reinforcement Fine-Tuning of Flow-Matching Policies for Vision-Language-Action Models

Mingyang Lyu[a,b,e,1], Yinqian Sun[a,b,d,f,1], Erliang Lin[a], Huangrui Li[d],
Ruolin Chen[a,b,e], Feifei Zhao[a,b,d,f,2], Yi Zeng[a,b,c,d,e,f,2]

***Abstract*— Vision-Language-Action (VLA) models such as OpenVLA, Octo, and π_0 have shown strong generalization by leveraging large-scale demonstrations, yet their performance is still fundamentally constrained by the quality and coverage of supervised data. Reinforcement learning (RL) provides a promising path for improving and fine-tuning VLAs through online interaction. However, conventional policy gradient methods are computationally infeasible in the context of flow-matching based models due to the intractability of the importance sampling process, which requires explicit computation of policy ratios. To overcome this limitation, we propose Flow Policy Optimization (FPO) algorithm, which reformulates importance sampling by leveraging per-sample changes in the conditional flow-matching objective. Furthermore, FPO achieves stable and scalable online reinforcement fine-tuning of the π_0 model by integrating structure-aware credit assignment to enhance gradient efficiency, clipped surrogate objectives to stabilize optimization, multi-step latent exploration to encourage diverse policy updates, and a Q-ensemble mechanism to provide robust value estimation. We evaluate FPO on the LIBERO benchmark and the ALOHA simulation task against supervised, preference-aligned, diffusion-based, autoregressive online RL, and π_0-FAST baselines, observing consistent improvements over the imitation prior and strong alternatives with stable learning under sparse rewards. In addition, ablation studies and analyses of the latent space dynamics further highlight the contributions of individual components within FPO, validating the effectiveness of the proposed computational modules and the stable convergence of the conditional flow-matching objective during online RL.**

I. INTRODUCTION

The pursuit of generalist robots capable of executing a diverse array of physical tasks has advanced significantly with the emergence of Vision-Language-Action (VLA) models. Recent architectures, such as OpenVLA [1] and Octo [2], have demonstrated that policies pre-trained on large-scale datasets of human demonstrations can acquire broad semantic understanding and effectively execute a wide spectrum of instructions. Notably, the π_0 [3] model implements action generation via a flow-matching technique [4], [5]. This method confers a unique advantage: it enables the generation of smooth, temporally coherent, high-frequency action segments, which is essential for achieving dexterous and long-horizon manipulation tasks that require more than isolated, single-step action predictions.

Drawing inspiration from the remarkable progress of Reinforcement Learning (RL) in enhancing Large Language Models (LLMs) beyond their supervised fine-tuning (SFT) performance [14]–[16], there is a growing trend to apply RL for post-training of embodied VLA models. Approaches such as online RL for auto-regressive VLAs [6], iterative RL+SL stabilization for large VLAs [7], and policy-gradient fine-tuning of diffusion/flow-matching policies [8], [9] have demonstrated that robotic agents can leverage online interaction to refine skills and discover strategies superior to those available in initial imitation datasets. This paradigm allows agents to overcome the inherent limitations of offline demonstration data quality and coverage, pushing performance beyond the imitation ceiling.

However, a core technical incompatibility arises when applying conventional RL techniques to flow-matching-based VLA models such as π_0 [3]. Commonly used policy gradient methods for reinforcement fine-tuning of VLA models, such as PPO [11] and TRPO [17], require importance sampling, i.e., the explicit computation of policy ratios. For flow-matching models, this computation is analytically intractable. It necessitates solving an underlying ordinary differential equation [18] and integrating a computationally prohibitive Jacobian trace term along the generation path [19]. This renders such methods computationally infeasible for the demands of online fine-tuning. While reward-weighted supervised learning approaches exist, they typically struggle with active exploration and the discovery of novel, out-of-distribution behaviors. These combined challenges have largely precluded the effective application of online RL for fine-tuning flow-matching generative policy-based VLA models.

In this paper, we introduce Flow Policy Optimization (FPO), a method designed to overcome the incompatibility between flow-matching policies and PPO-style updates by constructing a likelihood-free policy ratio based on per-sample changes in the conditional flow-matching objective. This formulation eliminates the need for explicit action likelihoods and ODE–Jacobian computations while preserving consistency with the policy's generative structure. Furthermore, we provide structure-aware credit assignment in the latent space by leveraging the model's training objective as a per-sample improvement signal. Combined with a clipped surrogate objective, multi-step latent exploration, and a Q-ensemble, FPO enables stable and efficient learning

Author affiliations: [a] Brain-inspired Cognitive AI Lab, Institute of Automation, Chinese Academy of Sciences, Beijing, China.; [b] Beijing Institute of AI Safety and Governance, China.; [c]State Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology; [d] Beijing Key Laboratory of Safe AI and Superalignment, China.; [e] University of Chinese Academy of Sciences (UCAS), Beijing, China.; [f] Long-term AI,Beijing,China.

[1] Co-first authors. [2] Co-corresponding authors. Correspondence: zhaofeifei2014@ia.ac.cn (Feifei Zhao), yi.zeng@ia.ac.cn (Yi Zeng). Additional contact: lvmingyang2024@ia.ac.cn (Mingyang Lyu).

even in environments with sparse rewards and contact-rich dynamics. The proposed method successfully facilitates online reinforcement fine-tuning of the π_0 model, with its effectiveness and superiority empirically validated on the LIBERO benchmark and the ALOHA Transfer Cube task. In summary, the main contributions of this work are summarized as follows:

- We propose FPO, a practical policy optimization framework that bridges flow-matching policies and PPO-style updates by introducing a likelihood-free policy ratio derived from per-sample changes in the conditional flow-matching objective. This formulation avoids explicit density estimation and complex Jacobian computations, while retaining structural consistency with the generative policy.
- We develop an online reinforcement fine-tuning algorithm for the π_0 model by integrating structure-aware credit assignment in the latent space with key RL components including a clipped surrogate objective, multi-step latent exploration, and a Q-ensemble. This combination ensures stable and efficient learning in challenging environments with sparse rewards and contact-rich dynamics.
- Extensive experiments on the LIBERO benchmark and the ALOHA Transfer Cube task demonstrate the superior performance of π_0-FPO over six strong baselines such as OpenVLA, Octo, Diffusion Policy, GRAPE, and π_0-FAST, achieving an average success rate of 87.2% on LIBERO, 65.3% on LIBERO-Long, and more than $1.5\times$ the baseline success rate on ALOHA-sim. Ablation studies validate the contribution of each component, while qualitative and latent-space analyses highlight improved correction of recurrent failure modes.

