VGM: Value-Gated Modulation for Minute-Level Residual Adaptation of VLA Policies

1st Zihan Wang
Tsinghua SIGS
Tsinghua University
Shenzhen, China
wangzihan25@mails.tsinghua.edu.cn

2nd Yizhe Li
Tsinghua SIGS
Tsinghua University
Shenzhen, China
li-yz23@mails.tsinghua.edu.cn

3rd Zhe Han
Queen Mary School Hainan
Beijing University of Posts and Telecommunications
Lingshui, China
2023213782@bupt.cn

4th Guoping Pan
Z-Lab
Zerith
Shenzhen, China
panguoping@zerith.com

5th Yi Cheng
Z-Lab
Zerith
Shenzhen, China
chengyi@zerith.com

6th Houde Liu*
Tsinghua SIGS
Tsinghua University
Shenzhen, China
liu.hd@sz.tsinghua.edu.cn

***Abstract*—Vision-Language-Action (VLA) policies demonstrate remarkable semantic understanding, yet their real-world deployment is often hindered by execution failures in last-mile tasks. Residual learning offers a non-intrusive adaptation scheme to enhance performance without altering the base architecture; however, its effectiveness is often limited by a lack of coordination between the base model and the corrective branch. Addressing common challenges such as Intent Misalignment and Stagnation Traps in VLA adaptation, we propose the Value-Gated Modulator (VGM)—a framework designed for efficient, minute-level adaptation of generalist VLA policies.**

The VGM leverages a multi-modal representation—integrating depth information alongside RGB and proprioception—to provide the necessary geometric grounding for precise interaction. Instead of unconstrained optimization, VGM functions as an adaptive arbitration layer that dynamically modulates the residual contribution by balancing strictly greedy optimization, preemptive regression detection, and forced recovery. Real-world experiments on a Franka robot arm demonstrate that VGM, initialized from only 20 demonstrations, enables a frozen VLA to converge to a 100% success rate within under 5 minutes of real-world interaction—a $3.5\times$ speedup over static residual baselines. Our approach provides a practical and data-efficient pathway for deploying general-purpose robot policies in high-precision scenarios.

***Index Terms*—VLA, residual adaptation, multi-modal perception, human-in-the-loop, rapid robot adaptation**

I. Introduction

Vision-Language-Action (VLA) models are becoming a dominant paradigm for general-purpose robot control, as they integrate visual perception, language understanding, and action generation into a unified policy [1], [3], [6], [7]. Their appeal lies in the ability to combine large-scale robot data with powerful pretrained vision-language backbones, enabling broad semantic grounding, instruction following, and transfer across tasks and embodiments [1], [3], [7]. Recent systems such as OpenVLA [3], π_0 [14] and $\pi_{0.5}$ [4], together with emerging post-training frameworks such as SOP [5] and training infrastructures like RLinf [31], have pushed VLAs toward more capable and adaptable real-world robot policies.

A recurring issue in real-world VLA deployment is that a policy may capture the correct high-level intent yet still fail in the final stage of execution. Recent robustness studies show that VLA models remain highly sensitive to physical variations, camera viewpoints, and robot initial states, even when they appear competent on standard benchmarks [11], [12]. Such failures suggest that the bottleneck often lies not in semantic understanding, but in the inability to apply precise, task-specific corrections under the embodiment and control conditions of the target robot. This gap has motivated residual adaptation as a corrective approach, yet naively adding a residual branch introduces its own coordination failures that prior work has not resolved.

As detailed in Section II, residual adaptation has emerged as a promising lightweight alternative to full fine-tuning, yet coordinating the corrective branch with the base policy's semantic intent remains an open challenge that existing methods do not explicitly address.

