Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

ROBOMETER: Scaling General-Purpose Robotic Reward Models via Trajectory Comparisons

Anthony Liang*[1], Yigit Korkmaz*[1], Jiahui Zhang[2], Minyoung Hwang[3], Abrar Anwar[1], Sidhant Kaushik[4]
Aditya Shah[5], Alex S. Huang[2], Luke Zettlemoyer[5], Dieter Fox[5,6], Yu Xiang[2], Anqi Li[7]
Andreea Bobu[3], Abhishek Gupta[5], Stephen Tu†[1], Erdem Bıyık†[1], Jesse Zhang†[5]

[1]Univ. of Southern California [2]UT Dallas [3]MIT [4]Indep. Researcher [5]Univ. of Washington [6]Ai2 [7]NVIDIA
*Equal contribution †Equal advising

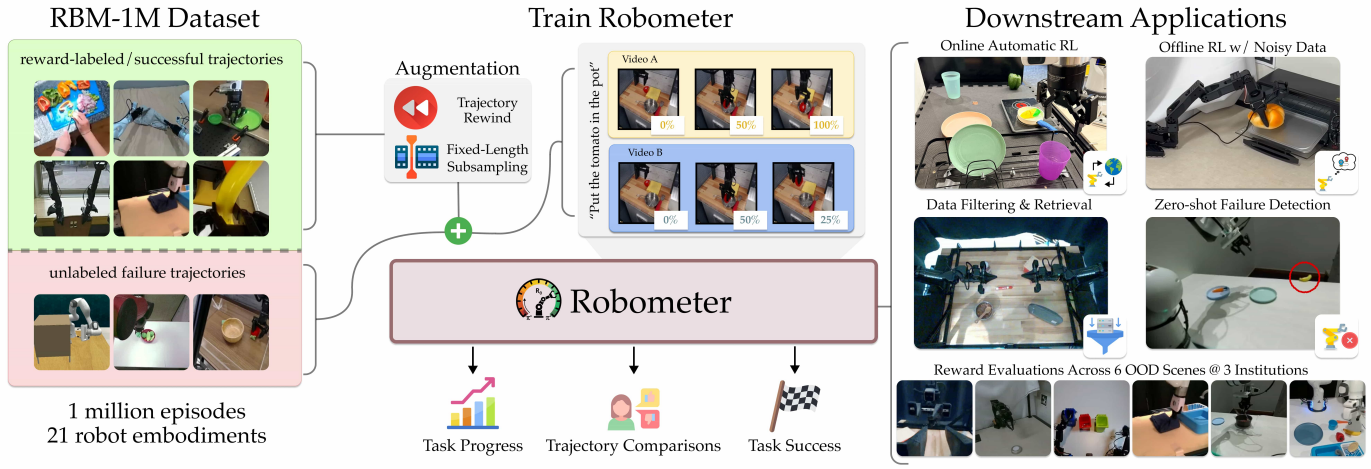


Fig. 1: **ROBOMETER Overview.** ROBOMETER is trained on RBM-1M, a 1M-trajectory dataset spanning 21 robot embodiments, containing both reward-labeled/expert trajectories and reward-unlabeled, failed trajectories. The model is supervised with a dual objective: predicting frame-level task progress (reward) and learning trajectory-level preferences from pairwise comparisons. To help with downstream RL, it is also trained to predict per-frame task success. This training recipe enables scalable reward learning, is validated on reward model evaluations from 6 out-of-distribution scenes collected at 3 institutions, and supports diverse downstream applications such as offline & online RL, imitation learning data filtering and retrieval, and automated failure detection.

***Abstract*—Current general-purpose robot reward models rely on frame-level progress labels from expert demonstrations. This approach scales poorly to large datasets, where suboptimal or failed trajectories are abundant and absolute progress is ambiguous. To address this, we introduce ROBOMETER, a scalable reward modeling framework combining intra-trajectory progress supervision with inter-trajectory preference supervision. ROBOMETER utilizes a dual objective: a frame-level loss that anchors reward magnitude to expert data, and a trajectory-comparison preference loss that imposes global ordering constraints. This enables effective learning from both successful and failed trajectories. To support this formulation at scale, we curate RBM-1M, a dataset of over one million multi-embodiment trajectories containing extensive suboptimal and failure data. Across benchmarks and real-world evaluations, ROBOMETER learns highly generalizable reward functions and improves downstream robot learning performance. Code, model weights, and videos at https://robometer.github.io/.**

I. INTRODUCTION

In human cognition, comparative judgments are a core mechanism for internalizing calibrated scales [1, 2, 3], enabling reasoning about relative progress and outcomes rather than isolated states. Analogously, the supervision signals used to train robotic reward models determine how well they internalize notions of task progress, enabling downstream applications such as online reinforcement learning (RL) [4, 5], imitation learning (IL) from noisy data [6, 7], automated failure detection [8], and offline RL [9]. Current general-purpose robot manipulation reward models rely exclusively on absolute progress labels derived from expert or reward-labeled demonstrations, providing pointwise, *trajectory-local* supervision [5, 10, 11, 12].

Such labels are easy to obtain for expert trajectories—for example, by linearly interpolating progress from 0 to 1—but become ill-defined and costly to annotate for failed attempts, where progress may fluctuate over time. As a result, large amounts of suboptimal data—ubiquitous in real-world robot learning—cannot be effectively leveraged [13]. This reliance on trajectory-*local* progress supervision limits both scalability and generalization. In this work, we address this limitation by training reward models with an additional *global* supervision signal that improves generalization across embodiments, scenes, and varying trajectory quality.

Our key insight is that *preference prediction* over trajectory pairs provides complementary supervision. While progress labels anchor reward values along individual trajectories, pairwise comparisons impose ordering constraints across diverse trajectories, tasks, robots, and viewpoints. This formulation en-

ables learning from previously unusable suboptimal data by requiring only relative comparisons—curated without additional human annotation—rather than absolute scores. Specifically, trajectory comparison supervision (1) enforces consistent ordering across trajectories, providing global grounding beyond individual rollouts, and (2) scales naturally to unlabeled failed trajectories where absolute progress is ambiguous, resulting in better-calibrated rewards.

To instantiate this insight, we propose a scalable training recipe for reward modeling based on a dual reward-prediction objective: a frame-level progress loss on expert demonstrations and a preference-prediction loss over trajectory comparisons. Using this recipe, we train ROBOMETER, a manipulation-centric reward model. Our training recipe reveals a strong mutual reinforcement effect between the two supervision signals. Even when trained only on expert demonstrations, preference supervision improves ROBOMETER's ability to distinguish successful from suboptimal trajectories, suggesting that global comparative constraints induce a more structured internal reward representation. Moreover, as additional unlabeled suboptimal data is introduced, ROBOMETER naturally scales to further improve performance.

To train ROBOMETER, we curate `RBM-1M`, a large-scale reward-learning dataset for robot manipulation containing over one million trajectories collected from 21 robot platforms, including single-arm, bimanual, and mobile manipulators, as well as human demonstrations (see Figure 1). Importantly, `RBM-1M` intentionally includes a substantial number of suboptimal and failed trajectories that naturally arise during real-world data collection but are difficult to leverage with purely absolute progress-based supervision. The scale and diversity of `RBM-1M` are therefore critical for learning globally consistent preference relations across embodiments, tasks, and viewpoints while fully exploiting failure data that would otherwise be discarded. Beyond real-world failures, we further construct preference pairs using a suite of augmentations—including video rewinding [5] and cross-task comparisons—that expose the model to diverse successful and suboptimal behaviors. Across external benchmarks and our own evaluation trajectories collected from six out-of-distribution scenes spanning three institutions, ROBOMETER outperforms state-of-the-art baselines by an average of 14% in reward rank correlation and achieves a 32% relative improvement in distinguishing successful from suboptimal trajectories.

Finally, we show that ROBOMETER outperforms relevant baselines in real-world robot learning applications across a diverse set of downstream applications: (1) automatic online RL, (2) offline RL with noisy and expert trajectories, (3) dataset filtering for imitation learning, and (4) zero-shot failure detection across multiple robot embodiments and institutions. Overall, policy learning experiments with ROBOMETER demonstrate $\mathbf{2.4 - 4.5}\times$ higher success rates than the best baseline in each category. We publicly release ROBOMETER, `RBM-1M`, and the training code at https://robometer.github.io, enabling the community to train their own models using our recipe or to directly use ROBOMETER for robot manipulation applications.

II. RELATED WORK

Learning reward functions is a central problem in reinforcement learning and robotics, and prior work has explored a wide range of approaches that differ in both the form of supervision and the scope of generalization.

Reward Learning from Demonstrations. Learning reward functions from human supervision is a long-studied topic in inverse RL (IRL), where reward functions are inferred from human demonstrations [14, 15, 16, 17, 18] or from expert and goal-state distributions [19, 20, 21, 22, 23, 24]. However, most IRL methods require task-specific expert demonstrations, necessitating the collection of new demonstrations whenever the task or reward specification changes. In contrast, we propose a framework for training reward models that can generalize effectively to new tasks.

Preference-Based Reward Learning. Prior work in psychology has observed that humans often use relative comparisons over absolute numerical scales when making judgments [1, 2, 3]. This insight has been central to modern reinforcement learning from human feedback (RLHF), where preference supervision from humans is used to learn reward models that are only identifiable up to monotone transformations, yet sufficient for effective policy optimization [25, 26, 27, 28, 29, 30, 31, 32, 33]. In contrast to these works, which treat preferences as the primary supervision signal to train domain-specific reward models, we generate preference supervision from both synthetic and real trajectories for which assigning progress labels is difficult. We use this preference signal as an *auxiliary* objective that complements direct progress prediction, enabling ROBOMETER to learn from large, heterogeneous datasets without additional human preference supervision. Closest in spirit, Kwok et al. [34] generate synthetic preference labels from action mean-squared-error to train a robot action verifier, and Venkataraman et al. [9], Wang et al. [35] use vision-language models (VLMs) to generate frame-level preferences as the primary supervision signal for task-specific reward models. In contrast, we generate preference supervision by comparing entire trajectories and use it as an auxiliary signal to learn a general-purpose reward function across tasks and embodiments.

