ROBOVALUE: A FINE-GRAINED SIM-AND-REAL BENCHMARK FOR UNIFIED EVALUATION OF ROBOTIC VALUE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

General-purpose robotic value models detect task-state changes from visual observations and instructions to score task progress across tasks and embodiments. Yet existing benchmarks test only episode outcomes or isolated process abilities, through model-specific interfaces and mostly on clean successful or failed rollouts, and thus cannot tell whether a model tracks task state or exploits shortcuts such as elapsed time and visual similarity. We introduce **RoboValue**, a benchmark with a unified protocol for fine-grained evaluation of robotic value models on simulation and real-world tasks. Shared interfaces and adapters enable comparison across heterogeneous models while preserving native value semantics. RoboValue evaluates four complementary dimensions: task-state understanding, temporal progress monitoring, failure and recovery reasoning, and value consistency. Beyond common successful and failed rollouts, it includes diagnostic trajectories with incomplete subtasks, effective and ineffective recovery, visually similar states with different execution histories, and multiple valid action orders. We evaluate 15 model variants from 9 families in zero-shot and one-shot settings, assessing performance under standard conditions and generalization across embodiment and environment shifts. RoboValue provides a public leaderboard and a diagnostic foundation for developing more reliable and general robotic value models. We propose RoboValue-Dataset, which comprises an evaluation set of 2,792 trajectories spanning 15 simulation and 20 real-world manipulation tasks across 4 embodiments, with corresponding training data. The website is available at https://robovalue.github.io/.

1 INTRODUCTION

Reliable assessment of task execution is essential for the curation of training data (Budzianowski et al., 2025), effective reinforcement learning of robotic foundation models (Tsao et al., 2026; Physical Intelligence et al., 2025; Fei et al., 2026; Liu et al., 2026a; Yong et al., 2026), and execution monitoring and failure diagnosis (Liang et al., 2026; Tan et al., 2025b; Ye et al., 2025; Park et al., 2026). Handcrafted dense rewards require task-specific design and generalize poorly (Mu et al., 2024; Kim et al., 2025), motivating general-purpose robotic value models that use visual observations and task instructions to identify changes in task state and assign scalar values that characterize execution (Alakuijala et al., 2024). Systematic evaluation is therefore essential to determine whether robotic value models reliably characterize task execution.

Existing evaluations lack a unified diagnostic protocol for jointly assessing instruction grounding, execution memory, recovery outcomes, and value consistency (Lee et al., 2026; Liang et al., 2026; Xia et al., 2026; Ji et al., 2026; Liu et al., 2026b). Differences in model interfaces and value semantics further complicate systematic comparison (Zhang et al., 2026b). As shown in Figure 1, predicted values may increase with elapsed time though visual evidence indicates regression, rebound during continued execution despite unresolved errors, or remain insensitive to execution history when current observations are visually similar. Without a unified diagnostic evaluation, such failures may go unnoticed and provide misleading optimization signals to downstream policies.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address these gaps, we introduce **RoboValue**, a unified sim-and-real benchmark for fine-grained evaluation of heterogeneous robotic value models. As illustrated in Figure 2, we ask *whether models understand task instructions and execution states*, *whether they track progress, regression and task-relevant history*, *when failures occur and whether recovery succeeds*, and *whether values remain stable over time and consistent across valid solutions*. These define four complementary capability dimensions: task-state understanding, temporal progress monitoring, failure and recovery reasoning, and value consistency.

RoboValue-Dataset contains 2,792 evaluation trajectories from 15 simulation and 20 real-world manipulation tasks across 4 embodiments, with corresponding training data (Section 3.2). Simulation provides diverse object assets and varied scene configurations under controlled conditions, while real-world data captures visual sensing and contact dynamics in complex physical environments. Beyond common successful and failed rollouts, we construct diagnostic trajectories with incomplete subtasks and effective or ineffective recovery, visually similar states with different execution histories, and alternative valid action orders. To support comparisons across heterogeneous models, RoboValue defines shared interfaces for scalar values, pairwise comparisons, and subtask descriptions, with model-specific adapters that preserve native value semantics (Section 3.3).

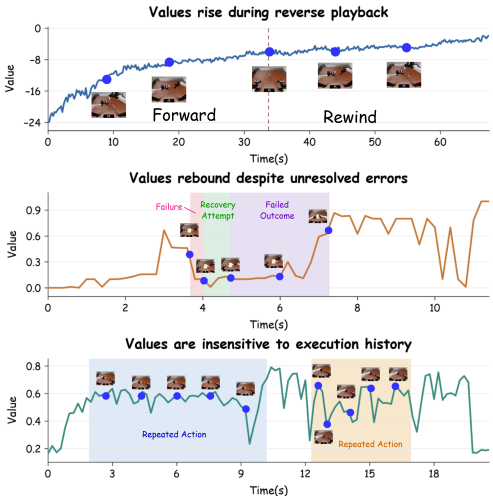

Figure 1: **Failure cases of value models.**

Our experiments cover 15 model variants from 9 families in zero-shot and one-shot settings, measuring performance under standard conditions and generalization across embodiment and environment shifts (Section 4). While several models demonstrate strong outcome discrimination and progress correlation, they still exhibit weaknesses in fine-grained instruction grounding, execution memory, and failure and recovery reasoning. These gaps remain evident in models with larger backbones and richer temporal context, suggesting that task semantics and execution history require more explicit supervision (Section 4.4). In summary, our main contributions are as follows:

- We introduce **RoboValue**, a **unified benchmark for fine-grained evaluation** of robotic value models with heterogeneous interfaces and value semantics across four capability dimensions, enabling systematic comparisons of how reliably models assess task execution beyond coarse outcome prediction.
- We design **broad sim-and-real evaluation scenarios** spanning diverse manipulation tasks, embodiments, and diagnostic execution patterns. Cross-embodiment and cross-environment shifts further test whether value judgments remain reliable under changes in robot morphology and background conditions.
- We evaluate 15 model variants from 9 families under zero-shot and one-shot settings, establish a **public leaderboard**, and provide **systematic analyses** of current limitations and priorities for future general-purpose robotic value models.