However, achieving minute-level 100% success with residual VLA is hindered by two critical bottlenecks. First, *Intent Misalignment*: unconstrained residual exploration often clashes with the VLA's high-level semantic intent, forcing the optimizer to collapse the residual branch to zero to maintain base performance. Second, *Stagnation Traps*: in sparse-reward settings, this failure unfolds in two stages. In the incipient stage, harmful perturbations during critical phases such as pre-contact approach actively degrade the value function before the system has stalled, yet no corrective signal is triggered because the instantaneous advantage remains ambiguous. If unchecked, this regression compounds into confirmed stagnation, where chronic under-progress prevents the collection of successful trajectories needed for convergence.

To address these, we introduce the Value-Gated Modulator

This work was supported by the Shenzhen Science and Technology Program (Grant No. RCJC20210706091946001, ZDCY20250901104207008)

(VGM). The residual branch leverages a multi-modal representation—integrating depth information alongside RGB and proprioception—to provide the geometric grounding that pure vision-language backbones often lack. VGM acts as an adaptive arbitration layer that balances strictly greedy optimization, preemptive regression detection, and forced recovery across three temporally complementary value signals, ensuring that the residual policy only asserts influence when it provides a clear gain, when value degradation is detected early, or when confirmed stagnation requires forced escape.

Our contributions are as follows:

- We introduce a multimodal residual branch integrating depth, RGB, and proprioception, providing the geometric grounding necessary for precise manipulation that RGB-only VLA backbones lack.
- We propose a three-gate value modulation scheme with distinct temporal scopes—instantaneous advantage, short-horizon regression, and long-horizon lag—offering a structured principle for coordinating residual intervention across different failure regimes.
- We demonstrate through real-world experiments on a Franka robot that VGM enables a frozen VLA to achieve perfect task reliability within minutes of interaction from minimal demonstration data.

II. Related Work

Residual RL for robot control. Residual RL augments a fixed base policy with a learned corrective branch, avoiding the cost of full retraining [22]. ResiP [8] demonstrates this for precise assembly from imitation priors, ResFiT [9] applies off-policy residual finetuning to behavior cloning policies, and PLD [10] combines residual RL with self-improving data generation for VLA policies. These works confirm that a lightweight corrective branch can close the gap between a general prior and task-specific execution, but none explicitly addresses the coordination problem between the base policy's semantic intent and the residual's exploratory gradient.

Efficient VLA adaptation. Adapting large VLA models without full fine-tuning has received growing attention [20], [21]. Our work is complementary: we freeze the backbone entirely and modulate a residual branch online, enabling minute-level adaptation from sparse rewards with as few as 20 demonstrations.

Value-based intervention and human-in-the-loop learning. Interactive imitation learning [23]–[25] uses human feedback to guide policy improvement. The escape gate in VGM is directly inspired by the lag-based progress monitoring in [26], which detects stagnation via value function tracking. We extend this by composing it with a greedy gate and a regression gate into a unified modulation framework.

III. Methodology

A. Problem Formulation and System Overview

We define the robotic task as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. The state $s_t \in \mathcal{S}$ is a multimodal tuple $(I_t^{\text{rgb}}, I_t^{\text{depth}}, s_t^{\text{proprio}}, L)$, and the action space $\mathcal{A}$ corresponds to the end-effector pose commands. The reward $r_t \in \{0, 1\}$ is a sparse signal provided via human keyboard input, where 1 is assigned only upon successful task completion. The system utilizes a demonstration dataset $\mathcal{D}_{\text{demo}}$, consisting of 20 expert trajectories, and an online dataset $\mathcal{D}_{\text{online}}$ generated during deployment. The final control command follows the law:

$$a_t = a_t^{\text{VLA}} + \lambda_t \cdot a_t^{\text{res}}, \tag{1}$$

where a_t^{VLA} is the current action from the VLA's action chunk and λ_t is the scalar scale computed by VGM.

B. Base Policy

Our system builds on a $\pi_{0.5}$-SFT VLA policy [4], which is fine-tuned on a small demonstration dataset $\mathcal{D}_{\text{demo}}$ of only 20 episodes to obtain a position-control policy. At inference time, the policy takes RGB images and language instructions as input and predicts an action chunk $\mathbf{a}_{t:t+k}^{\text{VLA}}$. This chunked prediction improves temporal smoothness and mitigates compounding errors in high-frequency control. During adaptation, the VLA backbone is kept frozen and provides the base action a_t^{VLA} for the residual branch.