II. RELATED WORK

A. Vision-Language-Action Models

VLA models are commonly trained via behavioral cloning on large-scale human demonstrations, coupling language and vision with end-to-end control [13]. Early systems predominantly used autoregressive, token-based action decoders, recent policies replace discrete heads with diffusion- or flow-matching controllers [38], [39], capturing continuous, multi-modal action distributions and enabling smooth, high-frequency control for dexterous manipulation. Representative systems include Octo and π_0 [2], [3], whose generative action heads are grounded in the modeling literature and applied to visuomotor control [4], [5], [27]. In parallel, OpenVLA provides an open-source generalist policy that scales across embodiments and supports efficient fine-tuning [1], orthogonal advances include frequency-space action tokenization (FAST) for high-rate control [28], BC-Z for zero-shot generalization from language goals [49], and preference alignment via GRAPE [29], DPO [41] and learning from human preferences [42]. Together, these developments situate diffusion- and flow-matching decoders within a broader VLA trajectory [13] and motivate online updates that improve beyond the imitation prior.

B. Reinforcement Learning for VLA Policies

Policy-gradient fine-tuning of generative policies faces two obstacles: intractable (or costly) likelihoods along generative trajectories and long-horizon credit assignment. For diffusion policies, PPO-style surrogates have been adapted to the denoising process with architecture-aware designs (DPPO) [9]. For flow-matching policies, recent work follows two lines. One line performs reward-weighted supervised updates that avoid explicit likelihoods by biasing training toward high-return samples (RWFM and variants) [35]. The other introduces stochastic relaxations or noise injection to enable sampling-based ratios and on-policy updates (Flow-GRPO [36], ReinFlow [8]). Related gradient estimators for flow models have also been explored from a policy-gradient perspective [22]. In VLA settings with autoregressive heads, VLA-RL reports procedures for scaling online RL with trajectory-level optimization [6], while preference-based tuning connects to advances in RL for large models [14]–[16]. A persistent difficulty for flow-based actors is that exact policy ratios generally require solving probability-flow ODEs with Jacobian-trace terms [39], rendering likelihoods and ratios expensive or intractable for online control [18], [19]. We address this by constructing a likelihood-free ratio from per-sample changes in the conditional flow-matching objective and performing clipped PPO-style updates aligned with the model's generative structure.

III. METHOD

The proposed FPO is an actor–critic framework that enables online fine-tuning of pretrained conditional flow-matching policies without requiring tractable action likelihoods. The central idea is to reformulate importance sampling by exploiting per-sample changes in the CFM objective as a structure-aligned signal, which is mapped to a likelihood-free ratio and used within a PPO-style clipped surrogate. To obtain stable and efficient updates, FPO integrates (i) structure-aware credit assignment in the action latent space, (ii) clipped surrogates for trust-region control, (iii) multi-step latent (Euler) exploration to produce smooth, temporally correlated perturbations, and (iv) a critic ensemble that supplies robust value estimates. Training alternates between *rollout* and *update* (Fig. 1b,c, Algorithm 1): rollout logs transitions and per-sample CFM losses into a small sliding-window buffer; updates recompute losses under the current actor, map their differences to a clipped surrogate, and train the critic ensemble; the updated actor is then used for the next rollout. Formal details appear in Sec. III-A and Sec. III-B.

A. FPO Pipeline and Problem Formulation

FPO steers a frozen base policy $\pi_0(a \mid s, x)$ with a flow-based actor $\pi_\theta(\cdot \mid s)$ operating in the action latent space. Let $x(u; s)$ denote the actor's latent at flow time $u \in [0, 1]$ for state s, and write $x_t := x(1; s_t)$ for the latent produced at environment step t. A frozen encoder maps observations to $s_t \in \mathbb{R}^d$; the actor samples $x_t \in \mathbb{R}^D$; the base decodes (s_t, x_t) to an action a_t; the environment yields reward r_t and next

