SPUR: Scaling Reward Learning from Human Demonstrations

Anonymous Author(s)
Affiliation
Address
email

Abstract: Learning reward functions from human demonstrations is critical for scalable robot learning, yet most approaches either require impractical ground-truth state access, costly online retraining, or yield domain-specific models with poor transferability. We propose SPUR, a unified reward modeling framework that combines a large pre-trained vision-language model (VLM) backbone fine-tuned to encode robot image sequences and language instructions, a progress-based reward objective trained on successful demonstrations augmented with video rewind to simulate failures, and a preference-learning objective over mismatched and rewound trajectories to enable training on failed executions without explicit progress labels. This design leverages the generalization of VLMs while integrating complementary progress and preference signals for improved robustness and generalization. Experiments on out-of-distribution tasks in LIBERO and Meta-World show that each component contributes to performance gains across a set of reward metrics, and their combination achieves state-of-the-art results compared to recent baselines, demonstrating scalable training of reward models.

Keywords: CoRL, Robots, Learning

1 Introduction

An important problem in robot learning is that of learning rewards from human demonstrations [1] to guide policy learning. When deploying robots in the real world, it's important that reward models *generalize* to new tasks so that humans do not need to provide additional demonstrations, which is expensive to scale, or train the reward models in tandem with the robot policies, which is sample-inefficient and time-consuming. In this work, we investigate how to train reward functions that can effectively *generalize* to new tasks without online training or additional demonstrations.

Prior works have attempted to develop generalizable reward functions, but they often assume access to ground-truth states that may be difficult to provide in the real world [2, 3, 4, 5, 6, 7] or the ability to train reward models from scratch in tandem with the policy [1, 8, 9], limiting their practical applicability.

Some recent works instead propose reward models that can be directly used at test time, conditioned solely on image observations and language instructions. One common approach is to leverage the generalization capabilities of large vision-language models (VLMs) by querying them for *task progress* to be used as reward [10, 11, 12, 13, 14], but these models have been shown to predict noisy rewards difficult to be directly used for training robot policies [12, 13, 15]. Another is to directly train a smaller reward model on human demonstrations. These methods use either a task-progress-based training objective [16, 17, 15], or a preference-based or contrastive objective [18, 19, 20], but they result in domain-specific reward models that are unlikely to generalize well to new domains. Instead, we aim to train a generalizable reward model that can provide useful rewards, even on sig-

Submitted to the 9th Conference on Robot Learning (CoRL 2025). Do not distribute.

nificantly out-of-distribution tasks and settings. We hypothesize that ideas from all three threads of work are useful, and combining them can lead to a reward model whose generalization is greater than the sum of its parts.

To this end, we investigate how to blend together large-scale VLM backbones, progress-based rewards, and preference-based rewards into one scalable, unified reward model we call SPUR (**S**calable **P**rogress and Preference **U**nified **R**eward). Firstly, we investigate the use of a large-scale pre-trained VLM backbone, not for zero-shot robot reward queries, but instead as a trainable backbone for encoding robot image sequences and language instruction tokens. SPUR then directly predicts task progress coming from successful demonstration trajectories, along with simulating failed trajectories with *video rewind* augmentation [15], to produce useful per-timestep rewards for robots. Finally, to help the model scale, SPUR also trains to predict binary *preferences* over mismatched and rewound video sequences. This preference objective complements the reward prediction objective while also allowing for training with trajectories with *failed execution*, which progress-based methods cannot train on without explicit progress labels for each failed trajectory.

Through reward analysis experiments on new tasks in LIBERO [21] and Metaworld [22], we demonstrate how each component complements the others for scalable training of generalizable reward models. We also outperform recent, state of the art baselines across comparisons in both domains.

2 Related Works

2.1 Learning Reward Functions

Several prior works explored learning reward functions from various forms of supervision. One line of research leverages direct human feedback, such as comparisons [23, 24, 25, 26, 27], rankings [28], language annotations [9], and trajectory corrections [29, 30], to infer rewards. While these methods can align reward functions with human intent, they typically require substantial human supervision and are often sample-inefficient.

