arXiv:2608.03875v1 [cs.LG] 4 Aug 2026

Enhancing VLM Reward Models Through Structure-Aware Fine-Tuning

Pyrros Koussios
ETH Zürich
pkoussios@ethz.ch

Chenhao Li
ETH Zürich
chhenhli@ethz.ch

Xin Chen
ETH Zürich
chexin@ethz.ch

Andreas Krause
ETH Zürich
krausea@ethz.ch

Abstract

Designing effective reward functions remains a major bottleneck in Reinforcement Learning (RL). Recent work uses large foundation Vision-Language Models (VLMs) as reward models, computing text-observation similarity to bypass manual reward engineering. Although promising, these rewards are often noisy and unreliable, limiting their direct utility during deployment. We present Structure-Aware Fine-Tuning (SAFT), a simple, self-supervised method that refines these imperfect reward signals online without access to ground-truth supervision. SAFT leverages intrinsic structural priors to regularize the VLM's latent space via LoRA adapters. We rigorously evaluate SAFT across a spectrum of base model capabilities to demonstrate its versatility. Our results show that SAFT consistently denoises the reward landscape, yielding faster policy convergence and substantially improved alignment (EPIC distance) relative to the underlying base model, suggesting that failures can often be attributed to structural brittleness rather than semantic misunderstanding. By replacing extensive human preference annotation with structural inductive biases inherent to the task, SAFT offers a scalable path for stabilizing text-conditioned RL and underscores the broader value of incorporating task structure as a general inductive bias.

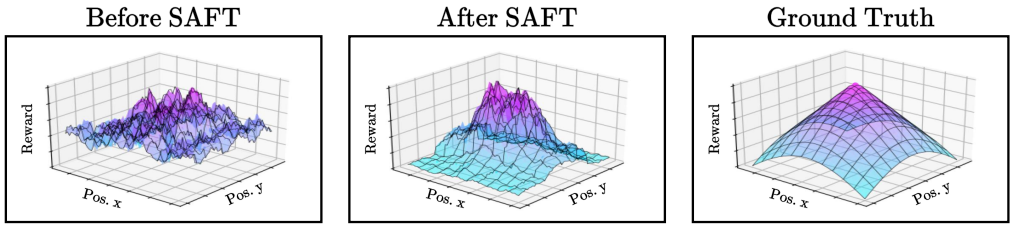

Figure 1: Structure-Aware Fine-Tuning (SAFT) improves VLM reward models for RL by enforcing inherent structural priors, without ground-truth labels.

1 Introduction

Deep Reinforcement Learning (RL) has achieved strong results across many environments and applications [Levine et al., 2016, Vinyals et al., 2019], but remains limited by the need for effective reward functions. As tasks become more complex, designing state-based rewards often requires substantial human effort and iterative refinement [Ng et al., 1999]. Reinforcement Learning from Human Feedback (RLHF) [Christiano et al., 2017], and more specifically Preference-based Reinforcement Learning (PbRL) [Abdelkareem et al., 2022], offers an alternative by learning rewards from human preference comparisons. However, due to the sparse nature of preference feedback, considerable human oversight is still required [Hejna and Sadigh, 2022].

Preprint.

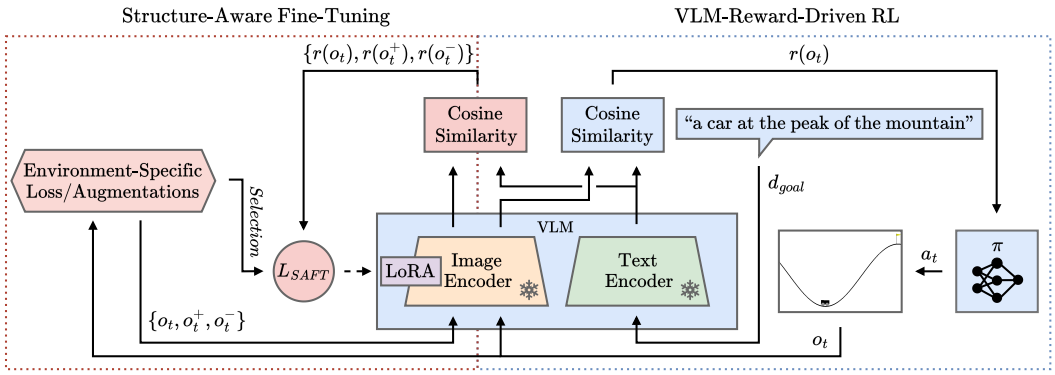

Figure 2: Structure-aware fine-tuning adapts VLM reward models online without any ground-truth rewards by updating LoRA modules with environment-specific augmentations and auxiliary losses, leveraging structural priors to improve alignment, sample efficiency, and reduce human feedback.