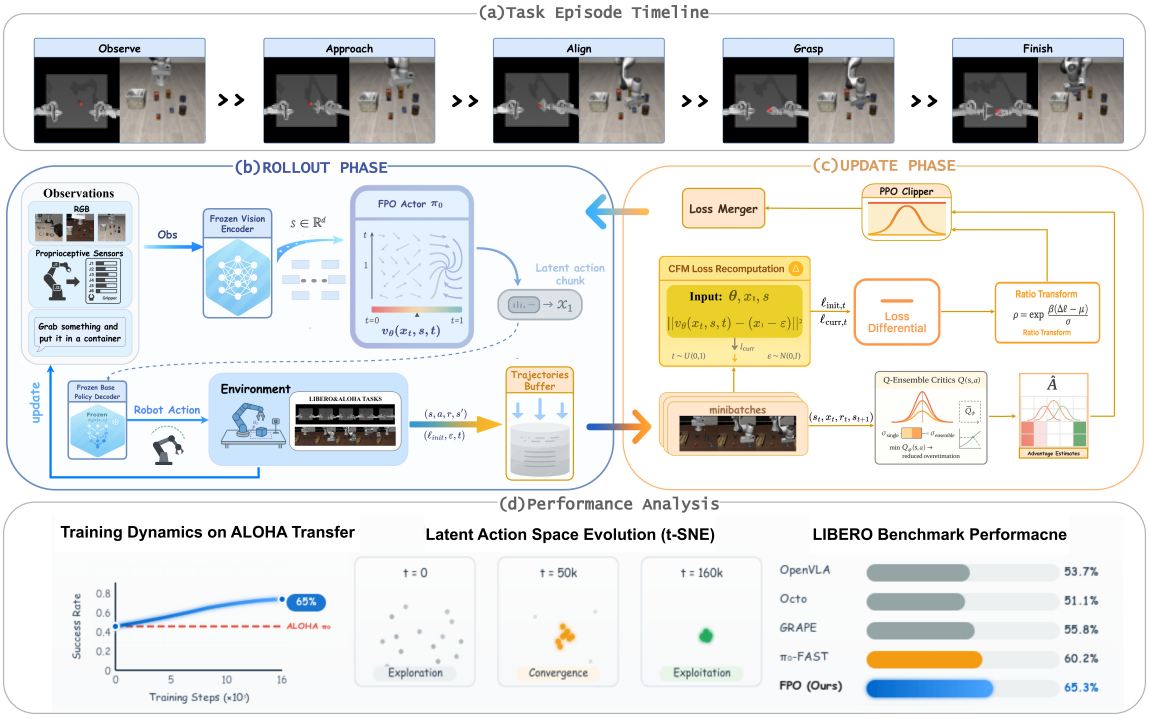


Fig. 1. Overview of Flow Policy Optimization (FPO). (a) Task episode timeline. (b) Rollout phase: a frozen encoder produces state s, the actor π_θ outputs a latent chunk x_1, the frozen base policy π_0 decodes (s, x_1) to control a, yielding (r, s'). We store the transition and cache the initial CFM loss in a near sliding-window trajectory buffer. (c) Update phase: Batch of trajectories are sampled, the CFM loss is recomputed to form a loss differential, which is mapped to a likelihood-free ratio. A Q-ensemble supplies advantages, and the actor is updated with a clipped surrogate. The updated policy feeds back to rollout. (d) Performance panels: example training curve, latent-space evolution (t-SNE), and LIBERO success rates.

state s_{t+1} (Fig. 1). The optimization objective is

$$J(\theta) \;=\; \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \, r_t\right], \qquad \gamma \in (0,1). \tag{1}$$

Because the actor is a conditional flow model, $\log \pi_\theta(x_t \mid s_t)$ is generally intractable, precluding direct policy-ratio computation.

a) Rollout and Data Recording: Training proceeds in alternating *rollout* and *update* phases (Fig. 1b,c; Algorithm 1). During interaction, a frozen rollout copy θ_{old} is used to generate experience so that logged quantities remain consistent with the data-collecting policy. At each step, the encoder produces s_t, the actor samples a latent chunk x_t (with optional short Euler perturbations in latent space for exploration), and the frozen base policy π_0 decodes (s_t, x_t) to low-level control a_t that is executed in the environment. The system records $(s_t, x_t, a_t, r_t, s_{t+1})$ and caches the per-sample CFM loss $\ell_{\text{cfm}}(x_t \mid s_t; \theta_{\text{old}})$ attached to the exact latent used for control. Transitions are stored in a small sliding-window *trajectory buffer* that retains only recent rollouts. This design preserves the linkage between cached losses and their originating policy, and bounds the distributional gap between the data-collecting policy and the subsequently updated actor.

b) Update Cycle: During updates, data are drawn from the trajectory buffer and the CFM loss is re-evaluated under the current actor θ on the same (s_t, x_t) pairs. The resulting per-sample loss differential is converted, via batch-standardization and a monotone mapping, into a *likelihood-free* ratio proxy that serves as the multiplicative factor in a PPO-style clipped surrogate. Advantages are supplied by a critic ensemble queried in latent space. Actor and critics are optimized for several SGD epochs per interaction batch while continuously evicting older trajectories to keep the training distribution close to recent behavior. Target networks for the critics are updated by Polyak averaging to stabilize bootstrapped targets. After the update cycle, parameters are synchronized by setting $\theta_{\text{old}} \leftarrow \theta$ before the next interaction phase. This schedule closes the interaction–update loop, maintains a tight coupling between the ratio proxy and the data-collecting policy, and yields steady improvement without requiring tractable action likelihoods.

B. Structure-Aligned Policy Update and Training Components

a) Likelihood-Free Ratio from CFM Loss: Because $\log \pi_\theta(x_t \mid s_t)$ is intractable for flow-based actors, FPO uses the actor's CFM objective as the update signal. Let $\ell_{\text{cfm}}(x_t \mid s_t; \theta)$ denote the per-sample CFM loss [4], [5]. For each stored pair (s_t, x_t), the loss reduction is:

$$\Delta\ell_{\text{cfm},t} \;=\; \ell_{\text{cfm}}(x_t \mid s_t; \theta_{\text{old}}) \;-\; \ell_{\text{cfm}}(x_t \mid s_t; \theta) \tag{2}$$

which measures improvement on the *same* sample relative to the rollout actor. Under a mild local monotonicity assumption—that per-sample CFM loss decreases coincide with increases in the actor's conditional density—we treat $\Delta\ell_{\text{cfm},t}$ as an order-preserving surrogate of the intractable importance ratio $\pi_\theta(x_t \,|\, s_t)/\pi_{\theta_{\text{old}}}(x_t \,|\, s_t)$. And the $\Delta\ell_{\text{cfm},t}$ is normalized

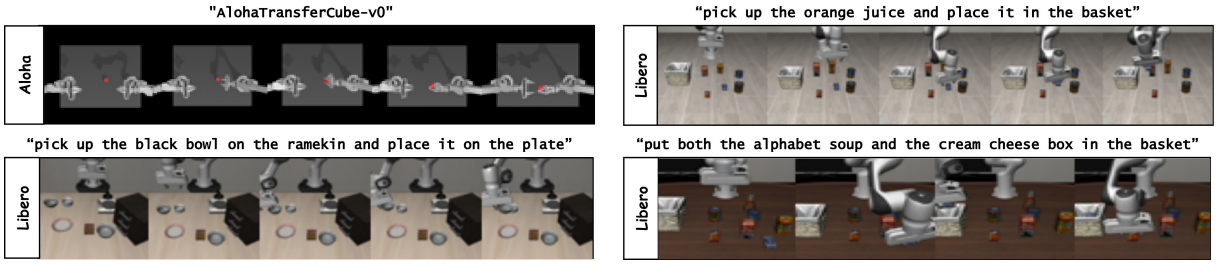


Fig. 2. An overview of the challenging visuomotor control environments used in our evaluation: the bimanual ALOHA Transfer Cube task and several multi-object manipulation tasks from the LIBERO suite. These environments require a combination of long-horizon reasoning, precise control, and generalization across different objects and initial conditions.

