Robo-ValueRL: Reliable Value Estimation for Offline-to-Online Reinforcement Learning

Wenke Xia[1,*], **Pei Ren**[2,*], **Wenbo Yu**[3], **Yizhuo Zhang**[1], **Jifan Li**[1], **Yixue Zhang**[4], **Yinuo Zhao**[2], **Qingyang Gao**[2], **Jianlong Fu**[5], **Jian Tang**[2], **Ji-Rong Wen**[1], **Zhengping Che**[2,✉], **Di Hu**[1,✉]

[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Beijing Innovation Center of Humanoid Robotics
[3]Beijing Forestry University, [4]Peking University, [5]Microsoft Research

[*]Equal contribution, ✉ Corresponding author

Abstract

Offline-to-online reinforcement learning is promising for generalizable robotic manipulation, yet its full-stack complexity obscures reproduction and diagnosis. Within such systems, value estimation plays a central role in prioritizing heterogeneous data for policy improvement. Despite its importance, the central question remains underexplored: *how value-function reliability shapes policy optimization in offline-to-online reinforcement learning*. To answer this question, we propose **Robo-ValueRL**, a unified framework that enables reliable value estimation and systematically traces its downstream effects on policy pretraining and online improvement. Concretely, Robo-ValueRL learns a history-conditioned value estimator and evaluates its reliability through global-progress and local-preference metrics. These resulting value estimates are propagated into quality-conditioned consistency-policy pretraining and a residual adaptation module on online rollouts, providing a unified testbed for analyzing how value reliability shapes downstream policy performance. Across 240 hours of offline demonstrations and over 3,000 online rollout trajectories, our extensive experiments show that downstream performance is strongly associated with value reliability. Reliable value functions provide better action-quality estimates, allowing **value-guided offline RL to scale more effectively than quality-agnostic behavior cloning**, and **stabilize online improvement by prioritizing high-quality rollout data**. Integrating reliable value guidance through offline pretraining with online improvement, our system achieves **86% success on millimeter-level precise chip insertion and 84% on generalizable block disassembly**. We hope these findings highlight the importance of value-guided data utilization for effective policy improvement from heterogeneous robotic experience.

Email: Wenke Xia at xiawenke2022@ruc.edu.cn, Di Hu at dihu@ruc.edu.cn,
Pei Ren at pei.ren@x-humanoid.com, Zhengping Che at z.che@x-humanoid.com
Project Page: https://gewu-lab.github.io/Robo-ValueRL/
Code: https://github.com/Open-X-Humanoid/Robo-ValueRL
Data&Model: https://huggingface.co/collections/X-Humanoid/robo-valuerl

1 Introduction

Offline-to-online reinforcement learning has emerged as a promising paradigm for generalizable robotic manipulation [1–3], bridging offline policy pretraining [4–10] with online improvement through real-world interaction [11–13]. However, unlike conventional benchmarks, it constitutes a full-stack embodied learning system shaped by coupled components, including *data curation*, *value estimation*, *offline pretraining*, and *online exploration* [14–16]. These interdependencies make performance difficult to reproduce and failures hard to diagnose under deployment conditions.

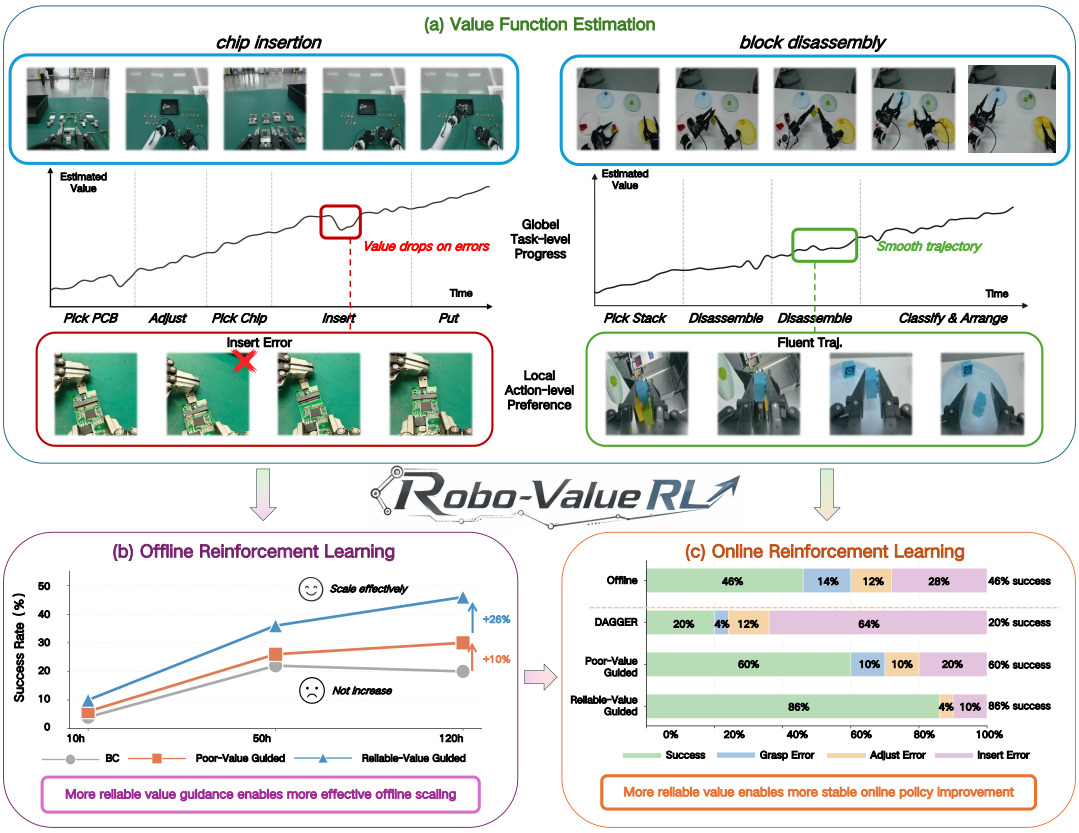


Figure 1 Robo-ValueRL improves offline-to-online RL via reliable value guidance. (a) Value functions are assessed with reliability metrics for downstream policy learning. (b) Reliable values enable scalable offline pretraining from mixed-quality demonstrations. (c) Value-guided online adaptation stabilizes policy improvement, reaching 86% millimeter-level chip-insertion success.

Among these factors, value estimation is central to offline-to-online robotic reinforcement learning [17, 18], as it evaluates heterogeneous experience, from suboptimal demonstrations to online rollouts, and determines how such data contribute to policy improvement [19–21]. Reliable values can prioritize informative experience and suppress low-quality data, whereas inaccurate values may misrank experience and destabilize learning. However, existing evidence on value-estimation reliability remains fragmented across the offline-to-online pipeline [3, 19, 22, 23]: algorithmic studies often analyze value optimization in isolation [2, 3], while robotic systems mainly report downstream performance without examining whether learned values capture task progress or action quality [19, 22, 24]. Thus, *the role of value-estimation reliability in offline pretraining and online improvement has yet to be systematically investigated.*

To investigate this role, we examine value-estimation reliability from two coupled perspectives: how reliable values are learned from heterogeneous robotic data, and how their reliability affects policy learning across offline pretraining and online improvement. To ground this investigation, we introduce Robo-ValueRL, a unified offline-to-online robotic reinforcement learning framework that organizes *reliable value function learning*, *value-guided offline policy pretraining*, and *real-world online improvement* into a coherent process. Beyond final success rates, Robo-ValueRL introduces value-reliability metrics that provide an analytical basis for relating value quality to downstream policy optimization.

Concretely, Robo-ValueRL first trains a history-conditioned value estimator on offline robotic data to reduce ambiguity from partial observations and visual occlusions. Its action-quality estimates are used to score offline experience for quality-conditioned consistency-policy pretraining [25], enabling efficient value-guided action generation through one-step denoising. The pretrained policy is then deployed to collect real-world rollouts with human-in-the-loop intervention. Robo-ValueRL further updates the value estimator on these rollouts and selects high-quality online segments to train a lightweight residual adaptation module, allowing targeted correction of failure modes while preserving prior knowledge. Across this offline-to-online process, our metrics assess value reliability from both global task-level progress to measure

whether values reflect long-horizon task advancement, and local action-level preference to estimate whether values capture fine-grained action quality. Together, Robo-ValueRL provides a unified testbed for analyzing how value reliability affects offline scaling and online improvement.

Leveraging Robo-ValueRL, we conduct an extensive real-world study on two challenging manipulation tasks: millimeter-level chip insertion and generalizable block disassembly, using 240 hours of offline robotic data and over 3,000 online rollout trajectories. As summarized in Figure 1, our results reveal a consistent relationship between downstream policy performance and value-estimation reliability: value functions that capture both *global task progress* and *local action preference* enable more effective utilization of heterogeneous robotic experience. During offline pretraining, **value-guided policy learning scales more effectively than quality-agnostic behavior cloning** under mixed-quality demonstrations, improving offline success rates by 26% on chip insertion and 34% on block disassembly. During online improvement, **reliable value estimates lead to more stable real-world adaptation** by prioritizing high-quality rollout segments and suppressing suboptimal interactions. Overall, Robo-ValueRL achieves **86% success on chip insertion and 84% on block disassembly**, demonstrating that reliable value-guided data utilization is critical for converting heterogeneous offline and online experience into effective robotic policy improvement.