Another major direction is inverse RL (IRL), where reward functions are inferred from demonstrations [1, 31, 32, 33] or implicitly from expert and goal-state distributions [34, 35, 36]. However, IRL methods struggle to scale to high-dimensional state–action spaces and usually require new demonstrations for every new task. In general, both human-feedback–based and IRL-based approaches lack effective transfer mechanisms: when faced with a novel task, they often need to be retrained from scratch. In contrast, our method leverages the semantic representations in VLM backbone to transfer learned reward functions to unseen tasks without requiring additional human supervision.

2.2 Large Vision and Language Models as Reward Functions

Recently, LLMs and VLMs have been applied to reward design through code generation [5, 4, 37], embedding-based reward estimation [38, 10], and preference-based feedback [8, 2]. However, most of these methods assume access to privileged state information that is rarely available in real-world settings. Another line of work employs VLMs as zero-shot success detectors, treating them as sparse reward models [39, 40, 41]. While promising, this approach provides only episodic feedback and misses the dense supervision signals present throughout the trajectory.

Some prior work explores task progress as a proxy reward, either by using VLMs as progress estimators [10, 11, 12, 13, 14] or by training task-specific models with progress-prediction objectives [16, 17, 15]. VLM-based estimators, however, often yield noisy outputs, while smaller per-task models tend to overfit to domain-specific dynamics, limiting their generalization to new domains. In this work, we combine progress prediction with preference feedback over video sequences to improve the reward learning objective. We further show that incorporating failure trajectory pairs improves generalization across tasks

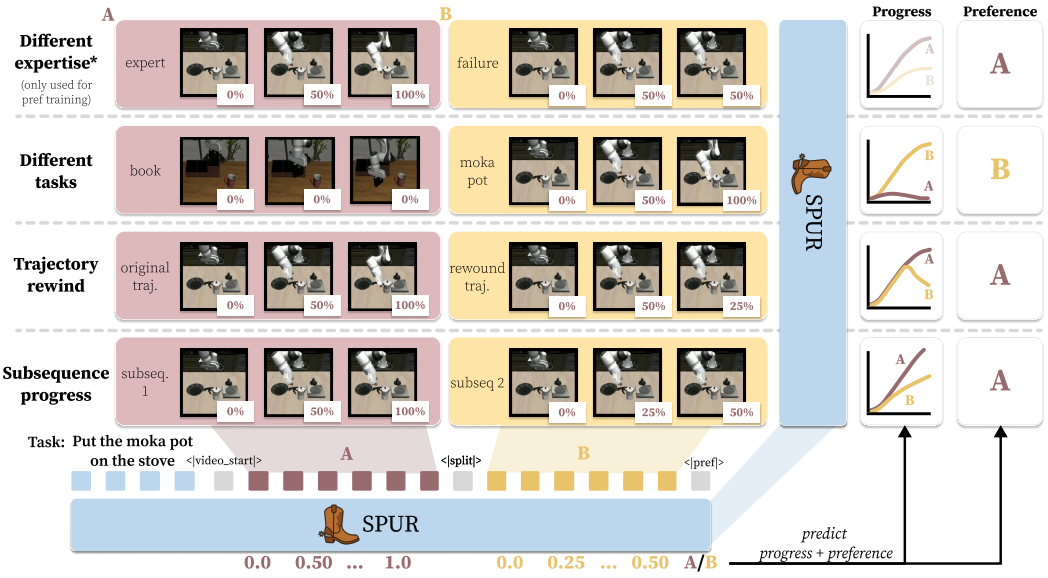

Figure 1: **SPUR.** Given two video trajectories, we train our VLM-based reward model, SPUR, to predict progress-based and preference-based rewards. We use four strategies (left) for curating training examples from our given datasets, which are further detailed in Section 3.2.

