VLM-AR3L: Vision-Language Models for Absolute and Relative Rewards in Reinforcement Learning

Kuan-Chen Chen[1], **Winston Chen**[1], **Wei-Fang Sun**[2] and **Min-Chun Hu**[1]
[1]Department of Computer Science, National Tsing Hua University
[2]NVIDIA AI Technology Center (NVAITC)

Abstract

Designing effective reward functions remains a major challenge in reinforcement learning (RL), particularly in open-ended environments where task goals are abstract and difficult to quantify. In this work, we present VLM-AR3L, a framework that leverages Vision-Language Models (VLMs) to provide both absolute and relative rewards for RL. VLM-AR3L interprets an agent's visual observations in the context of a natural language task goal, and learns both absolute and relative rewards from VLM-generated preference labels. The absolute reward model predicts scalar evaluations for individual states, while the relative reward model compares consecutive observations to infer progress or regression toward the task goal. Their integration combines the stability of state-based evaluation with the robustness of comparative supervision. We evaluate VLM-AR3L across benchmarks spanning classic control, manipulation, and open-world embodied tasks, with a particular focus on Minecraft given its visual complexity and long-horizon decision-making requirements. Experimental results show that VLM-AR3L consistently outperforms prior VLM-based reward learning methods. Videos and code are available on the project website: https://vlm-ar3l.github.io/.

1 Introduction

Designing effective reward functions remains a major challenge in reinforcement learning (RL) [Laud, 2004; Leike *et al.*, 2018; OpenAI, 2019; Gupta *et al.*, 2022]. This challenge is especially pronounced in open-ended or visually complex environments, where task goals are often abstract and difficult to specify precisely. For example, in open-world environments such as Minecraft, agents are often assigned high-level goals (e.g., "build a shelter") that require long-horizon planning and semantic understanding, while intermediate states provide little explicit reward signal. As a result, hand-crafting reward functions in such settings is typically impractical, due to ambiguous objectives, sparse feedback, and the absence of privileged state information. To address this bottleneck, recent approaches have explored leveraging large pre-trained models as sources of reward, enabling agents to learn from high-level task descriptions or multimodal alignment signals.

Large language models (LLMs) have been shown to provide structured code or textual feedback for RL agents in text-based or programmatic settings. However, these methods often rely on privileged state access or handcrafted programmatic interfaces, limiting their applicability in real-world visual domains. When working with visual observations, contrastive vision-language models (cVLMs) such as CLIP are widely used to compute similarity scores between observations and goals for reward shaping. While simple and scalable, these methods typically require task-specific fine-tuning or retraining to be effective across different domains.

To move beyond direct similarity-based alignment, preference-based reward modeling has emerged as a promising alternative. These methods learn reward functions by comparing observation pairs and inferring preferences that are typically obtained from human feedback or large pre-trained models. This formulation enables finer-grained feedback and greater robustness in visually diverse or ambiguous environments. For instance, RL-VLM-F [Wang *et al.*, 2024] leverages generative vision-language models (VLMs) to generate pairwise preferences, which are then used to train a reward model that assigns scalar values to individual states. We call this approach **absolute reward**. However, we observe that absolute reward signals can be inconsistent across training steps (Fig. 1), due to the continual exposure to newly encountered states as training progresses. These inconsistencies, in turn, make long-horizon tasks difficult to optimize. In particular, absolute reward formulations fundamentally collapse in cyclic task structures or under ambiguous state orderings. In these scenarios, progress cannot be uniquely defined by a global state ordering, a limitation illustrated in Fig. 2. To address these limitations, we propose VLM-AR3L (Fig. 3), a framework that leverages VLMs to provide both absolute and relative rewards in reinforcement learning, without requiring fine-tuning or access to privileged internal environment information. Our contributions are as follows:

- **Dual-reward Design:** We introduce a dual-reward framework that jointly learns absolute and relative reward from VLM-generated preferences, with the relative reward providing robustness to reward inconsistency during training (Fig. 1) and resolving tasks with ill-defined global state ordering (Fig. 2).

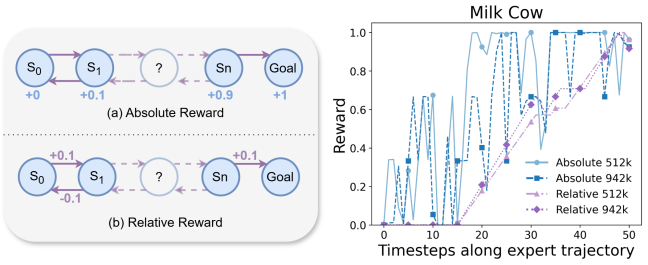

Figure 1: **Comparison of absolute and relative reward formulations. Left:** Conceptual illustration in a simple Markov process. Absolute reward assigns scalar values to states, where higher scores indicate proximity to the goal. Relative reward supervises transitions by assessing progress between state pairs. **Right:** Empirical reward trajectories evaluated along the same expert demonstration at different training steps (512k and 942k), showing that absolute reward varies substantially across training, whereas relative reward, after cumulative summation, exhibits greater temporal consistency.

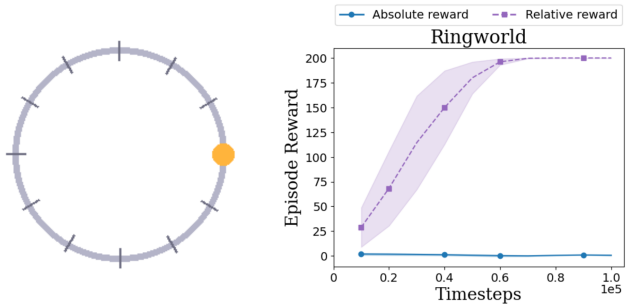

Figure 2: **RingWorld environment and reward comparison. Left:** Illustration of the RingWorld task, where the agent is required to move clockwise along a closed loop. **Right:** Learning curves for agents trained with absolute or relative reward, showing that absolute reward fails under the cyclic state ordering, whereas relative reward enables successful learning.