2 Related Work

2.1 Learning from Heterogeneous Robotic Data

A central challenge in scaling robotic learning is the heterogeneity of real robot data [26, 27]: human demonstrations contain mistakes, hesitation, and locally suboptimal actions, while online rollouts introduce exploratory failures and unstable interactions [16, 24]. Current large-scale VLA pipelines typically absorb such mixed-quality data through behavior cloning, which provides broad behavioral priors but lacks explicit action-quality discrimination [8, 28, 29]. Consequently, quality-agnostic imitation may fit both effective and suboptimal segments. Offline RL and recent offline-to-online methods address this limitation by using value functions to exploit higher-quality behaviors from offline data and online rollouts [3, 19, 22, 30, 31]. However, existing work mainly evaluates heterogeneous data usage through final policy performance, leaving unclear how value estimation is reliable and how such reliability affects downstream learning. We address this gap by treating value estimation as an interface between heterogeneous robotic data and policy learning, enabling mixed-quality trajectories to be incorporated into policy optimization.

2.2 Offline-to-Online Reinforcement Learning

Offline-to-online reinforcement learning combines the sample efficiency of offline RL with the adaptability of online interaction [1, 32]. A key challenge is distribution shift: once deployed, an offline-trained policy may visit poorly covered states and actions, causing critic miscalibration and unstable online updates [33, 34]. Existing methods mitigate this issue by constraining online adaptation around the offline policy and balancing offline-online data usage to preserve useful priors while adapting safely [3, 18]. However, these studies are often evaluated on small-scale benchmarks. Recent VLA-based robotic systems [19, 22, 24] extend offline-to-online learning to large generalist policies, but mainly report final performance, leaving the stability and design dependencies of value estimation underexplored. In this work, we build the Robo-ValueRL framework to systematically study how value estimation affects offline policy pretraining and online improvement, and whether value-function reliability can be diagnosed before downstream policy learning.

2.3 Value Estimation in Robotic Manipulation

Value estimation is central to robotic manipulation, where critics provide long-horizon signals for policy improvement. From temporal-difference learning and linear approximation to deep RL for visual control [21, 35–40], value functions have traditionally been learned as policy-coupled critics. Recent work instead trains general value functions to evaluate task progress and behavior quality across goals, tasks, and datasets [41–44], with goal-conditioned values, successor representations, and world-value models further improving transferable value prediction [23, 45–47]. However, evaluation has not kept pace: existing metrics such as Value-Order Correlation in GVL and smoothness in VIP mainly assess optimal trajectories [23, 44], while concurrent work studies diverse suboptimal value estimation but validates it primarily through filtered behavior cloning [47]. We instead train a history-conditioned value estimator and introduce global task-level progress and local action-level preference metrics to assess its role across offline-to-online learning.

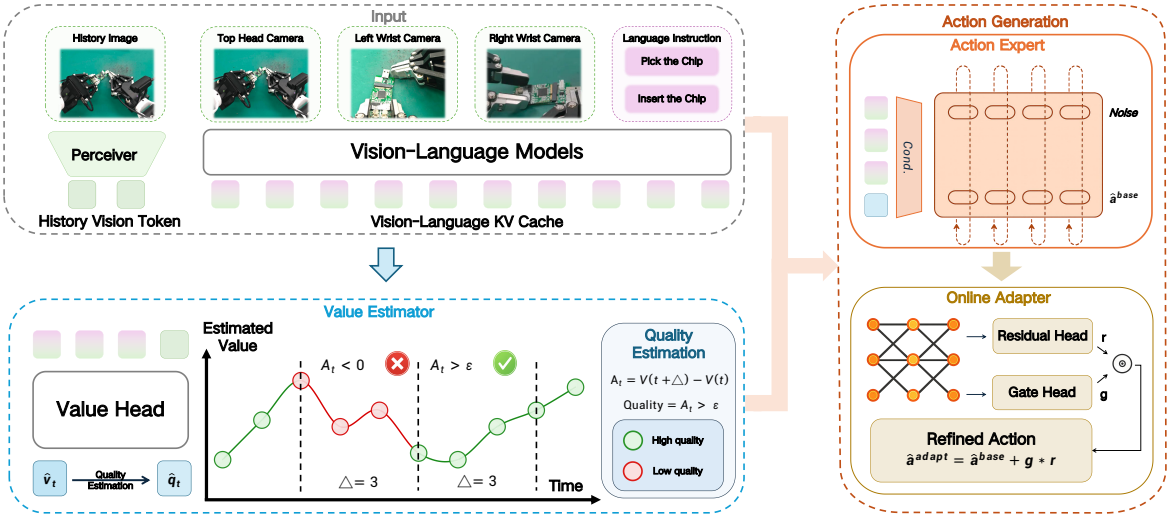


Figure 2 Our Robo-ValueRL offline-to-online reinforcement learning framework. The value estimator incorporates historical context to produce reliable value estimates, from which action-quality labels are derived to train a quality-conditioned consistency policy. During online interactions, the online adaptation module is further utilized to enable stable online improvement.

3 Robo-ValueRL

We propose Robo-ValueRL, a unified offline-to-online robotic reinforcement learning framework centered on reliable value estimation. Robo-ValueRL first trains a history-conditioned value estimator to infer normalized task progress from visual observations and history, and uses value differences to derive action-quality indicators for quality-conditioned VLA consistency-policy pretraining. For online improvement, it freezes the pretrained policy and learns a lightweight residual adapter from real-world rollouts, enabling targeted adaptation while preserving the offline prior. We further introduce value-reliability metrics that assess global task-stage ordering and local action-level preference, linking value quality to both offline scaling and online policy improvement.

3.1 History-Conditioned Value Estimator

Single-frame value estimation is inherently ambiguous in long-horizon robotic manipulation, where occlusions and repetitive visual patterns can make different task stages appear visually similar. To address this, we introduce a history-conditioned value estimator that augments the current visual observation with a compact representation of past observations for more reliable value prediction, with implementation details provided in Appendix C.1.

The value estimator uses a PaliGemma vision-language backbone [48] with a lightweight Transformer value head f_θ^v. At timestep t, the current three-view observation o_t and language instruction ℓ are processed by PaliGemma to produce a vision-language conditioning context $\mathbf{c}_t^v$, which serves as the VLM-generated prefix representation for subsequent value estimation. Further, the visual history h_t is encoded via a SigLIP encoder [49] and summarized into compact tokens $\tilde{\mathbf{H}}_t$ using a lightweight Perceiver module [50]. To predict the task value, we append a learnable value query $\mathbf{q}_v$ to $\tilde{\mathbf{H}}_t$ and employ a transformer-based value head conditioned on $\mathbf{c}_t^v$. The network outputs a categorical distribution over K discretized value bins $\{b_k\}_{k=1}^K$:

$$p_\theta(k \mid o_t, h_t, \ell) = \mathrm{softmax}_k \left(f_\theta^v \left(\mathbf{c}_t^v, \tilde{\mathbf{H}}_t, \mathbf{q}_v \right) \right). \tag{1}$$

Training Recipe. We supervise estimator using a normalized progress value $v_t^* \in [0, 1]$. We adopt an HL-Gaussian distributional representation [22, 51], the target v_t^* is projected onto the bins as a soft distribution $q_k(v_t^*) \propto \exp\left(-\frac{(v_t^* - b_k)^2}{2\sigma^2}\right)$. The value head is optimized via cross-entropy loss:

$$\mathcal{L}_v = -\sum_{k=1}^{K} q_k(v_t^*) \log p_\theta(k \mid o_t, h_t, \ell). \tag{2}$$

During inference, the final scalar value estimate is decoded as the expectation over the bins: $\hat{v}_t = V_\theta(o_t, h_t, \ell) = \sum_{k=1}^{K} b_k \cdot p_\theta(k \mid o_t, h_t, \ell)$.

Action Quality Indicator. The value estimator provides dense action-quality indicators to guide both offline reinforcement learning and online fine-tuning. Intuitively, an action is deemed beneficial if it increases the estimated task progress. Formally, for a transition from t to $t+\Delta$, we compute the value improvement as $\Delta v_t = V_\theta(o_{t+\Delta}, h_{t+\Delta}, \ell) - V_\theta(o_t, h_t, \ell)$. We then assign a discrete action quality indicator $q_t^{\text{act}} \in \{0, 1, 2\}$ (representing low-quality, neutral, and high-quality actions, respectively) by thresholding Δv_t with pre-defined positive and negative margins δ_+ and δ_-.

3.2 Quality-conditioned Vision-Language-Action Model

Given the action-quality indicators estimated from offline experience, we train a quality-conditioned VLA model π_θ with a consistency policy head f_θ^a on heterogeneous demonstrations. The policy learns to associate action patterns with estimated quality, separating high-quality behaviors from suboptimal segments. During inference, the desired quality prompt guides the policy toward high-quality action generation, while the consistency head enables one-step denoising for efficient real-time control. Additional implementation details are provided in Appendix C.2.

Concretely, our quality-conditioned VLA model π_θ uses PaliGemma as the vision-language backbone and a consistency policy head f_θ^a as action expert. At timestep t, to inject value-based quality information into the language-conditioned policy, we first convert the action-quality indicator q_t^{act} into a textual action-quality prompt ℓ_t^{act} and concatenate it with the task instruction to form $\ell_t^{\text{qc}} = [\ell; \ell_t^{\text{act}}]$. Further, the VLM model takes the multi-view observation o_t and quality-conditioned language input ℓ_t^{qc} as input to generate vision-language context $\mathbf{c}_t^a$ for action generation.

To enable efficient action generation, we instantiate action expert f_θ^a as a consistency policy [25]. Conditioned on the VLM-generated action context $\mathbf{c}_t^a$ and the robot state s_t, the policy maps a noisy action chunk $\mathbf{x}_{\sigma_i}$ at noise level σ directly to the corresponding clean action chunk: $\hat{\mathbf{x}}_0 = f_\theta^a(\mathbf{x}_{\sigma_i}, \sigma_i, s_t, \mathbf{c}_t^a)$. During inference, we sample $\mathbf{x}_{\sigma_{\max}} \sim \mathcal{N}(0, \sigma_{\max}^2 I)$ and obtain the final action through a single denoising step, enabling low-latency real-time robot execution.