Rewards from Foundation Models. Finally, recent work has sought to construct more general reward functions that operate directly on images, videos, and language. Large language models (LLMs) and VLMs have been applied to reward design, for example by generating executable reward code or shaping functions from natural language descriptions [36, 37, 38], directly producing reward functions [39], or guiding reward computation via language-conditioned state masking [40]. However, most of these approaches assume access to privileged state information, which is often unavailable or difficult to obtain in real-world robotic deployment settings.

To overcome this challenge, some approaches use task progress as a proxy reward, either by applying pre-trained VLMs as zero-shot progress or success estimators [6, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50], or by training domain-specific

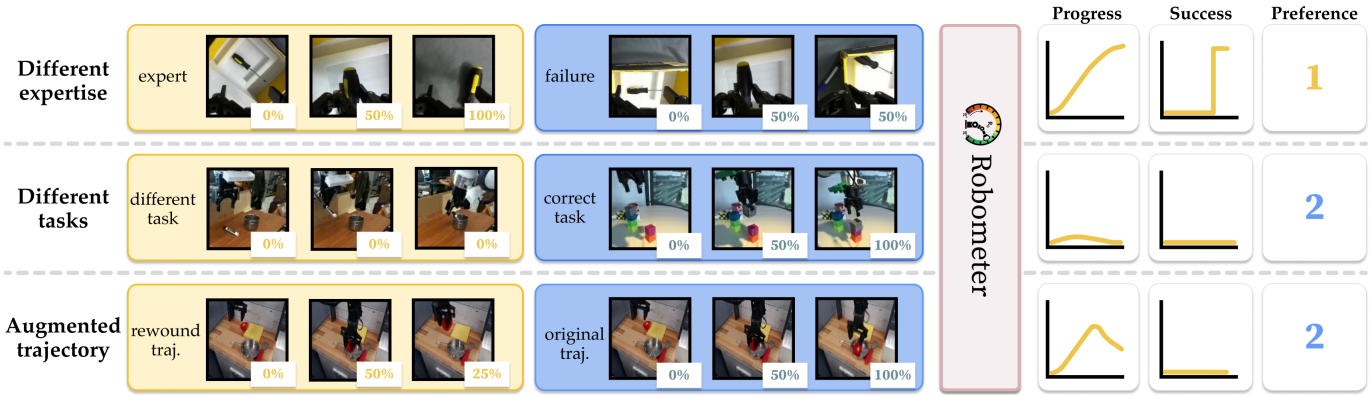


Fig. 2: **ROBOMETER** is a VLM-based reward model that predicts dense, per-frame progress-based rewards and success labels for the first of two video trajectories. To scalably train with failure data, we also predict which of the two video trajectories better completes the task (preference). We use three strategies for curating training examples from our given datasets, which are further detailed in Section III-D with model architecture shown in Appendix Figure 10. At inference, the model can input either up to two trajectories: with a single trajectory, it outputs per-frame progress & success predictions; with two, it additionally predicts a preference for which trajectory best performs the task.

models with progress-prediction objectives [4, 5, 7, 41, 51, 52, 53, 54, 55, 56]. However, directly using pre-trained VLMs for zero-shot video-language reward prediction often yields noisy or inconsistent signals [5, 11, 57], while smaller per-task models tend to overfit to domain-specific visual and semantic features, limiting generalization.

Recent methods address these limitations by fine-tuning VLMs or VLAs, enabling progress prediction to leverage visual–semantic representations learned from diverse data [10, 11, 12, 58, 59, 60] and demonstrating improved sample efficiency for RL. VLAC [10], for example, co-trains a VLA with relative progress difference targets, while $\pi^*_{0.6}$ [59] and Ghasemipour et al. [58] train distance-to-goal value functions that encode task progress. RoboReward [11] fine-tunes a VLM to predict discretized (1-5) progress labels generated via counterfactual instruction labeling by closed-source VLMs, and RoboDopamine [12] similarly fine-tunes a VLM for progress prediction but requires a goal image in addition to the language instruction and task-specific one-shot fine-tuning. In contrast, we introduce a more scalable training framework which uses an explicit auxiliary *preference prediction* objective, enabling us to scale to datasets containing suboptimal or failed trajectories without reward labels. Using such failed trajectories, even as just in-context learning examples for pre-trained VLMs, has been demonstrated to help reward models align more closely with human-specified rewards [13].

III. SCALABLE REWARD MODEL TRAINING

We propose a *scalable recipe* for training reward models, along with ROBOMETER, an instantiation of this recipe that provides dense rewards for robot manipulation. Our approach rests on three pillars: a diverse dataset which includes unlabeled failure trajectories, a pre-trained VLM backbone for generalization, and a training objective that combines **per-frame rewards** with **trajectory preferences**.

A. RBM-1M Dataset

Notation. We define the dataset $\mathcal{D} = \{\tau_i\}$ of trajectories, where each $\tau = \{o_{1:T}, l, p\}$ contains image observations o, a language instruction l, and a scalar progress label $p \in \{\text{None}, [0, 1]\}$ corresponding to the progress at the end of the trajectory. For expert demos, $p = 1.0$; for datasets with partial progress labels (e.g., RoboArena [61]), we use the provided score. For unlabeled failed trajectories, we set $p = \texttt{None}$.

Data Composition. Rather than maximizing trajectory quantity, RBM-1M focuses on viewpoint, scene, and embodiment diversity. We aggregate 1 million trajectories from: (1) **Expert robot data** from diverse, multi-robot sources such as Open-X [62] and subsets of high-quality, single-robot datasets such as AGIBotWorld [63]; (2) **Human videos** from datasets such as Epic-Kitchens [64] for scene diversity or human-robot paired datasets like RH20T [65] to promote embodiment-invariant representations; (3) **Simulation** data from sources like LIBERO [66]; and (4) **Failed trajectories** from automated policy rollouts [67] and failure-detection datasets [68].

Our dataset overall includes 21 robot embodiments and over 1 million trajectories, hence RBM-1M. We also construct two evaluation datasets, RBM-EVAL-ID and RBM-EVAL-OOD, detailed in Section A-3. For further details on dataset filtering and aggregation, see Appendix A. We also list all dataset categories and trajectory counts in Appendix Figure 9, Table VIII.

B. ROBOMETER Architecture and Tokenization

ROBOMETER buids on a causally masked VLM, QWEN3-VL-4B-INSTRUCT, to process either one video (for reward inference) or a pair of videos (for preference training).

Hidden Embedding Extraction. To extract rewards without disrupting the VLM's pre-trained internal representations, we insert new, learned tokens into the sequence. We interleave **progress tokens** ($\langle|\text{prog\_token}|\rangle$) within the first video sequence and a single **preference token** ($\langle|\text{pref\_token}|\rangle$) at the end of the multi-video prompt:

$$\begin{aligned}\text{Tok}(l, o^1, o^2) \rightarrow \text{Tok}(l)\langle|\text{video\_start}|\rangle \left[\text{Tok}(o^1_t)\langle|\text{prog\_token}|\rangle\right]_{t=1}^{T} \\ \langle|\text{split\_token}|\rangle \left[\text{Tok}(o^2_t)\right]_{t=1}^{T} \langle|\text{pref\_token}|\rangle,\end{aligned} \quad (1)$$

where $\langle|\text{video\_start}|\rangle$ is the model's default image-start delimiter and $\langle|\text{split\_token}|\rangle$ is a separator. The causal mask ensures

that $\langle|\text{prog\_token}|\rangle$ tokens attend only to current/previous frames of o^1, producing dense, frame-level progress estimates for online reward inference, while $\langle|\text{pref\_token}|\rangle$ attends to both trajectories to make a relative judgment. We fix both trajectories to length $T = 8$ to avoid preference predictions that rely on trajectory length as a proxy for quality. Progress tokens are inserted only for o^1 since at inference time, progress is predicted for a single trajectory; furthermore, if we insert progress tokens between o^2 frames, they would attend to o^1.

C. Training Objectives

We optimize ROBOMETER using a composite loss: $\mathcal{L} = \mathcal{L}_{\text{pref}} + \mathcal{L}_{\text{prog}} + \mathcal{L}_{\text{succ}}$. This allows the model to anchor rewards to absolute progress while learning to distinguish subtle quality differences through trajectory comparisons across the dataset.

Preference Prediction. We train a binary classifier, MLP_{pref}, on the hidden state $h_{\langle|\text{pref\_token}|\rangle}$ of the $\langle|\text{pref\_token}|\rangle$ to predict which trajectory better satisfies l:

$$\mathcal{L}_{\text{pref}} = -\Big[\mathbb{I}_{y=1}\ \log\sigma\big(\text{MLP}_{\text{pref}}(h_{\langle|\text{pref\_token}|\rangle})\big) + \mathbb{I}_{y=2}\ \log\big(1-\sigma\big(\text{MLP}_{\text{pref}}(h_{\langle|\text{pref\_token}|\rangle})\big)\big)\Big], \tag{2}$$

where y is the ground-truth preferred trajectory.

