UniSteer: Unified Noise Steering for Efficient Human-Guided VLA Adaptation

Junjie Lu[1*†], **Xinyao Qin**[2*†], **Yuhua Jiang**[2], **Kaixin Wang**[3‡], **Chuheng Zhang**[3‡]
Bin Liang[2], **Jun Yang**[2], **Min Xu**[1], **Li Zhao**[3§]
[1] University of Technology Sydney [2] Tsinghua University [3] Microsoft Research

Github: `https://github.com/microsoft/UniSteer`

Abstract

Diffusion-based vision-language-action (VLA) models have emerged as strong priors for robotic manipulation, yet adapting them to real-world distributions remains challenging. In particular, on-robot reinforcement learning (RL) is expensive and time-consuming, so effective adaptation depends on efficient policy improvement within a limited budget of real-world interactions. Noise-space RL lowers the cost by keeping the pretrained VLA fixed as a denoising generator while updating only a lightweight actor that predicts the noise. However, its performance is still limited due to inefficient autonomous exploration. Human corrective interventions can reduce this exploration burden, but they are naturally provided in action space, whereas noise-space finetuning requires supervision over noise variables. To address these challenges, we propose **UniSteer**, a **Uni**fied Noise **Steer**ing framework that combines human corrective guidance with noise-space RL through approximate action-to-noise inversion. Given a human corrective action, UniSteer inverts the frozen flow-matching decoder to recover a noise target, which provides supervised guidance for the same noise actor that is simultaneously optimized via reinforcement learning. Real-world experiments on diverse manipulation tasks show that UniSteer adapts more efficiently than strong noise-space RL and action-space human-in-the-loop baselines, improving the success rate from 20% to 90% in 66 minutes on average across four real-world adaptation tasks.

1 Introduction

Vision-language-action (VLA) models have recently shown strong capability across diverse robotic tasks, including single-arm manipulation [1–12], bimanual manipulation [13–19] and dexterous manipulation [20–23]. Among them, diffusion-based VLAs [e.g., 12, 14, 17] become popular due to their strong performance. However, when deployed in the real world, these pretrained models often encounter distribution shifts in scenes, objects, viewpoints, and contact dynamics. They can significantly degrade performance, especially in long-horizon, contact-rich, or high-precision manipulation tasks. While reinforcement learning (RL) provides a natural way to adapt VLA policies through task-specific interactions [24–35], on-robot RL remains costly: collecting real-world experience is expensive, failures consume substantial resources, and adaptation is constrained by tight wall-clock time budgets. This motivates the need for efficient VLA adaptation methods under limited on-robot interaction and wall-clock time budget.

One efficient way to avoid the high computational costs of adapting diffusion-based generative policies is *noise-space finetuning* [25, 33, 34, 36]. Specifically, we freeze the pretrained diffusion-based

[*]Equal contribution. [†]Work conducted at Microsoft Research. [‡]Corresponding authors. [§]Project lead.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

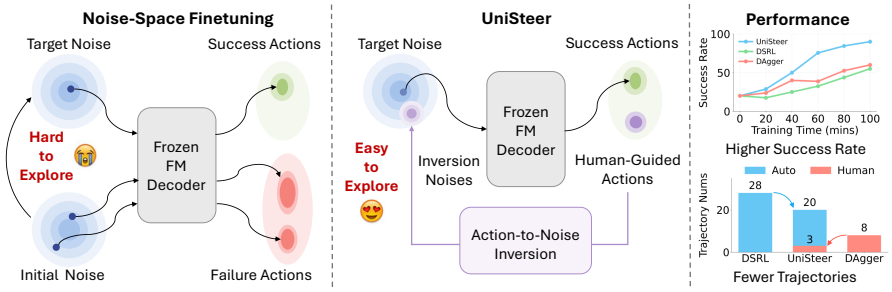


Figure 1: **UniSteer bridges human guidance and noise-space RL.** Noise-space finetuning relies on exploration from a distant initial noise distribution. UniSteer maps human corrective actions into noise-space targets, providing a useful prior for faster adaptation with fewer trajectories.

VLA as a denoising generator and steer a small noise actor to generate the noise. This confines adaptation to a small learnable module, reducing optimization cost while keeping the pretrained VLA frozen. As a result, the decoded actions naturally remain anchored to the pretrained policy prior, thus benefit from the priors in pretrained VLAs. However, since the VLA can degenerate under out-of-distribution (OOD) deployment conditions, policy improvement depends heavily on autonomous exploration, which can be inefficient under sparse rewards or large deployment shifts.

A natural way to mitigate this exploration bottleneck is to incorporate human guidance. While existing approaches for demonstrations [37], corrective interventions [38–40], and human-in-the-loop learning [24, 34, 41, 42] provide effective ways to refine robot behavior, they typically operate within the *action space*. Such action-space updates are often heavier and less stable compared with *noise-space finetuning*, since they require optimizing the action-generation pathway of a large generative policy. However, incorporating human corrective actions into noise-space is non-trivial. The mapping from noise to action is induced by a complex, learned transport process. Consequently, recovering a noise variable from a human-corrected action requires an implicit reverse Euler update whose velocity depends on the unknown preimage[43, 44]. Alternatively, backpropagating action-level supervision through the entire decoding chain is another option but it is computationally expensive.

To address this challenge, we propose **UniSteer**, a **Uni**fied Noise **Steer**ing framework that integrates human corrective guidance into noise-space online RL through approximate action-to-noise inversion. Although this inversion is not solved directly in closed form, we show that a tractable approximation yields low inversion error and provides reliable supervision in practice. Building on this inversion, we introduce a noise-space finetuning framework that unifies RL and human guidance through a shared noise-space interface, allowing the use of human corrections to supervise the same noise actor that is also optimized via RL. We validate this framework through extensive real-world robotic adaptation experiments, showing that it outperforms strong baselines with reduced time-to-performance and fewer human interventions. Our contributions are threefold:

(1) We introduce UniSteer, the first unified noise-space steering framework that incorporates human guidance into noise-space online RL for efficient real-world adaptation.

(2) We propose an approximate action-to-noise inversion method for flow-matching policies, enabling consistent noise supervision under a frozen generator.

(3) We demonstrate that our method achieves reduced time-to-performance with fewer human interventions than strong baselines across diverse real-world robotic manipulation tasks.

2 Related work

2.1 Vision-Language-Action Models

Vision-Language-Action (VLA) models have become a central paradigm for robot learning [1–19]. Early methods formulate action prediction as autoregressive token generation to integrate robot control into language-model-style sequence modeling [1–4, 10, 13, 18]. Other methods attach continuous action heads or diffusion-style decoders to vision-language models for better representation of high-dimensional continuous control [5, 17, 45, 46]. More recently, many VLA policies adopt flow-matching action heads, showing strong generative capability and promising performance in

real-world robotic manipulation [8, 12, 14–16, 19]. These policies generate action chunks by transporting initial noise variables to continuous actions through a learned state-conditioned velocity field. Our work focuses on this class of flow-matching VLA policies due to their competitive performance and leverages the initial noise variable as a natural interface for policy steering.

2.2 Adaptation of Flow-Matching VLA Policies

Reinforcement learning provides a natural approach for downstream VLA adaptation, but directly applying it to flow-matching policies is challenging. These policies generate actions through multi-step decoding, making end-to-end optimization costly and unstable. Moreover, their deterministic generation process makes the action log probabilities or likelihood ratios required by many RL algorithms difficult to obtain. Recent methods address this challenge by reformulating policy-gradient objectives for flow policies, such as RLinf [30] and FPO [47]. Other works avoid difficult online RL updates in the real world and instead improve policies through demonstrations, human interventions, or value-weighted behavior cloning, including HG-DAgger [38], AWR-style methods [48–50], and RECAP-style value-weighted imitation [26]. Another line of work keeps the pretrained generative policy frozen and adapts only a lightweight module, such as a residual correction network[51–54] or a noise-space policy [25, 33, 34, 36]. Compared with the above two lines of work, this paradigm offers a more lightweight and stable adaptation strategy by freezing the pretrained VLA and updating only a small module. Our method follows this lightweight adaptation paradigm, but differs from prior noise-space methods that mainly rely on autonomous exploration: we translate action-space human corrections into noise-space supervision targets, allowing human interventions to directly guide the compact noise actor while keeping the flow-matching VLA frozen. Inference-Time Policy Steering (ITPS) [55] also uses human input to steer frozen generative policies, but operates at inference time without updating model parameters.