C. Multimodal Residual Policy

The residual policy π_{res} is engineered to bridge the precision gap between the VLA and the real world.

To resolve the spatial misalignment inherent in RGB-only policies, we employ a modified `ResNet-18` backbone with a six-channel input layer. This layer integrates RGB images, relative depth maps from `Depth-Anything-v2` [15], and normalized spatial coordinate grids (`CoordConv`). This early-fusion strategy allows the encoder to learn position-dependent feature biases directly from the raw pixels. We adopt spatial learned embeddings [30] in place of standard global average pooling, enabling the encoder to retain spatially localized geometric features critical for precise manipulation.

The residual head, implemented as a lightweight MLP, receives a concatenated vector of visual features and the intention action of VLA $a_t^{\text{intent}} = a_t^{\text{VLA}} - s_t^{\text{pos}}$. To provide a robust behavioral prior, the entire residual policy is warmed up using the Cal-QL algorithm [17] on $\mathcal{D}_{\text{demo}}$. This offline pre-training ensures that π_{res} is initially aligned with the VLA's intent, preventing destructive exploration during the onset of real-world deployment.

D. Value-Gated Modulator (VGM)

The VGM computes the scalar modulation coefficient λ_t in Eq. 1 as a weighted sum of three temporally complementary value signals:

$$\lambda_t = \text{clip}\Big(\alpha \cdot g_t^{\text{greedy}} + \beta \cdot g_t^{\text{regress}} + \gamma \cdot g_t^{\text{escape}},\ 0,\ \lambda_{\max}\Big) \tag{2}$$

All three gates share a unified critic backbone motivated by [36], jointly producing $Q_\phi(s_t, \cdot)$ and $V_\phi(s_t)$ from a shared visual trunk trained online via Cal-QL [17].

Greedy Gate g_t^{greedy} addresses Intent Misalignment via strictly conservative arbitration. Defining $\hat{A}_t = Q_\phi(s_t, a_t^{\text{VLA}} +$

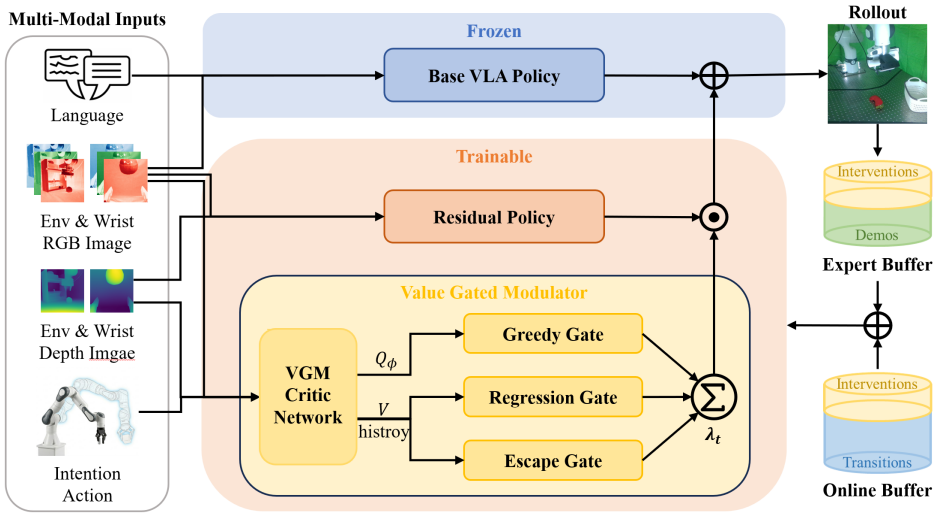


Fig. 1. VGM Architecture Overview.