- **Relative reward modeling:** We train a Siamese network with VLM-generated preference supervision to model relative reward, avoiding repeated VLM inference during policy optimization, which substantially reduces computational overhead and improves performance.
- **Comprehensive evaluation:** We evaluate VLM-AR3L across diverse domains, including structured robotic control and open-ended tasks. VLM-AR3L consistently outperforms prior VLM-based methods and even surpasses carefully human-designed reward functions. We further conduct detailed ablation studies and VLM accuracy analyses to assess the reliability and scalability of our approach.

2 Related Work

LLM-based Reward Learning. Recent research has explored the use of large pre-trained models, particularly large language models (LLMs), as reward functions for reinforcement learning agents. LLM-based approaches have demonstrated the ability to generate structured code for downstream training [Ma *et al.*, 2024; Xie *et al.*, 2024; Wang *et al.*, 2023a; Yu *et al.*, 2023; Wang *et al.*, 2023b], or to provide scalar reward signals from textual task descriptions in language-conditioned or text-based environments [Kwon *et al.*, 2023; Du *et al.*, 2023; Chu *et al.*, 2023]. However, many of these methods assume structured environments or require privileged state access, making them difficult to apply in high-dimensional, real-world visual domains. Additionally, for many open-ended or complex tasks, it is inherently challenging to specify reward logic in code, particularly when task-specific abstractions are involved or when full observability is lacking.

CLIP-based Reward Learning. When working with visual observations, a common strategy is to use CLIP-based vision-language models (cVLMs) to compute similarity-based scores between agent observations and language goals, typically via cosine similarity in the shared embedding space [Cui *et al.*, 2022; Rocamonde *et al.*, 2024; Sontakke *et al.*, 2023; Mahmoudieh *et al.*, 2022; Palo *et al.*, 2023; Nam *et al.*, 2023]. While conceptually simple, these scores are often noisy and fragile, exhibiting high sensitivity to the phrasing of goals. To mitigate this, methods such as MineDojo [Fan *et al.*, 2022] and CLIP4MC [Jiang *et al.*, 2024] fine-tune CLIP on domain-specific datasets (e.g., Minecraft videos) to generate denser and more accurate reward signals. Other approaches like LIV [Ma *et al.*, 2023] and FuRL [Fu *et al.*, 2024] propose tailored fine-tuning procedures to enhance reward quality. While these methods reduce dependence on explicit ground-truth reward annotations, they typically require task-specific fine-tuning to capture subtle variations in visual input. Without fine-tuning, CLIP-based methods often fail to provide nuanced or reliable reward signals, limiting their generalization to new tasks or environments.

Preference-based Reward Learning. Beyond similarity-based rewards, recent works explore preference-based reward learning, which uses generative VLMs with multimodal reasoning capabilities, such as Gemini or GPT. A notable direction involves generating preference-based supervision without human labels. For instance, Motif [Klissarov *et al.*, 2023] use LLMs to produce preference labels or intrinsic rewards by comparing model outputs under high-level alignment principles. Extending this paradigm to the visual domain, RL-VLM-F [Wang *et al.*, 2024] proposes learning reward functions by querying vision-language models to compare agent observation pairs and generate preference labels, which are then used to supervise reward learning. Several follow-up works further explore the use of VLMs for visual preference modeling and feedback generation in various embodied environments [Venkataraman *et al.*, 2024; Ghosh *et al.*, 2025; Liu *et al.*, 2025]. However, these methods largely adopt absolute reward formulations, which are fundamentally limited in cyclic task structures or under ambiguous state orderings, where a consistent global ordering of states is difficult to define. Motivated by this limitation, we explore a dual-reward formulation that augments absolute rewards with relative rewards, aiming to improve robustness and scalability in complex, open-ended environments.

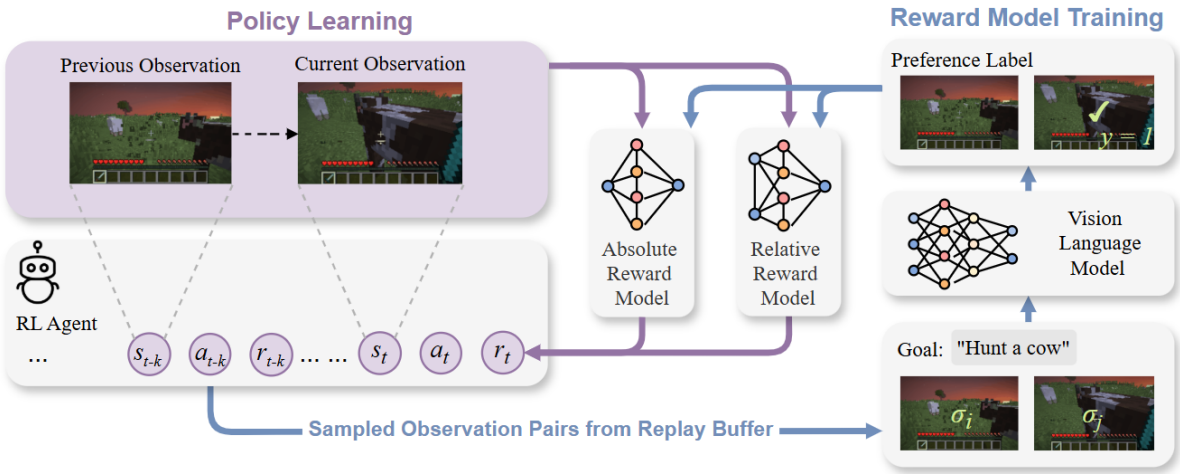

Figure 3: **Overview of the VLM-AR3L framework.** During reward model training (blue path), observation pairs are sampled from the agent's replay buffer and evaluated by a VLM to determine which observation better aligns with the task goal. The resulting preference labels are used to supervise both absolute and relative reward models. During policy learning (purple path), the trained absolute model evaluates individual observations, while the relative model compares the agent's current and previous observations to provide complementary absolute and relative reward signals.