2 Related Work

2.1 Value Models for Robotic Manipulation

Recent robotic value models vary in their interfaces and output semantics. Some models derive value estimates from assessments of task completion, using discrete ratings or the log-probability of an affirmative response indicating completion (Lee et al., 2026; Chen et al., 2026c). Other models estimate scalar values of task progress, using diverse architectures and supervision strategies (Liang et al., 2026; Feng et al., 2026; Chen et al., 2026a;b; Ma et al., 2025; Lv et al., 2026;

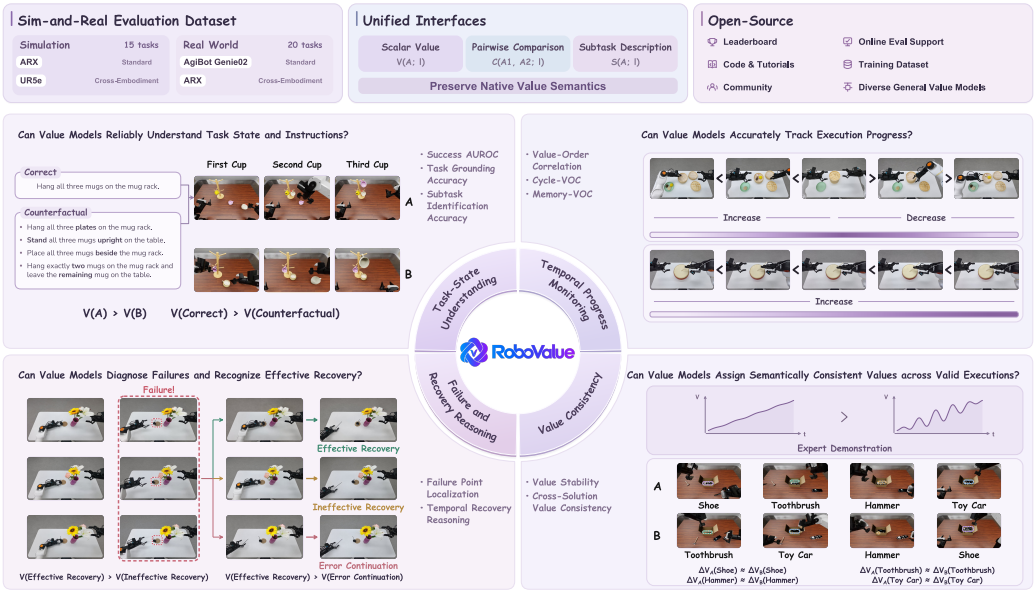

Figure 2: **Overview of RoboValue.** The benchmark combines simulation and real-world trajectories with shared evaluation interfaces to assess value models across four complementary capabilities.

Wang et al., 2026; Qin et al., 2026; Zhang et al., 2025; 2026a; Fang et al., 2026; Yan et al., 2026; Hung et al., 2025). A third group estimates temporal distances, including remaining time to task completion (Huang et al., 2026; Liu et al., 2025), while a fourth group estimates relative progress or advantage between states (Tan et al., 2025a; Xu et al., 2026; Zhai et al., 2025; Mao et al., 2026; Liu et al., 2026c). To fairly evaluate these heterogeneous models, RoboValue provides a unified multidimensional evaluation protocol that assesses their value estimation capabilities.

2.2 Evaluation of Robotic Value Models

Existing evaluations assess robotic value models at the outcome and process levels (Wang et al., 2025; ElMallah et al., 2025). One line of work evaluates task completion prediction, instruction grounding, reward correlation, and trajectory preference ranking (Lee et al., 2026; Liang et al., 2026). Another examines process understanding through diagnostic questions (Xia et al., 2026) and pairwise or interval-level progress discrimination (Ji et al., 2026; Liu et al., 2026b). However, differences in interfaces and evaluation criteria limit systematic comparison across models. RoboValue combines diagnostic trajectories and capability-specific metrics under a unified protocol to evaluate heterogeneous models across task-state understanding, temporal progress monitoring, failure and recovery reasoning, and value consistency.

3 RoboValue Benchmark

3.1 Preliminaries

In RoboValue, *robotic value models* identify task-state changes from visual observations and instructions and assign scalar scores that reflect task execution. The semantics and numerical scales of these scores depend on the model.

For an episode e with instruction l, let $\tau_e = (o_0, \ldots, o_{T-1})$ denote its length-T observation sequence. Each query A contains a visual observation and model-specific context, such as execution history, a reference demonstration, or robot states. We write the scalar execution score as $v(A; l) = f_\theta(A, l) \in \mathbb{R}$, where θ denotes the model parameters. RoboValue also defines a pairwise comparison $c(A, B; l)$ that favors B over A when positive, and a subtask description $s(A; l)$.



Figure 3: **RoboValue Benchmark Construction.** RoboValue comprises 2,792 trajectories across 15 simulation and 20 real-world manipulation tasks. Representative examples illustrate diagnostic trajectories and evaluation settings with embodiment and environment shifts.

3.2 Benchmark Construction

Dataset Composition. RoboValue-Dataset comprises training and evaluation sets. The evaluation set contains 2,792 trajectories across 35 manipulation tasks, including 1,231 from 15 simulation tasks and 1,561 from 20 real-world tasks. It supports in-domain (ID) evaluation under a standard setting and out-of-domain (OOD) evaluation under cross-embodiment and cross-environment settings. The 15 simulation tasks are adapted from RoboDojo (Chen et al., 2026d). Simulation enables scalable evaluation across diverse task configurations and execution conditions, while real-world data captures physical execution under realistic visual and interaction conditions. All tasks are performed on dual-arm platforms and cover diverse manipulation skills, with varying complexity and physical interaction requirements. For cross-embodiment evaluation, we change the robot embodiment. For cross-environment evaluation, we vary backgrounds, lighting, object appearance, and clutter configurations. All three settings follow the same evaluation protocol, with each OOD setting varying either the embodiment or the environment relative to the standard setting. Figure 3 shows the 35 tasks, the cross-embodiment and cross-environment settings in simulation and the real world, and representative diagnostic trajectories. Dataset statistics, task specifications, and OOD configurations and visualizations are provided in Appendices B.3 and D.

Trajectory Design. We organize six trajectory types into four categories to capture diverse execution patterns. (1) **Expert Demonstrations:** The robot executes the task fluently and without errors. (2) **Failure and Recovery Trajectories:** This category comprises three types. *Error Continuation:* After making an error during execution, the robot fails to recognize it and instead continues with the remaining task. *Effective Recovery:* After making an error during execution, the robot takes corrective actions and successfully recovers. *Ineffective Recovery:* After making an error during execution, the robot makes an unsuccessful recovery attempt and continues with the remaining task. (3) **Long-Horizon Temporal Trajectories:** The robot performs repeated actions and revisits visually similar states with different execution histories. (4) **Multi-Solution Trajectories:** The robot completes the task using alternative valid subtask orders that respect task dependencies.