3 Problem Formulation

3.1 Flow matching policies.

We consider VLA policies parameterized as conditional flow matching models. Given a state s, the policy maps an initial noise variable $z_0 \sim \mathcal{N}(0, I) \in \mathbb{R}^d$ to an action chunk $a \in \mathbb{R}^d$ by integrating a state-conditioned velocity field $v_\theta(z_t, t, s)$, where θ is the parameter of the network:

$$\frac{dz_t}{dt} = v_\theta(z_t, t, s).$$

At each flow step k, the network predicts a velocity update conditioned on the current noise, the flow time, and the state. Starting from z_0 at $t = 0$ and integrating forward to $t = 1$ produces the terminal sample deterministically

$$a = G_\theta(s, z_0) := z_1 = z_0 + \int_0^1 v_\theta(z_t, t, s)\, dt,$$

where a denotes the predicted action chunk. In practice, this continuous-time process is implemented using a K-step Euler discretization with step size Δt.

3.2 Noise-space policy optimization.

We formulate robotic policy learning as a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A} \subseteq \mathbb{R}^d$ is the continuous action space, $P(s' \mid s, a)$ is the transition dynamics, $r(s, a)$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. In our setting, the action is in fact the noise variable z because we freeze the decoding process $G_\theta(s, z)$. Specially, a lightweight policy $\psi_\phi(z \mid s)$ is trained to select the initial noise variable conditioned on the current state, and the frozen decoder maps it to an action chunk:

$$z \sim \psi_\phi(\cdot \mid s), \qquad a = G_\theta(s, z).$$

We further learn a noise-space critic $Q_\omega(s, z)$, which estimates the expected discounted return of the trajectory. $Q_\omega(s, z)$ is trained with the standard TD loss:

$$\mathcal{L}_{Q_\omega} = \left\| Q_\omega(s, z) - \left(r + \gamma \bar{Q}_\omega(s', z')\right) \right\|^2,$$

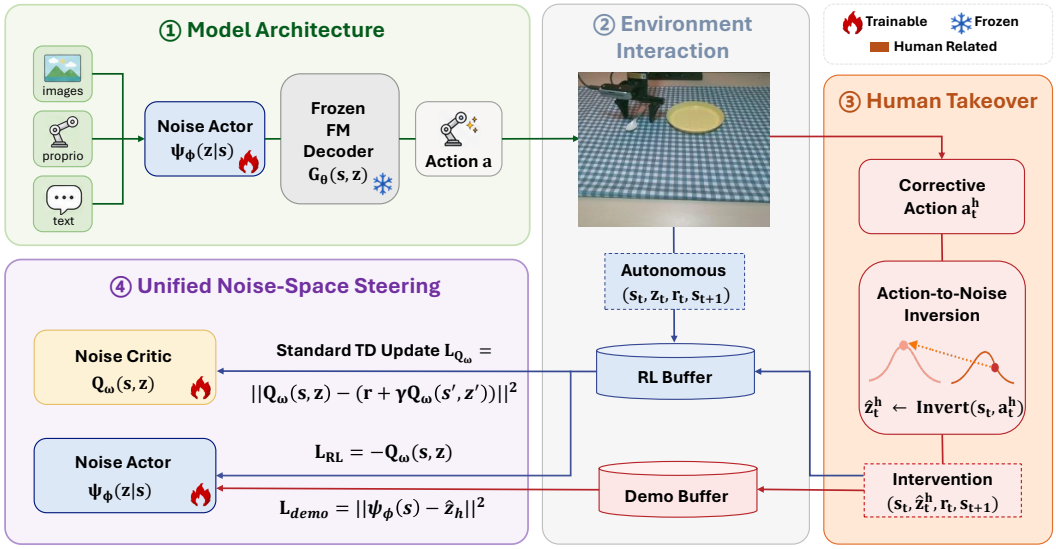


Figure 2: **Overview of UniSteer framework.** The noise actor maps current states into noise variables and generates actions through the frozen decoder. Autonomous rollouts are stored in the RL buffer, whereas human takeover actions are inverted into noise space and collected in both the demo buffer and the RL buffer. The two buffers jointly provide training signals for the critic and the actor.

The RL objective is therefore to optimize the policy with respect to this value function, while the pretrained flow decoder remains fixed.

$$\mathcal{L}_{\mathrm{RL}} = -Q_\omega(s,z), \qquad z \sim \psi_\phi(s).$$

4 Method

4.1 Action-to-Noise Inversion

Given the current state s and a human-provided corrective action chunk a^h, our goal is to recover a noise target $\hat{z}$ such that, the frozen flow-matching policy can decode it into an action chunk close to a^h. Ideally, the desired noise target is the inverse image $G_\theta(s,\cdot)^{-1}(a^h)$. However, since this noise-to-action mapping is realized through a multi-step nonlinear flow decoder, directly inverting it is difficult. We therefore exploit its stepwise Euler structure and invert the decoder one step at a time along the frozen decoding trajectory.

Proposition 1. *Suppose $v_\theta(z,t,s)$ is continuous in t and globally Lipschitz in z, uniformly for $t \in [0,1]$[1]. Then for a fixed state s, the map $G_\theta(s,\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ is bijective. In particular, for every action $a \in \mathbb{R}^d$, there exists a unique $z_0 \in \mathbb{R}^d$ such that $a = G_\theta(s, z_0)$.*

Remark. This proposition establishes the controllability of the noise-space actor. Intuitively, the global Lipschitz continuity of the velocity field prevents constant mapping or squashed mapping (such as $G_\theta(s,z_0) = \tanh(z_0)$).

For the frozen flow matching policy, forward decoding proceeds through K iterative Euler updates. At flow step k, the inverse of a single update can be written as a fixed-point equation:

$$x = y - \Delta t\, v_\theta(x, t_k, s) = g_y(x).$$

where x and y denote the input and output of that decoder step, respectively, and v_θ is the frozen velocity model.

Proposition 2. *Assume that $v_\theta(\cdot, t_k, s)$ is L-Lipschitz. If $\Delta t L < 1$, then the map $g_y(x) = y - \Delta t\, v_\theta(x, t_k, s)$ is contractive.*

[1]For each fixed state s, assume that $t \mapsto v_\theta(z,t,s)$ is continuous for every $z \in \mathbb{R}^d$, and that there exists a constant $L_s > 0$ such that $\|v_\theta(z,t,s) - v_\theta(z',t,s)\| \le L_s\|z - z'\|, \forall z, z' \in \mathbb{R}^d,\ \forall t \in [0,1]$.

Consequently, the inversion of the Euler step exists uniquely and can be recovered by fixed-point iteration. Starting from the corrective action chunk as the terminal state, we recover the noise states recursively as

$$\hat{z}_k = \mathrm{Inv}_k(\hat{z}_{k-1}, s), \qquad k = 1, \dots, K, \qquad \hat{z}_0 = a^h$$
$$\mathrm{Inv}_k(\hat{z}_{k-1}, s) \;:=\; \mathrm{fix}\Big(z \mapsto \hat{z}_{k-1} - \Delta t\, v_\theta(z, t_k, s)\Big).$$

Inv_k is defined as the fixed point of the inverse Euler map. In practice, we approximate this fixed point with M steps of iteration:

$$z_k^{(m+1)} = \hat{z}_{k-1} - \Delta t\, v_\theta\Big(z_k^{(m)}, t_k, s\Big), \quad m = 0, \dots, M-1, \qquad z_k^{(0)} = \hat{z}_{k-1}$$

After all K backward steps, we obtain the recovered initial noise $\hat{z} := \hat{z}_K$, which is used as the noise supervision target.