3 Background

We consider a standard reinforcement learning (RL) setting modeled as a partially observable Markov decision process (POMDP) [Kaelbling *et al.*, 1998], defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{O}, \mathcal{R}, \gamma)$. Here, $\mathcal{S}$ denotes the set of environment states, $\mathcal{A}$ denotes the set of possible actions, $\mathcal{T}$ denotes the (unknown) state transition dynamics, $\mathcal{O}$ denotes the observation space, $\mathcal{R}$ denotes the reward function, and $\gamma \in [0, 1)$ denotes the discount factor that prioritizes immediate rewards over long-term ones. At each time step t, the agent receives an observation $s_t \in \mathcal{O}$ (e.g., an RGB image) and selects an action $a_t \in \mathcal{A}$, which causes the environment state in $\mathcal{S}$ to evolve according to $\mathcal{T}$. The objective is to learn a policy π that maximizes the expected cumulative discounted reward over time.

Preference-Based Reinforcement Learning. Our work builds on the framework of preference-based reinforcement learning (pbRL), where agents learn reward functions from pairwise comparisons over trajectory segments or behavior snippets [Christiano *et al.*, 2017; Ibarz *et al.*, 2018; Lee *et al.*, 2021a; Lee *et al.*, 2021b]. In pbRL, a trajectory segment σ typically consists of a sequence of states $\{s_1, \ldots, s_H\}$, where H denotes the sequence length of the segment. In this paper, we consider the special case that $H = 1$, as multi-image or video-based VLM reasoning remains less reliable in complex visual environments. Given a pair of segments (σ_i, σ_j), an annotator provides a preference label $y \in \{0, 1, -1\}$, where $y = 0$ indicates that σ_i is preferred, $y = 1$ that σ_j is preferred, and $y = -1$ that the two are equally preferable or incomparable. Given a reward function r over the states, we follow the standard Bradley-Terry model [Bradley and Terry, 1952] to compute the preference probability of a pair of segments:

$$\mathbb{P}[\sigma_i \succ \sigma_j] = \frac{\exp(r(s_i))}{\sum_{k \in \{i,j\}} \exp(r(s_k))}. \tag{1}$$

The reward model is then trained by minimizing a cross-entropy loss over a dataset of annotated preferences:

$$\mathcal{L}_{\text{reward}}(\theta) = -\mathbb{E}_{(\sigma_i, \sigma_j, y) \sim \mathcal{D}} \Big[\mathbb{1}_{\{y=0\}} \log \mathbb{P}_\theta[\sigma_i \succ \sigma_j] + \mathbb{1}_{\{y=1\}} \log \mathbb{P}_\theta[\sigma_j \succ \sigma_i] \Big], \tag{2}$$

where θ denotes the learnable parameters of the reward model.

In typical pbRL, the reward model and policy are trained iteratively: the reward model is updated using newly collected preference annotations, and the policy is optimized with standard RL algorithms using the learned reward model.

4 Method

Fig. 3 illustrates the overall architecture of the proposed VLM-AR3L framework. The training process consists of two intertwined pipelines: (1) reward model training and (2) policy learning. In the reward model training phase, observation pairs (σ_i, σ_j) are sampled from the agent's replay buffer, along with a high-level language goal g. Each triplet (σ_i, σ_j, g) is passed to a frozen VLM, which outputs a preference label y indicating which observation better reflects progress toward the goal. These preference labels are used to supervise the training of both an **absolute reward model** and a **relative reward model**. During policy learning, the absolute reward model assigns state-based evaluations to the current observation s_t, while the relative reward model estimates progress by comparing the current observation s_t with a previous observation s_{t-k}, jointly providing dense and informative reward signals at each timestep. These signals guide the agent's policy updates using standard reinforcement learning algorithms, eliminating the need for hand-crafted rewards or privileged state information.

4.1 Absolute Reward Model

Absolute reward models assign scalar values to individual states, where higher scores reflect states that are closer to the goal. As illustrated in Fig. 1, this state-based formulation provides a stable evaluation signal when a global ordering over the state space is well-defined, as each state is mapped to a fixed reward value independent of the agent's trajectory.

Given a parameterized absolute reward function F_ψ^{abs} over the states, the model outputs a scalar reward value for each state following the formulation in Eq. (1):

$$F_\psi^{\text{abs}}(s_t) = r_\psi(s_t), \tag{3}$$

where $r_\psi(s_t) \in \mathbb{R}$ denotes the absolute reward assigned to state s_t. The absolute reward at time step t is then given by:

$$R_\psi^{\text{abs}}(s_t) = F_\psi^{\text{abs}}(s_t). \tag{4}$$

4.2 Relative Reward Model

In contrast to the absolute reward model, the relative reward model evaluates progress between pairs of states. This design reduces reliance on a global understanding of the task space and improves robustness in partially observable or long-horizon scenarios. When new states are introduced and the reward models are updated, the relative reward of previously seen transitions remains stable. In contrast, absolute reward often suffers from instability, where earlier states may receive drastically different scores due to shifts in global normalization or value scaling (Fig. 1). Such robustness is especially beneficial in long-horizon settings, where consistent reward signals are essential for learning.

To implement the relative model, we employ a Siamese neural network [Bromley *et al.*, 1993] to learn a relative function F_ϕ^{rel}. Given a pair of states (s_i, s_j), the network outputs the preference probability:

$$F_\phi^{\text{rel}}(s_i, s_j) = \mathbb{P}_\phi[\sigma_i \succ \sigma_j]. \tag{5}$$

The relative model is trained using the same loss defined in Eq. (2).