Data Collection. We collect trajectories through VR-based teleoperation using handheld controllers across all four embodiments: ARX and Franka Emika in the Isaac Sim (NVIDIA Corporation, 2026) simulation platform, and AgiBot Genie02 and ARX in the real world. Each trajectory includes

synchronized visual observations from head- and wrist-mounted cameras, together with robot states and executed actions. Platform configurations and collection details are provided in Appendix A.

Trajectory Annotation. We manually annotated all trajectories to support evaluation across different capability dimensions. For successful trajectories, we annotated the start and end timestamps of each subtask. For Failure and Recovery trajectories, we first annotated the beginning and end of the initial failure stage. Following this shared prefix, the robot continues without recovery, recovers through corrective actions, or makes an unsuccessful recovery attempt. For Effective Recovery and Ineffective Recovery trajectories, we additionally annotated the moment when the recovery outcome becomes observable. Finally, for all three types, we annotated the endpoint of the subtask in which the initial failure occurred.

3.3 Evaluation Protocol

Unified Interfaces and Adapters. We use model-specific adapters to map heterogeneous inputs and outputs to shared evaluation interfaces. For $v(A; l)$, adapters transform native scalar output or aggregate relative predictions into values. For $c(A_1, A_2; l)$, adapters use native pairwise predictions when available or derive comparisons as $c(A_1, A_2; l) = v(A_2; l) - v(A_1; l)$. For $s(A; l)$, we obtain subtask descriptions from supported textual outputs. See Appendix B.2 for model-specific adaptations and inference details.

Evaluation Settings. We evaluate models under zero-shot and one-shot settings. Zero-shot evaluation uses released model checkpoints without task-specific adaptation or reference demonstrations. One-shot evaluation uses a single demonstration per task, excluded from the evaluation set, for inference-time conditioning or task-specific adaptation. We report in-domain performance and out-of-domain generalization across environment and embodiment shifts, without further adaptation.

3.4 Capability Dimensions and Metrics

We organize the evaluation into four complementary dimensions: **Task-State Understanding** (SA, TGA), **Temporal Progress Monitoring** (VOC, Cycle-VOC, Memory-VOC), **Failure and Recovery Reasoning** (FPL, TRR), and **Value Consistency** (VS, CSVC). Additional Metric SIA are provided in Appendix G.1.

Success Accuracy (SA, ↑). For data curation, value models should distinguish fully successful executions from partial or failed ones. So, SA assesses whether episode-level scores ($\bar{v}$) rank successful episodes ($\mathcal{E}^+$) above partially successful and failed episodes ($\mathcal{E}^-$), within each task and evaluation condition: $\mathrm{SA} = \frac{\sum_{e^+ \in \mathcal{E}^+} \sum_{e^- \in \mathcal{E}^-} \mathbf{1}[v_{e^+}^- > v_{e^-}^-]}{|\mathcal{E}^+||\mathcal{E}^-|}$.

Task Grounding Accuracy (TGA, ↑). For instruction-conditioned policy optimization, value models should ground their value estimates in the given instruction, rather than rewarding progress toward other plausible goals. TGA therefore tests instruction grounding by assessing the value gain $\Delta(l) = c(A_{initial}, A_{terminal}; l)$ of the same trajectory under the correct and alternative instructions. There are two variants: Cross-Task (CT) and Counterfactual (CF). CT uses instructions from other tasks ($\mathcal{N}_{CT}$), while CF uses counterfactual instructions constructed by modifying different semantic aspects: objects, actions, placement, or constraints ($\mathcal{N}_{CF}$):
$\mathrm{TGA}_m = \frac{1}{|\mathcal{E}^+|} \sum_{e \in \mathcal{E}^+} \mathbf{1}[\Delta(l_k) > \max_{l \in \mathcal{N}_m} \Delta(l)], \quad m \in \{\mathrm{CT}, \mathrm{CF}\}.$

Value-Order Correlation (VOC, ↑). Value models should provide continuous and directionally correct feedback throughout execution, assigning higher values as the task progresses toward completion. VOC (Ma et al., 2025) therefore assesses whether predicted values increase with task advancement along expert trajectories, using temporal order as a progress proxy. We compute the Spearman correlation (ρ_{S}) between values and normalized temporal positions, assigning zero to constant predictions: $\mathrm{VOC} = \rho_{\mathrm{S}}\left((v(t_i))_{i=1}^N, (t_i/T)_{i=1}^N\right), \quad 0 \le t_1 < \cdots < t_N < T.$

Cycle-VOC (↑). Cycle-VOC assesses whether a model recognizes task regression rather than merely increasing values as time elapses, regardless of execution direction. We concatenate each expert trajectory with its reversed sequence, retaining forward context for history-aware models. The score averages the Spearman correlations against increasing and decreasing progress orders for the two halves, which share the turning point.

Memory-VOC (↑). For long-horizon and repetitive tasks, value models should use execution history to distinguish visually similar states, giving reasonable feedback. Memory-VOC therefore assesses whether a model can distinguish similar observations and assign appropriate values based on their execution histories. We evaluate expert trajectories in which similar observations recur, using the same scoring and aggregation protocol as VOC.

Failure-Point Localization (FPL, ↓). Value models should separate suboptimal trajectories into successful and failed stages, so that those data can be more fully utilized. FPL assesses how accurately declines in values localize the failure point (t^{GT}) in trajectories containing a single execution error. We extract declines using a ZigZag filter that ignores minor reversals and retains unfinished terminal declines. The peak of the largest decline is taken as the predicted point $\hat{t}$. We measure the error normalized by the final-frame timestamp T as $\mathrm{FPL} = |\hat{t} - t^{\mathrm{GT}}|/T$; if no decline is detected, we assign $\max(t^{\mathrm{GT}}, T - t^{\mathrm{GT}})/T$.

Trajectory Recovery Reasoning (TRR, ↑). Value models should recognize execution failures and provide positive feedback for effective corrective actions, while avoiding rewarding ineffective recovery attempts. TRR therefore tests whether predicted values change appropriately in response to failures, recovery attempts, and the resulting outcomes. Each matched group contains continued-error (CE), successful-recovery (SR), and failed-recovery (FR) trajectories sharing the same failure scenario. We evaluate value changes across the failure (f), continued-error (c), recovery-attempt (r), and outcome (o) stages. We measure the overall value trend using duration weights w_i for each sample, setting $D_s = 0$ for constant predictions ($s \in \{f, c, r, o\}$). Each trajectory type is scored based on its stage-wise trends:

$$D_s = \frac{\sum_{i<j} w_i w_j (v_e(j) - v_e(i))}{\sum_{i<j} w_i w_j |v_e(j) - v_e(i)|}, \quad \mathrm{TRR}_e = \mathbf{1}[D_f < 0] \begin{cases} \mathbf{1}[D_c \le 0], & \mathrm{CE}, \\ \mathbf{1}[D_r > 0 \wedge D_o > 0], & \mathrm{SR}, \\ \mathbf{1}[D_r > 0 \wedge D_o < 0], & \mathrm{FR}. \end{cases} \tag{1}$$

Value Stability (VS, ↑). For policy optimization, value models should provide stable and reliable feedback, avoiding both noisy reversals and uninformative flat predictions.