The condition $\Delta t L < 1$ is sufficient but not necessary for fixed-point convergence. Although our theory does not guarantee reliable inversion of arbitrary pretrained VLAs, Tables 2 and 3 empirically support accurate inversion and effective adaptation in our tested setting.

4.2 Unified Noise Steering Framework

With the inversion procedure, human corrective actions can be translated into noise-space supervision targets under the frozen flow matching policy. During online interaction, the noise actor ψ_ϕ selects an initial noise variable conditioned on the current state, and the frozen flow matching policy decodes it into an action chunk:

$$z_t \sim \psi_\phi(\cdot \mid s_t), \qquad a_t = G_\theta(s_t, z_t).$$

During autonomous interaction, we execute the decoded action chunk a_t and store the resulting transition (s_t, z_t, r_t, s_{t+1}) in the RL buffer $\mathcal{B}_{\mathrm{RL}}$. When a human takes over and provides a corrective action chunk a_t^h, we apply the inversion procedure in Sec4.1 to recover the corresponding noise target $\hat{z}_t^h$ and execute the corrective action. The corrected transition $(s_t, \hat{z}_t^h, r_t, s_{t+1})$ is stored in the RL buffer $\mathcal{B}_{\mathrm{RL}}$ for value learning and in the demo buffer $\mathcal{B}_{\mathrm{demo}}$ for correction supervision. This design allows human interventions to improve both critic learning and actor optimization under a unified noise-space representation.

For policy optimization, we first update the policy by sampling batches from demo buffer $\mathcal{B}_{\mathrm{demo}}$ by

$$\mathcal{L}_{\mathrm{demo}} = ||\psi_\phi(s) - \hat{z}_h||_2^2.$$

Then we sample from $\mathcal{B}_{\mathrm{RL}}$ and conduct optimization using $\mathcal{L}_{\mathrm{RL}}$. This design allows RL and human guidance to update the same noise actor through a shared noise interface. The RL branch improves the policy from reward feedback, while the correction branch injects action-space human guidance after translating it into noise-space targets through inversion. Algorithm 1 summarizes the overall training procedure.

5 Experiment

5.1 Experimental Setup

Tasks and Platform We evaluate our method on four real-world manipulation tasks: *pick up spoon*, *stack blocks*, *insert square*, and *fold towel*, as shown in Figure 3. These tasks cover a broad range of manipulation skills, including pick-and-place, precise contact-rich operations, and deformable-object manipulation. Together, they provide a compact benchmark for evaluating both broad spatial generalization and fine-grained manipulation refinement under distribution shift. All experiments are conducted on an AgileX Piper robot with a master–slave teleoperation setup for human takeover. For all tasks, the observations consist of two RGB images from a side camera and a wrist camera, together with the robot's 6D end-effector pose and gripper state. The action space is defined as the end-effector target pose with gripper openness, and the robot is controlled at 30 Hz. We first fine-tune a pretrained π_0 VLA with a flow-matching action head using 30 demonstrations per target task to establish initial task competence. UniSteer then performs task-specific online adaptation from this checkpoint while keeping the VLA frozen. All training runs are conducted on a single NVIDIA A100 GPU.

Baselines To fairly evaluate the effectiveness of our method, we compare against two representative baselines: DSRL[25], DAgger[37]. DSRL serves as the primary noise-space RL baseline, as it performs policy improvement under the same frozen generative policy backbone but without human intervention, thereby isolating the benefit of incorporating corrective guidance. DAgger represents a human-in-the-loop imitation learning approach, where corrective actions are aggregated and used for supervised policy updates in action space. In our implementation, DAgger updates all parameters of the VLA. Together, these baselines cover the relevant comparison axes for our setting: noise-space RL without human assistance, action-space imitation learning with human corrections. We initially considered HIL-SERL[41] as an additional human-in-the-loop online RL baseline. However, consistent with observations in RLToken [52], we found that HIL-SERL is poorly matched to our high-frequency control setting with sparse terminal rewards. In our preliminary experiments, HIL-SERL did not show consistent success-rate improvement; the setup differences and qualitative observations are discussed in Appendix E.

Evaluation Metrics We use real-world task success rate as the metric for evaluating each method's performance. For each task, the final policy is evaluated over 20 real-world trials, and the success rate is computed as the percentage of successful trials. For Pick up Spoon, Stack Blocks, and Insert Square, we randomly select 10 distinct initial object positions and evaluate each position twice. Among these positions, 20% are out-of-distribution cases not covered by the initial demonstration data, allowing us to evaluate both in-distribution adaptation and spatial generalization to unseen object placements. For Fold Towel, we run 20 consecutive trials, manually resetting the towel to the same nominal initial placement before each trial. We do not use the position grid because the towel occupies a large portion of the workspace and translating it may move part of it beyond the robot's reach. Placement, orientation, and deformation are not systematically varied; trial-to-trial variation arises only from manual resets.

5.2 Main Results

Table 1 reports the real-world adaptation performance of UniSteer and all baselines across four manipulation tasks. UniSteer consistently achieves the highest success rate on all tasks, improving the average success rate from 20% for the initial policy to 90% after adaptation. Compared with DSRL, which performs reinforcement learning only in the noise space without human guidance, UniSteer improves the average success rate by 35 percentage points. Compared with DAgger, which uses human corrections for supervised action-space imitation, UniSteer improves the average success rate by 30 percentage points. These results show that translating human corrections into noise-space supervision substantially improves the efficiency of noise-space policy optimization.

The improvement is consistent across tasks with different manipulation characteristics. On Pick up Spoon and Stack Blocks, UniSteer achieves 90% and 95% overall success rates, respectively, indicating strong adaptation performance on pick-and-place and spatial rearrangement tasks. On Insert Square, UniSteer reaches 100% success, showing that the proposed noise-space steering is particularly effective for precision-sensitive contact-rich manipulation. Fold Towel remains the most challenging task due to deformable-object dynamics and higher execution variability, but UniSteer still improves the success rate to 75%, outperforming both DSRL and DAgger by a large margin.

The ID/OOD split further shows that UniSteer improves not only in-distribution adaptation but also spatial generalization to unseen object placements. For all three position-based tasks, the initial

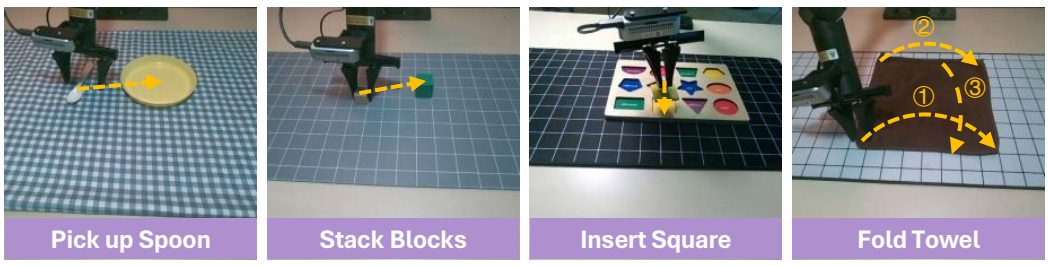
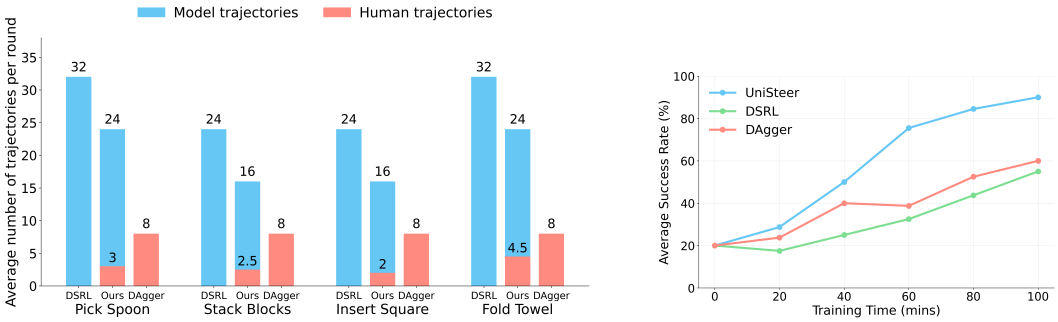


Figure 3: Overview of all real-world experimental tasks.