Progress and Success. For the first trajectory o^1, we attach an MLP head to each $h_{\langle|\text{prog\_token}|\rangle,t}$ to predict continuous progress p_t and binary success s_t. Similar to prior work [5, 7, 10], we define per-frame progress targets $p_{1:T}$ for expert demonstration data, where the final target progress $p = 1$. To better model multi-modal reward/progress distributions than continuous progress prediction, we discretize progress into $N = 10$ uniformly spaced bins over $[0, 1]$ and model progress prediction as a categorical distribution [59, 69, 70]. For a trajectory of length T, the ground-truth continuous progress target at frame t is defined as $p_t = t/T$ for $t \in \{1, \dots, T\}$. This scalar target is projected onto a categorical distribution over N bins using linear interpolation between the 2 neighboring bin centers. The progress head $\text{MLP}_{\text{progress}}$ outputs a categorical distribution $\hat{p}_t \in \Delta^N$, and the progress loss is computed using cross-entropy:

$$\mathcal{L}_{\text{prog}} = \frac{1}{T}\sum_{t=1}^{T}\text{CE}\big(\text{Project}(p_t),\ \text{MLP}_{\text{progress}}(h_{\langle|\text{prog\_token}|\rangle,t})\big)\,.$$

At inference time, a continuous progress estimate is recovered by taking the expectation over the bin centers, $\hat{p}_t = \sum_{i=1}^{N} z_i\,\hat{p}_{t,i}$, where $\{z_i\}_{i=1}^{N}$ denote the fixed bin centers. Per-frame success targets are defined such that $s_t = 0$ for $t < T$ and $s_t = 1$ for $t = T$. In some datasets, the human operator stops recording a trajectory several frames after the task has already been completed; to improve the accuracy of both progress and success prediction, we sample 10 trajectories per data source to determine the point at which the task actually ends (Appendix Section A-1). We train success prediction with binary cross-entropy on s, with balanced class weights adjusted per-batch to account for negative sample imbalance:

$$\mathcal{L}_{\text{succ}} = \text{BalancedBCE}(s_{1:T}, [\text{MLP}_{\text{success}}(h_{\langle|\text{prog\_token}|\rangle,t})]_{1:T}).$$

D. Data Sampling and Augmentation

Given these losses, the ideal training regime for ROBOMETER would rely on large-scale, preference-labeled robot trajectory datasets containing explicit progress-labeled failures. Such failures are particularly important because, at deployment time, reward models are frequently queried on out-of-distribution trajectories—e.g., failures induced by online RL exploration, compounding execution errors, or noisy data collection—that deviate substantially from the training distribution. In practice, however, preference annotations over robot trajectories are limited, and dense per-frame progress labels for failed executions are expensive and difficult to obtain. We address this limitation by constructing training inputs (l, o^1, o^2) and targets y dynamically from `RBM-1M` using three complementary strategies displayed in Figure 2:

1) **Progress-Based Comparisons (Different Expertise).** To teach the model to distinguish execution quality, we sample two trajectories τ_1, τ_2 sharing an instruction l but differing in outcome (e.g., an expert demonstration p=1 vs. an unlabeled failure p=`None`) or progress ($p^1 \neq p^2$). We set the preference target y=1 if $p^1 > p^2$ (or if τ_1 is the expert), and $y = 2$ otherwise. This allows ROBOMETER to leverage unlabeled failures by contrasting them against successful demonstrations.
2) **Instruction Negatives (Different Tasks).** To ensure rewards are grounded in the language command, we sample τ_1 and τ_2 with distinct instructions $l^1 \neq l^2$. We randomly select one instruction as the conditioning text l, set the preference label y to the trajectory corresponding to the selected instruction, and set the progress target p_t=0 for the other, enforcing that correct behavior for the wrong task yields no reward.
3) **Video Rewind (Augmented Failures).** To explicitly model "undoing" progress—a common failure mode in RL—we generate synthetic preferences from a single expert trajectory τ by reversing a segment of time. Prior work denotes this type of augmentation as *video rewind* [5, 7, 71]. We sample indices $1 \leq t_1 < t_2 < t_3 \leq T$ to form a *Chosen* forward sequence $o^c = o_{t_1:t_3}$ and a *Rejected* rewound sequence, either $o^r = [o_{t_1:t_3}, o_{t_3-1:t_2}]$ or $[o_{t_3:t_1}]$. The Chosen and Rejected sequences are randomly assigned to o^1 and o^2 to construct the preference label. While o^c targets linear forward progress, we explicitly penalize the reversal in o^r by assigning decreasing progress targets matched to their frame indices.

Fixed-length Subsampling. To avoid biasing toward longer trajectories, we construct fixed-length inputs by randomly selecting start and end indices in the trajectory, and uniformly sampling T frames between them. [1]

Summary. ROBOMETER builds upon a base VLM, QWEN3-VL-4B-INSTRUCT, and inserts additional learnable tokens to predict preference, progress, and success. We

[1] ROBOMETER is still a *dense*, per-frame reward model: during inference, we query over $o_{1:t}\ \forall t \in [1, ..., T]$ for a trajectory of length T to produce T rewards. Our subsampling always includes the first and current frame.

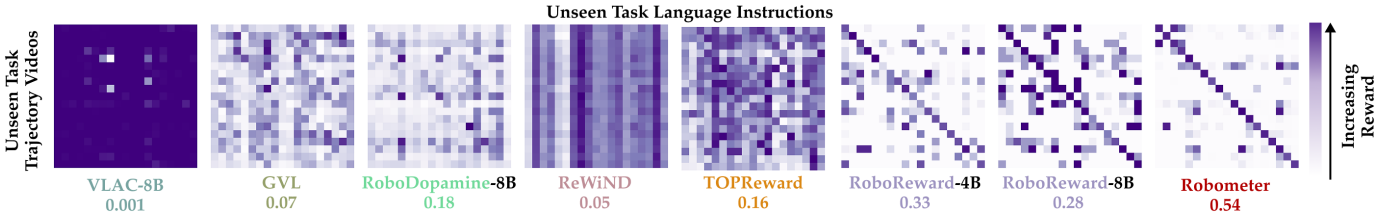


Fig. 3: **Video-Language Reward Confusion Matrix.** For each task sampled at random from *self-collected, unseen* data from RBM-EVAL-OOD, we compute rewards for all combinations of demonstration videos and language descriptions. ROBOMETER produces the most diagonal-heavy confusion matrix, indicating strong alignment between unseen demos and instructions. We also report the column-normalized diagonal mean under each model, which represents the fraction of the model's total reward for aligned task and video pairs.

		w/ Varied Training Data				w/ RoboReward Training Data			w/ our RBM-1M data	
		GVL	VLAC	TOPReward	RoboDopamine-8B	RoboReward-4B	RoboReward-8B	ROBOMETER	ReWiND	ROBOMETER
(a) VOC $r \uparrow$	RBM-EVAL-ID	0.16	0.16	0.42	0.79	0.77	0.82	0.84	0.46	**0.92**
	RBM-EVAL-OOD	0.21	0.17	0.40	0.80	0.88	0.88	0.93	0.51	**0.94**
(b) Kendall $\tau_a \uparrow$	RBM-EVAL-OOD	0.19	0.08	0.13	0.45	0.50	0.47	0.55	0.01	**0.64**

TABLE I: (a) Reward alignment (VOC Pearson r) and (b) trajectory ranking (Kendall τ_a) on RBM-EVAL datasets. Baselines are split into categories based on training data: GVL and TOPReward training data is unknown, VLAC is trained on a 300k-trajectory dataset, and RoboDopamine is trained on a 100k-trajectory dataset. We compare ROBOMETER against RoboReward-4B/8B with their own training data, and we also evaluate ReWiND and ROBOMETER trained with the full RBM-1M dataset. Kendall τ_a is not calculated for RBM-EVAL-ID due to it only having simulation failure data.

train ROBOMETER on RBM-1M, a dataset consisting of both progress-labeled and progress-unlabeled data, by sampling trajectory comparisons across the entire dataset. These comparisons come from failure trajectories, comparisons across different tasks, and rewound videos. At inference, ROBOMETER can operate on either one or two trajectories: with a single trajectory, it outputs per-frame progress and success predictions; with two trajectories, it additionally predicts a preference while still producing per-frame predictions for the first trajectory. For additional model implementation and training details, see Appendix B.

IV. EXPERIMENTS

Our experiments aim to study ROBOMETER's effectiveness in producing rewards for robot learning. Specifically, we organize our experiments to answer the following questions:
(**Q1**) **Reward Evaluation**: How well do ROBOMETER rewards reflect task progress on *unseen* tasks and embodiments?
(**Q2**) **Ablation + Analysis**: How much does each component of ROBOMETER contribute to reward performance?
(**Q3**) **Policy Learning**: How does ROBOMETER compare against baselines in enabling downstream robot learning?

Baselines. We compare ROBOMETER against the strongest set of video-language input, zero-shot-capable, and open-sourced or API-accessible reward baselines described in Section II. We list a dataset size comparison table for baselines and related papers in Appendix Table V.

- **VLAC-8B** [10] trains a VLA that predicts actions and rewards on a dataset of 300k human and robot trajectories. We compare against their larger 8B parameter checkpoint.
- **GVL** [6] prompts a pre-trained closed-source LLM with shuffled video frames to predict task progress for subsampled frames across the video sequence. We use GPT-5 mini as it is the best-performing closed-source model on the RoboRewardBench reward evaluation benchmark [11].
- **TOPReward** [50] prompts a pre-trained VLM (Qwen-3-VL 8B) with "does the trajectory complete the task?" and uses the log-likelihood of the "true" token as the reward.
- **ReWiND** [5] trains a small transformer-based network with a direct progress prediction objective along with video rewinding to simulate failed policy rollouts. We train ReWiND with RBM-1M to maximize its zero-shot capability.
- **RoboDopamine-8B** [12] fine-tunes a VLM for reward prediction via "frame hops" comparing forward and rewound frames. While designed for reward prediction conditioned on a *goal image* and instruction, we evaluate it in our zero-shot setting without a goal image for fair comparison. We use the latest, 8B parameter, RoboDopamine-2.0 checkpoint.
- **RoboReward-4B/8B** [11]: Fine-tunes a Qwen-3-VL 4B/8B VLM for discrete (1-5) progress prediction on a dataset consisting of data from OXE [62] and RoboArena evaluations [61]. Generates *counterfactual* instructions via closed-source VLMs to simulate failed trajectories.

Custom Evaluation Datasets. We train ROBOMETER, and certain baselines when applicable, on the aforementioned RBM-1M dataset. Prior large-scale reward modeling baselines mainly evaluate on validation or test set versions of the datasets they train on [5, 10, 11, 12], which contain in-distribution arms, camera angles, or scenes. Instead, we collect our own evaluation dataset, RBM-EVAL-OOD, consisting of 976 trajectories collected from 3 academic institutions that are guaranteed not to be in the training data of *any* baseline, consisting of 6 embodiments (including human hands), 3 of which are not in RBM-1M, collected across diverse camera angles. We also aggregate an in-distribution test split of unseen trajectories collected from datasets in RBM-1M, denoted RBM-EVAL-ID. See more evaluation dataset details in Section A-3.