VS therefore assesses whether predicted values follow a consistent trend along expert trajectories, while discounting apparent stability caused by prolonged flat segments. We combine Kaufman's Efficiency Ratio (ER) (Kaufman, 1995), the ratio of absolute net change to cumulative absolute variation, with non-flat time coverage (C). For a trajectory with N raw predictions v_i at timestamps t_i, ER uses $\tilde{v}_i$, which equals the respective mean within the first and last three samples and v_i elsewhere, reducing sensitivity to endpoint fluctuations: $\mathrm{ER} = \frac{|\tilde{v}_N - \tilde{v}_1|}{\sum_{i=1}^{N-1} |\tilde{v}_{i+1} - \tilde{v}_i|}$, $C = \frac{\sum_{i=1}^{N-1} (t_{i+1} - t_i) \mathbf{1}[v_{i+1} \neq v_i]}{t_N - t_1}$, and $\mathrm{VS} = \mathrm{ER} \cdot C$.

Cross-Solution Value Consistency (CSVC, ↑). Facing tasks with multiple valid solutions, value models should assign consistent gains across different execution orders, rather than overfitting to a particular order. CSVC therefore assesses whether a model assigns consistent value gains to the same semantic subtask across different valid solutions, rather than overfitting to a single execution order. For subtask a in solution q, we estimate its gain $c_{q,a}$ from paired comparisons $c(A_{\mathrm{suba_begin}}, A_{\mathrm{suba_end}}; l)$ near its annotated boundaries. Across all $a \in \mathcal{A}$ and $q \in \mathcal{S}_a$, we compare the magnitude of median gains $G = \sqrt{\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \widetilde{c}_a^2}$, with their cross-solution dispersion $D = \sqrt{\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \frac{1}{|\mathcal{S}_a|} \sum_{q \in \mathcal{S}_a} (c_{q,a} - \widetilde{c}_a)^2}$. The resulting score is $\mathrm{CSVC} = \frac{G - D}{G + D}$.

4 Experiments

4.1 Evaluated Value Models

We evaluate 15 model variants from 9 families with diverse value semantics. Progress signals include direct estimates (Liang et al., 2026; Feng et al., 2026), aggregated relative predictions (Tan et al., 2025a; Xu et al., 2026; Zhai et al., 2025), and assessments elicited with our rubric (Ye et al., 2025). Completion signals include discrete ratings (Lee et al., 2026) and the log-probability of an affirmative completion response (Chen et al., 2026c). We also evaluate negated remaining-time pre-

Table 1: **Main Results.** Performance across four capability dimensions under the standard setting. All metrics are on a $\times 100$ scale.

Model	Understanding			Tracking			Diagnosis		Consistency	
	SA	TGA-CT	TGA-CF	VOC	Cyc	Mem	FPL	TRR	VS	CSVC
Zero-Shot Evaluation										
LIV	54.44	9.10	6.17	8.33	8.33	1.50	31.39	24.25	3.36	-8.47
ProcVLM-2B	62.20	20.04	14.36	51.10	47.41	9.73	36.14	21.55	5.25	4.60
RoboFAC-7B	72.63	4.82	4.34	20.44	36.95	-25.23	34.64	**33.80**	3.34	-0.30
RoboMeter-4B	81.29	82.00	29.29	**86.05**	**89.43**	2.36	**29.80**	24.83	**18.17**	32.41
RoboReward-4B	28.10	35.97	9.12	82.76	44.23	81.46	32.17	3.07	-	-
RoboReward-8B	31.31	28.89	12.81	83.40	43.00	**81.77**	31.30	2.60	-	-
RynnValue-4B	**82.12**	**83.74**	**56.01**	73.19	81.74	-23.89	39.95	23.18	9.88	24.95
RynnValue-8B	80.41	72.82	46.21	73.52	83.06	-24.70	39.79	26.61	8.97	13.25
TOPReward[‡]	58.98	51.89	21.30	94.29[§]	13.50	95.51[§]	30.73	12.85	15.06	**45.45**
VLAC-2B	33.75	21.54	9.16	9.13	9.72	5.73	47.46	8.95	4.15	-
VLAC-8B	41.04	35.75	13.64	28.53	26.12	21.70	45.48	5.24	6.54	-
One-Shot Evaluation										
ProcVLM-2B	83.07	**59.25**	**51.16**	91.40	91.24	23.96	33.94	21.05	18.08	19.90
Robo-Dopamine-3B	73.77	7.44	3.03	76.30	76.40	16.69	34.32	32.02	12.56	17.65
Robo-Dopamine-4B[†]	**87.16**	3.34	0.77	**94.47**	**94.43**	13.73	**27.34**	33.54	**30.69**	35.40
Robo-Dopamine-8B	80.09	3.69	1.23	88.17	87.19	11.28	32.86	36.55	19.08	20.51
Robo-Dopamine-8B[†]	86.36	0.73	1.23	93.05	93.74	18.84	29.30	**38.75**	30.53	**39.48**
VLAC-2B	64.54	13.32	8.99	67.57	60.24	10.50	29.59	21.00	22.34	29.55
VLAC-8B	63.71	34.63	18.37	74.54	63.57	**51.12**	28.63	15.35	25.46	30.26

[†] 2.0 Preview checkpoints. [‡] Qwen3-8B backbone. [§] Additional note.

dictions (Huang et al., 2026) and image–text similarity scores (Ma et al., 2023). Model and adapter details are provided in Appendices B.1– B.2.

4.2 Main Results

Can Value Models Reliably Understand Task State and Instructions? Strong outcome discrimination does not guarantee reliable instruction grounding. As shown in Table 1, RynnValue-4B and RoboMeter combine strong outcome discrimination with task grounding, consistent with instruction-verification training and mismatched-task trajectory comparisons, respectively. But both are less accurate on counterfactual instructions, indicating that they can distinguish broadly different task goals but still struggle with fine-grained instruction changes involving the specified object, action, location, or constraints. Robo-Dopamine shows strong outcome discrimination but weak instruction grounding, suggesting that its value estimates are anchored more strongly to the demonstrated goal than to the instruction. (see Appendix G.2) In contrast, ProcVLM-2B more reliably distinguishes the intended task from alternative instructions. Its gains over the zero-shot checkpoint suggest that one-shot adaptation better aligns pretrained procedural knowledge with task-specific visual evidence. More details are given in Appendices C.2, D.2.