Training Recipe. The quality-conditioned VLA model π_θ is optimized with a joint objective that combines action reconstruction and adjacent-noise self-consistency. Given a ground-truth action chunk $\mathbf{x}_0$, we sample a noise level σ_i from a Karras noise schedule to construct a noisy action chunk: $\mathbf{x}_{\sigma_i} = \mathbf{x}_0 + \sigma_i \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. The model is first supervised by an action reconstruction loss:

$$\mathcal{L}_{\text{act}} = \|\pi_\theta(\mathbf{x}_{\sigma_i}, \sigma_i, s_t, o_t, \ell_t^{\text{qc}}) - \mathbf{x}_0\|_2^2. \tag{3}$$

We further impose self-consistency between adjacent noise levels σ_i and σ_{i+1} to align clean-action predictions along the denoising trajectory. The prediction at σ_i is propagated by a one-step solver to $\tilde{\mathbf{x}}_{\sigma_{i+1}}$, yielding the consistency loss:

$$\mathcal{L}_{\text{cons}} = w(\sigma_i)\|\pi_\theta(\mathbf{x}_{\sigma_i}, \sigma_i, s_t, o_t, \ell_t^{\text{qc}}) - \text{sg}[\pi_{\bar{\theta}}(\tilde{\mathbf{x}}_{\sigma_{i+1}}, \sigma_{i+1}, s_t, o_t, \ell_t^{\text{qc}})]\|_2^2, \tag{4}$$

where $w(\sigma_i)$ is a noise-dependent weighting term, $\text{sg}[\cdot]$ denotes the stop-gradient operation, and $\pi_{\bar{\theta}}$ is updated as an exponential moving average of the online model π_θ. This consistency regularization encourages the full VLA model to produce stable clean-action predictions across adjacent noise levels, enabling the learned policy to approximate multi-step denoising with a single denoising step during inference. The final joint training objective is balanced by a hyperparameter λ_{cons}:

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \lambda_{\text{cons}}\mathcal{L}_{\text{cons}}. \tag{5}$$

3.3 Online Residual Policy Adaptation

Online rollouts provide valuable interaction data for policy improvement, but directly fine-tuning the pretrained VLA policy on heterogeneous rollouts can destabilize optimization and induce catastrophic forgetting of the learned behavior prior. We therefore freeze the offline-trained base policy π_θ^{base} and introduce a lightweight residual adapter h_ϕ, which predicts bounded corrections to the base action to enable targeted adaptation from online experience while preserving the pretrained prior. Additional architectural and training details are provided in Appendix C.3.

Given the action chunk $\hat{\mathbf{x}}_t^{\text{base}}$ from the frozen base policy π_θ^{base}, the residual adapter h_ϕ predicts a bounded correction to refine it. Specifically, the adapter compresses the VLA action context $\mathbf{c}_t^a$ into compact latent context tokens using a

Perceiver module [50], concatenates them with the robot state s_t and base action prediction $\hat{\mathbf{x}}_t^{\text{base}}$, and feeds the result into a lightweight Transformer to predict an unconstrained residual $\tilde{\mathbf{r}}_t$ and a step-wise gate logit $\tilde{\mathbf{g}}_t$:

$$(\tilde{\mathbf{r}}_t, \tilde{\mathbf{g}}_t) = h_\phi \left(\mathbf{c}_t^a, s_t, \hat{\mathbf{x}}_t^{\text{base}} \right), \tag{6}$$

where we bound the residual as $\mathbf{r}_t = r_{\max} \tanh(\tilde{\mathbf{r}}_t)$ and constrain the gate with a sigmoid function as $\mathbf{g}_t = \sigma(\tilde{\mathbf{g}}_t)$, The final adapted action chunk is computed as:

$$\hat{\mathbf{x}}_t^{\text{adapt}} = \hat{\mathbf{x}}_t^{\text{base}} + \mathbf{g_t} \odot \mathbf{r_t}. \tag{7}$$

Training Recipe. We train only the residual adapter h_ϕ with a mixture of offline demonstrations $\mathcal{D}_{\text{off}}$ and online rollouts $\mathcal{D}_{\text{on}}$, while keeping base VLA policy π_θ^{base} frozen. We first use the learned value estimator to split online rollouts into high-quality samples $\mathcal{D}_{\text{on}}^+$ and low-quality samples $\mathcal{D}_{\text{on}}^-$ according to the estimated action-quality indicator q_t^{act}. For high-quality online samples from $\mathcal{D}_{\text{on}}^+$, the correction loss $\mathcal{L}_{\text{corr}} = \mathbb{E}_{\mathcal{D}_{\text{on}}^+} \left[\left|\left| \hat{\mathbf{x}}_t^{\text{adapt}} - \mathbf{x}_t \right|\right|_2^2 \right]$ encourages the adapted action to move toward successful online behaviors. For offline samples from $\mathcal{D}_{\text{off}}$, the behavior-preservation loss $\mathcal{L}_{\text{keep}} = \mathbb{E}_{\mathcal{D}_{\text{off}}} \left[\left|\left| \hat{\mathbf{x}}_t^{\text{adapt}} - \hat{\mathbf{x}}_t^{base} \right|\right|_2^2 \right]$ keeps the adapted action close to the frozen base action $\hat{\mathbf{x}}_t^{\text{base}}$, preventing the residual adapter from drifting away from the pretrained behavior prior. We further introduce a gate loss $\mathcal{L}_{\text{gate}} = \mathbb{E}_{\mathcal{D}_{\text{on}}^+ \cup \mathcal{D}_{\text{off}}} [\text{CE}(\mathbf{g}_t, y_t^g)]$ to regulate when the residual correction should be activated. The final adapter objective is:

$$\mathcal{L}_{\text{adapter}} = \mathcal{L}_{\text{corr}} + \lambda_{\text{keep}} \mathcal{L}_{\text{keep}} + \lambda_{\text{gate}} \mathcal{L}_{\text{gate}}. \tag{8}$$

3.4 Value-reliability Metrics

To systematically diagnose value-function reliability, we establish an evaluation protocol that comprehensively evaluates learned value functions across global task-level progress and local action-level preference. Given a predicted value sequence $\{\hat{v}_t\}$, this suite verifies the model's capacity to resolve task advancement, identify high-quality actions, and detect execution errors.

Global Task-level Progress. This diagnostic measures whether the value function preserves the coarse ordering of task progress. Given subgoal boundary timesteps $\{b_0, \ldots, b_K\}$, we extract the midpoint of each interval as $m_j = \hat{v}_{\lfloor (b_j + b_{j+1})/2 \rfloor}$. Since later intervals correspond to more advanced task stages, a reliable value function should assign higher values to later subgoals, i.e., $m_{j+1} > m_j$. We define the *Midpoint Ordering Rate* (MOR) as the fraction of adjacent subgoal transitions that satisfy this ordering: $\text{MOR} = \frac{1}{K-1} \sum_{j=0}^{K-2} \mathbb{1}[m_{j+1} > m_j]$.

Local Action-level Preference. This diagnostic evaluates whether the value function provides stable and accurate action-level estimates along local trajectories. A reliable value function should remain smooth during successful executions while sharply penalizing anomalous deviations from steady task progress. We characterize this behavior using two complementary metrics:

1) Fluency: Along successful execution paths, the predicted value sequence should increase smoothly, whereas abrupt downward jumps indicate noisy or unstable value estimates. To quantify this local fluency, we evaluate both the frequency and the severity of these temporal oscillations [23]. Specifically, we define the *Bump Ratio* as $\frac{1}{T-1} \sum_t \mathbb{1}[\hat{v}_{t+1} < \hat{v}_t - \epsilon_v]$, which measures how often the expected upward trend is violated beyond a small noise-tolerance threshold ϵ_v. Concurrently, we compute the *Bump Magnitude* as $\frac{1}{T-1} \sum_t \max(0, \hat{v}_t - \hat{v}_{t+1})$ to capture the average magnitude of these downward drops, to quantify the overall stability of the value signals.

2) Temporal Error Discrimination: While fluency measures stability along successful executions, temporal error discrimination evaluates whether the value function can reliably penalize deviations from normal task progress, thereby preventing erroneous behaviors from being assigned overly optimistic action-quality estimates. For each annotated error segment $\mathcal{E}$, we construct a normal-progress reference $\hat{v}_i^{\text{ref}}$ from the pre-error value trajectory, representing the value that would be expected under uninterrupted task progress. We compare this reference with the smoothed prediction $\tilde{v}_i$ and define the value deficit as $\delta_i = \hat{v}_i^{\text{ref}} - \tilde{v}_i$, where $\delta_i > 0$ means that the estimator assigns a lower value than expected under normal progress. We report *Error Sensitivity*, $S = \max_{i \in \mathcal{E}} \delta_i$, to measure the peak value deficit, and *Error Slope*, $\alpha = \text{slope}(\{\delta_i\}_{i \in \mathcal{E}})$, to measure whether this deficit is sustained over the erroneous execution.