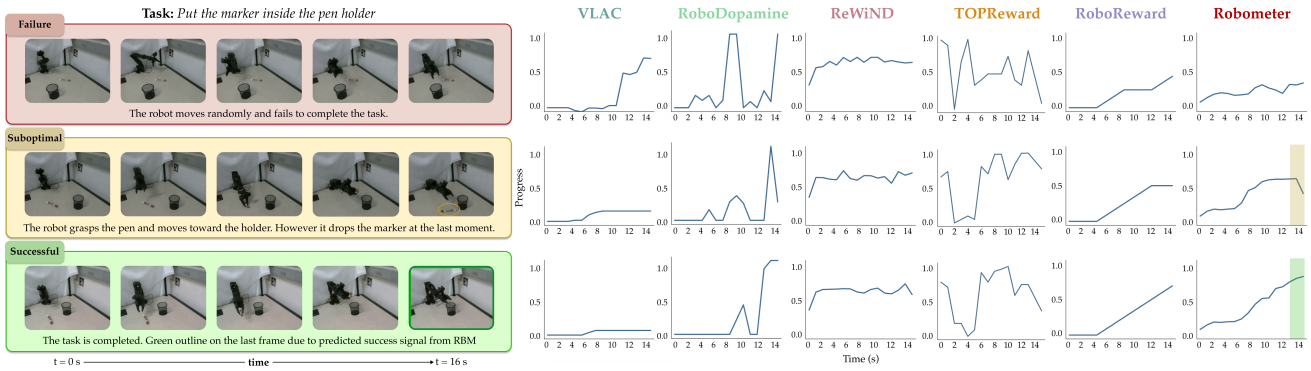


Fig. 4: **Qualitative Analysis of Failure, Suboptimal and Successful Trajectories.** We visualize the progress predictions for three trajectories of different quality for the same task. Notably, for the suboptimal trajectory, ROBOMETER predicts steadily increasing progress as the robot approaches the pen holder, but sharply reduces its progress estimate when the marker is dropped, correctly reflecting regression in task completion. In contrast, RoboReward continues to assign high progress despite the task failure. Finally, ROBOMETER is the only model that correctly predicts task success for the successful trajectory (i.e., high final progress value and explicit success prediction).

Q1: Reward Evaluation

As detailed in Section III-B, our focus is on training a reward model which: (1) generalizes to new tasks, embodiments, and domains while (2) providing reward feedback useful for *policy learning*. We structure this subsection to highlight ROBOMETER's strong performance across both criteria.

Trajectory Task Alignment. Our first main result demonstrates that ROBOMETER accurately distinguishes between different tasks in `RBM-EVAL-OOD`, which directly reflects its ability to assign rewards that align with task semantics, even across unseen robot embodiments, camera viewpoints, and scenes. We plot confusion matrices comparing unseen, successful trajectory videos versus their language instructions in Figure 3. Ideally, a **purple diagonal** indicates correct video-instruction pairs, with low (white) values elsewhere. ROBOMETER clearly produces the strongest disparity between the diagonal and off-diagonal elements, highlighting its superior ability to reward a robot for performing the *correct* task, which is especially important in cluttered, multi-task settings. This ability is in part due to how we sample *different-task* negative preference and progress examples across the entire dataset (cf. Section III-D).

Reward Alignment. Quantitatively, we evaluate the ability of baselines to predict increasing progress for *successful* robot videos from both `RBM-EVAL-OOD` and `RBM-EVAL-ID` in Table I(a). We report Value Order Correlation (VOC) [6] $\in [-1, 1]$, which calculates the Pearson correlation of predicted rewards for each trajectory video frame against their ground-truth timestep value. Overall, ROBOMETER performs the best across the board on both test sets, especially on `RBM-EVAL-OOD`. We break down per-dataset and per-subset results for both test sets in Appendix C, and evaluate on an external RoboRewardBench [11] benchmark in Section C-3.

Relative Trajectory Rankings for Mixed Expertise Data. Next, we quantitatively demonstrate that ROBOMETER is more effective than baselines at providing rewards useful for *policy learning*. For a robot policy to learn with rewards, the rewards should not only be high when performing the correct task, but also be low for incorrect execution. We measure this using the Kendall-τ_a coefficient [72], an ordering metric $\in [-1, 1]$ robust to ties. We calculate the alignment between model-assigned final rewards and the ground-truth ordering between failed, suboptimal, and successful trajectories for the same task. A higher τ_a value demonstrates that the reward model more accurately distinguishes between levels of policy performance and thus can provide proper reward signals to the policy for both low- and high-quality behaviors.

We report results in Table I(b). On `RBM-EVAL-OOD`, ROBOMETER achieves a Kendall-τ_a of 0.66, substantially outperforming RoboReward-4B (0.50) and RoboReward-8B (0.47), indicating that ROBOMETER more reliably recovers the correct ordering among failed, suboptimal, and successful trajectories. Notably, even when trained on the same data as RoboReward, ROBOMETER outperforms RoboReward in both policy ranking and reward alignment, highlighting the effectiveness of our data augmentation strategies.

To further illustrate this behavior, we visualize reward predictions over time for failed, suboptimal, and successful trajectories in Figure 4. ROBOMETER exhibits sharper separation in rewards between different levels of execution and accurately reflects regression in task progress. Additional results in Appendix C evaluating preference prediction accuracy.

Reward Fine-tuning. Finally, we demonstrate that ROBOMETER serves as a good initialization for domain-specific fine-tuning. We fine-tune on RoboFAC [73], a dataset of robotic failures and corrections spanning 16 tasks and 53 scenes (11k trajectories), including both simulated and real-world successes and failures. We adapt ROBOMETER via LoRA [74] adapters and full fine-tuning (FFT). We compare against fine-tuning the base VLM `Qwen/Qwen3-VL-4B-Instruct` in the same way.

In addition to the previous VOC r and Ranking Kendall τ_a, we also compare a *success - fail* metric measuring the difference in final reward between successful and failed trajectories of the same task. Qwen3 fine-tuned still performs worse than ROBOMETER zero-shot on 2 out of 3 metrics; fine-tuning from ROBOMETER yields substantially better reward evaluation results than training Qwen3-VL from scratch across

Method	VOC r ↑	Kendall τ ↑	Succ-Fail Diff ↑
ROBOMETER-4B (Zero-shot)	0.652	0.436	0.141
Qwen3-VL (LoRA)	0.701	0.067	0.005
Qwen3-VL (Full FT)	0.727	0.102	0.008
ROBOMETER-4B (LoRA)	0.875	0.786	0.271
ROBOMETER-4B (FFT)	**0.884**	**0.802**	**0.302**

TABLE II: **Finetuning ROBOMETER on RoboFAC dataset.** Zero-shot performance is strong, and fine-tuning the base Qwen3 VLM performs worse on Kendall τ and success - fail difference compared to zero-shot ROBOMETER. Meanwhile, ROBOMETER fine-tuned performs best, either with LoRA or FFT.

	(a) LIBERO-90			(b) RBM-EVAL-OOD		
Ablation	VOC r	Kendall τ	Suc − Fail	VOC r	Kendall τ	Suc − Fail
H1 **Prog. Only**	0.96	0.63	0.11	0.93	0.31	0.08
H1 **+Preference**	0.90	0.74	0.22	**0.95**	0.54	0.24
H2 **+Failed Data**	**0.98**	**0.92**	**0.46**	0.94	**0.64**	**0.32**
H3 **ReWiND Arch.**	0.48	-0.14	-0.02	0.51	0.01	0.02

TABLE III: Reward alignment (VOC Pearson r), policy ranking (Kendall τ), and average reward difference between successful and failed trajectories on LIBERO-90 and RBM-EVAL-OOD.

all of our ranking metrics (Table II). Importantly, LoRA and FFT perform similarly, demonstrating that ROBOMETER can be effectively fine-tuned with just 1 GPU. See further experiment details in Appendix F.

Overall, our results demonstrate that **ROBOMETER outperforms baselines** in both **generalization** and **distinguishing** successful / failed trajectories, and that it also serves as a **strong initialization for fine-tuning**. We next analyze *why*.

Q2: Ablations: Why does ROBOMETER *Perform so Well?*

Here, we investigate individual components of ROBOMETER to evaluate specific hypotheses about reward model training and its effects on downstream RL performance.

- **H1** Predicting **preferences** (Equation (2)), even without paired failure trajectories, improves reward performance.
- **H2** Scaling preference prediction with additional **failure data** leads to improved reward model performance.
- **H3** Fine-tuning from **pre-trained VLMs** helps with reward predictions on unseen tasks.

Our main analysis is performed via a controlled setting with data from the LIBERO [66] robot manipulation simulated benchmark. We train models with 1,709 successful demos from LIBERO-{10, Object, Goal, Spatial} and evaluate performance on a sample of the 8,262 unseen, paired, successful and failed trajectories from LIBERO-90.

Reward Model Ablations. To measure **H1**, we train ROBOMETER with *only* progress prediction and also ROBOMETER with both progress and preference prediction on the 1,709 demo dataset containing no failed trajectories. We then measure **H2**—scaling with failure data for preference training—by adding in 1,929 generated, failed LIBERO trajectories and train ROBOMETER with the full ROBOMETER training objective of progress and preference prediction on the larger dataset. These LIBERO ablations are trained with LoRA [74] due to the small dataset size. Finally, we verify the importance of a pre-trained VLM (**H3**) by training a larger,

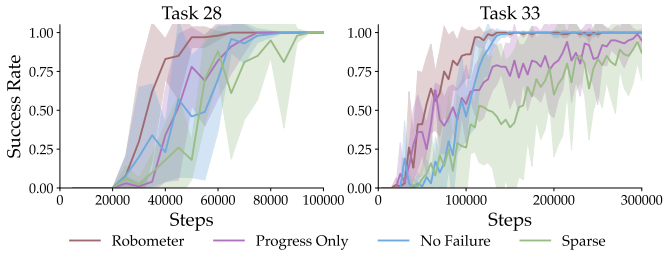


Fig. 5: **RL w/ Ablation Models** in LIBERO-90 tasks from scratch, corresponding to ablations trained only on LIBERO-10/Object/Goal/Spatial data from Table III. We report the average success rate ± standard deviation across 5 seeds.