Algorithm 1 Flow Policy Optimization (FPO)

Input: frozen base policy π_0, actor π_θ, critic ensemble $\{Q_{\phi_i}\}_{i=1}^{M}$, target critics $\{Q_{\bar{\phi}_i}\}_{i=1}^{M}$, buffer $\mathcal{B}$
for each iteration **do**
  $\theta_{\text{old}} \leftarrow \theta$ {Freeze actor for loss caching}
  // Rollout phase
  for $t = 0 \dots T_{\text{rollout}} - 1$ **do**
    observe s_t
    sample latent $x_t \sim \pi_\theta(\cdot \mid s_t)$
    decode action $a_t \sim \pi_0(\cdot \mid s_t, x_t)$
    step env, obtain (r_t, s_{t+1})
    cache $\ell_{\text{init},t} \leftarrow \ell_{\text{cfm}}(x_t \mid s_t; \theta_{\text{old}})$
    push $(s_t, x_t, a_t, r_t, s_{t+1}, \ell_{\text{init},t})$ into $\mathcal{B}$
  end for
  // Update phase
  for $k = 1 \dots K_{\text{update}}$ **do**
    sample batch $\mathcal{M} \subset \mathcal{B}$
    // Critic update (Eqs.5,6)
    for each $(s_t, x_t, r_t, s_{t+1}) \in \mathcal{M}$:
      sample $x'_{t+1} \sim \pi_\theta(\cdot \mid s_{t+1})$ and set
      $y_t \leftarrow r_t + \gamma \min_i Q_{\bar{\phi}_i}(s_{t+1}, x'_{t+1})$
    update $\{\phi_i\}$ by minimizing $\mathcal{L}_{\text{critic}}(\phi)$;
    Polyak update targets $Q_{\bar{\phi}_i}$
    // Actor update (Eqs.2,3,4)
    compute $\Delta\ell_{\text{cfm},t} \leftarrow \ell_{\text{cfm}}(x_t \mid s_t; \theta_{\text{old}})\, \ell_{\text{cfm}}(x_t \mid s_t; \theta)$
    standardize $z_t \leftarrow \text{standardize}(\Delta\ell_{\text{cfm},t})$
    map ratio proxy $\rho_t \leftarrow \exp(\beta z_t)$
    compute advantages $\hat{A}_t$ from the critic ensemble (e.g., GAE)
    update θ by minimizing $\mathcal{L}_{\text{actor}}(\theta)$
  end for
end for

with

$$z_t \;=\; \frac{\Delta\ell_{\text{cfm},t} - \mu_\Delta}{\sigma_\Delta}, \qquad \rho_t \;=\; \exp(\beta\, z_t) \tag{3}$$

where $(\mu_\Delta, \sigma_\Delta)$ are mean and standard deviation as batch statistics and $\beta > 0$ controls the sharpness of the mapping.

b) Clipped Surrogate and Actor Update: The actor is optimized with a PPO-style clipped surrogate [11] using advantages $\hat{A}_t$:

$$\mathcal{L}_{\text{actor}}(\theta) \;=\; -\mathbb{E}_t\Big[\min\big(\rho_t\,\hat{A}_t,\; \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)\,\hat{A}_t\big)\Big] \tag{4}$$

where $\epsilon > 0$ is the clipping parameter. This construction regulates update magnitude while preserving alignment with the actor's generative structure.In practice, we standardize $\hat{A}_t$ within each minibatch and stop gradients through ρ_t to reduce variance and avoid feedback instabilities.

c) Critic Ensemble and Advantage Estimation: We employ an ensemble of action–value functions $\{Q_{\phi_i}(s,x)\}_{i=1}^{M}$ to reduce overestimation and stabilize advantage estimates. Target critics $Q_{\bar{\phi}_i}$ are updated by Polyak averaging once per gradient step. For a transition (s_t, x_t, r_t, s_{t+1}), the temporal-difference target is:

$$y_t = r_t + \gamma \min_i Q_{\bar{\phi}_i}(s_{t+1}, x'_{t+1}), \quad x'_{t+1} \sim \pi_\theta(\cdot \mid s_{t+1}) \tag{5}$$

where the operation of minimizing introduces a conservative target that empirically curbs optimistic bias. For terminal s_{t+1} the bootstrap term is masked out. The critic loss is the squared TD error:

$$\mathcal{L}_{\text{critic}}(\phi) \;=\; \mathbb{E}\Big[\big(Q_\phi(s_t, x_t) - y_t\big)^2\Big] \tag{6}$$

Advantages are computed with generalized advantage estimation (GAE) [32]. The value baseline $V(s)$ is taken as a conservative estimate from the ensemble (the minimum across members). We reuse stored latents when available; otherwise a fresh latent is sampled from $\pi_\theta(\cdot \mid s)$.