Table 1: Success rate (%) after task-specific real-world adaptation. For position-based tasks, Overall is evaluated over 20 trajectories, while ID and OOD are evaluated over 16 in-distribution and 4 out-of-distribution trajectories, respectively. Numbers in parentheses denote absolute success-rate improvement over the initial policy for the same split. The Average row is computed across the four task-level Overall success rates.

Task	Training Time (mins)	Split	Success Rate (%)			
			Initial Policy	UniSteer	DSRL	DAgger
Pick up Spoon	45	ID	25.0%	**87.5% (+62.5%)**	62.5% (+37.5%)	68.8% (+43.8%)
		OOD	0.0%	**100.0% (+100.0%)**	0.0% (+0.0%)	75.0% (+75.0%)
		Overall	20.0%	**90.0% (+70.0%)**	50.0% (+30.0%)	70.0% (+50.0%)
Stack Blocks	60	ID	43.8%	**93.8% (+50.0%)**	75.0% (+31.3%)	62.5% (+18.8%)
		OOD	0.0%	**100.0% (+100.0%)**	0.0% (+0.0%)	**100.0% (+100.0%)**
		Overall	35.0%	**95.0% (+60.0%)**	60.0% (+25.0%)	70.0% (+35.0%)
Insert Square	60	ID	18.8%	**100.0% (+81.3%)**	81.3% (+62.5%)	62.5% (+43.8%)
		OOD	0.0%	**100.0% (+100.0%)**	25.0% (+25.0%)	25.0% (+25.0%)
		Overall	15.0%	**100.0% (+85.0%)**	70.0% (+55.0%)	55.0% (+40.0%)
Fold Towel	100	Overall	10.0%	**75.0% (+65.0%)**	40.0% (+30.0%)	45.0% (+35.0%)
Average	66	Overall	20.0%	**90.0% (+70.0%)**	55.0% (+35.0%)	60.0% (+40.0%)



(a) Trajectory composition per adaptation round.

(b) Average success rate over real-world adaptation time across four tasks.

Figure 4: Online adaptation efficiency of UniSteer. UniSteer uses human guidance more efficiently to accelerate noise-space policy adaptation.

policy completely fails on OOD positions, obtaining 0% OOD success. After adaptation, UniSteer achieves 100% OOD success on Pick up Spoon, Stack Blocks, and Insert Square. In contrast, DSRL remains weak on OOD positions, achieving 0%, 0%, and 25% success on the three tasks, respectively. DAgger improves OOD performance on some tasks but is less consistent, achieving 75%, 100%, and 25%. These results suggest that human-guided noise-space supervision helps the policy adapt beyond the demonstrated state distribution, while still preserving strong in-distribution performance.

Overall, these results show that UniSteer improves both sample-efficient policy adaptation and final task performance under limited real-world interaction budgets, while also enhancing spatial generalization to unseen initial object positions.

5.3 Analysis of Adaptation Efficiency

We further analyze the trajectory composition collected during each adaptation round and present the results in Figure 4a. Compared with DSRL, which relies entirely on autonomous model rollouts, UniSteer uses fewer model-only trajectories while achieving substantially higher final success rates. This suggests that the gain of UniSteer does not come from collecting more autonomous experience, but from using human corrective signals to guide exploration more effectively. In contrast, DAgger

Table 2: Comparison of different action-to-noise supervision strategies across all four tasks. Inv. Time denotes the time used to recover noise targets, Train Time denotes the time used to train the noise model, Act. Loss denotes the reconstructed action-chunk MSE between the decoded action and the target human action, and SR denotes the success rate over eight evaluation trials per task and method.

Task	Method	Inv. Time (s)	Train Time (s)	Act. Loss	Total Time (s)	SR
Pick up Spoon	Fixed-point inversion (Ours)	**23.05**	50.21	**0.00122**	**73.26**	**8/8**
	Optimization-based inv.	157.83	50.21	0.06516	208.04	3/8
	Direct action supervision	N/A	193.22	N/A	193.22	**8/8**
Stack Blocks	Fixed-point inversion (Ours)	**24.60**	51.39	**0.00234**	**75.99**	**8/8**
	Optimization-based inv.	168.83	51.39	0.07122	220.22	3/8
	Direct action supervision	N/A	201.77	N/A	201.77	7/8
Insert Square	Fixed-point inversion (Ours)	**10.32**	30.10	**0.00018**	**40.42**	**8/8**
	Optimization-based inv.	50.86	30.10	0.05624	80.96	4/8
	Direct action supervision	N/A	90.43	N/A	90.43	7/8
Fold Towel	Fixed-point inversion (Ours)	**118.66**	146.59	**0.00385**	**265.25**	**5/8**
	Optimization-based inv.	589.94	146.59	0.09737	736.53	1/8
	Direct action supervision	N/A	820.18	N/A	820.18	4/8

relies exclusively on human-collected trajectories, requiring 8 pure human trajectories per round for every task. UniSteer requires far fewer pure human trajectories, averaging only 0.98 per round across tasks, while additionally leveraging a small number of mixed trajectories where the model acts autonomously before human takeover.

This trajectory composition highlights the key efficiency advantage of UniSteer. Rather than treating human interventions only as action-space demonstrations, UniSteer converts them into noise-space supervision targets and uses them jointly with reinforcement learning. As a result, even a small number of human and mixed trajectories can steer the noise actor toward more useful exploration, leading to better adaptation performance than both purely autonomous noise-space RL and action-space imitation from human corrections.

Figure 4b shows the average online adaptation curve across the four real-world tasks. UniSteer improves more rapidly than both DSRL and DAgger throughout training, indicating better time-to-performance under the same real-world interaction budget. This suggests that human corrective guidance, when translated into noise-space supervision, accelerates policy improvement rather than merely improving the final converged performance.

The different learning trends also reflect the limitations of the two baselines. DSRL relies only on autonomous exploration in noise space, which makes improvement slow under sparse rewards and real-world interaction constraints. This issue is especially pronounced when the initial success rate is low: successful trajectories are rarely collected, so the policy receives limited useful reward signals and exploration becomes highly inefficient. DAgger benefits from human corrections early in training, but action-space imitation alone does not consistently translate corrections into robust policy improvement. In contrast, UniSteer uses human interventions to steer the noise actor toward more promising noise regions while continuing to optimize with reinforcement learning, leading to faster and more stable adaptation.

5.4 Analysis of Action-to-Noise Supervision Strategies

We compare fixed-point inversion with two alternative strategies for using action-space supervision. *Optimization-based inversion* treats the noise as an optimizable variable and searches for a target noise that reconstructs the human corrective action through the frozen decoder. *Direct action supervision* does not construct an explicit noise target; instead, it feeds the actor-predicted noise into the frozen decoder and updates the actor through action reconstruction loss.

Table 2 shows that fixed-point inversion achieves the best trade-off among efficiency, inversion accuracy, and downstream success. Compared with optimization-based inversion, it recovers more

Table 3: Fixed-point inversion performance under different numbers of fixed-point iterations M on 316 human correction action chunks from all four tasks. We report the average inversion time per sample and the mean, median, and 90th percentile of the reconstructed action loss, computed as the MSE between the decoded action chunk and the target human action chunk.

M	Time / Sample (s)	Mean Act. Loss	Median Act. Loss	P90 Act. Loss
4	0.0447	0.002526	0.000753	0.003939
8	0.0670	0.002096	0.000224	0.003683
16	0.1117	0.002017	0.000111	0.003631
32	0.1980	0.002004	0.000094	0.003618
64	0.3598	0.001982	0.000081	0.003589
128	0.6845	0.001977	0.000078	0.003571

Table 4: Comparison of different training-stage schedules across all four tasks. Overall is evaluated over 20 trials per task. For the three position-based tasks, ID and OOD are evaluated over 16 in-distribution and 4 out-of-distribution trials, respectively.