3 Method

We introduce SPUR, a generalizable reward model, as illustrated in Figure 1. We start with a dataset $\mathcal{D} = \{\tau_1, \tau_2, \tau_3, ...\}$ consisting of robot demonstration trajectories $\tau = \{o_{1:T}, l, \texttt{success}\}$ with image observations o, language instructions l, and a success label $\texttt{success} \in \{0, 1\}$. To enable generalization to unseen tasks, environments, and domains, we first instantiate the reward model with a large-scale, pre-trained vision-language model (VLM) backbone. Then, we fine-tune it on two objectives that complement each other: predicting *preferences* over pairs of video trajectories and predicting continuous task *progress* as rewards.

3.1 VLM Base Model

Our base model is QWEN2.5-VL-INSTRUCT-3B [42], a 3B parameter, open-source, image and video-input VLM which demonstrates strong zero-shot performance across various vision and language tasks. SPUR can incorporate any base VLM model which supports language and video input, but we found QWEN to be easy to tune and performant. SPUR uses this model to take as input a natural language task description l and up to two different video sequences, $o^1_{1:T}$ and $o^2_{1:T}$ of arbitrary length. SPUR encodes both the language and videos as a single sequence of tokens with the base model's tokenizer to construct its inputs as depicted below:

$$(l, o^1, o^2) \rightarrow \texttt{Token}(l)\ \langle|\text{video\_start}|\rangle\ \texttt{Token}(o^1)\ \langle|\text{split\_token}|\rangle\ \texttt{Token}(o^2)\ \langle|\text{pref\_token}|\rangle, \quad (1)$$

where $\langle|\text{split\_token}|\rangle$ is a special token that delinates the two video sequences. The VLM then produces a sequence of hidden states, which we use for preference and progress prediction, as detailed next.

3.2 Preference Prediction

To predict preferences, we attach an MLP head to the final hidden state corresponding to the special token $\langle|\text{pref\_token}|\rangle$ from Equation (1) to produce preference logits. The model is trained to discern which of the two video sequences, $o^1_{1:T}$ or $o^2_{1:T}$, is better aligned with the given natural language task description, l. We denote the preference label as y, where $y = 1$ if o^1 is preferred over o^2, and

$y = 0$ otherwise. Formally, the learned preference head MLP_{pref} produces a probability:

$$P(o^1 \succ o^2 \mid l) = \sigma\left(\text{MLP}_{\text{pref}}\left(h_{\langle|\text{pref\_token}|\rangle}\right)\right).$$

where σ is the sigmoid function and $h_{\langle|\text{pref\_token}|\rangle}$ is the hidden state corresponding to the location of the $\langle|\text{pref\_token}|\rangle$ in the input from Equation (1). The preference objective is optimized using the binary cross-entropy loss and backpropagated through MLP_{pref} and the VLM through $h_{\langle|\text{pref\_token}|\rangle}$:

$$\mathcal{L}_{\text{preference}} = -\Big[y \log P(o^1 \succ o^2 \mid l) + (1-y)\log(1 - P(o^1 \succ o^2 \mid l))\Big].$$

Preference Sample Construction. Large-scale preference datasets comparing robot trajectories are not widely available, especially for training generalizable reward models. Given the scarcity of such data, we instead propose a suite of strategies for scalably curating a larger set of preference samples from existing trajectories without needing manual human annotations.

We construct preference pairs $(l, o^{\text{chosen}}, o^{\text{rejected}}, y)$ for training by sampling trajectories from $\mathcal{D}$, always assigning o^{chosen} as the preferred observation sequence ($y = 1$). Given sampled trajectories $\tau = \{o_{1:T}, l, \texttt{success}\}$, we create batches of preference tuples sampled uniformly over the following four strategies:

1. **Different expertise.** Given a task instruction l, sample two trajectories $\tau_1, \tau_2 \sim \mathcal{D}$ with the same instruction where τ_1 has $\texttt{success} == 1$ and τ_2 has $\texttt{success} == 0$. We extract o^{chosen} from the observation sequence from τ_1.
2. **Different tasks.** Sample a trajectory $o^{\text{chosen}} \sim \mathcal{D}$ corresponding to the task instruction l and a trajectory o^{rejected} with a different instruction. These samples encourage the model to ground correct video and language pairs.
3. **Trajectory rewind.** Following the idea proposed by ReWiND [15] that generated failed trajectories for reward *progress* prediction by *rewinding* videos, we propose to rewind successful videos to generate negative preference pairs. For a given trajectory $o^{\text{chosen}} = o_{1:T}$ with $\texttt{success} == 1$, we first sample a random contiguous subsegment:

   $$o_{\text{sub}} = o_{1:t_{\text{end}}}, \quad 1 \le t_{\text{end}} \le T.$$

   We then generate a *rewound* trajectory o^{rejected} by reversing the last k frames of the o_{sub} where $k \sim \mathcal{U}(1, t_{\text{end}} - t_{\text{start}})$:

   $$o^{\text{rejected}} = \left[o_{1:t_{\text{end}}}, o_{t_{\text{end}}-1:t_{\text{end}}-k+1}\right],$$

   where $[\cdot]$ denotes concatenating the videos. This procedure ensures that o^{chosen} represents the full progress along the subsegment, while o^{rejected} exhibits backward progress at the end.
4. **Subsequence progress.** For the same trajectory τ with $\texttt{success} == 1$, sample two subsequences $o_{1:t_1}, o_{1:t_2}$ with $t_1 < t_2$. We assign $o^{\text{chosen}} = o_{1:t_2}$ as it is further along in the task.

In practice, for all of these samples, we also sample the first frame randomly from the first half of the trajectory so that in datasets where the robot's starting position is consistent across trajectories, SPUR does not overfit to the robot's starting position.

3.3 Task Progress Prediction

In addition to preference prediction, SPUR also predicts the per-frame *progress* for each video as it can more directly be used for rewarding policies downstream [15]. Given a video $o_{1:T}$ with language instruction l, SPUR predicts a continuous progress value $p \in [0, 1]$ indicating the fraction of the task completed at each frame. The tokenized prompt is the same as in Equation (1) except without the second video o^2.

Specifically, a progress prediction MLP head, $\text{MLP}_{\text{progress}}$, is attached to the hidden states $h_{\langle |o_i| \rangle}$ corresponding to each frame i, thereby producing per-frame progress predictions. We train SPUR on the same data as in Section 3.2, with the exception of "Different expertise" where failed trajectories are not used for progress training as they do not have a ground truth progress to use. For a given video from a sampled trajectory $o_{1:T}$ (which can also be a subsequence), the progress prediction loss is computed as the Mean Squared Error (MSE) between predicted and ground-truth progress values:

$$\mathcal{L}_{\text{progress}} = \begin{cases} \sum_{t=1}^{T} \left(\text{MLP}_{\text{progress}}(h_{\langle |o_t| \rangle}) - \underbrace{t/T}_{\text{ground truth progress}} \right)^2, & \text{if not rewound} \\ \sum_{t=1}^{T} \left(\underbrace{\text{MLP}_{\text{progress}}(h_{\langle |o_t| \rangle}) - 0}_{\text{0 progress for mismatched tasks}} \right)^2 & \text{if wrong task} \\ \underbrace{\sum_{t=1}^{t_{\text{end}}} \left(\text{MLP}_{\text{progress}}(h_{\langle |o_t| \rangle}) - \frac{t}{T} \right)^2}_{\text{Loss for original trajectory until } t_{\text{end}}} + \underbrace{\sum_{t=1}^{k} \left(\text{MLP}_{\text{progress}}(h_{\langle |o_t| \rangle}) - \frac{t_{\text{end}} - t}{T} \right)^2}_{\text{Rewound video for } k \text{ frames from } t_{\text{end}}-1}, & \text{if rewound.} \end{cases} \tag{2}$$

We compute progress losses only for `success` trajectories, ensuring that the model learns meaningful temporal progress where the task is at least partially completed.