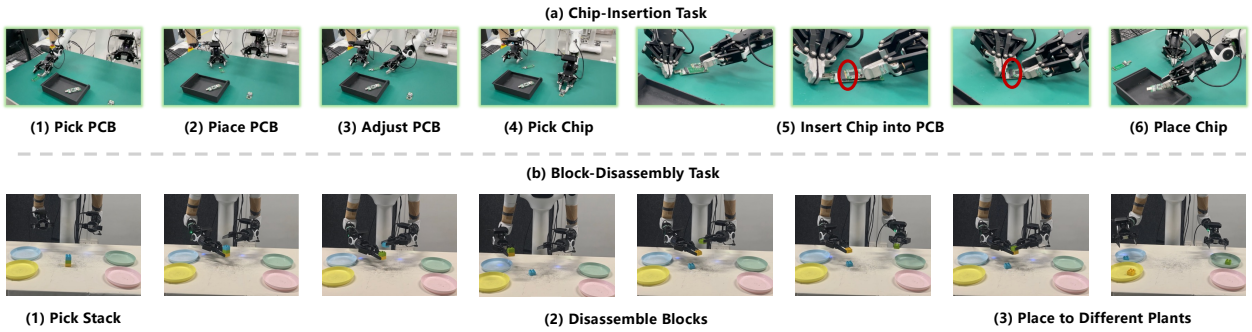


Figure 3 Overview of the two real-world manipulation tasks. (a) **Chip insertion**: a long-horizon, fine-grained task where the robot grasps a PCB, adjusts it to a feasible pose, and inserts a chip into a millimeter-scale slot within the recessed groove beneath the PCB base plate, as highlighted by the red circle. (b) **Block disassembly**: the robot disassembles randomly placed blocks and sorts them onto color-matched plates with randomized layouts, requiring generalizable bimanual coordination.

4 Experiments

In this work, we investigate *how reliable values can be learned from heterogeneous robotic data* and *how value reliability affects policy learning across offline pretraining and online improvement*. To this end, we build Robo-ValueRL, a unified framework for reliable value estimation that systematically traces its downstream effects on policy pretraining and online improvement. We evaluate Robo-ValueRL on two challenging real-world manipulation tasks with 240 hours of heterogeneous offline demonstrations and over 3,000 online rollout trajectories, as shown in Figure 3, covering both precision-critical fine-grained manipulation and generalizable bimanual coordination. Building on this experimental setting, we organize our investigation around three progressively connected questions:

(1) How can value reliability be diagnosed? In Section 4.1, we evaluate value-reliability metrics and examine their alignment with downstream policy success.

(2) How does value reliability affect offline pretraining? In Section 4.2, we compare value-guided pretraining across data scales and quality thresholds, showing how reliable values improve learning from heterogeneous demonstrations.

(3) How does value reliability affect online improvement? In Section 4.3, we study iterative online adaptation with value-guided rollout filtering, showing that reliable values stabilize policy improvement from real-world interaction.

Finally, in Section 4.4, we provide qualitative evidence that complements the quantitative results, showing that robots learn efficient execution and autonomous self-correction patterns through our Robo-ValueRL offline-to-online reinforcement learning framework.

4.1 Value Reliability and Its Downstream Effects

The lack of direct quantitative measures for value-function reliability makes it difficult to diagnose how value-estimation quality affects downstream policy optimization. We introduce value-reliability metrics to evaluate value functions and examine their alignment with downstream policy learning, and provide more visualization results in Appendix E.

Reliability Metrics for Value Estimation. We ablate the temporal context used by the value estimator: NO HISTORY uses only the current observation, while SHORT HISTORY and LONG HISTORY use 5-frame and 30-frame visual histories, respectively. The metrics columns in Table 1 show that SHORT HISTORY provides the strongest overall value reliability. These results indicate that moderate temporal context helps preserve global progress ordering, suppress local value fluctuations, and produce clearer responses to erroneous actions. Figure 4 visualizes the behaviors captured by these metrics. SHORT HISTORY produces smoother predictions during normal execution and a sharper drop after errors, corresponding to its stronger local fluency and error-response metrics. In contrast, other variants exhibit more oscillatory traces or weaker responses to erroneous actions, which are also reflected by their lower metrics scores. These traces support that our metrics quantitatively capture key properties of value reliability, including normal-trajectory stability and error sensitivity.

Table 1 Ablation study on value estimation diagnostics and downstream policy learning. SHORT HISTORY and LONG HISTORY denote 5-frame and 30-frame past visual histories. MOR is the Midpoint Ordering Rate for global task-level progress estimation.

Variant	Value Estimation Metrics					Downstream Success Rate	
	MOR. ↑	Bump Ratio ↓	Bump Mag. ↓	Sensitivity ↑	Slope ↑	Chip ↑	Block ↑
NO HISTORY	95.2	0.073	0.74	35.2	0.54	28.0%	44.0%
SHORT HISTORY	**95.6**	**0.067**	0.70	**41.1**	**0.62**	**46.0%**	**60.0%**
LONG HISTORY	95.1	0.069	**0.66**	35.8	0.58	30.0%	42.0%

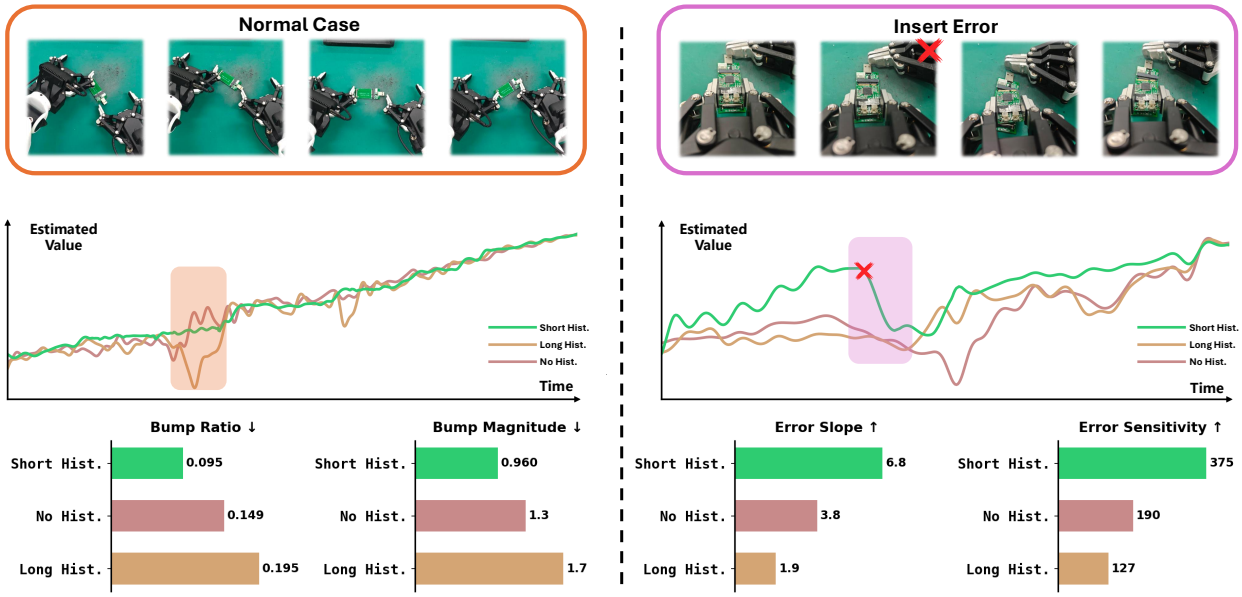


Figure 4 Estimated value curves with metrics. The metrics reflect stability in smooth trajectories and capture drops at error onset, demonstrating their effectiveness for measuring value reliability.

Diagnostic Validity for Policy Learning. We further examine whether the proposed metrics predict downstream policy learning. As shown in Table 1, the reliability ranking aligns with downstream success: SHORT HISTORY provides the strongest overall diagnostic performance, and also achieves the highest success rates on both precise chip insertion and generalizable block disassembly tasks. This alignment indicates that the metrics capture value-estimation properties that are relevant to policy optimization. This suggests that value functions capturing progress consistency, local smoothness, and error discrimination better guide policy optimization, making our metrics a lightweight criterion for value-estimator selection before expensive downstream training.

4.2 Value-Guided Offline Policy Pretraining

Offline policy pretraining on heterogeneous robotic data hinges on selecting useful experience from demonstrations, corrections, suboptimal actions, and failures. We study how different value estimators and action-quality selection strategies shape policy learning from such mixed-quality data.

Scaling with Heterogeneous Offline Data. We evaluate whether value-guided pretraining can better exploit heterogeneous offline data by training chip-insertion and block-disassembly policies on mixed-quality subsets of 10, 50, and 120 hours. We compare four variants: Behavior Cloning (BC) uses all data without value-based quality labels, treating every demonstrated action equally regardless of its contribution to task progress; Value-Guided Soft (VG-SOFT) derives action-quality labels from the SHORT HISTORY value estimator using a longer temporal window (Δ=60 frames) and relaxed positive margins to select the top 50% of samples, which retains more diverse behaviors including temporally extended manipulation sequences; Value-Guided Strict (VG-STRICT) uses the same value estimator but with a shorter temporal window (Δ=20 frames) and tighter margins to select the top 30% of samples, emphasizing immediate action-level progress; and Value-Guided No History (VG-NH) applies the same strict labeling rule as VG-STRICT but uses the NO HISTORY value estimator, isolating the effect of value-estimation reliability on downstream policy learning.

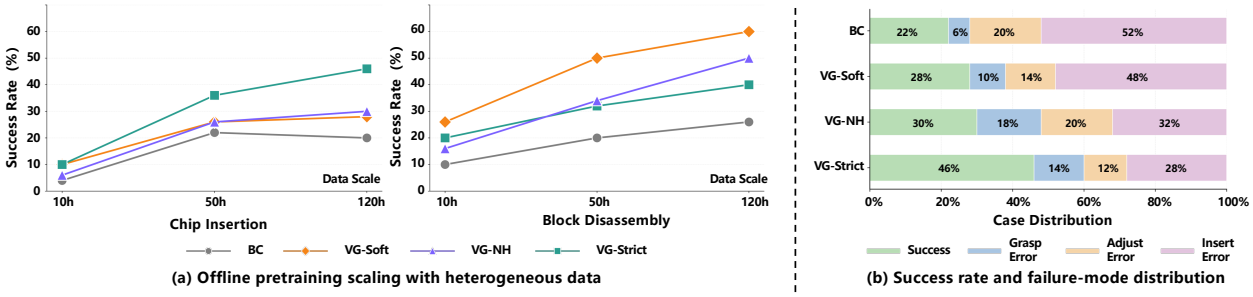


Figure 5 Value-guided offline pretraining performance and behavior. (a) BC saturates as mixed-quality data increases, while value-guided variants scale more effectively. (b) Value-estimation reliability directly shapes downstream policy behavior.