Task	Split	Success Rate (%)			
		SFT-then-RL	RL-then-SFT	Only SFT	Only RL
Pick up Spoon	ID	**87.5%**	68.8%	81.3%	62.5%
	OOD	**100.0%**	**100.0%**	75.0%	0.0%
	Overall	**90.0%**	75.0%	80.0%	50.0%
Stack Blocks	ID	**93.8%**	81.3%	87.5%	75.0%
	OOD	**100.0%**	75.0%	50.0%	0.0%
	Overall	**95.0%**	80.0%	80.0%	60.0%
Insert Square	ID	**100.0%**	81.3%	81.3%	81.3%
	OOD	**100.0%**	**100.0%**	**100.0%**	25.0%
	Overall	**100.0%**	85.0%	85.0%	70.0%
Fold Towel	Overall	**75.0%**	45.0%	50.0%	40.0%
Average	Overall	**90.0%**	71.3%	73.8%	55.0%

reliable noise targets with much lower computational cost. Compared with direct action supervision, it avoids expensive backpropagation through the frozen decoder during actor updates while achieving comparable success rates. These results support our design choice of using fixed-point inversion to construct explicit, model-consistent noise targets. This provides an efficient and stable supervision signal for learning from human corrective actions in noise space.

5.5 Analysis of Fixed-Point Inversion Iterations

We analyze the effect of the number of fixed-point iterations M on inversion quality and computation cost. For each recovered noise target, we decode it through the frozen flow-matching decoder and measure the reconstructed action-chunk MSE against the original human correction. Table 3 shows that fixed-point inversion already achieves low reconstruction error with a small number of iterations, and that increasing M yields quickly diminishing returns while steadily increasing computation. This indicates that the fixed-point iteration converges rapidly in the human-correction regime and produces model-consistent noise targets efficiently. We use $M = 16$ as a practical trade-off: increasing M from 16 to 128 reduces P90 reconstruction loss by only 1.65% while increasing inversion time by approximately $6.1\times$. Additional numerical-convergence and OOD inversion diagnostics are reported in Appendix K.

5.6 Analysis of Training Schedule

Table 4 ablates the effect of different training-stage schedules across all four tasks. In the Only-RL setting, online adaptation uses no human takeovers or corrective data, and the replay buffer contains only autonomous transitions. The SFT-then-RL schedule used in UniSteer achieves the best

overall performance, suggesting that the two stages play complementary roles. The supervised stage first pulls the noise actor toward target noise regions inferred from human corrections, providing a more effective starting point for exploration. The subsequent RL stage then further improves the policy through task-level rewards, leading to both more efficient exploration and stronger final performance.

When the order is reversed, the supervised update after RL can overwrite or distort the reward-optimized noise distribution learned during reinforcement learning, reducing the benefit of prior exploration. Using only SFT suffers from limitations similar to imitation-based methods: it relies solely on corrective data, can overfit to the collected correction distribution, and may forget previously learned behaviors, leading to weaker generalization. Using only RL avoids supervised overwriting, but exploration remains inefficient when the initial success rate is low and useful reward signals are sparse. Overall, these results show that supervised noise-space steering is most effective when used to guide exploration before RL, while RL is needed to consolidate and improve the policy under task-level rewards.

6 Conclusion

We presented **UniSteer**, the first unified noise-space finetuning framework that integrates human corrective guidance into online RL for flow-matching VLA adaptation. The key idea is to bridge the mismatch between action-space human corrections and noise-space policy optimization through approximate action-to-noise inversion, allowing both human guidance and reward-driven learning to operate through a shared noise interface while keeping the generative decoder frozen. This formulation preserves the efficiency and stability of lightweight noise-space adaptation, while addressing its reliance on inefficient autonomous exploration. More broadly, we show that human interventions can serve not only as action-space demonstrations, but also as structured noise supervision for steering frozen generative robot policies, offering a practical path toward more efficient and scalable real-world VLA adaptation.

Acknowledgement

We sincerely thank Tianxiang Zhang for assistance with the robot control code used in our experiments, and Yucen Wang for helpful discussions and constructive suggestions. We also thank the reviewers for their valuable feedback. We acknowledge the support of the UTS President's Scholarship for this work.

References

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

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

[3] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: Robot manipulation with multimodal prompts. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 14975–15022. PMLR, 2023. URL `https://proceedings.mlr.press/v202/jiang23b.html`.

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

[5] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[6] Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22508–22519, 2025.

[7] Wei Zhao, Pengxiang Ding, Min Zhang, Zhefei Gong, Shuanghao Bai, Han Zhao, and Donglin Wang. Vlas: Vision-language-action model with speech instructions for customized robot manipulation. *arXiv preprint arXiv:2502.13508*, 2025.

[8] Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, et al. Villa-x: enhancing latent action modeling in vision-language-action models. *arXiv preprint arXiv:2507.23682*, 2025.

[9] Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent. *arXiv preprint arXiv:2501.18867*, 2025.

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

[11] Peiyan Li, Yixiang Chen, Hongtao Wu, Xiao Ma, Xiangnan Wu, Yan Huang, Liang Wang, Tao Kong, and Tieniu Tan. Bridgevla: Input-output alignment for efficient 3d manipulation learning with vision-language models. *arXiv preprint arXiv:2506.07961*, 2025.

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

[13] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[14] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.010.

[15] Physical Intelligence. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL `https://arxiv.org/abs/2504.16054`.

[16] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[17] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

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

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

[20] Yifan Zhong, Xuchuan Huang, Ruochong Li, Ceyao Zhang, Zhang Chen, Tianrui Guan, Fanlian Zeng, Ka Num Lui, Yuyao Ye, Yitao Liang, et al. Dexgraspvla: A vision-language-action framework towards general dexterous grasping. *arXiv preprint arXiv:2502.20900*, 2025.

[21] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025.

[22] Gu Zhang, Qicheng Xu, Haozhe Zhang, Jianhan Ma, Long He, Yiming Bao, Zeyu Ping, Zhecheng Yuan, Chenhao Lu, Chengbo Yuan, et al. Unidex: A robot foundation suite for universal dexterous hand control from egocentric human videos. *arXiv preprint arXiv:2603.22264*, 2026.

[23] Guangqi Jiang, Yutong Liang, Jianglong Ye, Jia-Yang Huang, Changwei Jing, Rocky Duan, Pieter Abbeel, Xiaolong Wang, and Xueyan Zou. Cross-hand latent representation for vision-language-action models. *arXiv preprint arXiv:2603.10158*, 2026.

[24] Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. Conrft: A reinforced fine-tuning method for vla models via consistency policy. *arXiv preprint arXiv:2502.05450*, 2025.

[25] Andrew Wagenmaker, Mitsuhiko Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. *arXiv preprint arXiv:2506.15799*, 2025.

[26] Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: A VLA That Learns From Experience. *arXiv preprint arXiv:2511.14759*, 2025.

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

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

[29] Hongyin Zhang, Zifeng Zhuang, Han Zhao, Pengxiang Ding, Hongchao Lu, and Donglin Wang. Reinbot: Amplifying robot visual-language manipulation with reinforcement learning. *arXiv preprint arXiv:2505.07395*, 2025.

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

[31] Yanjiang Guo, Jianke Zhang, Xiaoyu Chen, Xiang Ji, Yen-Jen Wang, Yucheng Hu, and Jianyu Chen. Improving vision-language-action model with online reinforcement learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15665–15672. IEEE, 2025.

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

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

[34] Piaopiao Jin, Qi Wang, Guokang Sun, Ziwen Cai, Pinjia He, and Yangwei You. Dual-actor fine-tuning of vla models: A talk-and-tweak human-in-the-loop approach. *arXiv preprint arXiv:2509.13774*, 2025.