Overall, our final pretraining objective for SPUR is: $\mathcal{L}_{\text{preference}} + \mathcal{L}_{\text{progress}}$.

4 Experiments

Our experiments aim to study the efficacy of each component of SPUR and compare it against baseline across a wide array of reward metrics. To this end, we organize our experiments to answer the following experimental questions, in order:

- (**Q1**) Which components of SPUR contribute the most to generalizable reward prediction?
- (**Q2**) How does SPUR compare against baselines across a variety of reward metrics in unseen tasks?

Setup: We conduct experiments using the **LIBERO-90** dataset from the Lifelong Robot Learning Suite [21]. This dataset provides a diverse set of household manipulation tasks with various levels of distribution shift. Models are trained on demonstrations for 90 tasks in LIBERO-90 and evaluated on four benchmark splits: **LIBERO-10**, **Object**, **Spatial**, and **Goal**, which measure generalization across different dimensions such as goal, object, and spatial configurations. The original benchmark includes 4500 trajectories (50 per task) rendered at 128x128; following Kim et al. [43], we replay and re-render them at 256x256 and discard trajectories that did not replay successfully. We also include a corresponding set of failed trajectories constructed by replaying demonstration trajectories with added Gaussian noise on the actions.

We additionally compare on **MetaWorld** [22], specifically the 20-task training split consisting of 5 demonstrations each from Zhang et al. [15]. Correspondingly, we evaluate on the corresponding 17-task evaluation dataset across a variety of metrics proposed in Zhang et al. [15] that were shown to be reflective of downstream policy performance.

We list all dataset sizes in Table 4.

Baselines: We compare SPUR against several strong reward learning baselines:

- **ReWiND** [15]: trains a transformer-based network with a direct progress prediction objective using frozen language and image encoders along with video rewinding to simulate failed policy rollouts.
- **Generative Value Learning (GVL)** [14]: prompts a pre-trained Gemini LLM [44] with shuffled video frames to predict task progress for subsampled frames across the video sequence. We also

Table 1: **LIBERO Ablation Analysis.** Comparison of ablations across preference and progress accuracy metrics across unseen tasks in LIBERO-10, Object, Spatial, and Goal after training on LIBERO-90. – indicates metrics that are not applicable to the given model.

Category	Metric	Base Model	w/o Pref.	w/o Progress	w/o Fail. Traj.	SPUR
Preference Accuracy	Failed Trajs. ↑	0.5	0.64	0.82	0.69	**0.91**
Progress Accuracy	MSE ↓	–	0.04	–	0.04	**0.03**
	Reward Alignment ρ ↑	–	0.73	–	0.73	**0.81**

convert its progress predictions to preference predictions by comparing last-frame predicted task progress between queried trajectories.

- **RL-VLM-F** [8]: prompts a pre-trained LLM to obtain preference-based feedback predictions. We query Gemini for these preference predictions.

4.1 Q1: Which Components of SPUR Contribute the Most?

First, we ablate individual components of SPUR to measure the effect of each. For these experiments, we train exclusively on LIBERO-90 data (both success and failure) and evaluate on the unseen LIBERO-10, Object, Spatial, and Goal datasetes.

- **Base Model**: Uses the pre-trained QWEN-2.5-VL-INSTRUCT-3B model to produce preference and progress predictions via direct text prompticng.
- **w/o Preference**: Removes preference losses from the training objective. Preference accuracy is computed by using final-frame progress comparisons instead.
- **w/o Progress**: Removes progress losses from the training objective.
- **w/o Failure Data**: Removes unsuccessful trajectories from the preference training objective.

Reward Metrics. We compute: **preference accuracy** when comparing paired successful and failed trajectories, and **progress prediction accuracy** in terms of mean-squared-error (MSE) against the ground-truth progress target of successful trajectories and in terms of reward *alignment* in terms of spearman correlation (ρ), measuring how well the predicted progress is ordered with respect to the ground truth progress ordering of successful demonstrations.