During policy learning, the relative model aims to assess whether a transition indicates progress toward the goal g. To improve the model's sensitivity to meaningful changes, particularly when consecutive observations are highly similar, we introduce a temporal offset $k \geq 1$ and compare the current observation s_t with a past observation s_{t-k}. To ensure the reliability of the model's predictions, we apply a symmetric confidence check. Specifically, a positive reward is assigned only if the model's confidence in preferring s_t over s_{t-k} exceeds a threshold τ, and the reverse prediction also indicates that s_{t-k} is not preferred over s_t with confidence exceeding τ. This bidirectional check ensures both strong and consistent preference. Formally, the relative reward at time step t is computed as:

$$R_\phi^{\text{rel}}(s_t, s_{t-k}) = \begin{cases} 1, & \text{if } p_t^+ > \tau \text{ and } p_t^- < 1-\tau, \\ -1, & \text{if } p_t^- > \tau \text{ and } p_t^+ < 1-\tau, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

$$\text{where} \quad p_t^+ = F_\phi^{\text{rel}}(s_t, s_{t-k}), \quad p_t^- = F_\phi^{\text{rel}}(s_{t-k}, s_t).$$

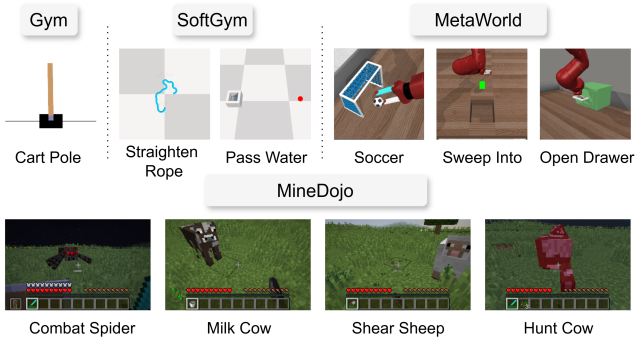

Figure 4: Environments and tasks used in our experiments, spanning structured control (Gym, SoftGym, MetaWorld) and open-ended visual domains (MineDojo).

4.3 Dual-Reward Design

Absolute and relative reward models offer complementary perspectives, balancing their individual strengths and weaknesses. Absolute rewards provide stable, state-based evaluations when a global ordering of the state space is well-defined, whereas relative rewards offer robust, transition-level supervision that remains reliable in cyclic, long-horizon, or partially observable settings. Motivated by this complementarity, we combine both reward signals to guide policy learning. The dual-reward is computed as:

$$R_{\psi,\phi}^{\text{dual}}(s_t, s_{t-k}) = \alpha R_\psi^{\text{abs}}(s_t) + (1-\alpha) R_\phi^{\text{rel}}(s_t, s_{t-k}), \tag{7}$$

where α is the weighting coefficient for the absolute reward signals, and both absolute and relative rewards are normalized to the same bounded range. Although the framework learns two reward models, both are trained using a single shared VLM querying pipeline.

5 Experiments

5.1 Setup

Environments and Tasks. To assess the proposed VLM-AR3L, we conduct evaluations on ten tasks across four distinct environments (Fig. 4): *Cart Pole* from Gym [Brockman *et al.*, 2016]; *Straighten Rope* and *Pass Water* from SoftGym [Lin *et al.*, 2020]; *Soccer*, *Sweep Into*, and *Drawer Open* from MetaWorld [Yu *et al.*, 2019]; and *Combat Spider*, *Milk Cow*, *Shear Sheep*, and *Hunt Cow* from MineDojo [Fan *et al.*, 2022]. MineDojo, in particular, features long-horizon, open-ended tasks built on top of Minecraft. These tasks demand multi-step reasoning, complex object interaction, and semantic goal understanding, thereby posing significant challenges for visual reward learning. Collectively, these benchmarks encompass both structured robotic manipulation and expansive open-world scenarios, facilitating a comprehensive evaluation of VLM-AR3L's generalization and adaptability.

Baselines. We evaluate our approach against several baselines that, like our framework, generate reward functions using only text goals and visual observations, without relying on privileged internal environment state information. In addition, we include oracle baselines using ground-truth rewards

Model	Cart Pole	Straighten Rope	Pass Water	Soccer	Sweep Into	Drawer Open	Combat Spider	Milk Cow	Shear Sheep	Hunt Cow
Gemini-2.0-Flash [DeepMind, 2024]	**0.91**	**0.82**	0.70	**0.87**	**0.87**	**0.84**	0.82	0.78	0.80	**0.76**
GPT-4.1-nano [OpenAI, 2025]	0.66	0.81	0.65	0.60	0.81	0.65	0.64	0.62	0.66	0.66
Phi-3.5-Vision-Instruct	0.56	0.67	0.62	0.52	0.69	0.54	0.63	**0.90**	**0.90**	0.66
MiniCPM-o-2.6 [OpenBMB, 2025]	0.53	0.61	**0.73**	0.65	0.79	0.56	0.59	0.61	0.62	0.64
DeepSeek-VL2-Tiny [Wu *et al.*, 2024]	0.50	0.50	0.50	0.50	0.50	0.56	**0.85**	0.68	0.66	0.67
Qwen2.5-VL-7B-Instruct [Team, 2025b]	0.50	0.76	0.70	0.55	0.52	0.67	0.64	0.59	0.72	0.64
InternVL2.5-8B	0.32	0.61	0.50	0.50	0.50	0.50	0.59	0.41	0.66	0.69
Gemma-3-12b-it [Team, 2025a]	0.52	0.51	0.50	0.50	0.50	0.50	0.50	0.50	0.54	0.50
CLIP (ViT-L/14@336px) [Radford *et al.*, 2021]	0.67	0.54	0.48	0.56	0.47	0.07	0.50	0.62	0.64	0.59
MineCLIP (Attn) [Fan *et al.*, 2022]	–	–	–	–	–	–	0.45	0.56	0.55	0.39

Table 1: Accuracy of different VLMs in predicting preference labels across tasks. Each task has 50 positive and 50 negative label queries.