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

[36] Omkar Patil, Ondrej Biza, Thomas Weng, Karl Schmeckpeper, Wil Thomason, Xiaohan Zhang, Robin Walters, Nakul Gopalan, Sebastian Castro, and Eric Rosen. You've got a golden ticket: Improving generative robot policies with a single noise vector. *arXiv preprint arXiv:2603.15757*, 2026.

[37] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[38] Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083. IEEE, 2019.

[39] Zheyuan Hu, Robyn Wu, Naveen Enock, Jasmine Li, Riya Kadakia, Zackory Erickson, and Aviral Kumar. Rac: Robot learning for long-horizon tasks by scaling recovery and correction. *arXiv preprint arXiv:2509.07953*, 2025.

[40] Guanxing Lu, Rui Zhao, Haitao Lin, He Zhang, and Yansong Tang. Human-in-the-loop online rejection sampling for robotic manipulation. *arXiv preprint arXiv:2510.26406*, 2025.

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

[42] Wenke Xia, Yichu Yang, Hongtao Wu, Xiao Ma, Tao Kong, and Di Hu. Human-assisted robotic policy refinement via action preference optimization. *arXiv preprint arXiv:2506.07127*, 2025.

[43] Chieh-Hsin Lai, Yang Song, Dongjun Kim, Yuki Mitsufuji, and Stefano Ermon. The principles of diffusion models. *arXiv preprint arXiv:2510.21890*, 2025.

[44] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[45] Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, et al. Dita: Scaling diffusion transformer for generalist vision-language-action policy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7686–7697, 2025.

[46] Junjie Wen, Minjie Zhu, Yichen Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Chengmeng Li, Xiaoyu Liu, Yaxin Peng, Chaomin Shen, et al. Diffusion-vla: Generalizable and interpretable robot foundation model via self-generated reasoning. *arXiv preprint arXiv:2412.03293*, 2024.

[47] David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. *arXiv preprint arXiv:2507.21053*, 2025.

[48] Zhian Su, Weijie Kong, Haonan Dong, and Huixu Dong. Ig-rft: An interaction-guided rl framework for vla models in long-horizon robotic manipulation. *arXiv preprint arXiv:2602.20715*, 2026.

[49] Rushuai Yang, Hecheng Wang, Chiming Liu, Xiaohan Yan, Yunlong Wang, Xuan Du, Shuoyu Yue, Yongcheng Liu, Chuheng Zhang, Lizhe Qi, et al. Aloe: Action-level off-policy evaluation for vision-language-action model post-training. *arXiv preprint arXiv:2602.12691*, 2026.

[50] Yiming Mao, Zixi Yu, Weixin Mao, Yinhao Li, Qirui Hu, Zihan Lan, Minzhao Zhu, and Hua Chen. Arm: Advantage reward modeling for long-horizon manipulation. *arXiv preprint arXiv:2604.03037*, 2026.

[51] Wenli Xiao, Haotian Lin, Andy Peng, Haoru Xue, Tairan He, Yuqi Xie, Fengyuan Hu, Jimmy Wu, Zhengyi Luo, Linxi Fan, et al. Self-improving vision-language-action models with data generation via residual rl. *arXiv preprint arXiv:2511.00091*, 2025.

[52] Charles Xu, Jost Tobias Springenberg, Michael Equi, Ali Amin, Adnan Esmail, Sergey Levine, and Liyiming Ke. Rl token: Bootstrapping online rl with vision-language-action models, 2026. URL `https://arxiv.org/abs/2604.23073`.

[53] Perry Dong, Qiyang Li, Dorsa Sadigh, and Chelsea Finn. EXPO: Stable reinforcement learning with expressive policies. *arXiv preprint arXiv:2507.07986*, 2025.

[54] Zhanyi Sun and Shuran Song. From prior to pro: Efficient skill mastery via distribution contractive RL finetuning. *arXiv preprint arXiv:2603.10263*, 2026.

[55] Yanwei Wang, Lirui Wang, Yilun Du, Balakumar Sundaralingam, Xuning Yang, Yu-Wei Chao, Claudia Pérez-D'Arpino, Dieter Fox, and Julie Shah. Inference-time policy steering through human interactions. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15626–15633, 2025. doi: 10.1109/ICRA55743.2025.11127931.

A Theoretical Analysis and Proofs

A.1 Invertibility of the Continuous Flow Decoder

Setup and statement. For a fixed state s, consider the continuous-time flow decoder

$$\frac{dz_t}{dt} = v_\theta(z_t, t, s), \qquad t \in [0, 1],$$

and denote the induced noise-to-action map by $G_\theta(s, z_0) = z_1$. We show that this map is bijective under standard Lipschitz regularity.

Proposition A.1. Assume that $v_\theta(\cdot, t, s)$ is Lipschitz continuous for all $t \in [0, 1]$. Then, for any fixed s, $G_\theta(s, \cdot)$ is bijective.

Proof. The Lipschitz assumption guarantees existence and uniqueness of the ODE solution for every initial condition. For surjectivity, fix any target action $a \in \mathbb{R}^d$ and solve the reverse-time ODE initialized at a:

$$\frac{dy_\tau}{d\tau} = -v_\theta(y_\tau, 1 - \tau, s), \qquad y_0 = a,$$
$$z_0 := y_1, \qquad z_t := y_{1-t}.$$

Then

$$\frac{dz_t}{dt} = -\frac{dy_\tau}{d\tau}\Big|_{\tau=1-t} = v_\theta(y_{1-t}, t, s) = v_\theta(z_t, t, s), \qquad z_1 = y_0 = a.$$

Thus z_t is a valid forward trajectory of the original ODE and $G_\theta(s, z_0) = a$, proving surjectivity. For injectivity, if $G_\theta(s, z_0) = G_\theta(s, z_0') = a$, then the reverse-time ODE from the same terminal value a has a unique solution, so the recovered initial values must coincide: $z_0 = z_0'$. Therefore, $G_\theta(s, \cdot)$ is bijective. □

A.2 One-Step Fixed-Point Inversion

Setup. We analyze the inverse of one Euler step of the frozen flow decoder. The forward step, inverse equation, and associated fixed-point map are

$$y = x + \Delta t\, v_\theta(x, t_k, s),$$
$$x = y - \Delta t\, v_\theta(x, t_k, s),$$
$$g_y(x) := y - \Delta t\, v_\theta(x, t_k, s).$$

Proposition A.2. Assume that $v_\theta(\cdot, t_k, s)$ is L-Lipschitz and $\Delta t L < 1$. Then g_y is a contraction, admits a unique fixed point $x^\star$, and the iteration

$$x^{(m+1)} = g_y(x^{(m)})$$

converges to $x^\star$ from any initialization.

Proof. For any x_1, x_2, the Lipschitz constant of g_y is bounded by

$$\|g_y(x_1) - g_y(x_2)\| = \Delta t \|v_\theta(x_1, t_k, s) - v_\theta(x_2, t_k, s)\|$$
$$\leq \Delta t L \|x_1 - x_2\|.$$

Since $\Delta t L < 1$, g_y is a contraction. By the Banach fixed-point theorem, it admits a unique fixed point $x^\star$, and the Picard iteration converges to $x^\star$ from any initialization. □

A.3 Finite-Step Approximation Error

Setup and statement. In practice, the exact fixed point is approximated by a finite number of fixed-point iterations. Let $x^\star$ be the unique fixed point of g_y, and let $\rho := \Delta t L < 1$. Then the M-step approximation satisfies

$$\|x^{(M)} - x^\star\| \leq \rho^M \|x^{(0)} - x^\star\|.$$

Proof. The contraction property gives

$$\begin{aligned}\|x^{(m+1)} - x^\star\| &= \|g_y(x^{(m)}) - g_y(x^\star)\| \\ &\le \rho\|x^{(m)} - x^\star\|.\end{aligned}$$

Recursively applying this inequality yields

$$\|x^{(M)} - x^\star\| \le \rho^M \|x^{(0)} - x^\star\|.$$

An alternative a posteriori bound is

$$\|x^{(M)} - x^\star\| \le \frac{\rho^M}{1-\rho}\|x^{(1)} - x^{(0)}\|.$$

Therefore, the finite-step inversion error decays geometrically with the number of iterations. □