Results averaged across our 4 unseen task distributions are displayed in Table 1, where the base model performs at random chance on predicting preferences. We found it almost always produced deterministically increasing progress predictions, so we do not include progress accuracy metrics. Meanwhile, removing preference predictions hurts the progress accuracy and reward alignment compared to SPUR, and removing progress predictions hurts the preference accuracy relative to SPUR. Removing failed trajectories also predictably hurts unseen failed trajectory preference accuracy. Overall, we demonstrate that SPUR performs the best across all comparisons and that each component we ablate complements each other to increase overall performance.

4.2 Q2: Reward Function Analysis in Unseen Tasks

Table 2: **LIBERO Metrics.** Baseline comparison across preference and progress accuracy metrics across unseen tasks in LIBERO-10, Object, Spatial, and Goal after training on LIBERO-90.

Category	Metric	RL-VLM-F	GVL	SPUR
Preference Accuracy	Failed Trajs.	0.39	0.65	**0.91**
Progress Accuracy	MSE ↓	–	0.07	**0.03**
	Reward Alignment ρ ↑	–	0.68	**0.81**

Now, we compare SPUR against reward model baselines across unseen tasks in both LIBERO and Metaworld. We first list **LIBERO** comparisons in Table 2 to GVL and Rl-VLM-F. All methods are

trained on the same LIBERO-90 datasets where applicable (GVL and RL-VLM-F instead prompt pre-trained, closed-source generative models). We can see that SPUR outperforms RL-VLM-F by **2.9x** and GVL by **1.4x** on preference accuracy. Additionally, it outperforms GVL with less than half the progress prediction MSE and **1.19x** improvement on reward alignment correlation.

Table 3: **Metaworld Reward Metrics.** Comparison of reward models in terms of reward alignment (ρ) on Metaworld. Baseline results taken from ReWiND [15].

Category	Metric	LIV-FT	RoboCLIP	VLC	GVL	ReWiND w/o OXE	ReWiND w/ OXE	SPUR
Reward Alignment	$\rho \uparrow$	0.55	-0.01	0.62	0.57	0.64	0.79	**0.83**

Next we compare **MetaWorld** performance against an additional set of baselines on the MetaWorld evaluation dataset from ReWiND [15]. For a more comprehensive comparison, we also include additional baselines listed in Zhang et al. [15], namely LIV-FT [16], VLC [19], and RoboCLIP [10], along with ReWiND trained with and without the Open X-Embodiment (OXE) Dataset [45] as proposed in Zhang et al. [15]. We refer readers to Zhang et al. [15] for additional details about these baselines. Results in Table 3 indicate that SPUR outperforms the best-performing model, beating ReWiND even when it's trained with additional data from OXE, and beating ReWiND's performance by **1.29x** when both models are trained on the same data (w/o OXE).

5 Conclusion

We studied the problem of learning reward functions that generalize to unseen tasks without relying on additional demonstrations or online training. To address these challenges, we introduced SPUR a unified reward learning framework that leverages a large-scale VLM backbone together with both progress-based and preference-based objectives. By combining per-timestep progress prediction with preference supervision over mismatched and rewound trajectories, SPUR learns from both successful and failed executions while producing denser and more transferable rewards. Our experiments on LIBERO and Metaworld show that each component of SPUR contributes to improved generalization, and that the full model consistently outperforms recent state-of-the-art baselines across diverse reward metrics.

Looking forward, we believe that scalable reward learning frameworks such as SPUR offer a promising path toward reducing reliance on costly demonstrations and enabling more robust robot policy training in real-world settings. Future directions include extending our framework to longer-horizon tasks, enabling cross-embodiment reward transfer including human videos, and evaluating deployment in real-robot experiments.

References

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

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

[3] H. Hu and D. Sadigh. Language instructed reinforcement learning for human-ai coordination. In *International Conference on Machine Learning (ICML)*, 2023.