Can Value Models Accurately Track Execution Progress? Our results show that high forward-progress correlation does not imply a reliable understanding of progress regression and execution history. As shown in Table 1, TOPReward leads zero-shot VOC and Memory-VOC but performs poorly on Cycle-VOC. We find that this discrepancy stems from TOPReward's bias toward temporal progression rather than a genuine understanding of task progress. It relies heavily on the timestamp provided to its base model Qwen3-VL, and increases its predicted values as the video advances even when the visual evidence indicates task regression. Conversely, RoboMeter and RynnValue handle reversals more reliably, consistent with rewind augmentation and signed temporal supervision, respectively. Robo-Dopamine 2.0 and adapted ProcVLM also track both execution directions well, in line with their supervision on signed relative progress and procedural stages. Yet these models struggle when visually similar states recur, indicating that they can accurately track progress changes without reliably remembering which subgoals have been completed.

Table 2: **Generalization Results.** Performance across four capability dimensions under embodiment shifts (left values) and environment shifts (right values). All metrics are on a $\times 100$ scale.

Model	Understanding			Tracking			Diagnosis	Consistency
	SA	TGA-CT	TGA-CF	VOC	Cyc	Mem	FPL	VS
Zero-Shot Evaluation								
LIV	55.90 / 52.61	17.34 / 8.36	9.40 / 10.36	13.65 / 5.52	13.65 / 5.52	-2.51 / 2.73	**32.13** / 32.79	2.48 / 2.84
ProcVLM-2B	66.75 / 55.71	31.26 / 17.50	17.69 / 16.05	53.71 / 45.50	47.06 / 40.66	8.84 / 15.31	37.61 / 35.72	4.31 / 3.64
RoboFAC-7B	78.22 / 73.15	6.16 / 3.85	7.20 / 3.99	27.12 / 5.99	36.74 / 26.38	-16.75 / -42.24	39.28 / 39.42	3.41 / 5.55
RoboMeter-4B	**85.42** / 79.14	79.53 / **91.13**	28.63 / 20.33	**86.79** / 81.38	**90.04 / 85.50**	-7.57 / -6.22	35.14 / **30.38**	**15.48 / 15.91**
RoboReward-4B	25.03 / 29.69	33.13 / 28.08	11.93 / 6.84	79.24 / **83.51**	41.02 / 43.83	79.71 / **83.27**	35.15 / 34.15	-
RoboReward-8B	32.12 / 26.81	40.17 / 27.86	10.61 / 9.23	79.91 / 82.89	44.30 / 41.55	**82.42** / 81.11	37.41 / 35.83	-
RynnValue-4B	79.74 / 76.47	**91.60** / 85.80	**57.61 / 53.19**	77.87 / 66.74	83.84 / 77.03	-23.17 / -16.50	43.62 / 44.19	9.78 / 8.58
RynnValue-8B	79.18 / **79.41**	76.96 / 79.23	53.49 / 49.07	77.59 / 65.61	84.97 / 78.04	-31.88 / -22.45	43.50 / 42.10	7.78 / 8.50
TOPReward‡	68.85 / 53.40	51.05 / 49.10	18.60 / 23.90	93.91§ / 92.36§	18.83 / 8.23	96.24§ / 94.51§	37.78 / 34.90	14.13 / 14.07
VLAC-2B	42.90 / 36.92	39.81 / 15.99	10.74 / 3.03	10.28 / 14.77	16.41 / 15.40	-2.00 / -7.83	39.86 / 49.97	5.29 / 5.06
VLAC-8B	51.92 / 38.96	56.87 / 41.46	18.96 / 16.98	41.92 / 28.56	31.58 / 26.19	9.04 / -6.21	41.47 / 47.39	7.54 / 6.80
One-Shot Evaluation								
ProcVLM-2B	83.77 / 72.27	**48.82 / 50.03**	**44.65 / 38.22**	86.66 / 78.36	86.35 / 77.04	14.86 / 21.23	32.99 / 33.76	12.49 / 12.24
Robo-Dopamine-3B	80.45 / 61.73	7.75 / 17.86	5.47 / 3.36	79.68 / 66.55	79.20 / 65.79	21.47 / 18.55	30.08 / 34.23	10.68 / 7.62
Robo-Dopamine-4B†	85.03 / **80.33**	1.51 / 4.12	4.76 / 2.26	**92.48** / 89.84	**92.68** / 89.40	2.05 / 12.71	29.45 / 31.45	**27.99** / 24.34
Robo-Dopamine-8B	80.26 / 71.95	6.29 / 9.07	1.16 / 4.07	85.23 / 76.81	84.05 / 76.46	22.86 / 16.19	30.46 / 32.28	13.25 / 11.13
Robo-Dopamine-8B†	**86.99** / 79.88	1.46 / 3.79	2.92 / 0.36	91.76 / **90.82**	92.66 / **91.04**	-4.34 / 12.73	**26.56** / 33.27	**27.99 / 24.41**
VLAC-2B	62.81 / 60.10	11.26 / 13.71	15.77 / 2.97	66.37 / 59.66	58.45 / 56.61	30.27 / 14.11	29.15 / **29.99**	17.77 / 17.70
VLAC-8B	70.22 / 58.93	28.22 / 34.43	21.19 / 17.34	80.35 / 64.94	70.12 / 58.17	**41.96 / 23.62**	28.68 / 31.52	22.26 / 22.44

† 2.0 Preview checkpoints. ‡ Qwen3-8B backbone. § Additional note.

Can Value Models Diagnose Failures and Recognize Effective Recovery? Reliable feedback requires value models to localize failures, track unresolved errors, and update values according to recovery outcomes. Yet current value models struggle to meet these requirements. As shown in Table 1, mean FPL error is approximately 34% of the trajectory length, slightly above the 33% random baseline (which uniformly samples a timestamp as failure onset from each failure trajectory and applies the same FPL scoring protocol), and TRR scores are generally low. RoboMeter localizes failures more accurately, whereas RoboFAC achieves higher TRR, particularly on continued-error and failed-recovery branches, consistent with its failure-analysis and correction training (see Appendix G.3). This difference reflects that responding to failure and precisely locating its onset are distinct capabilities. Robo-Dopamine 2.0 achieves relatively strong results in both, consistent with its supervision across both failure and recovery transitions. We further find that current value models may respond more strongly to other trajectory changes than to the target failure or fail to produce a distinct failure response. Even after correctly reflecting the initial failure, values may rebound or fail to decline when errors persist or recovery attempts fail. More details are given in Appendix G.3.