d) Latent-Space Exploration and Data Handling: Exploration is induced by multi-step Euler integration in the actor's latent dynamics. Starting from a sampled latent $x_t^{(0)} \sim \pi_\theta(\cdot \mid s_t)$ and the CFM velocity field v_θ, we apply K short steps:

$$x_t^{(k+1)} = x_t^{(k)} + \eta\, v_\theta\big(x_t^{(k)}, \tau^{(k)} \mid s_t\big), \quad k = 0, \dots, K-1 \tag{7}$$

where $\{\tau^{(k)}\}$ is a discretization of the flow time and $\eta > 0$ is a small step size. The final $x_t^{(K)}$ is decoded by the frozen base. This procedure yields smooth, temporally correlated perturbations that remain aligned with the actor's generative field. Transitions are stored in a compact sliding-window *trajectory buffer* $\mathcal{B}$ that retains only recent rollouts. Each update draws sample batches from $\mathcal{B}$ and performs several SGD epochs while evicting older entries, which limits

TABLE I

OVERALL SUCCESS RATE (SR, %) AND PER-SUITE RANK ON LIBERO BENCHMARKS. RANKS ARE COMPUTED AMONG BASELINE METHODS ONLY (EXCLUDING π_0-FAST). AVG RANK IS THE MEAN OF PER-SUITE RANKS FOR EACH BASELINE. BEST IN **BOLD**.

Method	LIBERO-Spatial		LIBERO-Object		LIBERO-Goal		LIBERO-Long		Average	
	SR (%)	Rank	SR (%)	Rank	SR (%)	Rank	SR (%)	Rank	SR (%)	Avg Rank
Diffusion Policy [27]	78.3	6	92.5	2	68.3	6	50.5	6	72.4	5.0
GRAPE (DPO) [29]	87.6	3	91.2	4	82.2	3	55.8	3	79.2	3.3
Octo (SFT) [2]	78.9	5	85.7	6	84.6	2	51.1	5	75.1	4.5
OpenVLA (SFT) [1]	84.7	4	88.4	5	79.2	5	53.7	4	76.5	4.5
VLA-RL [6]	90.2	2	91.8	3	82.2	3	59.8	2	81.0	2.5
π_0-FAST [3]	96.4	–	96.8	–	88.6	–	60.2	–	85.5	–
π_0-FPO (Ours)	**97.2**	**1**	**97.3**	**1**	**89.4**	**1**	**65.3**	**1**	**87.2**	**1**

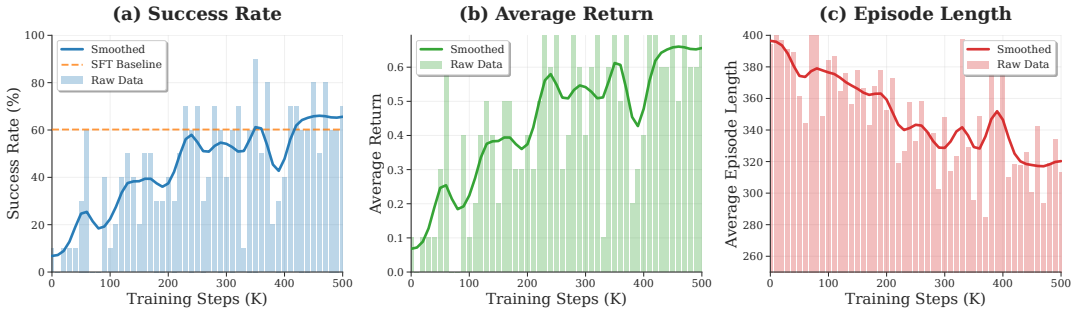


Fig. 3. LIBERO-Long simulation: online fine-tuning curves on a representative task.

distributional drift between the update policy and the data-collecting policy and keeps the loss differential $\Delta\ell_{\text{cfm},t}$ (Eq. 2) evaluated under a distribution close to behavior, stabilizing the ratio mapping in Eq. 3.

IV. EXPERIMENTS

In this section, we empirically evaluate FPO along three axes: (i) final performance on standard manipulation benchmarks relative to strong baselines. (ii) learning dynamics under online interaction (improvement curves and exploration behavior). and (iii) ablations that isolate the contribution of each component.

A. Experimental Setup

a) Tasks and Evaluation: The proposed FPO algorithm was evaluated on two simulated visuomotor benchmarks: LIBERO [20] and ALOHA Transfer Cube [21] (as shown in Fig. 2). LIBERO [20] comprises four sub-suites—Spatial, Object, Goal, and LIBERO-Long. ALOHA Transfer Cube [21] is a bimanual manipulation task with contact-rich dynamics. We follow the official success criteria and report per-suite success rate (SR, %).

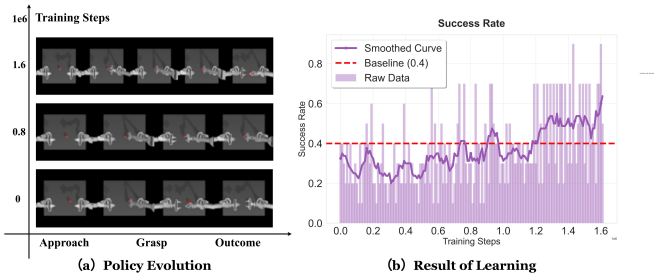
b) Baselines and Protocol: We compare against π_0-FAST [28], GRAPE [29], Diffusion Policy [27], OpenVLA [1], Octo [2], and VLA-RL [6], covering supervised steering of π_0, preference alignment, diffusion-based control, large-scale SFT VLAs, and online RL with autoregressive heads. Evaluations follow the official success metrics and protocols. We use public checkpoints when available, otherwise authors' reference implementations with reported settings. Task definitions, observation/action interfaces, and evaluation seeds are matched across methods. Our runs initialize from the released π_0 checkpoint, keep the π_0 decoder frozen, and update only the flow actor and an ensemble critic online.