Methods	Cart Pole	Straighten Rope	Pass Water	Soccer	Sweep Into	Drawer Open	Combat Spider	Milk Cow	Shear Sheep	Hunt Cow
GT Dense (Oracle)	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.04	0.38 ± 0.15	0.67 ± 0.25	0.58 ± 0.33
GT Sparse (Oracle)	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.84 ± 0.10	0.37 ± 0.01	0.47 ± 0.09	0.15 ± 0.13
CLIP	1.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	0.67 ± 0.47	0.33 ± 0.47	0.59 ± 0.16	0.37 ± 0.00	0.40 ± 0.16	0.02 ± 0.02
MineCLIP	-	-	-	-	-	-	0.11 ± 0.09	0.31 ± 0.09	0.20 ± 0.00	0.09 ± 0.08
RL-VLM-F	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.53 ± 0.36	0.50 ± 0.32	0.47 ± 0.38	0.29 ± 0.22
VLM-AR3L (Ours)	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.85 \pm 0.05}$	$\mathbf{0.95 \pm 0.00}$	$\mathbf{0.70 \pm 0.10}$	$\mathbf{0.52 \pm 0.02}$

Table 2: Success rate (mean $\pm$ standard error) of all evaluated methods across tasks. Results are averaged over 3 random seeds with 5 evaluation episodes per best checkpoint. MineCLIP is only evaluated on MineDojo tasks due to its domain-specific training.

to assess whether VLM-generated rewards can serve as practical alternatives to human-designed reward functions.

- **RL-VLM-F [Wang *et al.*, 2024]:** A state-of-the-art preference-based method that queries a VLM for pairwise preferences between observations to train a scalar-valued reward function over individual states.
- **CLIP [Rocamonde *et al.*, 2024]:** A similarity-based method using cosine similarity between CLIP embeddings of the current observation and the language goal for reward shaping.
- **MineCLIP [Fan *et al.*, 2022]:** A domain-adapted CLIP variant fine-tuned on Minecraft videos, as proposed in MineDojo. Results are not directly comparable to the original paper, as self-imitation learning is omitted to ensure a fair and consistent training protocol across baselines.
- **GT Dense Reward (Oracle):** Hand-crafted, incremental reward functions based on full access to environment ground truth.
- **GT Sparse Reward (Oracle):** Binary success/failure signals given only at episode completion.

We exclude FuRL [Fu *et al.*, 2024], RoboCLIP [Sontakke *et al.*, 2023] and CLIP4MC [Jiang *et al.*, 2024] from our comparisons for the following reasons: FuRL relies on privileged reward signals, which falls outside our problem setting; RoboCLIP has been reported to perform poorly under comparable assumptions (as reported in RL-VLM-F); and CLIP4MC does not release its fine-tuned CLIP model, precluding an identical reproduction. For policy optimization, we employ Soft Actor-Critic (SAC) [Haarnoja *et al.*, 2017] for Gym, SoftGym, and MetaWorld tasks, and Proximal Policy Optimization (PPO) [Schulman *et al.*, 2017] for MineDojo tasks, selected based on environment characteristics.

5.2 VLM's Label Accuracy

To assess the reliability of using VLMs as supervision, we evaluate several VLMs on multi-image reasoning across our tasks. For each task, we construct 100 image pairs with balanced pseudo preference labels (a 50/50 split) and prompt each VLM to identify the image that better aligns with the task goal.

Table 1 summarizes the accuracy of the tested VLMs across tasks. This evaluation serves to identify which models are most suitable as automated annotators for learning relative rewards. Gemini-2.0-Flash stands out for its consistently superior performance, achieving over 70% accuracy in all tasks, reflecting its robust multimodal reasoning capabilities. In contrast, GPT-4.1-nano underperforms relative to Gemini, showing noticeably lower accuracy across most tasks. Besides Gemini and GPT models, several open-source VLMs demonstrate competitive performance on specific tasks despite lacking general robustness. For example, Phi-3.5-Vision-Instruct achieves 90% accuracy on the Milk Cow and Shear Sheep tasks, while DeepSeek-VL2-Tiny excels in the Combat Spider task with an accuracy of 85%. MiniCPM-o-2.6 performs well in Pass Water, reaching 73%. On the other hand, CLIP and MineCLIP (evaluated only in MineDojo tasks) exhibit weak performance, underscoring their limitations in modeling multi-step interactions and temporal dependencies.

5.3 Comparison with Baselines

Building on the evaluation in Section 5.2, we adopt Gemini-2.0-Flash as the primary VLM for both VLM-AR3L and RL-VLM-F, given its consistently superior annotation performance across tasks. Table 4.3 reports the best success rates of all evaluated methods, and Fig. 5 presents the corresponding learning curves. We first examine similarity-based baselines such as CLIP and MineCLIP. While CLIP performs well in the low-dimensional Cart Pole environment, it struggles significantly in more complex environments. Notably, MineCLIP fails to generate reliable reward signals despite being fine-tuned on Minecraft videos. These results align with prior findings [Fu *et al.*, 2024; Jiang *et al.*, 2024; Wang *et al.*, 2024], suggesting that CLIP-based models often lack the granularity necessary to distinguish subtle visual dif-