500M-parameter version of ReWiND's transformer model (originally designed for low-data regimes) with our preference and progress objectives on the paired-failure LIBERO dataset.

We depict results on LIBERO, and separately, trained on the full RBM-1M and evaluated on RBM-EVAL-OOD (with failed data removed for +Preference and +Failed Data), in Table III. First, comparing **H1 Prog. Only** to **H1 +Preference**, adding preference supervision consistently improves policy ranking performance, increasing Kendall-τ_a on both LIBERO-90 and RBM-EVAL-OOD. Second, **incorporating failed trajectories** for preference training (**H2 +Failed Data**) yields the **largest gains** across all ranking-based metrics. On LIBERO-90, Kendall-τ improves to 0.92 and the average difference in final rewards between success–failure increases more than 4× relative to progress-only training. Similar trends hold on RBM-EVAL-OOD, where Kendall-τ and success–failure separation improve significantly. Finally, replacing the pretrained VLM backbone with a scaled variant of ReWiND's architecture (**H3 ReWiND Arch.**) results in a severe degradation across all metrics, confirming that large-scale multimodal pretrained backbone is essential for learning generalizable reward representations. See Appendix D for additional ablations.

Ablation Results on RL Performance. Before moving to our full policy learning experiments, we verify that the reward evaluation trends observed in our ablation studies hold for policy learning. We train policies via online RL using ablated reward models on two tasks from the unseen LIBERO-90 suite. These tasks were selected because sparse-reward RL eventually achieves a near-100% success rate, allowing us to directly compare *sample efficiency* rather than success rates.

Results in Figure 5 demonstrate that improvements in reward evaluation metrics consistently transfer to policy success rates. For both tasks, policies trained using ROBOMETER (the **H2** LIBERO model) demonstrate better sample efficiency than ablations (**H1** LIBERO models) and sparse reward, highlighting the importance of dense, well-calibrated reward signals for efficient policy optimization. Overall, ROBOMETER trained only on LIBERO achieves $\mathbf{2-4\times}$ **better sample efficiency** than sparse reward on these unseen tasks. Additional details of the RL training setup are provided in Appendix E.

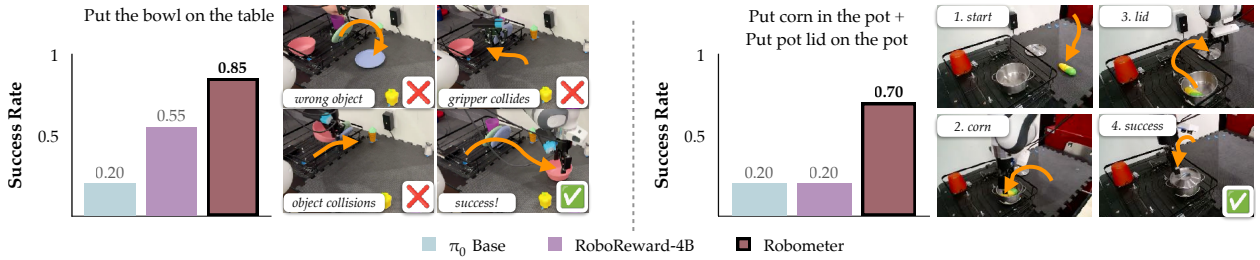


Fig. 6: **Automatic online RL** with DSRL [75] on a DROID [76] setup with ROBOMETER improves π_0 from 20% to 85% on a single-stage task and 20% to 70% on a two-stage task, outperforming RoboReward's overall success rate by 2.5×. DSRL with ROBOMETER learns to avoid base π_0 errors such as collisions or moving the wrong object. The setup is deemed "automatic" because success detection and stage advancement are handled automatically by the reward model, requiring human intervention only for physical scene resets.

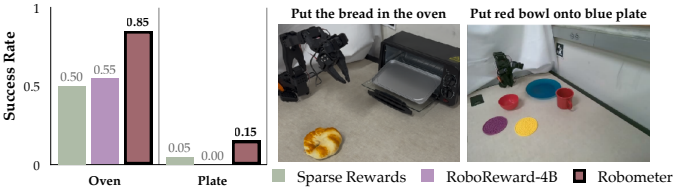


Fig. 7: **Offline RL** results using IQL on a mixture of Noisy and Expert trajectories. ROBOMETER rewards consistently outperform both RoboReward and sparse rewards: 2.4× average success rate improvement over the best baseline for each task.

Q3: Accelerating Robot Learning with Generalizable Rewards

We evaluate whether ROBOMETER's dense and generalizable reward signals can be used **zero-shot** into improved downstream robot learning across four settings, including both prehensile and non-prehensile tasks: (1) automatic online RL, (2) offline RL with mixed-expertise data, (3) data filtering and retrieval for policy improvement, and (4) out-of-distribution failure detection. Across all experiments, we compare against RoboReward-4B—the strongest baseline reward model in our offline evaluations—to assess how ROBOMETER's dense, instruction-aligned rewards affect learning stability, robustness, and sample efficiency. We also compare against strong, relevant, non-reward-model baselines for each setting where applicable. All policy learning results are averaged over 20 evaluation trials. Additional details and finer-grained results on each experiment can be found in Appendix E.

Automatic Online RL. First, we evaluate ROBOMETER in an *automated* online RL setting by training DSRL [75] from scratch on a π_0 base policy [77] pre-trained on DROID [76]. ROBOMETER enables autonomous RL by providing dense rewards and explicit *success predictions*, which we use to automate episode termination; manual human intervention is required only for physical scene resets. For comparison, RoboReward's discrete scores are also used for both reward shaping and success detection. As shown in Figure 6 (left), DSRL+ROBOMETER improves success from 20% to 85% in ≤ 45 **minutes** (10k timesteps), outperforming RoboReward's 55%. This gap arises from a key failure mode of RoboReward: it frequently assigns maximum rewards for unrelated tasks (e.g., picking up the wrong object), leading to premature resets and reinforcing incorrect behaviors. In contrast, ROBOMETER provides a more reliable learning signal.

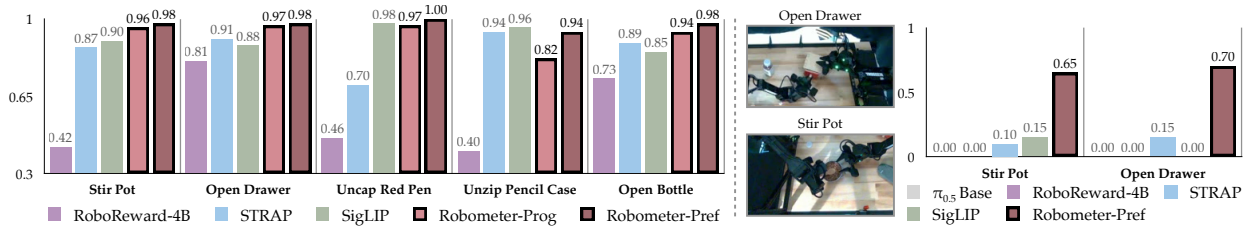
Next, we evaluate a **longer-horizon RL** setting in Figure 6 (right), where ROBOMETER's success predictions trigger progression between stages of a multi-stage task. Unlike methods that explicitly train with multi-stage rewards and thus require stage labels [7, 56], we simply decompose tasks into stages at inference time using a pre-trained VLM and use ROBOMETER to advance stages automatically. In this setting, DSRL+ROBOMETER improves π_0's success from 20% to 70% over 10k timesteps, outperforming RoboReward's 20%, which suffers from inaccurate rewards and unreliable stage transitions. Across both setups, ROBOMETER outperforms RoboReward's overall success rate by an average of $\mathbf{2.5}\times$.

Finally, we perform an additional online RL experiment—*model-based RL* integrating ROBOMETER into DreamZero [78]—where ROBOMETER improves success rate from 20% to 70%. See details and results in Appendix G.

Combining Noisy and Expert Data via Offline RL. We consider an offline RL setting with mixed-expertise data for two tasks on an SO-101 robot (SO-101 is not in `RBM-1M`), combining expert and noisy, suboptimal demos, as shown in Figure 7. We train policies with Implicit Q-Learning (IQL) [79] to study how dense rewards from ROBOMETER improve learning stability and policy extraction in offline RL.

Accurate, dense reward signals can provide informative intermediate feedback, reducing reliance on long-horizon credit assignment and enabling trajectory "stitching" with smaller discount factors γ, thereby reducing value function variance. For each of sparse reward, RoboReward, and ROBOMETER, we sweep $\gamma \in 0.90, 0.95, 0.99$ and report the best-performing checkpoint over 30,000 offline training steps. We observe that ROBOMETER, which provides dense, temporally aligned rewards, performs best at a lower discount factor $\gamma = 0.9$ and outperforms both baselines across both tasks with a $\mathbf{2.4}\times$ success rate improvement over the best baseline in each. RoboReward performs similarly to sparse rewards across the γ sweep, as its categorical (1–5) outputs provide less dense guidance and often assign high rewards to suboptimal trajectories, leading to noisy training signals.

Data Filtering & Retrieval. We next evaluate ROBOMETER as a mechanism for unsupervised data filtering and retrieval. Using a bimanual "play" dataset [80] of unannotated, multi-task trajectories collected on a Trossen AI setup (not in



(a) Proportion of Task-Relevant Subtrajectories Retrieved **(b) Policy Success Rates**

Fig. 8: **(a): Proportion of task-relevant subtrajectories** out of 100 retrieval queries. Our method consistently retrieves a high number of relevant subtrajectories using either the preference or progress objective. **(b): Success rates** of LoRA-finetuned $\pi_{0.5}$ policies using the retrieved trajectories from each method. Small amounts of suboptimal & unrelated data retrieved by other baselines degrade policy-learning performance: ROBOMETER-retrieval attains an average $4.5\times$ success rate improvement over the best baseline.