$a_t^{\text{res}}) - Q_\phi(s_t, a_t^{\text{VLA}})$, a naive sigmoid is insufficient: when $\hat{A}_t = 0$, $\sigma(0) = 0.5$ yields a non-zero scale even when the residual provides no benefit; when $\hat{A}_t < 0$, $\sigma(\hat{A}_t) > 0$ still permits harmful residual injection. We therefore apply a shifted and rectified transformation:

$$g_t^{\text{greedy}} = 2\,\text{ReLU}\left[\sigma\left(\frac{\hat{A}_t}{\tau}\right) - \frac{1}{2}\right] \tag{3}$$

The shift re-centers the activation threshold at $\hat{A}_t = 0$, and the ReLU hard-zeros all non-positive cases, implementing a strict safety lock: the residual branch is completely suppressed unless the critic certifies a positive gain. The factor of 2 restores the output range to $[0, 1]$, and τ controls activation steepness to avoid discontinuous switching.

Regression Gate g_t^{regress} targets the incipient stage of Stagnation Traps, providing preemptive intervention when V_ϕ is actively decreasing over a short look-back window Δ—a regime not captured by the instantaneous advantage signal of g_t^{greedy}:

$$g_t^{\text{regress}} = \text{ReLU}\left(\frac{V_{t-\Delta} - V_t}{\tau_r \cdot \Delta}\right) \tag{4}$$

This gate activates proportionally to the degradation rate, enabling early intervention before harmful perturbations render the state unrecoverable.

Escape Gate g_t^{escape} resolves confirmed Stagnation Traps by monitoring cumulative progress deficit over a long horizon W. Following [26]:

$$\mathcal{L}_t = \rho \cdot W - (V_t - V_{t-W}), \qquad g_t^{\text{escape}} = \text{ReLU}(\mathcal{L}_t) \tag{5}$$

where ρ is the expected per-step value gain. Critically, this gate operates independently of g_t^{greedy}: even when the greedy gate is fully suppressed, a sufficiently large $\mathcal{L}_t$ forces a non-zero λ_t, compelling the residual branch to attempt recovery actions that break the deadlock.

The three gates cover complementary temporal scopes: g^{greedy} arbitrates at the current frame, g^{regress} detects degradation over $\Delta \in [3, 5]$ steps, and g^{escape} accumulates evidence over $W = 20$ steps.

IV. Experiments

In this section, we conduct real-world experiments on a Franka FR3 robot arm to evaluate the proposed framework. Our evaluation is structured around two core questions:

1) How should depth information be acquired to satisfy the requirements of both real-time control and geometric precision?
2) What is the impact of different module configurations on the overall model performance?

A. Hardware and Tasks

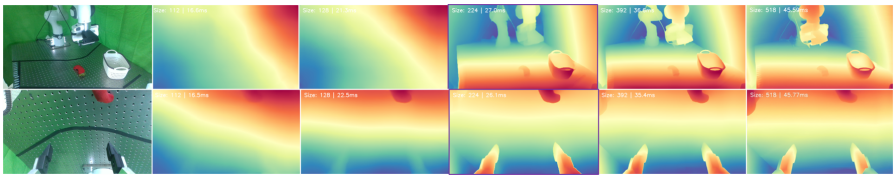
We utilize a Franka FR3 robot arm equipped with a RealSense D435i camera for visual observation and a SpaceMouse for human-in-the-loop intervention. The evaluation task involves picking up a red chili pepper from the tabletop and placing it into a designated basket. Due to the elongated and asymmetric geometry of the chili pepper, the task necessitates high-precision grasp orientations and precise contact-phase coordination.

We compare our proposed framework against the following:

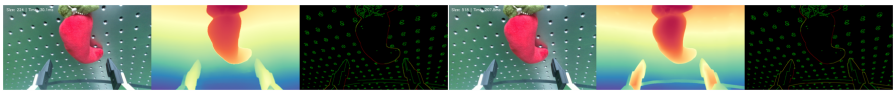
- **SFT-VLA (Baseline)**: The base $\pi_{0.5}$ model fine-tuned on demonstration dataset $\mathcal{D}_{\text{demo}}$ (20 trajectories).
- **Static-Residual**: The VLA policy augmented with a residual branch but utilizing a fixed scaling factor λ.
- **VGM-Residual (Ours)**: The proposed framework incorporating the VGM for dynamic, on-demand residual intervention.