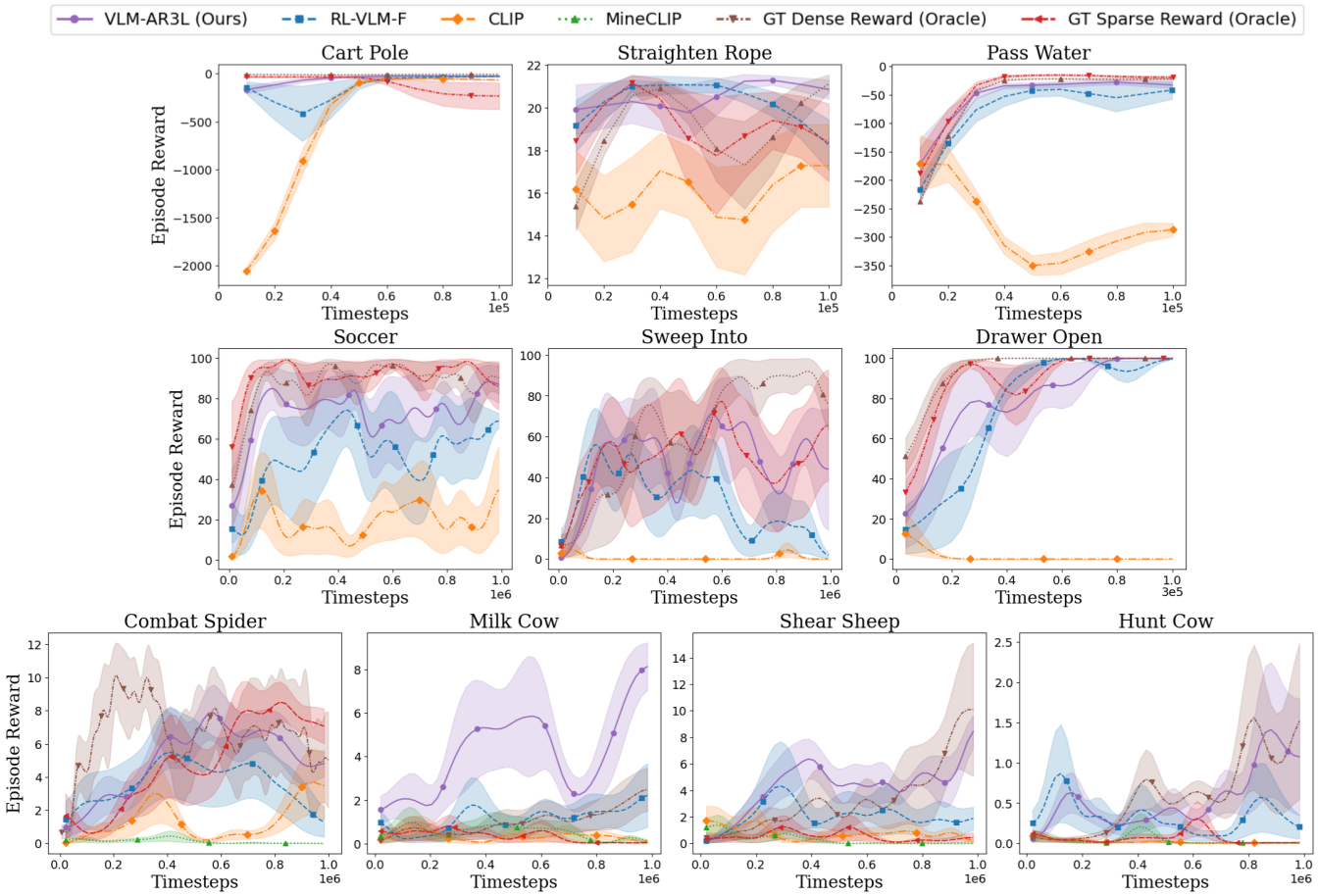

Figure 5: **Learning curves of all evaluated methods across tasks.** The x-axis represents training timesteps. The y-axis denotes episode rewards measured by the ground-truth dense reward. Results are averaged over 3 random seeds with 5 evaluation episodes per checkpoint. Shaded regions indicate standard error.

ferences, resulting in noisy and unstable reward signals during training.

Next, we evaluate RL-VLM-F, a preference-based method that learns scalar rewards over individual states. While RL-VLM-F achieves performance comparable to our method on simpler tasks, its efficacy degrades substantially in long-horizon and visually complex domains such as MineDojo. We attribute this degradation to its reliance on absolute state-based rewards, which requires a global understanding of the task space and often lacks robustness to novel or unseen state variations. In contrast, our proposed VLM-AR3L leverages relative comparisons between transitions, enabling it to generate more stable and goal-consistent rewards even as the state distribution evolves during training.

Finally, when compared against ground-truth dense and sparse rewards, our VLM-AR3L shows strong potential as a practical alternative. In most tasks, VLM-AR3L matches the performance of GT Dense Reward. Moreover, in tasks such as Milk Cow, Shear Sheep, and Hunt Cow, where GT Sparse Reward fails to provide sufficient guidance for learning, VLM-AR3L significantly outperforms the baseline, demonstrating its ability to handle sparse feedback and challenging long-horizon scenarios without requiring handcrafted rewards. These results underscore the practical applicability of our method for real-world settings where reward design is often infeasible.

5.4 Ablation Studies

The Impact of VLM. While Section 5.2 evaluates VLMs as annotators in terms of preference-label accuracy, we further study how the choice of VLM affects end-to-end RL performance when used to generate training supervision for VLM-AR3L. Specifically, we instantiate the same training pipeline and only vary the annotator VLM (Gemini-2.0, Phi-3.5, MiniCPM-o 2.6) across all tasks (Fig. 6). Overall, the performance trends are broadly consistent with the label accuracy results in Section 5.2. While Phi-3.5 and MiniCPM-o 2.6 do not excel across all tasks, both exhibit clear task-specific strengths. Given their relatively small model sizes and open-source availability, they offer promising potential for broader adoption in future applications.

The Impact of Relative Reward Architecture. An alternative design choice for relative reward, instead of explicitly training a Siamese model (denoted as Siamese in Fig. 7), is

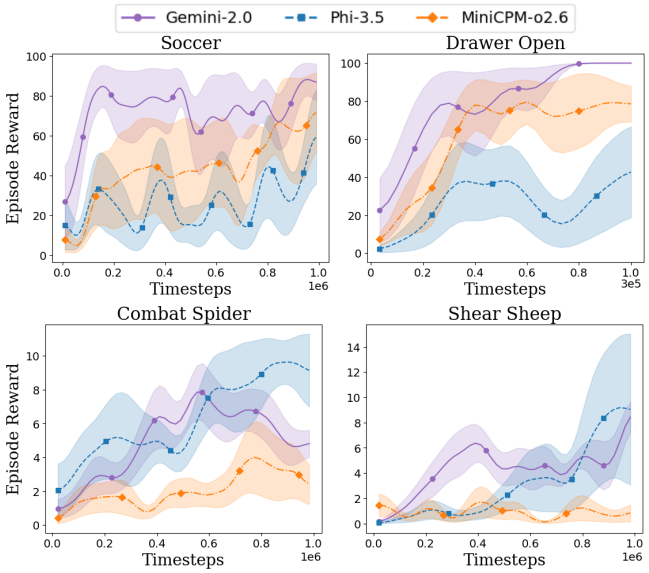

Figure 6: Ablation results with different vision-language models.