As shown in Figure 5(a), BC yields limited gains beyond 50 hours, since BC treats all samples uniformly; additional data may introduce low-quality supervision from suboptimal or failed behaviors. In contrast, value-guided pretraining emphasizes high-value samples, enabling the policy to learn high-quality action distributions and scale effectively with heterogeneous offline data. These results in Figure 5(b) show that quality threshold is task-dependent. VG-STRICT performs better on chip insertion, where progress depends on short precise motions, whereas VG-SOFT is better suited to block disassembly, where progress spans longer manipulation segments and requires broader behavioral coverage. The comparison between VG-STRICT and VG-NH further connects these results to our value-reliability metrics in Sec. 4.1. Both methods use the same strict labeling rule, but differ in the value estimator used to assign action-quality labels. The stronger performance of VG-STRICT indicates that the more reliable SHORT HISTORY estimator produces more accurate quality labels than the NO HISTORY estimator. This shows that value-estimation reliability directly affects downstream offline reinforcement learning by determining which actions are selected as beneficial training signals.

Failure Case Analysis. We further analyze how value-based quality criteria reshape policy behavior. BC assumes all demonstrated actions are valid, thereby preserving routine behaviors well: because human demonstrations rarely fail during grasping, BC yields the lowest grasp-error rate. However, it also imitates hesitation and trial-and-error corrections during insertion, leading to frequent insertion failures. Value-guided pretraining shifts this behavior distribution by selecting high-quality action chunks. Stricter filtering suppresses noisy insertion-stage corrections, reducing insertion errors and improving success, but may also remove supervision for routine stages such as grasping and coarse adjustment, increasing early-stage failures. The gap between VG-STRICT and VG-NH further shows that this trade-off depends on value reliability: unreliable estimates can discard useful chunks and amplify early failures.

4.3 Value-Guided Online Improvement

Online rollouts help bridge the distribution gap between offline pretraining data and real-world interaction, but their heterogeneous nature makes them difficult to exploit effectively. We therefore study how value guidance can stabilize online policy optimization by identifying high-quality interaction data and suppressing suboptimal updates.

Multi-Iteration Online Improvement. Starting from the VG-STRICT offline-RL-pretrained policy, which achieves a 46% success rate (SR) on the chip insertion task, we conduct three rounds of online rollout collection and policy improvement, using 500 rollout trajectories per iteration. We compare against Dataset Aggregation (DAGGER) [52], a widely used online imitation learning method that aggregates human-corrected trajectories into the training set and retrains the policy; here it directly treats all human corrections as positive supervision without distinguishing action quality. In contrast, Value-Guided Online Strict (ONLINE-STRICT) initializes the value model from VG-STRICT and further fine-tunes it during each online iteration using the same strict value-estimation protocol. The updated value model is then used to score online rollout segments and select high-quality actions for policy improvement.

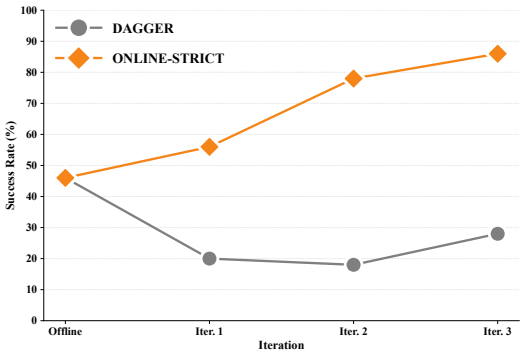


Figure 6 Iterative online improvement on the chip insertion task.

As shown in Figure 6, ONLINE-STRICT improves from 46% to 86% over three online iterations. This suggests that the iteratively fine-tuned value model can identify useful correction signals from heterogeneous rollouts and convert them into stable policy improvement. In contrast, DAGGER drops sharply, and its adapted performance approaches the offline BC baseline. This suggests that aggregating corrective trajectories without quality assessment harms online improvement due to the suboptimal behaviors present in human correction data.

Value-Guided Online Soft (ONLINE-SOFT) and Value-Guided Online No History (ONLINE-NH) follow the same online pipeline as ONLINE-STRICT, but initialize the value model from VG-SOFT and VG-NH, respectively. As shown in Table 2, all value-guided online variants outperform DAGGER, highlighting the benefit of value-based quality filtering for online rollout data. The performance ranking remains task-dependent: ONLINE-STRICT achieves the best success rate on precision-critical chip insertion, while ONLINE-SOFT performs best on block disassembly, which requires broader temporal coverage to preserve diverse manipulation sequences. The gap between ONLINE-STRICT and ONLINE-NH further demonstrates that value-estimation reliability directly impacts online improvement quality.

Table 2 Online policy improvement under different value estimation settings after one iteration of improvement.

Method	Chip	Block
DAGGER	20%	66%
ONLINE-SOFT	32%	**84%**
ONLINE-STRICT	**56%**	70%
ONLINE-NH	40%	70%

4.4 Qualitative Analysis

Beyond the quantitative results presented above, we provide qualitative analysis to demonstrate the effectiveness of our Robo-ValueRL framework. We visualize the gate activation patterns of the online residual adapter and present representative behavioral patterns acquired by the robot, including both optimal behaviors and self-correction capabilities.

Gate Activation of Online Residual Adapter. We visualize the gate activation pattern of the online residual adapter during task execution in Figure 7. For most steps, the offline policy already performs reliably, so the gate remains inactive and the residual adapter does not intervene. At critical error-prone states, such as PCB orientation adjustment and fine-grained chip insertion, the gate is activated to enable targeted refinement. The left panel shows correction of erroneous PCB adjustment, while the right panel shows fine-grained action refinement during precision insertion. This selective activation indicates that the gate learns to identify when the base policy is insufficient, preserving the pretrained behavior prior while enabling precise online correction.

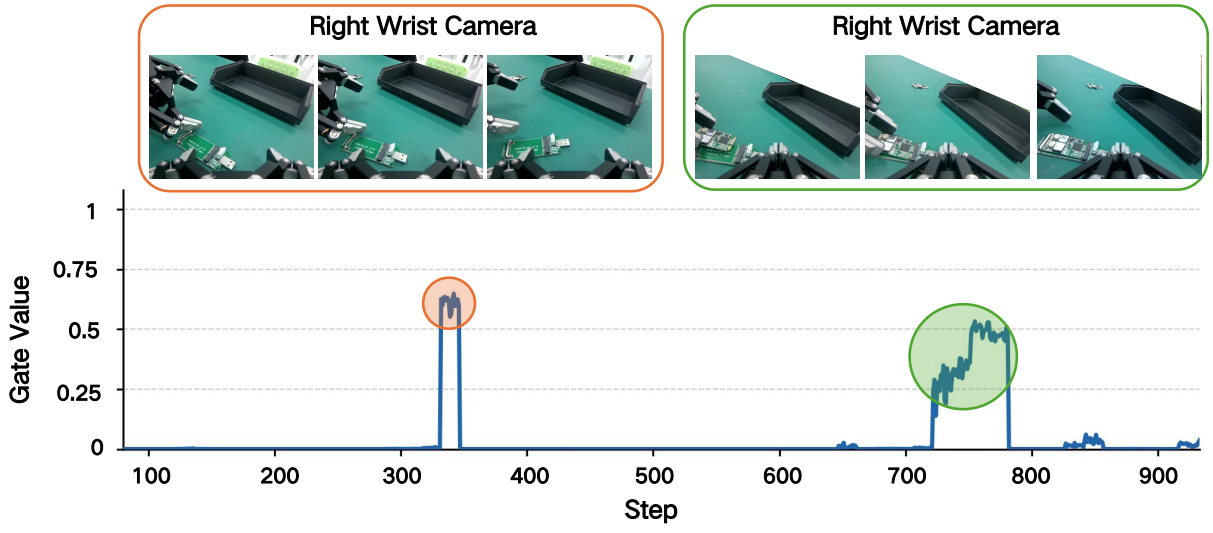
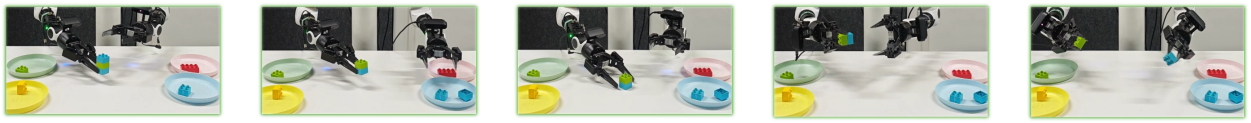
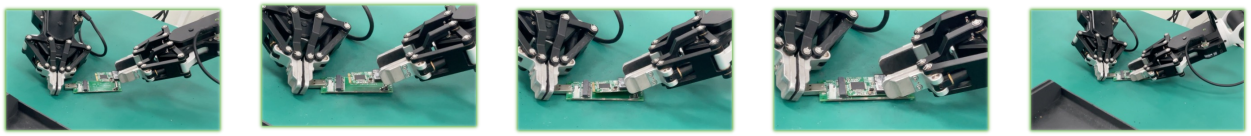


Figure 7 Gate activation pattern of the online residual adapter over time. The online module mostly remains inactive when the offline policy is sufficient. It is selectively triggered at critical error-prone stages, such as PCB pose adjustment (left) and chip insertion (right), to provide targeted action refinement.