A.4 Error Accumulation in Recursive Inversion

Setup. We analyze the cumulative error induced by applying approximate inversion over K Euler steps. Let F_k^{-1} denote the exact inverse of the k-th Euler step and $\mathrm{Inv}_k^{(M)}$ denote its M-step fixed-point approximation. Starting from the human corrective action a^h, the exact and approximate recursive inversions are

$$\begin{aligned} z_0^\star &= a^h, \quad z_k^\star = F_k^{-1}(z_{k-1}^\star, s), \qquad k = 1, \dots, K, \\ \hat{z}_0 &= a^h, \quad \hat{z}_k = \mathrm{Inv}_k^{(M)}(\hat{z}_{k-1}, s), \quad k = 1, \dots, K. \end{aligned}$$

Assumptions. For each step k, assume that the exact inverse is Lipschitz and the finite-step approximation error is geometrically bounded:

$$\begin{aligned} &\|F_k^{-1}(y_1, s) - F_k^{-1}(y_2, s)\| \le c_k\|y_1 - y_2\|, \\ &\|\mathrm{Inv}_k^{(M)}(y, s) - F_k^{-1}(y, s)\| \le C_k\rho_k^M, \qquad 0 < \rho_k < 1. \end{aligned}$$

Proposition A.3. Under the above assumptions, the recursive inversion error satisfies

$$\|\hat{z}_K - z_K^\star\| \le \sum_{k=1}^{K} \left(C_k \rho_k^M \prod_{\ell=k+1}^{K} c_\ell \right).$$

In particular, for any $\delta > 0$, there exists M_0 such that for all $M \ge M_0$,

$$\|\hat{z}_K - z_K^\star\| < \delta.$$

Proof. Define the stepwise error $e_k := \|\hat{z}_k - z_k^\star\|$. Since both recursions start from the same terminal action, $e_0 = 0$. For each step $k \ge 1$, we have

$$\begin{aligned} e_k &= \|\mathrm{Inv}_k^{(M)}(\hat{z}_{k-1}, s) - F_k^{-1}(z_{k-1}^\star, s)\| \\ &\le \|\mathrm{Inv}_k^{(M)}(\hat{z}_{k-1}, s) - F_k^{-1}(\hat{z}_{k-1}, s)\| \\ &\quad + \|F_k^{-1}(\hat{z}_{k-1}, s) - F_k^{-1}(z_{k-1}^\star, s)\| \\ &\le C_k\rho_k^M + c_k e_{k-1}. \end{aligned}$$

Here the first inequality adds and subtracts $F_k^{-1}(\hat{z}_{k-1}, s)$, and the second inequality uses the approximation bound and Lipschitz continuity. Unrolling the recursion from $e_0 = 0$ gives

$$e_K \le \sum_{k=1}^{K} \left(C_k \rho_k^M \prod_{\ell=k+1}^{K} c_\ell \right).$$

Since $\rho_k \in (0, 1)$ for every fixed k, each term converges to zero as $M \to \infty$. The right-hand side is a finite sum, so it also converges to zero. Therefore, for any $\delta > 0$, there exists M_0 such that $M \ge M_0$ implies $\|\hat{z}_K - z_K^\star\| < \delta$. □

B Limitations

UniSteer still has several limitations. First, its action-to-noise inversion depends on the frozen flow-matching decoder, and the recovered noise targets may be slightly shifted from the initial distribution of the noise actor. Second, our experiments are conducted on four tasks with one robot platform, and broader evaluation across more robots, tasks, and longer-horizon settings is needed.

C Algorithm of UniSteer Framework

In each adaptation round, we perform all SFT updates before RL updates. The numbers of updates are $N_{\mathrm{SFT}} = 1.0\, n_h$ and $N_{\mathrm{RL}} = 2.0\, n_m$, where n_h and n_m count newly collected human-corrected and autonomously executed model chunks, respectively.

Algorithm 1 Algorithm of UniSteer Framework

Require: Frozen flow matching policy G_θ, noise actor ψ_ϕ, noise-space critic Q_ω, RL buffer $\mathcal{B}_{\mathrm{RL}}$, demo buffer $\mathcal{B}_{\mathrm{demo}}$

for each adaptation round **do**
  for each time step t **do**
    Sample noise $z_t \sim \psi_\phi(\cdot \mid s_t)$ and decode action chunk $a_t = G_\theta(s_t, z_t)$
    if human takeover with corrective action chunk a_t^h **then**
      Recover noise target $\hat{z}_t^h \leftarrow \mathrm{Invert}(s_t, a_t^h)$
      Execute a_t^h and observe (r_t, s_{t+1})
      Store $(s_t, \hat{z}_t^h, r_t, s_{t+1})$ in $\mathcal{B}_{\mathrm{RL}}$ and $\mathcal{B}_{\mathrm{demo}}$
    else
      Execute a_t and observe (r_t, s_{t+1})
      Store (s_t, z_t, r_t, s_{t+1}) in $\mathcal{B}_{\mathrm{RL}}$
    end if
  end for
  Compute $N_{\mathrm{SFT}} = 1.0\, n_h$ and $N_{\mathrm{RL}} = 2.0\, n_m$ from this round's collected chunks
  for $i = 1, \ldots, N_{\mathrm{SFT}}$ **do**
    Update ψ_ϕ with $\mathcal{L}_{\mathrm{demo}}$ on $\mathcal{D}_{demo} \sim \mathcal{B}_{demo}$
  end for
  for $j = 1, \ldots, N_{\mathrm{RL}}$ **do**
    Update Q_ω , ψ_ϕ with $\mathcal{L}_{Q_\omega}$, $\mathcal{L}_{RL}$ on $\mathcal{D}_{RL} \sim \mathcal{B}_{RL}$
  end for
end for

D Hyperparameters

Table 5: Implementation hyperparameters used in our experiments.

Component	Value	Component	Value
π_0 denoising steps	10	Action horizon	16
Action dimension	7	State dimension	2055
Image size	$224 \times 224 \times 6$	Noise representation dim	$16 \times 7 = 112$
Actor hidden dims	$[1024, 1024, 1024]$	Critic hidden dims	$[1024, 1024, 1024]$
CNN features	$[32, 32, 32, 32]$	CNN strides	$[3, 2, 2, 2]$
Replay batch size	256	Replay capacity	33333
Discount γ	0.99	Target update rate τ	0.005
Actor LR	1×10^{-4}	Critic LR	3×10^{-4}
Temperature LR	3×10^{-4}	Actor-SFT LR	6×10^{-5}
Learnable temperature	True	Initial temperature	1.0
Target entropy	0.0	$\log \sigma_{\min}$	-20.0

E Preliminary HIL-SERL Observations

Standard HIL-SERL uses step-wise actor–critic learning, whereas UniSteer treats each 16-step action chunk, executed at 30 Hz, as one RL transition. At the same control frequency, single-step learning involves approximately 16 times as many transitions over the same physical duration. Chunk-level adaptation would require changes to the actor action space, critic transitions, and handling of human interventions.

In our preliminary runs, success rates did not improve consistently, and the robot sometimes repeated local actions near the goal after human interventions without completing the task. These are preliminary qualitative observations, not a controlled comparison, and do not establish that HIL-SERL is generally ineffective or unstable.

F Case Study



(a) DSRL rollout during early adaptation.



(b) UniSteer rollout during early adaptation with human-guided noise supervision.

Figure 5: Qualitative comparison of early exploration behavior on Pick up Spoon. DSRL relies on autonomous noise-space exploration and repeatedly approaches the spoon from above, which rarely leads to successful grasping. In contrast, UniSteer uses action-to-noise inversion to convert human corrective actions into noise-space supervision, guiding the noise actor toward lower and more successful grasping motions.

Figure 5 provides a qualitative comparison of early exploration behavior on the Pick up Spoon task. During early adaptation, DSRL relies entirely on autonomous noise-space exploration. As shown in Figure 5a, the robot repeatedly approaches the spoon from above but fails to reach a graspable pose. This leads to almost no successful trajectories, making reward-driven improvement inefficient under sparse task rewards.