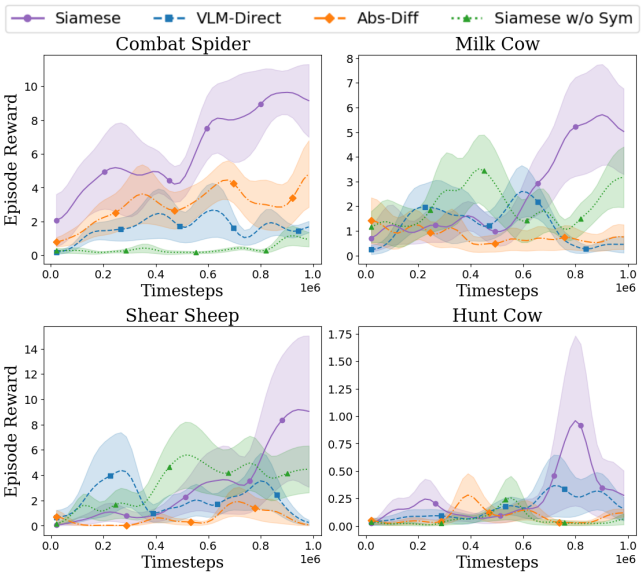

Figure 7: Ablation studies of the relative reward architecture across all experiments, using Phi-3.5 as the VLM.

to directly query a VLM at each timestep to evaluate progress (denoted as VLM-Direct). Another baseline derives relative rewards based on differences between absolute reward predictions (denoted as Abs-Diff). As shown in Fig. 7, explicitly training a Siamese model consistently outperforms direct VLM querying during rollouts (VLM-Direct). This improvement stems from preference-based training, which reduces noise in VLM judgments by learning a more stable comparison function, while also lowering computational overhead by avoiding repeated VLM queries at inference time. We observe that computing relative rewards from absolute reward estimates (Abs-Diff) leads to inferior performance. This is because when the absolute reward model fails to assign reliable scores to individual states, the resulting differences become noisy and unreliable, leading to inaccurate relative assessments. We also examine the role of the symmetric comparison rule in Eq. (6) by evaluating a variant that omits this consistency check (Siamese w/o Sym). Removing the rule leads to notable performance drops across tasks, as it no longer filters inconsistent VLM judgments or prevents biased rewards when observations are visually similar.

The Impact of Absolute and Relative rewards. Fig. 8 analyzes how absolute and relative rewards contribute to each task. Absolute rewards are most beneficial in tasks with a well-defined global progress measure. In contrast, relative rewards become critical in tasks with ambiguous state ordering, cyclic structure, or long-horizon exploration. Combining the two consistently leads to further improvements, indicating that the reward signals are complementary.

The Impact of Hyper-parameters. All results above are obtained using a single shared set of hyperparameters, with $\alpha = 0.5$, $\tau = 0.52$, and $k = 16$, without task-specific tuning.

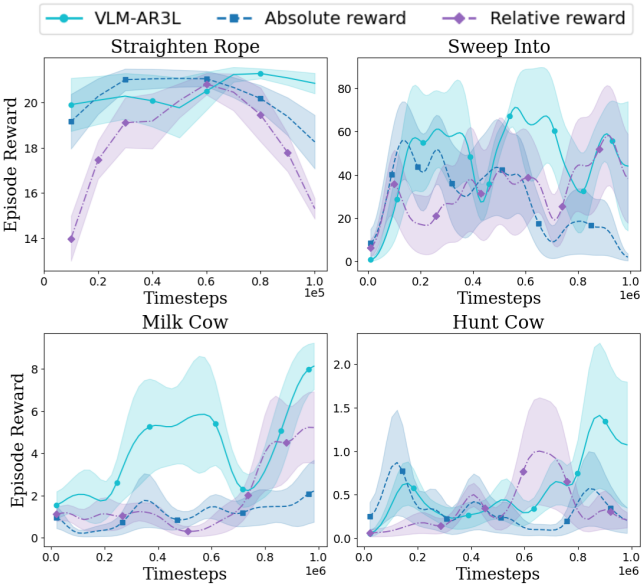

Figure 8: Ablation studies of absolute and relative rewards.

6 Conclusion

Our results demonstrate that absolute and relative rewards provide complementary supervision for reinforcement learning with VLM-generated preferences. Absolute rewards are effective when a global notion of progress is well defined, while relative rewards are more robust in scenarios involving ambiguous, cyclic, or long-horizon state transitions. Integrating both reward types leads to more stable optimization and consistently outperforms existing VLM-based reward learning methods. Notably, in several tasks, our approach even surpasses manually engineered human reward functions.

Acknowledgements

This research was funded by the National Science and Technology Council (NSTC) under grant numbers 114-2218-E-007-019. The last author was partially supported by the Hon Hai Research Institute. We express our sincere appreciation to the NVIDIA AI Technology Center (NVAITC) for the technical support and access to the Taipei-1 supercomputer.

References

[Bradley and Terry, 1952] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[Brockman *et al.*, 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[Bromley *et al.*, 1993] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, 1993.

[Christiano *et al.*, 2017] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[Chu *et al.*, 2023] Kun Chu, Xufeng Zhao, Cornelius Weber, Mengdi Li, and Stefan Wermter. Accelerating reinforcement learning of robotic manipulations via feedback from large language models, 2023.