B. Performance Evaluation and Analysis

a) Performance Advantages of FPO on the LIBERO Benchmark: FPO (denoted as π_0-**FPO**) achieves state-of-the-art performance across all four task suites of the LIBERO benchmark, as shown in Table I. It attains suite-leading success rates with an overall average of 87.2%, outperforming all baseline methods. On the LIBERO-Long suite, FPO attains 65.3% SR. This corresponds to improvements of +5.5 percentage points over the RL baseline VLA-RL (59.8%), +9.5 over GRAPE (55.8%), and +5.1 over π_0-FAST (60.2%), as reported in Table I. These comparisons situate FPO ahead of both online RL and offline-trained baselines under the same evaluation protocol.

This improvement indicates that even foundation models trained on large-scale offline datasets retain unused capacity. FPO leverages an online, reward-driven update mechanism to exploit this capacity, addressing the lack of adaptability and exploration inherent to purely offline training. These results demonstrate that FPO can effectively learn from sparse rewards and refine goal-directed behaviors, extending performance beyond the limits of imitation-based methods.

b) FPO Enables Stable and Efficient Online Learning: To further assess the effectiveness of FPO in improving model performance through online reinforcement learning, we examine its learning dynamics on the LIBERO benchmark and the ALOHA Transfer Cube task. Starting from an SFT baseline, FPO consistently increases both success rate and



(a) Policy Evolution (b) Result of Learning

Fig. 4. FPO online learning on the ALOHA Transfer Cube task. **(a)** Policy evolution at 0/0.8M/1.6M training steps: the baseline side-grasp failure mode is corrected to a robust top-down grasp that consistently completes the task. **(b)** Success rate (SR) curve: the smoothed trajectory (purple) steadily improves, surpassing the 40% baseline (red dashed) and reaching 65%, mirroring the behavioural change in (a).

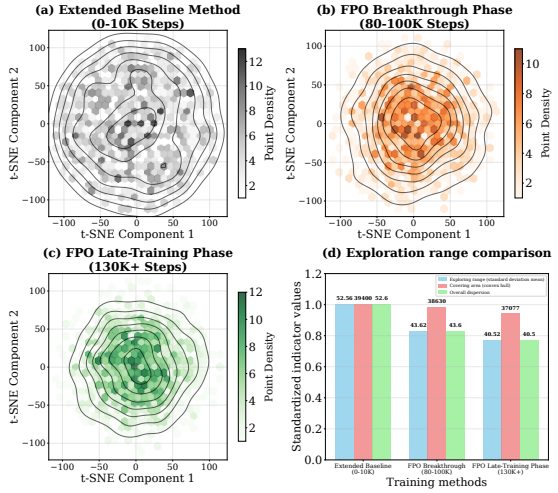


Fig. 5. FPO latent action space evolution. Visualized via t-SNE, this figure shows the policy's latent action distribution transitioning from broad exploration to focused exploitation across training stages. **(a)** Initial policy: wide, high-variance exploration. **(b)** Breakthrough phase: distribution concentrates around successful sequences. **(c)** Late-Training Phase: highly focused, low-variance exploitation of optimal regions. **(d)** Bar chart: quantifies reduced exploration range and dispersion, confirming convergence to refined behaviors.

average return throughout training, as shown in Fig. 3. The return curve exhibits a similar upward trend, while episode length remains steady downward trend, indicating that gains arise from discovering more direct and efficient strategies rather than extending trial duration.

Comparable behavior is observed in the ALOHA Transfer Cube task (Fig. 4). Beginning with the π_0 model at $\sim 40\%$ success, FPO exceeds 65% after a comparable training budget. The smoothed trajectory improves monotonically and avoids the instability typically seen in online RL under sparse rewards. Together, these results indicate stable online learning across distinct manipulation domains, supporting FPO as an effective fine-tuning framework for VLA policies.

C. Analysis of FPO's Internal Learning Mechanism

To analyze how FPO achieves its performance, we examined the evolution of its internal behavior during training by visualizing the distribution of latent action chunks across different stages, shown as Fig. 5. Using t-SNE for dimensionality reduction, the results reveal a clear trajectory from broad exploration to focused exploitation in the latent space.

In the initial phase (Fig. 5(a)), the policy, guided by the imitation prior, explores a wide region of the latent space, enabling the discovery of rewarding areas beyond the baseline policy. During the breakthrough phase (Fig. 5(b)), exploration becomes more structured and concentrated around successful action sequences, indicating prioritization of high-value regions. At convergence period (Fig. 5(c)), the distribution narrows into a low-variance cluster, reflecting efficient exploitation of the optimal region. The bar chart in Fig. 5(d) quantifies this transition, showing a marked reduction in dispersion and variance over training. These results demonstrate that FPO supports gradient-driven exploration beyond imitation priors and enables stable convergence to efficient task-solving behaviors.

In addition, the Fig. 6 visually demonstrates FPO's ability to resolve specific, recurring failure modes. The pre-trained π_0 policy often fails in a representative LIBERO task by attempting a suboptimal side grasp, leading to object instability (Fig. 6, top). FPO, after online fine-tuning, fundamentally alters this approach, consistently executing a robust top-down grasp from the same initial state that previously led to failure (Fig. 6, bottom). This illustrates FPO's capacity to learn physically grounded, effective trajectories through active online interaction, fixing nuanced, contact-rich errors that are challenging for offline methods alone.

TABLE II

ABLATIONS ON LIBERO-90 TASK *pick up the butter and put it in the tray*. NUMBERS ARE FINAL SUCCESS RATE (%). EACH ROW REMOVES ONE COMPONENT FROM FPO.

Method	Success Rate (%)
FPO (complete)	78.5
– without CFM ratio proxy	32.4
– without PPO clipping	45.1
– single-step integration (K=1)	61.7
– single critic (without ensemble)	71.2

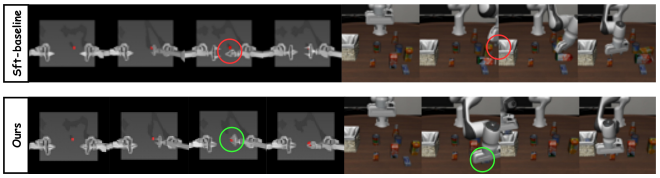


Fig. 6. FPO's ability to correct suboptimal behaviors. **(Top)** The SFT baseline policy consistently fails the task due to a suboptimal grasping approach inherited from the imitation prior. **(Bottom)** After online fine-tuning with FPO, the policy discovers a novel and successful trajectory from the same initial state, showcasing effective online correction.