Recent work has explored leveraging Vision-Language Models (VLMs) as a bridge between human intent and reward specification [Wang et al., 2024, Mahmoudieh et al., 2022]. VLMs with shared vision-language embedding spaces can act as drop-in reward models by computing similarity between textual goal embeddings and visual observation embeddings. Prior work has used such rewards to derive dense signals from sparse rewards [Fu et al., 2024], improve learning from demonstrations [Sontakke et al., 2023], and train text-conditioned policies without explicit rewards or demonstrations [Rocamonde et al., 2024]. However, a critical limitation of these approaches is their dependence on VLMs to generate consistent goal-distance estimates without fine-grained task understanding, demanding generalized high-fidelity scene perception capabilities that current models cannot achieve [Sontakke et al., 2023, Rocamonde et al., 2024, Fu et al., 2024].

A central observation of this work is that such failures need not arise solely from semantic misunderstanding. In many cases, VLM rewards contain useful task information but fail to respect simple structural properties of the environment. Rather than correcting rewards with ground-truth labels or preference annotations, this suggests a broader alternative to human supervision in which reward models are improved by imposing structural priors already inherent to the task. Building on this idea, we present *Structure-Aware Fine-Tuning (SAFT)*, an online, self-supervised method for efficiently fine-tuning VLM reward models with task-inherent structural priors. SAFT enables agents to solve previously intractable tasks, accelerates policy convergence, and improves alignment with ground-truth rewards relative to non-finetuned VLM baselines, while substantially reducing the amount of human labeling required compared to preference-based alternatives.

Our contributions are threefold. **(i)** First, we propose structure-aware fine-tuning using augmentations and auxiliary losses applied to VLM reward models during online training, achieving significant sample efficiency gains across a comprehensive set of environments. **(ii)** Second, we demonstrate substantial reductions in human labeling requirements through preference-based learning evaluation, while allowing smaller models to solve previously intractable tasks. **(iii)** Third, we show an improved alignment between fine-tuned VLM reward models and ground-truth reward functions, with benefits extending beyond online performance gains.

Our experiments across two standard control benchmarks and two high-dimensional robotic manipulation tasks show that incorporating task structure into VLM reward models is useful, regardless of the particular mechanism used to impose it. SAFT instantiates this paradigm through CAL and RLR, improving sample efficiency, reducing human labeling requirements, enabling smaller models to approach the performance of larger variants, and bringing reward predictions closer to the ground truth.

2 Background

2.1 Reinforcement Learning from human feedback

Reinforcement Learning from Human Feedback (RLHF) [Christiano et al., 2017] replaces a manually specified reward function with a learned reward model inferred from human feedback. In Preference-

based Reinforcement Learning (PbRL) [Wirth et al., 2016, Christiano et al., 2017], this feedback takes the form of trajectory comparisons $(\tau_1, \tau_2, \tau_1 \succ \tau_2)$ used to train a parametric reward model $\hat{r}_\phi$.

PbRL commonly uses the Bradley-Terry likelihood

$$\mathbb{P}_\phi[\tau_1 \succ \tau_2] = \frac{\exp(\hat{R}_\phi(\tau_1))}{\exp(\hat{R}_\phi(\tau_1)) + \exp(\hat{R}_\phi(\tau_2))}, \quad \hat{R}_\phi(\tau) = \sum_{t=0}^{T-1} \hat{r}_\phi(s_t, a_t), \tag{1}$$

and minimizes the negative log-likelihood

$$L_{\text{pref}}(\phi) = -\sum_{(\tau_1,\tau_2,\tau_1\succ\tau_2)} \log \mathbb{P}_\phi[\tau_1 \succ \tau_2]. \tag{2}$$

The learned reward is then used as a proxy objective for standard RL. In observation-based VLM reward modeling, $\hat{r}_\phi$ is instantiated over observations rather than explicit state-action pairs.