[4] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. Gonzalez Arenas, H.-T. Lewis Chiang, T. Erez, L. Hasenclever, J. Humplik, B. Ichter, T. Xiao, P. Xu, A. Zeng, T. Zhang, N. Heess, D. Sadigh, J. Tan, Y. Tassa, and F. Xia. Language to rewards for robotic skill synthesis. In *Conference on Robot Learning (CoRL)*, 2023.

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

[6] Y. J. Ma, W. Liang, H. Wang, S. Wang, Y. Zhu, L. Fan, O. Bastani, and D. Jayaraman. Dreureka: Language model guided sim-to-real transfer. In *Robotics: Science and Systems (RSS)*, 2024.

[7] W. Liang, S. Wang, H.-J. Wang, O. Bastani, D. Jayaraman, and Y. J. Ma. Environment curriculum generation via large language models. In *Conference on Robot Learning (CoRL)*, 2024.

[8] 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.

[9] 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.

[10] 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 *NeurIPS*, 2023.

[11] 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.

[12] A. Adeniji, A. Xie, C. Sferrazza, Y. Seo, S. James, and P. Abbeel. Language reward modulation for pretraining reinforcement learning. In *In RLC Reinforcement Learning Beyond Rewards Workshop 2024*, 2024. URL https://arxiv.org/abs/2308.12270.

[13] 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*, 2024.

[14] Y. J. Ma, J. Hejna, A. Wahid, C. Fu, D. Shah, J. Liang, Z. Xu, S. Kirmani, P. Xu, D. Driess, T. Xiao, J. Tompson, O. Bastani, D. Jayaraman, W. Yu, T. Zhang, D. Sadigh, and F. Xia. Vision language models are in-context value learners. In *International Conference on Learning Representations (ICLR)*, 2025.

[15] 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 *9th Annual Conference on Robot Learning*, 2025. URL https://openreview.net/forum?id=XjjXLxfPou.

[16] 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.

[17] 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.

[18] 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.

[19] M. Alakuijala, R. McLean, I. Woungang, N. Farsad, S. Kaski, P. Marttinen, and K. Yuan. Video-language critic: Transferable reward functions for language-conditioned robotics. In *Transactions on Machine Learning Research (TMLR)*, 2025.

[20] 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.

[21] 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.

[22] T. Yu, D. Quillen, Z. He, R. Julian, A. Narayan, H. Shively, A. Bellathur, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019.

[23] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In *NeurIPS*, 2017.

[24] 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.

[25] 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.

[26] 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.

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

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

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

[30] 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.

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

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

[33] 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.

[34] J. Ho and S. Ermon. Generative adversarial imitation learning. In *NeurIPS*, 2016.

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

[36] 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 S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *NeurIPS*, 2018.

[37] 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.

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

[39] 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.

[40] 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 *Proceedings of The 2nd Conference on Lifelong Learning Agents*, pages 120–136, 2023.

[41] 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*, 2024.

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

[43] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

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

[45] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Bharadhwaj, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Vakil, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala,

K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart'in-Mart'in, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Tulsiani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Kumar, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. In *International Conference on Robotics and Automation (ICRA)*, 2024.

Table 4: Dataset

Dataset Splits	
Dataset	**Num Trajectories**
LIBERO90	3950
LIBERO10	388
LIBERO-Goal	432
LIBERO-Spatial	433
LIBERO-Object	456
LIBERO90 Failure	4312
LIBERO10 Failure	498
MetaWorld Train	100
MetaWorld Eval	85

Table 5: Configuration Parameters for SPUR Training

Training Configuration for RFM	
Parameter	**Value**
Base Model	`Qwen/Qwen2.5-VL-3B-Instruct`
Max frames (downsampled)	16
Per device training batch size	16
Learning rate	2e-5
Training steps	5000
Max sequence length	1024
LR scheduler	Cosine
Warmup ratio	0.1
Expertise / Task / Rewind / Subsequence ratio	[0.3, 0.3, 0.4, 0.0]