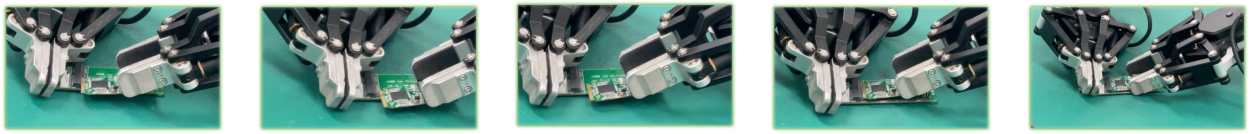


(a) Block Disassembly: Optimal Behavior

(b) Block Disassembly : Self Correction for a set of blocks in OOD scenarios



(c) Chip Insertion: Optimal Behavior



(d) Chip Insertion: Self-Correction from Insert Failure

Figure 8 Representative learned behavioral patterns in block disassembly and chip insertion.

Learned Behavioral Patterns. Figure 8 presents representative behaviors in block disassembly and chip insertion, highlighting two capabilities learned through our offline-to-online reinforcement learning framework: efficient optimal execution and autonomous self-correction. Additional patterns are provided in Appendix D.

Optimal Behavior. As shown in Figure 8(a), the robot learns non-greedy disassembly strategies that anticipate future placement constraints. Although immediately disassembling the next block may appear locally optimal, it would require extra tabletop placement and hand-switching before final placement. Instead, the robot first releases the blue block, then assigns the green and blue blocks to the hands aligned with their target trays, enabling direct placement after disassembly. This shows that value-guided reinforcement learning favors long-horizon efficiency over greedy local progress. In chip insertion (Figure 8(c)), the robot further completes millimeter-level insertion in a single attempt, whereas human teleoperators typically require 2–3 attempts, yielding over 1.5$\times$ higher execution efficiency.

Self-Correction. As shown in Figure 8(b), when a block falls to an unexpected location, the robot autonomously retrieves it and returns it to the correct tray. In chip insertion (Figure 8(d)), when an insertion failure occurs, the robot lifts the gripper and performs a second insertion attempt. These recovery behaviors emerge from online improvement with value-filtered rollout data, where human-corrected trajectories teach the policy to associate failure states with corrective actions, enabling autonomous error recovery during deployment.

5 Conclusion

We presented Robo-ValueRL, an offline-to-online reinforcement learning framework that enables scalable offline pretraining and stable online improvement through reliable value estimation. Robo-ValueRL learns a history-conditioned value estimator, propagates its predictions into quality-conditioned consistency-policy pretraining, and uses online rollouts to train a lightweight residual adapter for targeted adaptation while preserving the pretrained prior. We assess value reliability from global-progress and local-preference perspectives, showing that reliability strongly predicts downstream performance: more reliable values lead to better offline scaling and more stable online improvement. By integrating value guidance across offline pretraining and online adaptation, Robo-ValueRL achieves 86% success on millimeter-level chip insertion and 84% on generalizable block disassembly. We hope this study encourages future robotic learning systems to move beyond simple data accumulation and toward reliable value-guided data utilization.

6 Acknowledgement

This work is supported by Beijing Natural Science Foundation (4262050). This work was also supported by Beijing Innovation Center of Humanoid Robotics (X-Humanoid)

References

[1] A. Nair, A. Gupta, M. Dalal, and S. Levine, "Awac: Accelerating online reinforcement learning with offline datasets," *arXiv preprint arXiv:2006.09359*, 2020.

[2] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," *Advances in neural information processing systems*, vol. 33, pp. 1179–1191, 2020.

[3] M. Nakamoto, S. Zhai, A. Singh, M. Sobol Mark, Y. Ma, C. Finn, A. Kumar, and S. Levine, "Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 62 244–62 269, 2023.

[4] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. S. Ryoo, G. Salazar, P. R. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. H. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "RT-1: Robotics Transformer for Real-World Control at Scale," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[5] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183.

[6] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *2024 IEEE International Conference on Robotics and Automation*. IEEE, 2024, pp. 6892–6903.

[7] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "Openvla: An open-source vision-language-action model," in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, P. Agrawal, O. Kroemer, and W. Burgard, Eds., vol. 270. PMLR, 06–09 Nov 2025, pp. 2679–2713. [Online]. Available: https://proceedings.mlr.press/v270/kim25c.html

[8] K. Black, N. Brown, D. Driess, A. Esmail, M. R. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, L. Smith, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, "π_0: A Vision-Language-Action Flow Model for General Robot Control," in *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025.

[9] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang *et al.*, "Gr00t n1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025.

[10] A. Kumar, A. Singh, F. D. Ebert, M. Nakamoto, Y. Yang, C. Finn, and S. Levine, "Pre-Training for Robots: Offline RL Enables Learning New Tasks in a Handful of Trials," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[11] J. Luo, C. Xu, J. Wu, and S. Levine, "Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning," *Science Robotics*, vol. 10, no. 105, p. eads5033, 2025.

[12] J. Yang, M. S. Mark, B. Vu, A. Sharma, J. Bohg, and C. Finn, "Robot fine-tuning made easy: Pre-training rewards and policies for autonomous real-world reinforcement learning," in *2024 IEEE International Conference on Robotics and Automation*. IEEE, 2024, pp. 4804–4811.

[13] H. Li, Y. Zuo, J. Yu, Y. Zhang, Y. Zhaohui, K. Zhang, X. Zhu, Y. Zhang, T. Chen, G. Cui, D. Wang, D. Luo, Y. Fan, Y. Sun, J. Zeng, J. Pang, S. Zhang, Y. Wang, Y. Mu, B. Zhou, and N. Ding, "SimpleVLA-RL: Scaling VLA training via reinforcement learning," in *The Fourteenth International Conference on Learning Representations*, 2026. [Online]. Available: https://openreview.net/forum?id=TQhSodCM4r

[14] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[15] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare, "Deep reinforcement learning at the edge of the statistical precipice," *Advances in neural information processing systems*, vol. 34, pp. 29 304–29 320, 2021.

[16] Y. Wang, X. Li, P. Xie, P. Yang, B. Nie, Y. Cai, Q. Zhang, C. Qu, J. Wu, J. Song *et al.*, "Learning while deploying: Fleet-scale reinforcement learning for generalist robot policies," *arXiv preprint arXiv:2605.00416*, 2026.

[17] J. Feng, M. Feng, H. Song, W. Zhou, and H. Li, "Suf: Stabilized unconstrained fine-tuning for offline-to-online reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 11, 2024, pp. 11 961–11 969.

[18] S. Lee, Y. Seo, K. Lee, P. Abbeel, and J. Shin, "Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble," in *Conference on Robot Learning*. PMLR, 2022, pp. 1702–1712.

[19] P. Intelligence, A. Amin, R. Aniceto, A. Balakrishna, K. Black, K. Conley, G. Connors, J. Darpinian, K. Dhabalia, J. DiCarlo *et al.*, "pi*0.6: a vla that learns from experience," *arXiv preprint arXiv:2511.14759*, 2025.

[20] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 1678–1690. [Online]. Available: https://proceedings.mlr.press/v164/mandlekar22a.html

[21] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on robot learning*. PMLR, 2018, pp. 651–673.

[22] Y. Li, X. Ma, J. Xu, Y. Cui, Z. Cui, Z. Han, L. Huang, T. Kong, Y. Liu, H. Niu *et al.*, "Gr-rl: Going dexterous and precise for long-horizon robotic manipulation," *arXiv preprint arXiv:2512.01801*, 2025.

[23] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang, "VIP: Towards universal visual reward and representation via value-implicit pre-training," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=YJ7o2wetJ2

[24] Y. Chen, S. Tian, S. Liu, Y. Zhou, H. Li, and D. Zhao, "Conrft: A reinforced fine-tuning method for vla models via consistency policy," in *Proceedings of Robotics: Science and Systems, 2025, Los Angeles, CA, USA, Jun 21-25, 2025*, 2025.

[25] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg, "Consistency policy: Accelerated visuomotor policies via consistency distillation," in *Robotics: Science and Systems*, 2024.

[26] Z. Cao and D. Sadigh, "Learning from imperfect demonstrations from agents with varying dynamics," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5231–5238, 2021.

[27] A. S. Chen, A. M. Lessing, Y. Liu, and C. Finn, "Curating Demonstrations using Online Experience," in *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

[28] D. Ghosh, H. R. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, J. Luo, Y. L. Tan, L. Y. Chen, Q. Vuong, T. Xiao, P. R. Sanketi, D. Sadigh, C. Finn, and S. Levine, "Octo: An Open-Source Generalist Robot Policy," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.

[29] S. Fan, K. Wu, Z. Che, X. Wang, D. Wu, F. Liao, N. Liu, Y. Zhang, Z. Zhao, Z. Xu *et al.*, "Xr-1: Towards versatile vision-language-action models via learning unified vision-motion representations," in *Proceedings of the International Conference on Machine Learning*, 2026.

[30] R. F. Prudencio, M. R. Maximo, and E. L. Colombini, "A survey on offline reinforcement learning: Taxonomy, review, and open problems," *IEEE transactions on neural networks and learning systems*, vol. 35, no. 8, pp. 10 237–10 257, 2023.

[31] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," in *International Conference on Learning Representations*, 2022.

[32] H. Zhang, W. Xu, and H. Yu, "Policy expansion for bridging offline-to-online reinforcement learning," in *The Eleventh International Conference on Learning Representations*, 2023.

[33] Y. Chebotar, Q. Vuong, K. Hausman, F. Xia, Y. Lu, A. Irpan, A. Kumar, T. Yu, A. Herzog, K. Pertsch *et al.*, "Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions," in *Conference on Robot Learning*. PMLR, 2023, pp. 3909–3928.

[34] Y. Luo, J. Kay, E. Grefenstette, and M. P. Deisenroth, "Finetuning from offline reinforcement learning: Challenges, trade-offs and practical solutions," *arXiv preprint arXiv:2303.17396*, 2023.