Can Value Models Assign Semantically Consistent Values across Valid Executions? Our results reveal that providing stable and informative values and assigning consistent gains to the same semantic subtasks across valid solutions remain challenging. As shown in Table 1, TOPReward and RoboMeter provide comparatively stable local feedback and consistent subtask gains across solutions among zero-shot models. Robo-Dopamine 2.0 achieves relatively strong performance in both metrics under one-shot evaluation. We further find that CSVC tends to be higher for repetitive manipulation than for reordered semantic subtasks: Value gains remain similar across valid solutions for repeated placements but vary when the same semantic subtask occurs at different execution positions. These findings suggest that global progress tracking does not guarantee reliable local feedback and reveal limited robustness to changes in execution order when valuing the same semantic subtask (detailed analysis in Appendix G.4). We omit RoboReward, whose discrete outputs yield many zero gains, and zero-shot VLAC, whose frequent constant gains limit fine-grained CSVC analysis.

4.3 Generalization Results

Cross-Embodiment Generalization. Cross-embodiment generalization requires value models to judge task execution from object states and interaction outcomes relative to the task goal, rather than relying on the appearance or motion patterns of a particular robot. As shown in Table 2, models largely retain their in-domain strengths, suggesting that task understanding can transfer across robot morphologies and execution patterns. RoboMeter retains strong outcome discrimination, while RynnValue-4B maintains strong instruction grounding. Their transfer is consistent with

training across diverse embodiments, coupled with trajectory-preference supervision for RoboMeter and instruction verification for RynnValue. ProcVLM's adapted checkpoint retains strong outcome discrimination but declines in grounding and progress tracking, suggesting that calibration from a single demonstration transfers only partially.

Cross-Environment Generalization. Cross-environment generalization requires value models to distinguish actual changes in task state from irrelevant visual variation while maintaining accurate recognition of target objects and actions. As shown in Table 2, environmental shifts cause broader degradation across models than embodiment shifts, revealing limited robustness to visual variation. ProcVLM and VLAC-8B decline in outcome discrimination and progress tracking despite access to a task demonstration. VLAC-8B's advantage over ProcVLM on history-dependent executions also narrows. These declines suggest that task adaptation in ProcVLM and reference conditioning in VLAC leave judgments tied to the demonstration's visual context. Robo-Dopamine 2.0 better preserves progress tracking and retains the highest value stability, with its tracking advantage over version 1.0 widening after the shift. This resilience is consistent with its explicit training to preserve progress targets under visual perturbations that leave task semantics unchanged, suggesting that robustness benefits from learning which visual changes should not affect value estimates.

4.4 Analysis

Does Temporal Context Enable History-Aware Value Estimation? Reliable progress tracking requires not only multiple input frames but also memory of which subgoals have been completed (Zhang et al., 2026c; Chen et al., 2026e). RoboMeter and RynnValue track forward and reversed executions much better with sampled histories, yet their near-zero or negative Memory-VOC indicates difficulty distinguishing visually similar observations at different task stages. This contrast suggests that simply exposing earlier observations does not ensure that models retain which subgoals have been completed.

How Does Supervision Construction Shape Performance? Our results suggest that different supervision signals are associated with distinct capability gains rather than uniform improvements in execution understanding. Robo-Dopamine 2.0 better preserves execution tracking under environmental shifts than 1.0, consistent with supervision that holds targets constant under visual perturbations that preserve task semantics. RoboMeter and RynnValue, in contrast, exhibit more reliable instruction grounding with explicit instruction-mismatch supervision and track regressions well with rewind and temporal-order perturbation training. However, their remaining weakness in fine-grained counterfactual grounding and execution memory indicates that the capabilities induced by these targeted supervision signals do not automatically transfer to more demanding semantic or history-dependent reasoning.

Do Larger or Shared Backbones Yield Better Execution Understanding? Our comparisons show that neither increasing model scale nor sharing the same backbone consistently determines fine-grained execution understanding. RynnValue-8B only slightly improves regression tracking over 4B while performing worse in instruction grounding and cross-solution consistency, indicating that scaling does not yield uniform capability gains. Similarly, RoboMeter outperforms RoboReward-4B in outcome discrimination and regression tracking despite their shared Qwen3-VL-4B backbone. These comparisons suggest that scaling alone is unlikely to close fine-grained execution-understanding gaps; improving these capabilities may instead require training data and supervision that explicitly cover the execution patterns and reasoning skills being evaluated.

5 Conclusion

We presented **RoboValue**, a unified sim-and-real benchmark for fine-grained evaluation of heterogeneous robotic value models. Shared interfaces and diagnostic trajectories enable systematic assessment of execution understanding while preserving native value semantics. Across 15 model variants from 9 families, strong outcome discrimination and progress correlation coexist with weaknesses in fine-grained instruction grounding, execution memory, and failure and recovery reasoning. These gaps persist in models with larger backbones or richer temporal context, motivating more explicit supervision of task semantics and execution history. RoboValue-Dataset and the public leaderboard support the development of more reliable and general robotic value models.

AI USE STATEMENT

We used large language models to assist with literature lookup, manuscript drafting, and language editing, and to provide feedback on evaluation methodology and the interpretation of experimental results. This assistance included checking spelling and grammar, improving clarity, and checking consistency between reported results and the accompanying discussion. The authors are responsible for verifying all AI-assisted content and take full responsibility for the final manuscript, including its references, methodology, data, experimental results, and conclusions.

REFERENCES

Minttu Alakuijala, Reginald McLean, Isaac Woungang, Nariman Farsad, Samuel Kaski, Pekka Marttinen, and Kai Yuan. Video-language critic: Transferable reward functions for language-conditioned robotics. *arXiv preprint arXiv:2405.19988*, 2024.

Paweł Budzianowski, Emilia Wiśnios, Michał Tyrolski, Gracjan Góral, Igor Kulakov, Viktor Petrenko, and Krzysztof Walas. Opengvl–benchmarking visual temporal progress for data curation. *arXiv preprint arXiv:2509.17321*, 2025.

Qianzhong Chen, Justin Yu, Mac Schwager, Pieter Abbeel, Fred Shentu, and Philipp Wu. Sarm: Stage-aware reward modeling for long horizon robot manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 17091–17116, 2026a.