D. Ablation Studies

To assess which components drive FPO's performance, we conduct ablations on the LIBERO-90 task *pick up the butter and put it in the tray*. Table II reports final success rate (%)

after training. Each variant disables exactly one component while keeping the training budget, architectures, and hyperparameters fixed. We consider four interventions: substituting the CFM-based ratio with an SAC-style latent-space update, removing PPO-style clipping, reducing exploration to single-step integration, and replacing the critic ensemble with a single critic. All ablations degrade performance relative to the complete method: replacing the CFM-based ratio causes the most substantial drop, removing clipping also leads to a marked reduction, limiting exploration depth yields a smaller but consistent decline, and using a single critic has the least impact yet remains non-negligible. Taken together, these results indicate that all components are consequential—the structure-aligned ratio and trust-region control account for a large portion of the gains, while exploration depth and value ensembling contribute additional stability and data efficiency.

a) Importance of the CFM-based Ratio Proxy.: Replacing the CFM-based policy ratio proxy with standard Soft Actor-Critic (SAC) in the latent space significantly reduced the success rate from 78.5% to 32.4%. This indicates that FPO's performance depends critically on the structurally-aware update rule that leverages the generative loss. Conventional RL in latent space is insufficient to achieve comparable results.

b) Effect of PPO-style Clipping.: Removing PPO-style clipping caused instability and reduced success to 45.1% (-33.4%). This confirms the role of clipping as a trust-region mechanism [11], preventing uncontrolled updates from the strong CFM signal and avoiding policy collapse.

c) Role of Multi-step Exploration.: Disabling multi-step Euler integration ($K = 1$) lowered success to 61.7% (-16.8%). This shows the benefit of generating temporally correlated latent trajectories, which improve the discovery of viable action sequences and enable stable execution in contact-rich tasks.

d) Contribution of the Q-Ensemble.: Using a single critic instead of a Q-ensemble reduced success to 71.2%. Although the effect is smaller (-7.2%), the ensemble improves stability by providing more reliable advantage estimates, which is particularly useful in sparse-reward settings.

E. Latent Space Characteristics of Successful Policies

To examine latent space characteristics distinguishing successful from failed policy executions, we conducted a statistical analysis of latent action chunks from the untrained imitation prior, failed rollouts after partial training, and successful rollouts after extended training. Fig. 7 summarizes the statistical differences across these groups. The t-SNE and PCA projections (Fig. 7a,b) show that successful trajectories occupy a compact and well-defined region of the latent space, whereas actions from the initial policy and failed rollouts are more dispersed. This separation indicates that FPO guides the policy toward a higher-performing latent subspace.

Analysis of action magnitudes (Fig. 7c) reveals that successful trajectories fall within a narrower range, suggesting avoidance of extreme actions and convergence toward an effective magnitude. The per-dimension variance plot (Fig. 7d) further shows reduced variance across most dimensions, confirming that successful policies act with greater precision by suppressing unnecessary exploratory noise and focusing capacity on reliable execution. This transition reflects effective skill acquisition in the latent space.

Quantitatively, successful rollouts exhibit higher silhouette scores and larger Mahalanobis distance to the success centroid, alongside lower within-cluster variance and reduced first-order temporal differences, indicating concentration of probability mass in a stable high-value latent region.

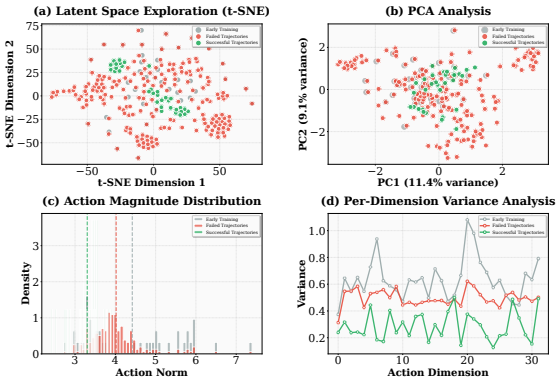


Fig. 7. Latent space analysis on initial policy, failed and successful Trajectories. The t-SNE and PCA plots (top row) demonstrate that successful trajectories (green) converge to a distinct, highly structured region of the latent space, sharply contrasting with the diffuse distributions of the initial policy (blue) and failed attempts (red). The action magnitude distribution (bottom-left) shows successful trajectories favoring a narrower, optimal range of action norms. Critically, the per-dimension variance plot (bottom-right) reveals significantly lower variance across most latent dimensions for successful trajectories, indicating learned precision and intentionality.

V. CONCLUSION

This work introduced a FPO algorithm for online fine-tuning of flow-matching VLA policies. FPO resolves the incompatibility with conventional policy-gradient methods by replacing explicit likelihood ratios with a likelihood-free proxy derived from per-sample changes in the conditional flow-matching objective, thereby enabling PPO-style clipped updates without Jacobian or density evaluation. The method incorporates structure-aware credit assignment in the latent space, a clipped surrogate objective, multi-step latent exploration, and a Q-ensemble, enabling stable and efficient optimization in sparse-reward and contact-rich environments. Experiments on the LIBERO benchmark and the ALOHA Transfer Cube task demonstrate that π_0-FPO consistently outperforms imitation-trained priors and strong baselines, including OpenVLA, Octo, Diffusion Policy, GRAPE, and π_0-FAST. Ablation experiments and latent space dynamics analysis not only confirm the efficacy of individual FPO components, but also demonstrate enhanced mitigation of recurring failure patterns through qualitative assessment. In the future, we will further enhance the few-shot adaptation capability based on limited online interactions, enabling faster learning and transfer while minimizing additional data requirements.