B. Perception Analysis: Depth Acquisition for Real-time RL

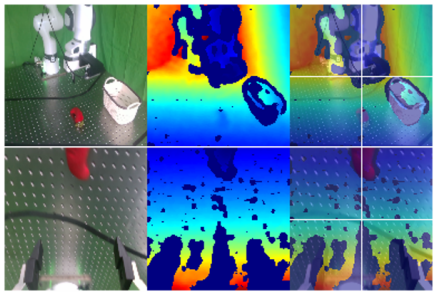
To ensure real-time geometric grounding, we evaluate `Depth-Anything-V2-Small` as a replacement for native



(a) Multi-scale depth estimation. From left to right: 112, 128, 224, 392, 518 resolution.



(b) Geometric alignment check (224 vs. 518 resolution).



(c) Raw depth by D435i camera.

Fig. 2. Comparative analysis of perceptual streams. The left column illustrates the consistency of model-based depth across scales and viewpoints. The right column highlights the structural failures of native active-stereo depth in high-precision interaction zones. In fig. 2b, the red line indicates raw sensor depth edges, the green line indicates predicted depth edges, and the yellow area indicates perfect alignment where the red and green lines overlap.

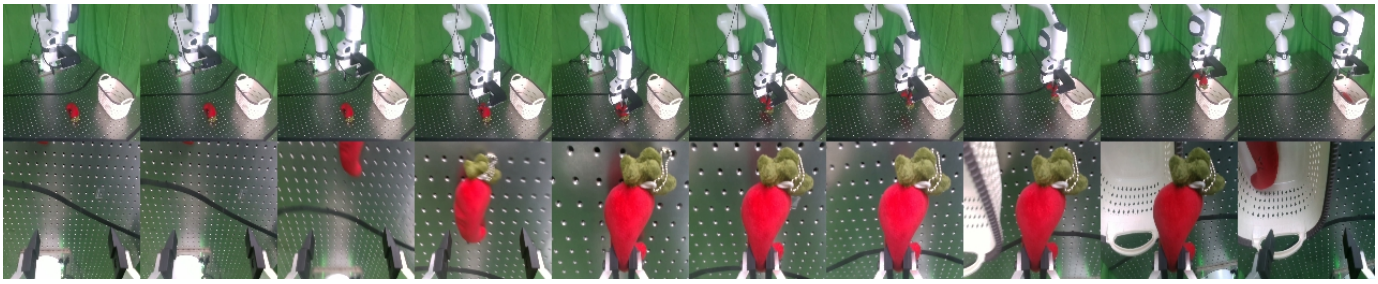


Fig. 3. Real-World Evaluation of VGM.

active-stereo depth. As shown in fig. 2, model-based estimation effectively recovers structural features (e.g., gripper fingers and table) lost in raw RealSense data due to specular reflections and near-field blind spots. We select 224×224 as the operating resolution to balance structural fidelity and computational overhead. This configuration achieves a stable 26.1ms inference latency—significantly outperforming the 45.6ms required for the 518-resolution reference—while avoiding the geometric degradation observed at lower resolutions. This ensures a suitable control frequency around 30Hz, which is essential for robust real-world RL deployment.

C. Performance Evaluation under Different Module Settings

To address the second research question, we evaluate the success rate and adaptation efficiency across different baseline configurations. The results are summarized in Table I.

TABLE I
PERFORMANCE EVALUATION ON THE CHILI PICK-AND-PLACE TASK.