[Cui *et al.*, 2022] Yuchen Cui, Scott Niekum, Abhinav Gupta, Vikash Kumar, and Aravind Rajeswaran. Can foundation models perform zero-shot task specification for robot manipulation? In *Learning for Dynamics and Control Conference (L4DC)*, 2022.

[DeepMind, 2024] Google DeepMind. Gemini 2.0: Introducing our new ai model for the agentic era. https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/, 2024.

[Du *et al.*, 2023] Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[Fan *et al.*, 2022] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[Fu *et al.*, 2024] Yuwei Fu, Haichao Zhang, Di Wu, Wei Xu, and Benoit Boulet. Furl: Visual-language models as fuzzy rewards for reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[Ghosh *et al.*, 2025] Udita Ghosh, Dripta S. Raychaudhuri, Jiachen Li, Konstantinos Karydis, and Amit Roy-Chowdhury. Preference vlm: Leveraging vlms for scalable preference-based reinforcement learning, 2025.

[Gupta *et al.*, 2022] Abhishek Gupta, Aldo Pacchiano, Yuexiang Zhai, Sham Kakade, and Sergey Levine. Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. In *Advances in Neural Information Processing Systems*, 2022.

[Haarnoja *et al.*, 2017] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *https://arxiv.org/abs/1801.01290*, 2017.

[Ibarz *et al.*, 2018] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. In *Advances in Neural Information Processing Systems*, 2018.

[Jiang *et al.*, 2024] Haobin Jiang, Junpeng Yue, Hao Luo, Ziluo Ding, and Zongqing Lu. Reinforcement learning friendly vision-language model for minecraft. In *European Conference on Computer Vision (ECCV)*, 2024.

[Kaelbling *et al.*, 1998] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998.

[Klissarov *et al.*, 2023] Martin Klissarov, Pierluca D'Oro, Shagun Sodhani, Roberta Raileanu, Pierre-Luc Bacon, Pascal Vincent, Amy Zhang, and Mikael Henaff. Motif: Intrinsic motivation from artificial intelligence feedback. *arXiv preprint arXiv:2310.00166*, 2023.

[Kwon *et al.*, 2023] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[Laud, 2004] Adam Daniel Laud. *Theory and application of reward shaping in reinforcement learning*. PhD thesis, University of Illinois at Urbana-Champaign, USA, 2004. AAI3130966.

[Lee *et al.*, 2021a] Kimin Lee, Laura Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *International Conference on Machine Learning*, 2021.

[Lee *et al.*, 2021b] Kimin Lee, Laura Smith, Anca Dragan, and Pieter Abbeel. B-pref: Benchmarking preference-based reinforcement learning. *arXiv preprint arXiv:2111.03026*, 2021.

[Leike *et al.*, 2018] Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction, 2018.

[Lin *et al.*, 2020] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, 2020.

[Liu *et al.*, 2025] Runze Liu, Chenjia Bai, Jiafei Lyu, Shengjie Sun, Yali Du, and Xiu Li. Vlp: Vision-language preference learning for embodied manipulation, 2025.

[Ma *et al.*, 2023] Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. *arXiv preprint arXiv:2306.00958*, 2023.

[Ma *et al.*, 2024] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models, 2024.

[Mahmoudieh *et al.*, 2022] Parsa Mahmoudieh, Sayna Ebrahimi, Deepak Pathak, and Trevor Darrell. Zero-shot reward specification via grounded natural language, 2022.

[Nam *et al.*, 2023] Taewook Nam, Juyong Lee, Jesse Zhang, Sung Ju Hwang, Joseph J. Lim, and Karl Pertsch. Lift: Unsupervised reinforcement learning with foundation models as teachers, 2023.

[OpenAI, 2019] OpenAI. Dota 2 with large scale deep reinforcement learning, 2019.

[OpenAI, 2025] OpenAI. Introducing gpt-4.1 in the api. https://openai.com/index/gpt-4-1/, 2025. Accessed: 2025-05-07.

[OpenBMB, 2025] OpenBMB. Minicpm-o 2.6: A gpt-4o-level mllm for vision, speech, and multimodal live streaming on your phone, 2025. Accessed: 2025-05-07.

[Palo *et al.*, 2023] Norman Di Palo, Arunkumar Byravan, Leonard Hasenclever, Markus Wulfmeier, Nicolas Heess, and Martin Riedmiller. Towards a unified agent with foundation models, 2023.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[Rocamonde *et al.*, 2024] Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[Sontakke *et al.*, 2023] Saurabh A Sontakke, Jiapeng Zhang, Spencer Arnold, Karl Pertsch, Emre Biyik, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[Team, 2025a] Gemma Team. Gemma 3, 2025.

[Team, 2025b] Qwen Team. Qwen2.5-vl, January 2025.

[Venkataraman *et al.*, 2024] Sreyas Venkataraman, Yufei Wang, Ziyu Wang, Zackory Erickson, and David Held. Real-world offline reinforcement learning from vision language model feedback, 2024.

[Wang *et al.*, 2023a] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv: Arxiv-2305.16291*, 2023.

[Wang *et al.*, 2023b] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. *arXiv preprint arXiv:2311.01455*, 2023.

[Wang *et al.*, 2024] Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[Wu *et al.*, 2024] Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, Zhenda Xie, Yu Wu, Kai Hu, Jiawei Wang, Yaofeng Sun, Yukun Li, Yishi Piao, Kang Guan, Aixin Liu, Xin Xie, Yuxiang You, Kai Dong, Xingkai Yu, Haowei Zhang, Liang Zhao, Yisong Wang, and Chong Ruan. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding, 2024.

[Xie *et al.*, 2024] Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2reward: Reward shaping with language models for reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[Yu *et al.*, 2019] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019.

[Yu *et al.*, 2023] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, Brian Ichter, Ted Xiao, Peng Xu, Andy Zeng, Tingnan Zhang, Nicolas Heess, Dorsa Sadigh, Jie Tan, Yuval Tassa, and Fei Xia. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*, 2023.