In contrast, UniSteer uses human corrective actions to provide noise-space supervision through action-to-noise inversion. As shown in Figure 5b, after receiving human-guided noise targets, the robot explores lower motions around the spoon and is able to grasp it successfully. This qualitative behavior illustrates the key advantage of UniSteer: human corrections steer the noise actor toward more promising exploration regions, enabling the policy to collect successful trajectories earlier and improving the efficiency of subsequent reinforcement learning.

Figure 6 illustrates a typical forgetting behavior of DAgger during online adaptation. As shown in Figure 6a, after the second adaptation round, DAgger learns to grasp the upright spoon configuration. However, when evaluated on the tilted spoon configuration that was learned in the previous round, the policy fails to preserve the earlier behavior, as shown in Figure 6b. This suggests that directly aggregating corrective actions in action space can introduce interference between different correction distributions, causing newly learned behaviors to overwrite earlier ones. In contrast, UniSteer translates human corrections into noise-space supervision and combines them with reward-driven



(a) After the second adaptation round, DAgger learns to grasp the upright spoon.



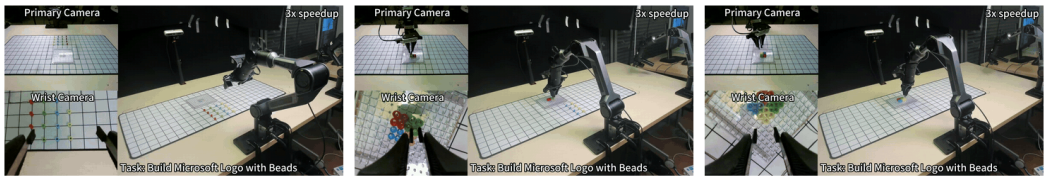
(b) After the second adaptation round, DAgger fails to reproduce the tilted-spoon grasping behavior observed after the first round.

Figure 6: Qualitative example of a possible interference failure mode in DAgger on Pick up Spoon. After the second adaptation round, DAgger succeeds on the upright-spoon configuration but fails on the tilted-spoon configuration it previously handled successfully.

optimization, which helps reduce such forgetting while improving exploration. This case study provides a qualitative illustration of a possible interference failure mode.

G Bead-Placement Demonstration

We further present a demonstration of UniSteer on a bead-placement task, arranging 16 beads of 5 mm radius into the Microsoft logo. This task involves repeated precise placement of small objects. Two stage-specific noise actors handle the first and last eight placements, respectively. Figure 7 shows representative frames from the demonstration.



(a) Stage 1: start (b) Stage 1: final frame (c) Stage 2: final frame

Figure 7: Preliminary qualitative demonstration of the Microsoft-logo bead-placement task. Frames show the start of stage 1, the end of stage 1, and the end of stage 2. Demonstration videos: `https://github.com/microsoft/UniSteer`.

H Real-world experimental setup

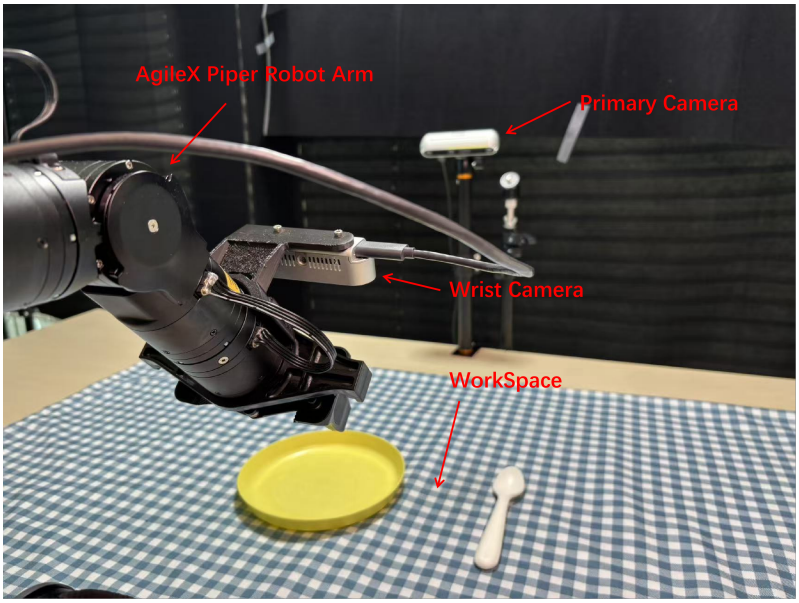


Figure 8: We use an AgileX Piper robot arm equipped with a wrist camera, together with a primary camera observing the tabletop workspace. The policy takes both camera views, the robot end-effector pose, and gripper state as observations for real-world manipulation.

I Additional Evaluation Results

We conduct 20 additional evaluation trials for each task and method using the same trained policies and evaluation protocol as in the main experiments. We report the combined results over the original and additional trials, totaling 40 trials per task and method, with Wilson 95% confidence intervals.

Table 6: Combined evaluation results over 40 trials per task and method. Each cell reports success rate (%) and the Wilson 95% confidence interval in brackets.

Task	Success Rate (%)			
	Initial Policy	UniSteer	DSRL	DAgger
Pick up Spoon	20.0% [10.5, 34.8]	**92.5%** [80.1, 97.4]	52.5% [37.5, 67.1]	70.0% [54.6, 81.9]
Stack Blocks	35.0% [22.1, 50.5]	**95.0%** [83.5, 98.6]	60.0% [44.6, 73.7]	70.0% [54.6, 81.9]
Insert Square	15.0% [7.1, 29.1]	**100.0%** [91.2, 100.0]	67.5% [52.0, 79.9]	55.0% [39.8, 69.3]
Fold Towel	10.0% [4.0, 23.1]	**77.5%** [62.5, 87.7]	42.5% [28.5, 57.8]	45.0% [30.7, 60.2]
Pooled across tasks	20.0% [14.5, 26.9]	**91.3%** [85.8, 94.7]	55.6% [47.9, 63.1]	60.0% [52.3, 67.3]

J Human Effort and Training Budget

We report success rates together with cumulative active human-control time under the same training budget on a single NVIDIA A100 GPU. Human time includes only active teleoperation or takeover time and excludes waiting time and the 30 initialization demonstrations per task shared by all methods.

Table 7: Success rate (SR) and cumulative active human-control time at different GPU training budgets. All times are in minutes.

GPU training time (min)	UniSteer		DSRL		DAgger	
	SR (%)	Human time	SR (%)	Human time	SR (%)	Human time
0	20.0	0.0	20.0	0.0	20.0	0.0
20	**28.8**	3.0	17.5	0.0	23.8	6.7
40	**50.0**	5.1	25.0	0.0	40.0	11.7
60	**75.5**	6.7	32.5	0.0	38.8	16.9
80	**84.5**	7.5	43.8	0.0	52.5	21.2
100	**90.0**	8.2	55.0	0.0	60.0	24.3

K Additional Inversion Diagnostics

Numerical convergence. We define a numerical inversion failure as a non-finite output or non-convergent residuals. Across the human correction chunks from all four tasks, inversion with $M = 16$ produced finite outputs and decreasing residuals, with an observed numerical failure rate of 0%.

Correction magnitude and task phase. On 205 human correction chunks from Insert Square and Stack Blocks, reconstruction MSE was weakly correlated with normalized action-correction RMS ($r = 0.19$ and $r = 0.14$, respectively). Higher-error samples occurred more often in the early and late phases of Insert Square and in the early phases of Stack Blocks, suggesting that inversion error also depends on task context.

Unseen configurations. We additionally evaluated eight complete human trajectories comprising 112 correction chunks from configurations absent from the SFT demonstrations: unseen object positions in Stack Blocks and a 45° square-slot misalignment in Insert Square. The normalized action-chunk reconstruction MSEs were 0.002114 and 0.000174, respectively. All 112 chunks produced finite noise targets and decreasing residuals, with zero numerical inversion failures.