[35] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[36] J. Tsitsiklis and B. Van Roy, "Analysis of temporal-diffference learning with function approximation," *Advances in neural information processing systems*, vol. 9, 1996.

[37] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 993–1000.

[38] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[39] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International conference on machine learning*. PMLR, 2016, pp. 1329–1338.

[40] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. Pmlr, 2018, pp. 1861–1870.

[41] S. Zhai, Q. Zhang, T. Zhang, F. Huang, H. Zhang, M. Zhou, S. Zhang, L. Liu, S. Lin, and J. Pang, "A vision-language-action-critic model for robotic real-world reinforcement learning," *arXiv preprint arXiv:2509.15937*, 2025.

[42] Y. Wang, Z. Sun, J. Zhang, Z. Xian, E. Biyik, D. Held, and Z. Erickson, "RL-VLM-f: Reinforcement learning from vision language foundation model feedback," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 2024, pp. 51 484–51 501.

[43] D. Yang, D. Tjia, J. Berg, D. Damen, P. Agrawal, and A. Gupta, "Rank2reward: Learning shaped reward functions from passive video," in *2024 IEEE International Conference on Robotics and Automation*. IEEE, 2024, pp. 2806–2813.

[44] Y. J. Ma, J. Hejna, C. Fu, D. Shah, J. Liang, Z. Xu, S. Kirmani, P. Xu, D. Driess, T. Xiao *et al.*, "Vision language models are in-context value learners," in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 33 984–34 009.

[45] T. Schaul, D. Horgan, K. Gregor, and D. Silver, "Universal value function approximators," in *International conference on machine learning*. PMLR, 2015, pp. 1312–1320.

[46] J. Lv, H. Li, J. Li, Y. Nie, F. Kong, Y. Wang, X. Wang, Z. Zhu, C. Ni, Q. Deng *et al.*, "Viva: A video-generative value model for robot reinforcement learning," *arXiv preprint arXiv:2604.08168*, 2026.

[47] Z. Wang, J. Li, Y. Cui, Y. Gao, X. Zhan, J. Yu, and X. Ma, "World value models for robotic manipulation," *arXiv preprint arXiv:2606.24742*, 2026.

[48] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello *et al.*, "Paligemma: A versatile 3b vlm for transfer," *arXiv preprint arXiv:2407.07726*, 2024.

[49] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 11 975–11 986.

[50] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, "Perceiver: General perception with iterative attention," in *International conference on machine learning*. PMLR, 2021, pp. 4651–4664.

[51] J. Farebrother, J. Orbay, Q. Vuong, A. Ali Taiga, Y. Chebotar, T. Xiao, A. Irpan, S. Levine, P. S. Castro, A. Faust, A. Kumar, and R. Agarwal, "Stop regressing: Training value functions via classification for scalable deep RL," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds., vol. 235. PMLR, 21–27 Jul 2024, pp. 13 049–13 071. [Online]. Available: https://proceedings.mlr.press/v235/farebrother24a.html

[52] S. Ross, G. Gordon, and D. 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*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 627–635. [Online]. Available: https://proceedings.mlr.press/v15/ross11a.html

Appendix

A Overview

In this work, we investigate *how value-function reliability affects value-guided learning across offline pretraining and online improvement*. To answer this question, we build a complete offline-to-online reinforcement learning pipeline. In the supplementary materials, we first present the performance of the model trained through the complete pipeline on two representative real-world tasks: a millimeter-level precise chip-insertion task and a generalizable block-disassembly task. In Section B and the supplementary video, we provide **long-duration, one-take real-world videos of over 35 minutes for each task**. The block-disassembly task achieves an **82.9%** success rate **(58/70)**, while the chip-insertion task achieves an **85.7%** success rate **(30/35)**. We also provide detailed implementation details to support reproducibility in Section C. Beyond these quantitative results, in Section D, we further demonstrate several surprising generalization and self-correction capabilities acquired by the robot through our offline-to-online framework. These include recovering from mistakes, completing tasks in out-of-distribution scenarios, and exhibiting more efficient action patterns than skilled human demonstrators. Next, in Section E, we provide visualizations of the value estimator and the corresponding action quality, showing that our framework can effectively estimate task progress in manipulation tasks and further predict action quality to guide policy learning.

B One-take Real-world Experiment Video

We introduce the task setting and provide one-take real-world experiment videos in the supplementary material. As shown in Figure 3, we evaluate our system on two tasks: a millimeter-level chip-insertion task, where the robot must precisely insert a chip into a narrow slot, and a generalizable block-disassembly task, where the robot disassembles randomly placed blocks and sorts them into color-matched plates under randomized layouts.

The supplementary video includes continuous one-take real-world experiments for both tasks. For block disassembly, the video is played at 15$\times$ speed, and the robot achieves an 82.9% success rate (58/70) over a continuous 35-minute experiment. For chip insertion, the video is played at 10$\times$ speed, and the robot achieves an 85.7% success rate (30/35) over a continuous 40-minute experiment. Several frame-drop events occurred during chip insertion due to instability of the Orbbec Gemini 336 camera SDK, causing intermittent pauses and slightly affecting the final success rate.

C Implementation Details

C.1 Value Estimator

We build the value estimator using a PaliGemma VLM backbone [48] and a transformer-based value head. To leverage the knowledge learned from large-scale robotic datasets, we initialize the PaliGemma backbone from the π_0 [8] VLM checkpoint and keep it frozen during training. We mix the OpenX dataset [6], our self-collected data, the chip-insertion and block-disassembly datasets, which are important for preventing overfitting since the supervised value-prediction objective can otherwise be easily overfit. We train the value estimator for two epochs with an initial learning rate of 1×10^{-4}, which is decayed to 5×10^{-6} using a cosine scheduler. During online improvement, we collect rollout data and mix it with the offline datasets at a fixed ratio of 3:1, followed by another two epochs of value-estimator training.

For value estimation, we use a discrete distributional head and formulate value prediction as classification over $K = 256$ fixed bins. We first compute a remaining-time target for each timestep t in trajectory τ. For successful trajectories, the remaining time is the number of steps before task completion. For failed trajectories, we add a failure penalty C_{fail} to assign lower values to all timesteps:

$$r_t^* = \begin{cases} T_\tau - t, & \text{if } \tau \text{ is successful,} \\ T_\tau - t + C_{\text{fail}}, & \text{otherwise,} \end{cases} \tag{9}$$

where T_τ denotes the trajectory length. We then normalize the remaining-time target into a progress-style value target in $[0, 1]$:

$$v_t^* = \text{clip}\left(1 - \frac{r_t^*}{T_{\max}}, 0, 1\right), \tag{10}$$

where $T_{\max} = 5000$. This formulation assigns larger values to states closer to successful completion, while failed trajectories receive lower values due to the additional penalty.

To train the distributional value head, we discretize $[0, 1]$ into K equally spaced bins with centers $c_k = k/(K-1)$ for $k = 0, \ldots, K-1$. Instead of using a hard one-hot label, we convert v_t^* into a soft target distribution:

$$p_k(v_t^*) = \frac{\exp\left(-\frac{(v_t^* - c_k)^2}{2\sigma^2}\right)}{\sum_{j=0}^{K-1} \exp\left(-\frac{(v_t^* - c_j)^2}{2\sigma^2}\right)}, \quad \sigma = \frac{1}{K-1}. \tag{11}$$

The value estimator is trained by minimizing the cross-entropy between the predicted distribution and this soft target distribution.

C.2 Quality-conditioned Consistency Policy

To convert the action-quality indicator q_t^{act} into a textual action-quality prompt ℓ_t^{act}, we use a rule-based mapping:

$$\ell_t^{\text{act}} = \begin{cases} \texttt{Quality: Low}, & q_t^{\text{act}} = 0, \\ \texttt{Quality: Medium}, & q_t^{\text{act}} = 1 \\ \texttt{Quality: High}, & q_t^{\text{act}} = 2. \end{cases} \tag{12}$$

During training, we further apply quality-condition dropout: with a probability of 10%, we replace ℓ_t^{act} with `Medium`, regardless of the original quality label.

For the action expert, we instantiate f_θ^a as a consistency policy, where the noise level σ is sampled from $[2, 80]$ during training. To improve training stability, we adopt a noise-level curriculum that starts with low-noise samples and gradually expands to higher noise levels. The full VLA model π_θ is trained for 50,000 steps on 32 A100 GPUs, with the learning rate decayed from 2×10^{-4} to 5×10^{-6} with a cosine scheduler.

C.3 Online Adapter

During online improvement, we freeze the offline policy and train only an online adapter. The adapter is zero-initialized so that, at the beginning of online training, it behaves as a near-identity correction and does not significantly deviate from the offline policy. In each update round, we mix online and offline data with a 3:1 ratio and limit training to 900 gradient steps, preventing the gate from overfitting to the limited online rollouts.

During online interaction, we find that the robot often fails by getting stuck. For example, in chip insertion, the chip may get stuck on the PCB base plate and fail to lift, while in block disassembly, the gripper may push against the inner block and fail to grasp it. These failure modes make the representations of human-corrected states highly similar to those of prolonged stuck states. Therefore, we use high-quality samples from the online interaction dataset $\mathcal{D}_{\text{on}}$ to train the gate to open, while using only offline data to train the gate to remain closed. This allows the model to acquire corrective behaviors for previous low-quality states through representation similarity, while avoiding the ill-posed supervision problem of learning corrections from low-quality samples without ground-truth corrected actions.

D Learned Pattern from Our Methods

We present the representative behavioral patterns in Figure 9 and Figure 10, we also provide the corresponding videos to demonstrate our robots' behaviors.