ACKNOWLEDGMENT

This study is supported by the State Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology (Grant No. JS202401), the funding from Institute of Automation, Chinese Academy of Sciences (Grant No. E411230101), and the National Natural Science Foundation of China (Grant No. 62576341 and No. 32441109).

REFERENCES

[1] M. J. Kim *et al.*, "OpenVLA: An Open-Source Vision-Language-Action Model," *arXiv:2406.09246*, 2024.

[2] Octo Model 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:2405.12213*, 2024.

[3] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, *et al.*, "π_0: A Vision-Language-Action Flow Model for General Robot Control," *arXiv:2410.24164*, 2024.

[4] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow Matching for Generative Modeling," *arXiv:2210.02747*, 2022.

[5] A. Tong *et al.*, "Improving and generalizing flow-based generative models with minibatch optimal transport," *Transactions on Machine Learning Research (TMLR)*, 2024. *arXiv:2302.00482*

[6] 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:2505.18719*, 2025.

[7] 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," *ICRA*, 2025.*arXiv:2501.16664*

[8] T. Zhang, C. Yu, S. Su, and Y. Wang, "ReinFlow: Fine-tuning Flow Matching Policy with Online Reinforcement Learning," *arXiv:2505.22094*, 2025.

[9] 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:2409.00588*, 2024.

[10] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to Control: Learning Behaviors by Latent Imagination," *arXiv:1912.01603*, 2019.

[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv:1707.06347*, 2017.

[12] L. Chen, R. Paleja, and M. Gombolay, "Learning from Sub-optimal Demonstration via Self-Supervised Reward Regression," arXiv:2010.11723, 2020.

[13] M. U. D. Waseem Akram, L. S. Saoud, J. Rosell, and I. Hussain, "Vision Language Action Models in Robotic Manipulation: A Systematic Review," *arXiv:2507.10672*, 2025.

[14] L. Ouyang *et al.*, "Training Language Models to Follow Instructions with Human Feedback," *arXiv:2203.02155*, 2022.

[15] Y. Bai *et al.*, "Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback," *arXiv:2204.05862*, 2022.

[16] N. Stiennon *et al.*, "Learning to Summarize from Human Feedback," *NeurIPS*, 2020.*arXiv:2009.01325*

[17] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust Region Policy Optimization," *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, PMLR 37:1889–1897, 2015.

[18] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural Ordinary Differential Equations," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[19] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud, "FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models," *arXiv:1810.01367*, 2018.

[20] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning," in *NeurIPS Datasets & Benchmarks Track*, 2023.

[21] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid, "ALOHA Unleashed: A Simple Recipe for Robot Dexterity," *arXiv:2410.13126*, 2024.

[22] David McAllister *et al.*, "Flow Matching Policy Gradients," arXiv:2507.21053, 2025.

[23] A. Wagenmaker *et al.*, "Steering Your Diffusion Policy with Latent Space Reinforcement Learning," *arXiv:2506.15799*, 2025.

[24] A. X. Lee, A. Nagabandi, P. Abbeel, and S. Levine, "Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model," in *NeurIPS*, 2020. *arXiv:1907.00953*

[25] H. Wang, S. Lin, and J. Zhang, "Adaptive Ensemble Q-learning: Minimizing Estimation Bias via Error Feedback," in *NeurIPS*, 2021.

[26] A. Nikulin *et al.*, "Q-Ensemble for Offline RL: Don't Scale the Ensemble, Scale the Batch Size," *arXiv:2211.11092*, 2022.

[27] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion," in *Robotics: Science and Systems (RSS)*, 2023.

[28] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine, "FAST: Efficient Action Tokenization for Vision-Language-Action Models," *arXiv:2501.09747*, 2025.

[29] 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:2411.19309*, 2024.

[30] Y. Wang, H. He, C. Wen, and X. Tan, "Truly Proximal Policy Optimization," *arXiv:1903.07940*, 2019.

[31] G. Chen, Y. Peng, and M. Zhang, "An Adaptive Clipping Approach for Proximal Policy Optimization," *arXiv:1804.06461*, 2018.

[32] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," *arXiv:1506.02438*, 2015.

[33] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. G. Bellemare, "Reincarnating Reinforcement Learning: Reusing Prior Computation to Accelerate Progress," in *NeurIPS 2022*, pp. 28955–28971.

[34] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[35] S. Pfrommer, Y. Huang, and S. Sojoudi, "Reinforcement Learning for Flow-Matching Policies," arXiv:2507.15073, 2025.

[36] J. Liu, G. Liu, J. Liang, *et al.*, "Flow-GRPO: Training Flow Matching Models via Online RL," arXiv:2505.05470, 2025.

[37] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *ICLR*, 2022.

[38] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. *arXiv:2006.11239*

[39] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-Based Generative Modeling through Stochastic Differential Equations," in *International Conference on Learning Representations (ICLR)*, 2021.

[40] Open X-Embodiment Collaboration, "Open X-Embodiment: Robotic Learning Datasets and RT-X Models," *arXiv:2310.08864*, 2023.

[41] 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," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[42] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep Reinforcement Learning from Human Preferences," *arXiv:1706.03741*, 2017.

[43] B. T. Polyak and A. B. Juditsky, "Acceleration of Stochastic Approximation by Averaging," *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

[44] T. P. Lillicrap, J. J. Hunt, A. Pritzel, *et al.*, "Continuous Control with Deep Reinforcement Learning," *arXiv:1509.02971*, 2015.

[45] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, "Deep Exploration via Bootstrapped DQN," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[46] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[47] A. Brohan *et al.*, "RT-1: Robotics Transformer for Real-World Control at Scale," *arXiv:2212.06817*, 2022.

[48] A. Brohan *et al.*, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," *arXiv:2307.15818*, 2023.

[49] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "BC-Z: Zero-Shot Task Generalization with Robotic Imitation Learning," in *CoRL*, 2022.