Qianzhong Chen, Hau Zheng, Justin Yu, Suning Huang, Jiankai Sun, Ken Goldberg, Chuan Wen, Pieter Abbeel, Yide Shentu, Philipp Wu, et al. Sarm2: Multi-task stage aware reward modeling for self improving robotic manipulation. *arXiv preprint arXiv:2606.10305*, 2026b.

Shirui Chen, Cole Harrison, Ying-Chun Lee, Angela Jin Yang, Zhongzheng Ren, Lillian J Ratliff, Jiafei Duan, Dieter Fox, and Ranjay Krishna. Topreward: Token probabilities as hidden zero-shot rewards for robotics. *arXiv preprint arXiv:2602.19313*, 2026c.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, et al. Robodojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026d.

Tianxing Chen, Yuran Wang, Mingleyang Li, Yan Qin, Hao Shi, Zixuan Li, Yifan Hu, Yingsheng Zhang, Kaixuan Wang, Yue Chen, et al. Rmbench: Memory-dependent robotic manipulation benchmark with insights into policy design. *arXiv preprint arXiv:2603.01229*, 2026e.

Ramy ElMallah, Krish Chhajer, and Chi-Guhn Lee. Score the steps, not just the goal: Vlm-based subgoal evaluation for robotic manipulation. *arXiv preprint arXiv:2509.19524*, 2025.

Yu Fang, Wanxi Dong, Jiaqi Liu, Yue Yang, Mingxiao Huo, Yao Mu, Huaxiu Yao, Li Erran Li, Daniel Szafir, and Mingyu Ding. Densereward: Dense reward learning via failure synthesis for robotic manipulation. *arXiv preprint arXiv:2607.13033*, 2026.

Senyu Fei, Siyin Wang, Li Ji, Ao Li, Shiduo Zhang, Liming Liu, Jinlong Hou, Jingjing Gong, Xianzhong Zhao, and Xipeng Qiu. Srpo: Self-referential policy optimization for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6718–6728, 2026.

Youhe Feng, Hansen Shi, Haoyang Li, Xinlei Guo, Yang Wang, Chengyang Zhang, Jinkai Zhang, Xiaohan Zhang, Jie Tang, and Jing Zhang. ProcVLM: Learning procedure-grounded progress rewards for robotic manipulation. *arXiv preprint arXiv:2605.08774*, 2026. URL https://arxiv.org/abs/2605.08774.

Dongchi Huang, Hongyin Zhang, Bohan Hou, Siteng Huang, Zhian Su, Hang Guo, Tong Lu, Zhaofeng Xu, Jiahao Tang, Jianfei Yang, Donglin Wang, Peixi Peng, Mingxiu Chen, Deli Zhao, and Xin Li. RynnValue: Scaling robotic value foundation models with temporal distance. *arXiv preprint arXiv:2608.09853*, 2026. URL https://arxiv.org/abs/2608.09853.

Kuo-Han Hung, Pang-Chi Lo, Jia-Fong Yeh, Han-Yuan Hsu, Yi-Ting Chen, and Winston Hsu. Victor: Learning hierarchical vision-instruction correlation rewards for long-horizon manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 17550–17577, 2025.

Yuheng Ji, Yuyang Liu, Huajie Tan, Xuchuan Huang, Fanding Huang, Yijie Xu, Cheng Chi, Yuting Zhao, Huaihai Lyu, Peterson Co, Mingyu Cao, Qiongyu Zhang, Zhe Li, Enshen Zhou, Pengwei Wang, Zhongyuan Wang, Shanghang Zhang, and Xiaolong Zheng. PRM-as-a-Judge: A dense evaluation paradigm for fine-grained robotic auditing. *arXiv preprint arXiv:2603.21669*, 2026. URL https://arxiv.org/abs/2603.21669.

P.J. Kaufman. *Smarter Trading: Improving Performance in Changing Markets*. McGraw-Hill, 1995. ISBN 9780070340022. URL https://books.google.com/books?id=ndq_21wRJjEC.

Changyeon Kim, Minho Heo, Doohyun Lee, Honglak Lee, Jinwoo Shin, Joseph Lim, and Kimin Lee. Subtask-aware visual reward learning from segmented demonstrations. In *International Conference on Learning Representations*, volume 2025, pp. 15142–15165, 2025.

Tony Lee, Andrew Wagenmaker, Karl Pertsch, Percy Liang, Sergey Levine, and Chelsea Finn. RoboReward: General-purpose vision-language reward models for robotics. *arXiv preprint arXiv:2601.00675*, 2026. URL https://arxiv.org/abs/2601.00675.

Anthony Liang, Yigit Korkmaz, Jiahui Zhang, Minyoung Hwang, Abrar Anwar, Sidhant Kaushik, Aditya Shah, Alex S. Huang, Luke Zettlemoyer, Dieter Fox, Yu Xiang, Anqi Li, Andreea Bobu, Abhishek Gupta, Stephen Tu, Erdem Biyik, and Jesse Zhang. Robometer: Scaling general-purpose robotic reward models via trajectory comparisons. *arXiv preprint arXiv:2603.02115*, 2026. URL https://arxiv.org/abs/2603.02115.

Shengbang Liu, Yueru Jia, Yuyang Yan, Jiaming Liu, Xinran Zhang, Qiuxuan Feng, Yandong Guo, Shiji Zhou, Boxin Shi, and Shanghang Zhang. Taco: Tactile world model as a self-corrector forscalable vla post-training. *arXiv preprint arXiv:2607.02840*, 2026a.

Yuyang Liu, Chuan Wen, Yihang Hu, Dinesh Jayaraman, and Yang Gao. Timerewarder: Learning dense reward from passive videos via frame-wise temporal distance. *arXiv preprint arXiv:2509.26627*, 2025.

Yuyang Liu, Yanqing Shen, Ruike Chen, Jifan Zhao, Yuxuan Tian, Yichi Zhang, Tianfeng Long, Zixuan Yin, Yipu Wang, Ziheng Qin, Wenxing Tan, Yang Shi, Mingyu Cao, Runze Xiao, Ziqi Wang, Zhixin Yin, Shiwei Chu, Yi-Fan Zhang, Yao Mu, Yuheng Ji, Yihao Wang, Jun Yan, Zhongyuan Wang, Pengwei Wang, and Xiaolong Zheng. PRM-as-a-Judge 1.5: A toolkit for robot process assessment. *arXiv preprint arXiv:2608.14284*, 2026b. URL https://arxiv.org/abs/2608.14284.

Zhihao Liu, Qiuyi Gu, Yitao Wang, Dongming Qiao, Yixian Zhang, Shuaihang Chen, Liangzhi Shi, Tianxing Zhou, Zefang Huang, Kang Chen, et al. Steam: Self-supervised temporal ensemble advantage modeling for real-world robot learning. *arXiv preprint arXiv:2606.29834*, 2026c.