For the block-disassembly task, we present five representative behavioral patterns exhibited by our agent in Figure 9. As shown in (a), the robot learns optimal disassembly sequences, completing the task efficiently without redundant actions. After the robot finishes disassembling the previous block, its left hand is holding the blue block. If it proceeds with disassembly at this point, the left hand will still hold the blue block while the right hand will hold the green block. However, since the blue tray is on the right and the green tray is on the left, the blocks cannot be placed directly into their corresponding plates. **Instead, the robot chooses to put down the block first.** It then uses its left hand to grasp

the upper green block and its right hand to disassemble the upper blue block, so that after disassembly, each block can be directly placed into the tray with the matching color. By using task progress as guidance for RL learning, the robot learns optimal actions that maximize task-completion efficiency. In (b), when the target plate is suddenly moved to the opposite side during reaching, the agent immediately detects the change, retracts its extended hand, and switches to the other hand to complete the placement. In (c), The robot grasps the middle of the three-block assembly and first disassembles the bottom red block by pulling it horizontally. However, the same action pattern can no longer continue the disassembly. Therefore, the robot switches to a different manipulation mode, reorients the blocks upright, and then disassembles another block. This avoids unnecessary hand swapping. In (d), when a single block falls to a distant out-of-distribution location during disassembly, the agent recognizes the error and retrieves the block back to the correct plate. In (e), when a whole set of blocks is placed into the plate without being disassembled first, the agent detects the mistake, picks them up again, and performs the correct disassembly sequence. These patterns collectively demonstrate the agent's robustness, adaptability, and strong generalization ability

Figure 9 We demonstrate several robot self-correction and optimal-action segments in the block-disassembly task. These include autonomously selecting the most appropriate disassembly strategy to maximize efficiency, rapidly responding to scene changes, and self-correcting under out-of-distribution conditions.

For the chip-insertion task, we demonstrate diverse robot behaviors in Figure 10. As shown in (a), our robot can perform chip insertion highly efficiently. Unlike human teleoperators, who typically require 2–3 attempts, our robot can complete precise millimeter-level insertion in a single attempt, achieving more than 1.5× the efficiency of human teleoperators. (b) and (c) show that when grasping failures occur, the robot can autonomously correct its behavior and re-grasp the chip. (d) shows that when an error occurs, the robot can lift the gripper and perform a second insertion attempt.

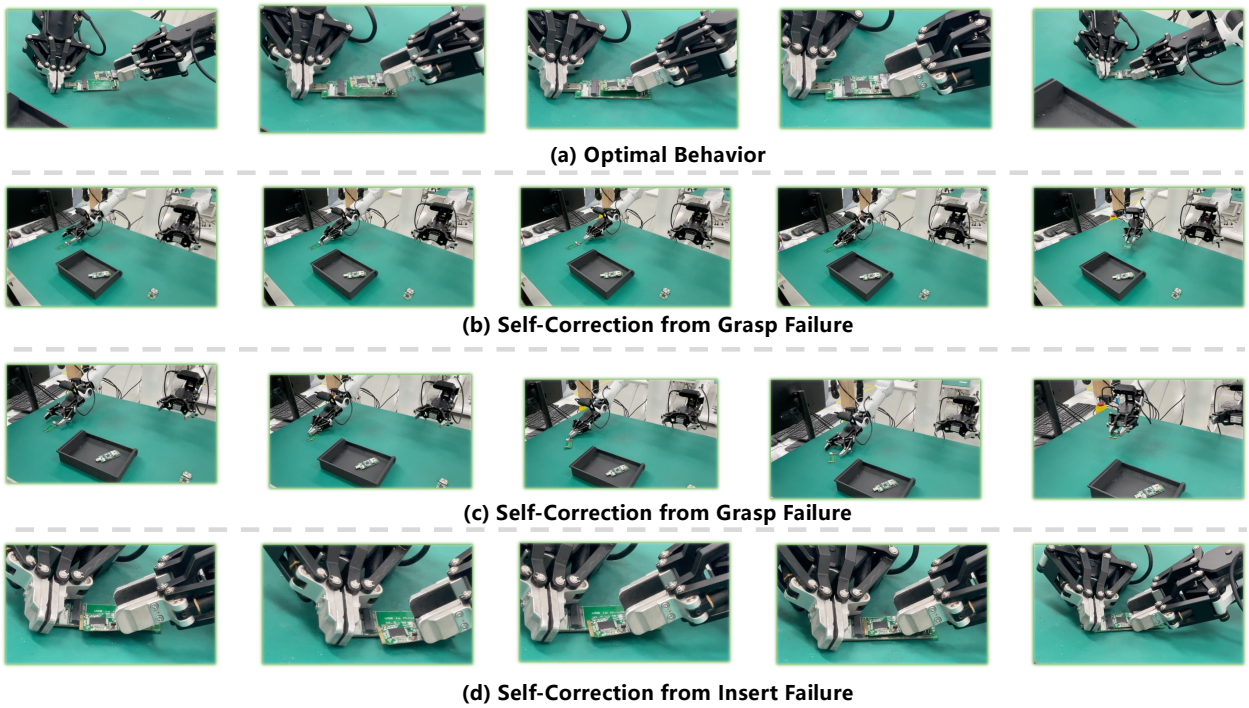
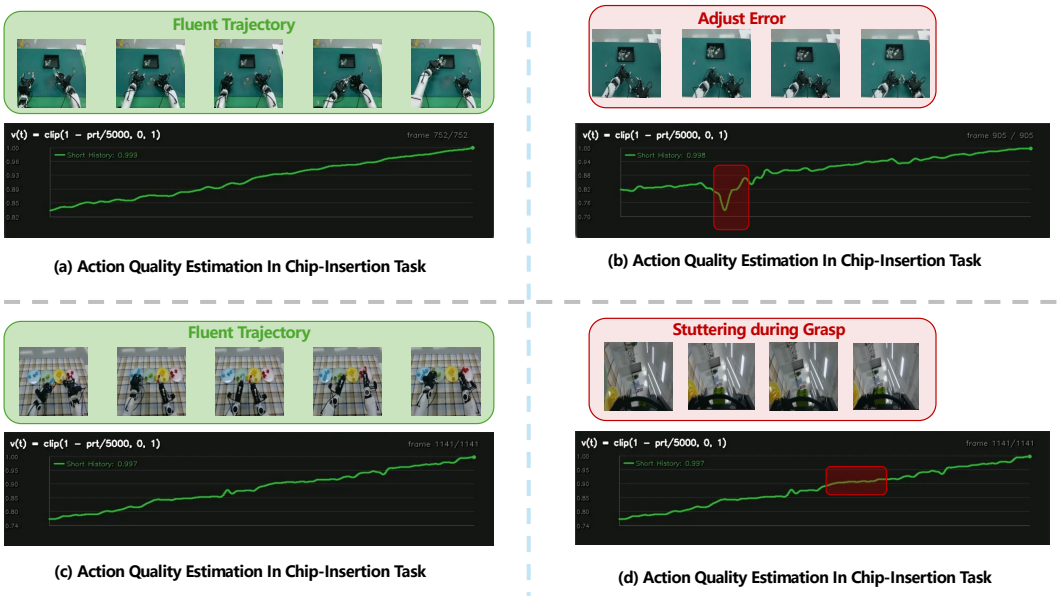


Figure 10 We demonstrate robot self-correction and optimal-action segments in the chip-insertion task. The robot adapts its manipulation strategy under execution disturbances and recovers from unstable intermediate states.

E Value Estimator and Action Quality Estimation Performance

We present the results of value estimation and analyze how action quality affects action filtering with the corresponding video. In Figure 11, we show the value-function estimation results on the chip-insertion task and the block-sorting task. As shown in Figures 11(a) and (c), when a trajectory can smoothly complete the task, the estimated value curve remains stable and smooth throughout the execution. In contrast, when errors occur, our model can sensitively capture their impact on task progress and reflect this change in the value curve. For example, in Figure 11(b), when an adjustment error occurs and the left hand fails to properly align the PCB base to the desired pose, the estimated value drops sharply. It then recovers as the base is manually corrected to the proper position. Figure 11(d) further shows that when the model remains stuck in the grasping stage for an extended period, the value function only fluctuates within a small range, indicating that the task progress is not being advanced. We further visualize the effect of different action-quality thresholds under the same value-function estimates in Figure 12. Specifically, the strict criterion selects the top 30% of samples based on $v(t+20) - v(t)$, whereas the soft criterion selects the top 50% of samples based on $v(t+60) - v(t)$. As shown in the figure, chip insertion is a highly precise manipulation task, where even a short action segment of around 10 frames can determine whether the insertion succeeds. Therefore, the strict threshold can effectively identify meaningful and efficient actions. In contrast, the soft threshold tends to label many suboptimal pre-insertion actions as good actions, which can negatively affect policy learning. The block task exhibits the opposite behavior. Its disassembly process consists of long-horizon actions that require stronger generalization but lower precision. In this case, an overly strict threshold may incorrectly filter out many useful actions as bad ones, thereby degrading the learned policy.



(a) Action Quality Estimation In Chip-Insertion Task

(b) Action Quality Estimation In Chip-Insertion Task

(c) Action Quality Estimation In Chip-Insertion Task

(d) Action Quality Estimation In Chip-Insertion Task

Figure 11 Value-estimation results. We visualize value estimates on chip insertion and block manipulation tasks. Successful trajectories produce smooth and stable value curves, while execution errors or stagnation are reflected by sharp value drops or low-amplitude fluctuations, showing that the value estimator captures task progress and failure recovery.

(a) Action Quality Estimation In Chip-Insertion Task

(b) Action Quality Estimation In Block-Disassembly Task

Figure 12 Effect of action-quality filtering. We compare strict and soft action-quality thresholds under the same value estimates. Strict filtering better identifies effective short-horizon actions in the precise chip-insertion task, whereas soft filtering is more suitable for the long-horizon block task by preserving more useful but temporally extended actions.