Task	T.U.	VLAC	GPT-5-mini	RoboReward-4B	ROBOMETER
move banana	0.53	0.45	0.48	0.91	**0.94**
move mouse	0.50	0.00	0.89	0.80	**0.91**
pour pebble	0.32	0.00	0.25	0.73	**0.83**
fold towel	**0.58**	0.16	0.27	0.40	**0.58**
pull tissue	0.43	0.00	0.00	0.57	**0.76**
put spoon	0.22	0.00	0.25	**0.73**	**0.73**
stir pot	0.47	0.00	0.17	**0.95**	0.90
Average	0.48	0.16	0.33	0.74	**0.81**

TABLE IV: **Failure detection performance**. ROBOMETER achieves the highest average F1 score. T.U. is the token-uncertainty baseline.

RBM-1M), we retrieve the top 100 subtrajectories for a given task instruction. We compare retrieval relevance against RoboReward, pre-trained SigLIP [81], and a retrieval-specific baseline, STRAP [82]. For ROBOMETER we retrieve subtrajectories using (i) the preference objective via pairwise trajectory comparisons, or (ii) the progress objective by computing each trajectory's value–order correlation. As shown in Figure 8(a), ROBOMETER consistently achieves higher retrieval relevance across five tasks. Finally, we LoRA-finetune $\pi_{0.5}$ [74, 83, 84] on these retrieved segments. Policies trained on ROBOMETER-filtered data vastly outperform those using baseline-retrieved data on `Stir the Pot` and `Open the Red Drawer` (Figure 8(b)), demonstrating its efficacy for targeted imitation learning. Low baseline success rates despite high retrieval rates stem from their retrieval of more failed and suboptimal, yet task-relevant, subtrajectories. Overall, ROBOMETER-retrieval averages a $\mathbf{4.5}\times$ higher success rate than the best baseline.

Failure Detection. Detecting failures during online deployment is critical for safe deployment. Thus, we evaluate ROBOMETER's zero-shot failure detection on 100 manipulation trajectories from a Franka Panda DROID robot (30 successful, 70 failed) spanning 7 tasks collected in scenes not in RBM-1M. Failures are evenly split between irreversible failures (e.g., drops or spills) and insufficient-progress failures, where execution stalls or terminates prematurely. We compare our method against: the token-uncertainty [8] of π_0-FAST-DROID [85] as proposed by Gu et al. [8] for zero-shot failure detection; VLAC, which reports failure detection results; GPT-5-mini; and RoboReward-4B. Failures are detected via temporal inconsistencies in predicted per-frame rewards.

As shown in Table IV, ROBOMETER achieves the highest average F1 score, effectively balancing true positive and true negative rates (TPR and TNR); the full breakdown with TPR and TNR is provided in Appendix Table XXVII. VLAC frequently flags trajectories as failures, achieving high TPR but low TNR, resulting in lower F1 scores. RoboReward-4B performs competitively but underperforms ROBOMETER, particularly on tasks with subtle failure modes such as *fold towel* and *pull tissue*. As detailed and visualized in Section E-5, ROBOMETER robustly detects irreversible, insufficient-progress, and non-terminal failures (e.g., hovering, oscillation, or partial completion), fully zero-shot across tasks and environments—unlike prior methods that require task-specific thresholds, calibration, or test-time interaction [8, 86, 87].

V. LIMITATIONS AND FUTURE WORK

ROBOMETER operates as a dense, per-frame reward model over subsampled video inputs, enabling scalable training but limiting its ability to memorize specific past key events for very long-horizon or partially observed tasks. In addition, real-world robot executions exhibit a wide diversity of failure modes, many of which are rare, subtle, or task-specific, and the current training data may not fully capture this breadth. As a result, ROBOMETER may fail to recognize or correctly reward certain failure cases that fall outside the dominant patterns seen during training. We hope that our training recipe enables future work to build upon larger reward modeling datasets.

As a vision-language-based model, ROBOMETER also lacks direct access to latent physical state such as contact forces, grasp stability, or compliance, and may fail to recognize or correctly reward failure cases driven by these factors until they become visually observable. Future work could address these limitations by incorporating denser temporal modeling, VQA-style supervision to reason about task structure and completion criteria, and off-domain data for better generalization [88, 89], as well as by developing more systematically curated failure datasets that better reflect the diversity of real-world failure modes [13]. Finally, real-world executions may also contain safety-critical states (e.g., in a constrained MDP formulation) that ROBOMETER is not pre-trained to predict. Future work can train ROBOMETER with an additional safety head, similar to the success prediction head, on provided safety labels.

References

[1] D. R. J. Laming, "The relativity of 'absolute' judgements," *British Journal of Mathematical and Statistical Psychology*, vol. 37, pp. 152–183, 1984.

[2] N. Stewart, G. D. A. Brown, and N. Chater, "Absolute identification by relative judgment." *Psychological review*, vol. 112 4, pp. 881–911, 2005.

[3] M. A. Sharif and D. M. Oppenheimer, "The effect of relative encoding on memory-based judgments," *Psychological Science*, vol. 27, no. 8, pp. 1136–1145, 2016.

[4] D. Yang, D. Tjia, J. Berg, D. Damen, P. Agrawal, and A. Gupta, "Rank2reward: Learning shaped reward functions from passive video," in *International Conference on Robotics and Automation (ICRA)*, 2024.

[5] J. Zhang, Y. Luo, A. Anwar, S. A. Sontakke, J. J. Lim, J. Thomason, E. Biyik, and J. Zhang, "ReWiND: Language-guided rewards teach robot policies without new demonstrations," in *Conference on Robot Learning (CoRL)*, 2025.

[6] Y. J. Ma, J. Hejna, A. Wahid, C. Fu, D. Shah, J. Liang, Z. Xu, S. Kirmani *et al.*, "Vision language models are in-context value learners," in *International Conference on Learning Representations (ICLR)*, 2025.

[7] Q. Chen, J. Yu, M. Schwager, P. Abbeel, F. Shentu, and P. Wu, "Sarm: Stage-aware reward modeling for long horizon robot manipulation," *arXiv preprint arXiv:2509.25358*, 2025.

[8] Q. Gu, Y. Ju, S. Sun, I. Gilitschenski, H. Nishimura, M. Itkina, and F. Shkurti, "SAFE: Multitask failure detection for vision-language-action models," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[9] S. Venkataraman, Y. Wang, Z. Wang, N. S. Ravie, Z. Erickson, and D. Held, "Real-world offline reinforcement learning from vision language model feedback," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

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

[11] T. Lee, A. Wagenmaker, K. Pertsch, P. Liang, S. Levine, and C. Finn, "Roboreward: General-purpose vision-language reward models for robotics," *arXiv preprint arXiv:2601.00675*, 2026.

[12] H. Tan, S. Chen, Y. Xu, Z. Wang, Y. Ji, C. Chi, Y. Lyu, Z. Zhao *et al.*, "Robo-dopamine: General process reward modeling for high-precision robotic manipulation," *arXiv preprint arXiv:2512.23703*, 2025.

[13] R. Tian, Y. Wu, and A. Bajcsy, "Position: Good embodied reward models need bad behavior data," in *International Conference on Machine Learning (ICML)*, 2026.

[14] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *International Conference on Machine Learning (ICML)*, 2000.

[15] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *International Conference on Machine Learning (ICML)*, 2004.

[16] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *AAAI Conference on Artificial Intelligence*, 2008.

[17] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *International Conference on Machine Learning (ICML)*, 2016.

[18] A. Bobu, M. Wiggert, C. Tomlin, and A. D. Dragan, "Feature expansive reward learning: Rethinking human input," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2021.

[19] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[20] L. Tai, J. Zhang, M. Liu, and W. Burgard, "Socially compliant navigation through raw depth inputs with generative adversarial imitation learning," in *International Conference on Robotics and Automation (ICRA)*, 2018.

[21] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2018.

[22] J. Fu, A. Singh, D. Ghosh, L. Yang, and S. Levine, "Variational inverse control with events: A general framework for data-driven reward definition," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[23] Y. Fu, H. Zhang, D. Wu, W. Xu, and B. Boulet, "Robot policy learning with temporal optimal transport reward," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[24] A. K. Jain, V. Mohta, S. Kim, A. Bhardwaj, J. Ren, Y. Feng, S. Choudhury, and G. Swamy, "A smooth sea never made a skilled SAILOR: Robust imitation via learning to search," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[25] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[26] D. Sadigh, A. D. Dragan, S. S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions," in *Robotics: Science and Systems (RSS)*, 2017.

[27] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan, "Learning from physical human corrections, one feature at a time," in *International Conference on Human-Robot Interaction (HRI)*, 2018.

[28] E. Biyik, N. Huynh, M. J. Kochenderfer, and D. Sadigh, "Active preference-based gaussian process regression for reward learning," in *Robotics: Science and Systems (RSS)*, 2020.

[29] K. Lee, L. Smith, and P. Abbeel, "Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training," in *International Conference on Machine Learning (ICML)*, 2021.

[30] J. Hejna and D. Sadigh, "Few-shot preference learning for human-in-the-loop rl," in *Conference on Robot Learning (CoRL)*, 2022.

[31] V. Myers, E. Biyik, N. Anari, and D. Sadigh, "Learning multimodal rewards from rankings," in *Conference on Robot Learning (CoRL)*, 2021.

[32] Z. Yang, M. Jun, J. Tien, S. J. Russell, A. Dragan, and E. Bıyık, "Trajectory improvement and reward learning from comparative language feedback," in *Conference on Robot Learning (CoRL)*, 2024.

[33] Y. Korkmaz and E. Bıyık, "Mile: Model-based intervention learning," in *International Conference on Robotics and Automation (ICRA)*, 2025.

[34] J. Kwok, C. Agia, R. Sinha, M. Foutter, S. Li, I. Stoica, A. Mirhoseini, and M. Pavone, "Robomonkey: Scaling test-time sampling and verification for vision-language-action models," in *Conference on Robot Learning (CoRL)*, 2025.

[35] 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 *International Conference on Machine Learning (ICML)*, 2024.

[36] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan *et al.*, "Eureka: Human-level reward design via coding large language models," in *International Conference on Learning Representations (ICLR)*, 2024.

[37] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. Gonzalez Arenas, H.-T. Lewis Chiang, T. Erez *et al.*, "Language to rewards for robotic skill synthesis," in *Conference on Robot Learning (CoRL)*, 2023.