Jindi Lv, Hao Li, Jie Li, Fankun Kong, Yang Wang, Pengfei Yi, Yifei Nie, Xiaofeng Wang, Zheng Zhu, Chaojun Ni, et al. Viva: A video-generative value model for robot reinforcement learning. *arXiv preprint arXiv:2604.08168*, 2026.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning*, pp. 23301–23320. PMLR, 2023.

Yecheng Jason Ma, Joey Hejna, Chuyuan Fu, Dhruv Shah, Jacky Liang, Zhuo Xu, Sean Kirmani, Peng Xu, Danny Driess, Ted Xiao, et al. Vision language models are in-context value learners. In *International Conference on Learning Representations*, volume 2025, pp. 33984–34009, 2025.

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

Tongzhou Mu, Minghua Liu, and Hao Su. Drs: Learning reusable dense rewards for multi-stage tasks. In *International Conference on Learning Representations*, volume 2024, pp. 8470–8482, 2024.

NVIDIA Corporation. NVIDIA Isaac Sim: Robotics simulation and synthetic data generation. https://developer.nvidia.com/isaac/sim/, 2026.

Seongheon Park, Wendi Li, Changdae Oh, Samuel Yeh, Zsolt Kira, Michael Hagenow, and Sharon Li. Hide-and-seek in trajectories: Discovering failure signals for vla runtime monitoring. *arXiv preprint arXiv:2605.30834*, 2026.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubzak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kamil Stachowicz, Alex Stoecklein, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi^*_{0.6}$: A VLA that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025. URL https://arxiv.org/abs/2511.14759.

Xinyao Qin, Junjie Lu, Kaixin Wang, Chuheng Zhang, Sinjae Kang, Kimin Lee, Min Xu, Bin Liang, Jun Yang, and Li Zhao. Beyond monotonic progress: Retry-supervised value learning for robot imitation. *arXiv preprint arXiv:2606.24633*, 2026.

Huajie Tan, Sixiang Chen, Yijie Xu, Zixiao Wang, Yuheng Ji, Cheng Chi, Yaoxu Lyu, Zhongxia Zhao, Xiansheng Chen, Peterson Co, Shaoxuan Xie, Guocai Yao, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Robo-Dopamine: General process reward modeling for high-precision robotic manipulation. *arXiv preprint arXiv:2512.23703*, 2025a. URL https://arxiv.org/abs/2512.23703.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025b. URL https://arxiv.org/abs/2505.17016.

Raymond Tsao, Andrew Wagenmaker, and Sergey Levine. Learning process rewards via success visitation matching for efficient rl. *arXiv preprint arXiv:2606.23640*, 2026.

Yi Ru Wang, Carter Ung, Christopher Tan, Grant Tannert, Jiafei Duan, Josephine Li, Anh Le, Rishabh Oswal, Markus Grotz, Wilbert Pumacay, et al. Roboeval: Where robotic manipulation meets structured and scalable evaluation. *arXiv preprint arXiv:2507.00435*, 2025.

Zhihao Wang, Jianxiong Li, Yu Cui, Yuan Gao, Xianyuan Zhan, Junzhi Yu, and Xiao Ma. World value models for robotic manipulation. *arXiv preprint arXiv:2606.24742*, 2026.

Dayu Xia, Yue Shi, Yao Mu, Huiting Ji, Chaofan Ma, Yingjie Zhou, Hua Chen, Yang Liu, Jiezhang Cao, and Guangtao Zhai. RoboProcessBench: Benchmarking process-aware understanding in vision-language robotic manipulation. *arXiv preprint arXiv:2606.13040*, 2026. URL https://arxiv.org/abs/2606.13040.

Yijie Xu, Haopeng Jin, Run Zhou, Shengbang Liu, Sixiang Chen, Hongyang Cheng, Sicheng Hu, Peterson Co, Jinwen Luo, Huajie Tan, et al. Robo-Dopamine 2.0: History-conditioned and OOD-aware process reward modeling for robotic manipulation. *arXiv preprint arXiv:2608.15680*, 2026. URL https://arxiv.org/abs/2608.15680.

Hongyu Yan, Qiwei Li, Jiaolong Yang, and Yadong Mu. Progressvla: Progress-guided diffusion policy for vision-language robotic manipulation. *arXiv preprint arXiv:2603.27670*, 2026.

Zewei Ye, Weifeng Lu, Minghao Ye, Tao Lin, Shuo Yang, Junchi Yan, and Bo Zhao. Robofac: A comprehensive framework for robotic failure analysis and correction. *arXiv preprint arXiv:2505.12224*, 2025.

Silong Yong, Stephen Sheng, Carl Qi, Xiaojie Wang, Evan Sheehan, Anurag Shivaprasad, Yaqi Xie, Katia Sycara, and Yesh Dattatreya. Generalizable dense reward for long-horizon robotic tasks. *arXiv preprint arXiv:2604.00055*, 2026.

Shaopeng Zhai, Qi Zhang, Tianyi Zhang, Fuxian Huang, Haoran Zhang, Ming Zhou, Shengzhe Zhang, Litao Liu, Sixu Lin, and Jiangmiao Pang. A vision-language-action-critic model for robotic real-world reinforcement learning. *arXiv preprint arXiv:2509.15937*, 2025. URL https://arxiv.org/abs/2509.15937.

Jiahui Zhang, Yusen Luo, Abrar Anwar, Sumedh Anand Sontakke, Joseph J Lim, Jesse Thomason, Erdem Biyik, and Jesse Zhang. Rewind: Language-guided rewards teach robot policies without new demonstrations. *arXiv preprint arXiv:2505.10911*, 2025.

Jianshu Zhang, Chengxuan Qian, Haosen Sun, Haoran Lu, Dingcheng Wang, Letian Xue, and Han Liu. Progresslm: Towards progress reasoning in vision-language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11243–11271, 2026a.

Jianshu Zhang, Keliang Wu, Haoran Lu, Anbang Liu, Ce Zhang, Weijie Yin, Chengxuan Qian, Xiyuan Yang, Zhenyu Pan, Guo Ye, et al. Progress reward modeling for robotic learning: A comprehensive survey. *arXiv preprint arXiv:2607.21655*, 2026b.

Yuelin Zhang, Sijie Cheng, Chen Li, Zongzhao Li, Yuxin Huang, Yang Liu, and Wenbing Huang. Recurrent reasoning with vision-language models for estimating long-horizon embodied task progress. *arXiv preprint arXiv:2603.17312*, 2026c.