Method	Training Time (mins)	Success Rate
SFT-VLA (Baseline)	N/A	0% (0/20)
Static-Residual (λ fixed)	16.67	90% (18/20)
VGM-Residual (Ours)	**4.76**	**100% (20/20)**

Quantitative Results. As detailed in Table I, the SFT-VLA baseline fails to complete the task (0% success), primarily due to its inability to manage the precise contact requirements of the chili's irregular geometry. While adding a Static-Residual branch improves the success rate to 90%, it requires a lengthy interaction period of 16.67 minutes to converge. In stark contrast, our framework achieves a perfect success rate (100%) in only 4.76 minutes. This represents a $3.5\times$ reduction in training time compared to the static baseline, demonstrating the superior data efficiency of dynamic trust modulation in real-world adaptation.

Effect of Dynamic Scale Modulation. The significant gap in both efficiency and success rate underscores the necessity of the dynamic scale λ_t. During experiments, the Static-Residual often falls into *Stagnation Traps* where the robot ceases movement near the target. By monitoring execution lag and advantage, the VGM dynamically amplifies λ_t to provide the necessary corrective force. This mechanism not only accelerates convergence by filtering out unhelpful residual noise in early stages but also ensures 100% reliability by forcing recovery in critical failure modes.

V. Discussion and Future Work

We presented the Value-Gated Modulator (VGM), a framework for rapid residual adaptation of VLA policies. By integrating multimodal depth perception and a three-gate modulation scheme, VGM addresses Intent Misalignment and Stagnation Traps in real-world deployment. Experiments on a Franka robot demonstrate that VGM enables a frozen VLA to achieve 100% success on high-precision tasks within 5 minutes—a $3.5\times$ speedup over static baselines. Future work will explore dimension-wise modulation and adaptive gate parameter tuning.

REFERENCES

[1] A. Brohan *et al.*, "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *Proc. 7th Conf. Robot Learning (CoRL)*, vol. 229, pp. 2165–2183, 2023.

[2] Octo Model Team *et al.*, "Octo: An open-source generalist robot policy," in *Robotics: Science and Systems (RSS)*, 2024.

[3] M. J. Kim *et al.*, "OpenVLA: An open-source vision-language-action model," in *Proc. 8th Conf. Robot Learning (CoRL)*, vol. 270, pp. 2679–2713, 2025.

[4] K. Black *et al.*, "$\pi_{0.5}$: A vision-language-action model with open-world generalization," in *Proc. 9th Conf. Robot Learning (CoRL)*, vol. 305, 2025.

[5] M. Pan *et al.*, "SOP: A scalable online post-training system for vision-language-action models," unpublished, arXiv:2601.03044, 2026.

[6] K. Kawaharazuka, J. Oh, J. Yamada, I. Posner, and Y. Zhu, "Vision-language-action models for robotics: A review towards real-world applications," unpublished, arXiv:2510.07077, 2025.

[7] X. Li *et al.*, "What matters in building vision-language-action models for generalist robots," *Nature Machine Intelligence*, vol. 8, pp. 158–172, 2026.

[8] L. Ankile, A. Simeonov, I. Shenfeld, M. Torne, and P. Agrawal, "From imitation to refinement: Residual RL for precise assembly," unpublished, arXiv:2407.16677, 2024.

[9] L. Ankile, Z. Jiang, R. Duan, G. Shi, P. Abbeel, and A. Nagabandi, "Residual off-policy RL for finetuning behavior cloning policies," unpublished, arXiv:2509.19301, 2025.

[10] W. Xiao *et al.*, "Self-improving vision-language-action models with data generation via residual RL," unpublished, arXiv:2511.00091, 2025.

[11] H. Liu *et al.*, "Eva-VLA: Evaluating vision-language-action models' robustness under physical variations," unpublished, arXiv:2509.18953, 2025.

[12] S. Fei *et al.*, "LIBERO-Plus: In-depth robustness analysis of vision-language-action models," unpublished, arXiv:2510.13626, 2025.

[13] Y. Zhang, Y. Qi, and X. Zheng, "Experiences from benchmarking vision-language-action models for robotic manipulation," unpublished, arXiv:2511.11298, 2025.

[14] K. Black *et al.*, "π_0: A vision-language-action flow model for general robot control," unpublished, arXiv:2410.24164, 2024.