[38] T. Xie, S. Zhao, C. H. Wu, Y. Liu, Q. Luo, V. Zhong, Y. Yang, and T. Yu, "Text2reward: Reward shaping with language models for reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2024.

[39] M. Kwon, S. M. Xie, K. Bullard, and D. Sadigh, "Reward design with language models," in *International Conference on Learning Representations (ICLR)*, 2023.

[40] M. Hwang, A. Forsey-Smerek, N. Dennler, and A. Bobu, "Masked irl: Llm-guided reward disambiguation from demonstrations and language," *arXiv preprint arXiv:2511.14565*, 2025.

[41] Y. Cui, S. Niekum, A. Gupta, V. Kumar, and A. Rajeswaran, "Can foundation models perform zero-shot task specification for robot manipulation?" in *Learning for Dynamics and Control Conference (L4DC)*, 2022.

[42] P. Mahmoudieh, D. Pathak, and T. Darrell, "Zero-shot reward specification via grounded natural language," in *International Conference on Machine Learning (ICML)*, 2022.

[43] S. A. Sontakke, J. Zhang, S. Arnold, K. Pertsch, E. Biyik, D. Sadigh, C. Finn, and L. Itti, "Roboclip: One demonstration is enough to learn robot policies," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[44] Y. Du, K. Konyushkova, M. Denil, A. Raju, J. Landon, F. Hill, N. de Freitas, and S. Cabi, "Vision-language models as success detectors," in *Conference on Lifelong Learning Agents*, 2023.

[45] 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 *International Conference on Learning Representations (ICLR)*, 2023.

[46] A. Adeniji, A. Xie, C. Sferrazza, Y. Seo, S. James, and P. Abbeel, "Language reward modulation for pretraining reinforcement learning," *arXiv preprint arXiv:2308.12270*, 2024.

[47] Y. Fu, H. Zhang, D. Wu, W. Xu, and B. Boulet, "FuRL: Visual-language models as fuzzy rewards for reinforcement learning," in *Internatonal Conference on Machine Learning (ICML)*, 2024.

[48] L. Guan, Y. Zhou, D. Liu, Y. Zha, H. B. Amor, and S. Kambhampati, "Task success is not enough: Investigating the use of video-language models as behavior critics for catching undesirable agent behaviors," in *Conference on Language Modeling (COLM)*, 2024.

[49] J. Rocamonde, V. Montesinos, E. Nava, E. Perez, and D. Lindner, "Vision-language models are zero-shot reward models for reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2024.

[50] S. Chen, C. Harrison, Y.-C. Lee, A. J. Yang, Z. Ren, L. J. Ratliff, J. Duan, D. Fox *et al.*, "Topreward: Token probabilities as hidden zero-shot rewards for robotics," *arXiv preprint arXiv:2602.19313*, 2026.

[51] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang *et al.*, "Minedojo: Building open-ended embodied agents with internet-scale knowledge," in *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[52] K. Nottingham, P. Ammanabrolu, A. Suhr, Y. Choi, H. Hajishirzi, S. Singh, and R. Fox, "Do embodied agents dream of pixelated sheep?: Embodied decision making using language guided world modelling," in *International Conference on Machine Learning (ICML)*, 2023.

[53] T. Nam, J. Lee, J. Zhang, S. J. Hwang, J. J. Lim, and K. Pertsch, "Lift: Unsupervised reinforcement learning with foundation models as teachers," *arXiv preprint arXiv:2312.08958*, 2023.

[54] Y. J. Ma, W. Liang, V. Som, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman, "Liv: Language-image representations and rewards for robotic control," in

International Conference on Machine Learning (ICML), 2023.

[55] K.-H. Hung, P.-C. Lo, J.-F. Yeh, H.-Y. Hsu, Y.-T. Chen, and W. H. Hsu, "VICtor: Learning hierarchical vision-instruction correlation rewards for long-horizon manipulation," in *International Conference on Learning Representations (ICLR)*, 2025.

[56] C. Kim, M. Heo, D. Lee, H. Lee, J. Shin, J. J. Lim, and K. Lee, "Subtask-aware visual reward learning from segmented demonstrations," in *International Conference on Learning Representations (ICLR)*, 2025.

[57] P. Budzianowski, E. Wiśnios, G. Góral, I. Kulakov, V. Petrenko, and K. Walas, "Opengvl: Benchmarking visual temporal progress for data curation," *arXiv preprint arXiv:2509.17321*, 2025.

[58] S. K. S. Ghasemipour, A. Wahid, J. Tompson, P. R. Sanketi, and I. Mordatch, "Self-improving embodied foundation models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[59] P. Intelligence, A. Amin, R. Aniceto, A. Balakrishna, K. Black, K. Conley, G. Connors, J. Darpinian *et al.*, "$\pi^*_{0.6}$: A vla that learns from experience," *arXiv:2511.14759*, 2025.

[60] J. Zhang, C. Qian, H. Sun, H. Lu, D. Wang, L. Xue, and H. Liu, "Progresslm: Towards progress reasoning in vision-language models," *arXiv preprint arXiv:2601.15224*, 2026.

[61] P. Atreya, K. Pertsch, T. Lee, M. J. Kim, A. Jain, A. Kuramshin, C. Eppner, C. Neary *et al.*, "Roboarena: Distributed real-world evaluation of generalist robot policies," in *Conference on Robot Learning (CoRL)*, 2025.

[62] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee *et al.*, "Open X-Embodiment: Robotic learning datasets and RT-X models," in *International Conference on Robotics and Automation (ICRA)*, 2024.

[63] A. W. C. contributors, "Agibot world colosseum," 2024.

[64] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, J. Ma, E. Kazakos, D. Moltisanti, J. Munro *et al.*, "Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100," *International Journal of Computer Vision (IJCV)*, vol. 130, p. 33–55, 2022.

[65] H.-S. Fang, H. Fang, Z. Tang, J. Liu, C. Wang, J. Wang, H. Zhu, and C. Lu, "Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot," *arXiv preprint arXiv:2307.00595*, 2023.

[66] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *arXiv preprint arXiv:2306.03310*, 2023.

[67] Z. Zhou, P. Atreya, A. Lee, H. R. Walke, O. Mees, and S. Levine, "Autonomous improvement of instruction following skills via foundation models," in *Conference on Robot Learning (CoRL)*, 2024.

[68] Z. Lin, J. Duan, H. Fang, D. Fox, R. Krishna, C. Tan, and B. Wen, "Failsafe: Reasoning and recovery from failures in vision-language-action models," *arXiv preprint arXiv:2510.01642*, 2025.

[69] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *International Conference on Machine Learning (ICML)*, 2017.

[70] J. Farebrother, J. Orbay, Q. Vuong, A. A. Taiga, Y. Chebotar, T. Xiao, A. Irpan, S. Levine *et al.*, "Stop regressing: Training value functions via classification for scalable deep RL," in *International Conference on Machine Learning (ICML)*, 2024.

[71] Y. Huang, S. Zou, J. Zhang, X. Liu, R. Hu, and K. Xu, "Adapower: Specializing world foundation models for predictive manipulation," *arXiv preprint arXiv:2512.035358*, 2025.

[72] C. Muslimani, K. Johnstonbaugh, S. Chandramouli, S. Booth, W. B. Knox, and M. E. Taylor, "Towards improving reward design in RL: A reward alignment metric for RL practitioners," in *Reinforcement Learning Conference (RLC)*, 2025.

[73] W. Lu, M. Ye, Z. Ye, R. Tao, S. Yang, and B. Zhao, "Robofac: A comprehensive framework for robotic failure analysis and correction," *arXiv preprint arXiv:2505.12224*, 2025.

[74] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations (ICLR)*, 2022.

[75] A. Wagenmaker, M. Nakamoto, Y. Zhang, S. Park, W. Yagoub, A. Nagabandi, A. Gupta, and S. Levine, "Steering your diffusion policy with latent space reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2025.

[76] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[77] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arxiv:2410.24164*, 2024.

[78] S. Ye, Y. Ge, K. Zheng, S. Gao, S. Yu, G. Kurian, S. Indupuru, Y. L. Tan *et al.*, "World action models are zero-shot policies," *arXiv preprint arXiv:2602.15922*, 2026.

[79] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," *arXiv preprint arXiv:2110.06169*, 2021.

[80] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet, "Learning latent plans from play," *Conference on Robot Learning (CoRL)*, 2019.

[81] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *International Conference on Computer Vision (ICCV)*, 2023.

[82] M. Memmel, J. Berg, B. Chen, A. Gupta, and J. Francis, "Strap: Robot sub-trajectory retrieval for augmented policy learning," in *International Conference on Learning Representations (ICLR)*, 2025.

[83] Z. Liu, J. Zhang, K. Asadi, Y. Liu, D. Zhao, S. Sabach, and R. Fakoor, "TAIL: Task-specific adapters for imitation learning with large pretrained models," in *International Conference on Learning Representations (ICLR)*, 2024.

[84] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi *et al.*, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," *arXiv preprint arXiv:2504.16054*, 2025.

[85] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn *et al.*, "Fast: Efficient action tokenization for vision-language-action models," in *Robotics: Science and Systems (RSS)*, 2025.

[86] C. Xu, T. K. Nguyen, E. Dixon, C. Rodriguez, P. Miller, R. Lee, P. Shah, R. Ambrus *et al.*, "Can we detect failures without failure data? Uncertainty-aware runtime failure detection for imitation learning policies," in *Robotics: Science and Systems (RSS)*, 2025.

[87] C. Agia, R. Sinha, J. Yang, Z. Cao, R. Antonova, M. Pavone, and J. Bohg, "Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress," in *Conference on Robot Learning (CoRL)*, 2025.

[88] Y. Li, Y. Deng, J. Zhang, J. Jang, M. Memmel, C. R. Garrett, F. Ramos, D. Fox *et al.*, "Hamster: Hierarchical action models for open-world robot manipulation," in *International Conference on Learning Representations (ICLR)*, 2025.

[89] J. Zhang, M. Memmel, K. Kim, D. Fox, J. Thomason, F. Ramos, E. Bıyık, A. Gupta *et al.*, "Peek: Guiding and minimal image representations for zero-shot generalization of robot manipulation policies," in *International Conference on Robotics and Automation (ICRA)*, 2026.