[15] L. Yang *et al.*, "Depth anything V2," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 37, pp. 21875–21911, 2024.

[16] S. Liu *et al.*, "RDT-1B: A diffusion foundation model for bimanual manipulation," unpublished, arXiv:2410.07864, 2024.

[17] M. Nakamoto *et al.*, "Cal-QL: Calibrated offline RL pre-training for efficient online fine-tuning," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023.

[18] T. Motoda, R. Hanai, R. Nakajo, M. Murooka, F. Erich, and Y. Domae, "Learning bimanual manipulation via action chunking and inter-arm coordination with transformers," unpublished, arXiv:2503.13916, 2025.

[19] Y. Zhao *et al.*, "AnyPlace: Learning generalized object placement for robot manipulation," unpublished, arXiv:2502.04531, 2025.

[20] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," unpublished, arXiv:2502.19645, 2025.

[21] Y. Yadav, Z. Zhou, A. Wagenmaker, K. Pertsch, and S. Levine, "Robust finetuning of vision-language-action robot policies via parameter merging," unpublished, arXiv:2512.08333, 2025.

[22] T. Johannink *et al.*, "Residual reinforcement learning for robot control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 6023–6029, 2019.

[23] A. Mandlekar, D. Xu, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese, "Human-in-the-loop imitation learning using remote teleoperation," unpublished, arXiv:2012.06733, 2020.

[24] C. Celemin *et al.*, "Interactive imitation learning in robotics: A survey," *Foundations and Trends in Robotics*, vol. 10, no. 1–2, pp. 1–197, 2022.

[25] J. Luo, P. Dong, Y. Zhai, Y. Ma, and S. Levine, "RLIF: Interactive imitation learning as reinforcement learning," in *Int. Conf. Learn. Representations (ICLR)*, 2024.

[26] M. Du, A. Khazatsky, T. Gerstenberg, and C. Finn, "To err is robotic: Rapid value-based trial-and-error during deployment," unpublished, arXiv:2406.15917, 2024.

[27] H. Liu, S. Dass, R. Martín-Martín, and Y. Zhu, "Model-based runtime monitoring with interactive imitation learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024.

[28] B. Zhang, Y. Zhang, J. Ji, Y. Lei, J. Dai, Y. Chen, and Y. Yang, "SafeVLA: Towards safety alignment of vision-language-action model via constrained learning," unpublished, arXiv:2503.03480, 2025.

[29] H. Wang, G. Zhang, Y. Yan, Y. Shang, R. R. Kompella, and G. Liu, "Real-time robot execution with masked action chunking," unpublished, arXiv:2601.20130, 2026.

[30] J. Luo *et al.*, "SERL: A software suite for sample-efficient robotic reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 16961–16969, 2024.

[31] C. Yu *et al.*, "RLinf: Flexible and efficient large-scale reinforcement learning via macro-to-micro flow transformation," unpublished, arXiv:2509.15965, 2025.

[32] Y. Weng, X. Zhang, Y. Mu, Y. Zhu, Y. Li, and Q. Liu, "Temporal action selection for action chunking," unpublished, arXiv:2511.04421, 2025.

[33] Y. Liu, J. I. Hamid, A. Xie, Y. Lee, M. Du, and C. Finn, "Bidirectional decoding: Improving action chunking via closed-loop resampling," in *Int. Conf. Learn. Representations (ICLR)*, 2025.

[34] Z. Xue, B. An, and S. Yan, "Policy optimization under imperfect human interactions with agent-gated shared autonomy," in *Int. Conf. Learn. Representations (ICLR)*, 2025.

[35] X. Xu, Y. Hou, Z. Liu, and S. Song, "Compliant residual DAgger: Improving real-world contact-rich manipulation with human corrections," unpublished, arXiv:2506.16685, 2025.

[36] S. Zhai *et al.*, "A vision-language-action-critic model for robotic real-world reinforcement learning," unpublished, arXiv:2509.15937, 2025.