[90] Y. Dai, J. Lee, N. Fazeli, and J. Chai, "Racer: Rich language-guided failure recovery policies for imitation learning," in *International Conference on Robotics and Automation (ICRA)*, 2025.

[91] T. Yu, D. Quillen, Z. He, R. Julian, A. Narayan, H. Shively, A. Bellathur, K. Hausman *et al.*, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2019.

[92] T. Jiang, T. Yuan, Y. Liu, C. Lu, J. Cui, X. Liu, S. Cheng, J. Gao *et al.*, "Galaxea open-world dataset and g0 dual-system vla model," *arXiv preprint arXiv:2509.00576*, 2025.

[93] J. Lee, J. Duan, H. Fang, Y. Deng, S. Liu, B. Li, B. Fang, J. Zhang *et al.*, "Molmoact: Action reasoning models that can reason in space," *arXiv preprint arXiv:2508.07917*, 2025.

[94] Z. Zhao, H. Jing, X. Liu, J. Mao, A. Jha, H. Yang, R. Xue, S. Zakharor *et al.*, "Humanoid everyday: A comprehensive robotic dataset for open-world humanoid manipulation," *arXiv preprint arXiv:2510.08807*, 2025.

[95] M. Hwang, J. Hejna, D. Sadigh, and Y. Bisk, "Motif: Motion instruction fine-tuning," *IEEE Robotics and Automation Letters (RA-L)*, 2025.

[96] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, D. J. Yoon *et al.*, "Humanoid policy˜ human policy," *arXiv preprint arXiv:2503.13441*, 2025.

[97] S. Xie, H. Cao, Z. Weng, Z. Xing, H. Chen, S. Shen, J. Leng, Z. Wu *et al.*, "Human2robot: Learning robot actions from paired human-robot videos," *arXiv preprint arXiv:2502.16587*, 2025.

[98] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin *et al.*, "Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai," in *Robotics: Science and Systems (RSS)*, 2025.

[99] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters (RA-L)*, 2020.

[100] Z. Zhou, P. Atreya, Y. L. Tan, K. Pertsch, and S. Levine, "Autoeval: Autonomous evaluation of generalist robot manipulation policies in the real world," *arXiv preprint arXiv:2503.24278*, 2025.

[101] A. Inceoglu, E. E. Aksoy, A. C. Ak, and S. Sariel, "Fino-net: A deep multimodal sensor fusion framework for manipulation failure detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[102] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman *et al.*, "Rt-1: Robotics transformer for real-world control at scale," in *Robotics: Science and Systems (RSS)*, 2023.

[103] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers *et al.*, "Bridgedata v2: A dataset for robot learning at scale," in *Conference on Robot Learning (CoRL)*, 2023.

[104] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence, "Interactive language: Talking to robots in real time," *IEEE Robotics and Automation Letters (RA-L)*, 2023.

[105] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "BC-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning (CoRL)*, 2021.

[106] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar, "Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking," *arXiv preprint arXiv:2309.01918*, 2023.

[107] M. Heo, Y. Lee, D. Lee, and J. J. Lim, "Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation," in *Robotics: Science and Systems (RSS)*, 2023.

[108] R. Shah, R. Martín-Martín, and Y. Zhu, "MUTEX: Learning unified policies from multimodal task specifications," in *Conference on Robot Learning (CoRL)*, 2023.

[109] J. Luo, C. Xu, X. Geng, G. Feng, K. Fang, L. Tan, S. Schaal, and S. Levine, "Multi-stage cable routing through hierarchical imitation learning," *arXiv preprint arXiv:2307.08927*, 2023.

[110] S. Dass, J. Yapeter, J. Zhang, J. Zhang, K. Pertsch, S. Nikolaidis, and J. J. Lim, "Clvr jaco play dataset," 2023.

[111] I. Radosavovic, B. Shi, L. Fu, K. Goldberg, T. Darrell, and J. Malik, "Robot learning with sensorimotor pre-training," *arXiv preprint arXiv:2306.10007*, 2023.

[112] G. Zhou, V. Dean, M. K. Srirama, A. Rajeswaran, J. Pari, K. Hatch, A. Jain, T. Yu *et al.*, "Train offline, test online: A real robot learning benchmark," in *International Conference on Robotics and Automation (ICRA)*, 2023.

[113] S. Saxena, M. Sharma, and O. Kroemer, "Multi-resolution sensing for real-time control with vision-language models," in *Conference on Robot Learning (CoRL)*, 2023.

[114] S. Belkhale, Y. Cui, and D. Sadigh, "Hydra: Hybrid robot actions for imitation learning," in *Conference on Robot Learning (CoRL)*, 2023.

[115] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell, "Real-world robot learning with masked visual pre-training," in *Conference on Robot Learning (CoRL)*, 2022.

[116] X. Zhu, R. Tian, C. Xu, M. Ding, W. Zhan, and M. Tomizuka, "Fanuc manipulation: A dataset for learning-based manipulation with fanuc mate 200id robot," 2023.

[117] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *Conference on Robot Learning (CoRL)*, 2024.

[118] G. Yan, K. Wu, and X. Wang, "Ucsd kitchens dataset," August 2023.

[119] Y. Zhu, P. Stone, and Y. Zhu, "Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation," *IEEE Robotics and Automation Letters (RA-L)*, 2022.

[120] J. Vogel, A. Hagengruber, M. Iskandar, G. Quere, U. Leipscher, S. Bustamante, A. Dietrich, H. Hoeppner *et al.*, "Edan - an emg-controlled daily assistant to help people with physical disabilities," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[121] G. Quere, A. Hagengruber, M. Iskandar, S. Bustamante, D. Leidner, F. Stulp, and J. Vogel, "Shared Control Templates for Assistive Robotics," in *International Conference on Robotics and Automation (ICRA)*, 2020.

[122] T. Osa, "Motion planning by learning the solution manifold in trajectory optimization," *The International Journal of Robotics Research*, 2022.

[123] S. Haldar, V. Mathur, D. Yarats, and L. Pinto, "Watch and match: Supercharging imitation with regularized optimal transport," in *Conference on Robot Learning (CoRL)*, 2023.

[124] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster *et al.*, "OpenVLA: An open-source vision-language-action model," in *Conference on Robot Learning (CoRL)*, 2024.

[125] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding *et al.*, "Qwen3-vl technical report," *arXiv preprint arXiv:2511.21631*, 2025.

[126] R. Hoque, P. Huang, D. J. Yoon, M. Sivapurapu, and J. Zhang, "Egodex: Learning dexterous manipulation from large-scale egocentric video," *arXiv preprint arXiv:2505.11709*, 2025.

[127] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang *et al.*, "Qwen2. 5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025.

[128] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, p. 324, 1952.

[129] W. Monroe, R. X. Hawkins, N. Goodman, and C. Potts, "Colors in context: A pragmatic neural model for grounded language understanding," *Transactions of the Association for Computational Linguistics (TACL)*, 2017.

[130] C. Mitra, A. Anwar, R. Corona, D. Klein, T. Darrell, and J. Thomason, "Which one? leveraging context between objects and multiple views for language grounding," in *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024.

[131] Y. Bao, S. Ghosh, and J. Chai, "Learning to mediate disparities towards pragmatic communication," *arXiv preprint arXiv:2203.13685*, 2022.

[132] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang *et al.*, "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[133] G. Team, "Gemini: A family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2024.

[134] C. Clark, J. Zhang, Z. Ma, J. S. Park, M. Salehi, R. Tripathi, S. Lee, Z. Ren *et al.*, "Molmo2: Open weights and data for vision-language models with video understanding and grounding," *arXiv preprint arXiv:2601.10611*, 2026.

[135] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2024.

[136] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[137] 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 (ICML)*, 2018.
[138] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," in *International Conference on Learning Representations (ICLR)*, 2022.
[139] X. B. Peng, A. Kumar, G. Zhang, and S. Levine, "Advantage-weighted regression: Simple and scalable off-policy reinforcement learning," in *arXiv preprint arXiv:1910.00177*, 2019.
[140] M. Hong, A. Liang, K. Kim, H. Rajaprakash, J. Thomason, E. Bıyık, and J. Zhang, "Hand me the data: Fast robot adaptation via hand path retrieval," in *International Conference on Robotics and Automation (ICRA)*, 2026.
[141] C. Agia, R. Sinha, J. Yang, R. Antonova, M. Pavone, H. Nishimura, M. Itkina, and J. Bohg, "Cupid: Curating data your robot loves with influence functions," in *Conference on Robot Learning (CoRL)*, 2025.
[142] A. Xie, R. Chand, D. Sadigh, and J. Hejna, "Data retrieval with importance weights for few-shot imitation learning," in *Conference on Robot Learning (CoRL)*, 2025.
[143] Y. Zhang, Y. Xie, H. Liu, R. Shah, M. Wan, L. Fan, and Y. Zhu, "Scizor: Self-supervised data curation for large-scale imitation learning," in *International Conference on Robotics and Automation (ICRA)*, 2026.
[144] J. Hejna, S. Mirchandani, A. Balakrishna, A. Xie, A. Wahid, J. Tompson, P. Sanketi, D. Shah *et al.*, "Robot data curation with mutual information estimators," in *Robotics: Science and Systems (RSS)*, 2025.
[145] M. Du, S. Nair, D. Sadigh, and C. Finn, "Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets," in *Robotics: Science and Systems (RSS)*, 2023.
[146] L.-H. Lin, Y. Cui, A. Xie, T. Hua, and D. Sadigh, "Flowretrieval: Flow-guided data retrieval for few-shot imitation learning," in *Conference on Robot Learning (CoRL)*, 2024.
[147] A. Liang, I. Singh, K. Pertsch, and J. Thomason, "Transformer adapters for robot learning," in *CoRL 2022 Workshop on Pre-training Robot Learning*, 2022.
[148] Y. Huang, J. Song, Z. Wang, S. Zhao, H. Chen, F. Juefei-Xu, and L. Ma, "Look before you leap: An exploratory study of uncertainty measurement for large language models," *arXiv preprint arXiv:2307.10236*, 2023.