2.2 CLIP-trained vision-language models

Vision-Language Models (VLMs) learn joint representations of visual and textual modalities. CLIP-trained VLMs [Radford et al., 2021] use dual encoders: a vision encoder f_{image} and a text encoder f_{text}. Given an image I and text T, the encoders produce embeddings $v = f_{\text{image}}(I)$ and $w = f_{\text{text}}(T)$ in a shared d-dimensional space. Training maximizes cosine similarity for matching pairs and minimizes it for non-matching pairs via the symmetric cross-entropy loss:

$$L_{\text{CLIP}} = -\frac{1}{2N}\sum_{i=1}^{N}\left[\log\frac{\exp(v_i \cdot w_i/\kappa)}{\sum_{j=1}^{N}\exp(v_i \cdot w_j/\kappa)} + \log\frac{\exp(v_i \cdot w_i/\kappa)}{\sum_{j=1}^{N}\exp(v_j \cdot w_i/\kappa)}\right] \tag{3}$$

where κ is a temperature parameter and N the batch size. Trained jointly on large-scale image-text pairs without task-specific supervision, the shared embedding space enables zero-shot transfer by comparing image embeddings with textual prompts.

2.3 VLM reward models

Consider a goal-based RL task specified by a textual goal description d_{goal}. At each time step t, the VLM encodes the observation o_t and goal text as $\boldsymbol{v}_t = f_{\text{image}}(o_t)$ and $\boldsymbol{w} = f_{\text{text}}(d_{\text{goal}})$. The reward is then given by their cosine similarity:

$$r_t = \cos(\boldsymbol{v}_t, \boldsymbol{w}) = \frac{\boldsymbol{v}_t \cdot \boldsymbol{w}}{\|\boldsymbol{v}_t\|\|\boldsymbol{w}\|}. \tag{4}$$

This reward can replace the true reward in standard RL, enabling the training of text-conditioned policies without modification to the underlying algorithm. Following prior work [Sontakke et al., 2023, Rocamonde et al., 2024, Fu et al., 2024], policy learning still uses the underlying state s, while the observation-based VLM reward acts as the task reward.

3 Structure-aware fine-tuning for VLM reward models

3.1 Structure-aware fine-tuning

Previous work has shown that reward models derived from foundational VLMs often fail catastrophically, with weak scene understanding and reward signals that collapse under even minor visual changes [Rocamonde et al., 2024, Fu et al., 2024]. These brittleness issues make fine-tuning essential for building a stable and accurate reward landscape. Injecting inductive bias is a natural way to encourage consistency, motivating our central question of whether task-inherent structural priors can mitigate VLM reward failures without ground-truth labels during fine-tuning.

Accordingly, we introduce Structure-Aware Fine-Tuning (SAFT), a simple LoRA-based [Hu et al., 2021] procedure that strengthens VLM reward models using auxiliary self-supervised objectives that encode the invariances and relational patterns of the target environment. SAFT implements this idea through two complementary auxiliary objectives: one that enforces invariance to task-preserving

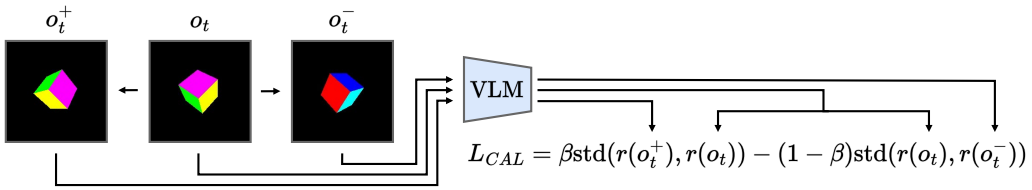

Figure 3: Contrastive Augmentation Loss, defined in (5), rewards invariance for positive augmentations while increasing separation from negative examples, preserving task semantics and avoiding representational collapse. Visualization pertains to the ReposeCube environment.

transformations and another that promotes proportionality between state changes and reward changes. Together, these losses yield reward models that respect environment symmetries, are less noisy, and align more closely with the true reward. We freeze the base VLM and insert LoRA adapters into the image encoder, fine-tuning them online with one of the two objectives described below. Figure 2 shows this process. Hardware requirements and wall-clock time increases due to fine-tuning are detailed in Appendix A.7.

3.1.1 Invariance via Contrastive Augmentation Loss (CAL)

Reinforcement learning environments often exhibit structural symmetries that should be preserved in the reward function. For instance, in CartPole, a pole tilted 15 degrees left or right from vertical corresponds to equivalent states that should yield similar rewards.

We exploit these symmetries through task-specific augmentations. Positive augmentations $\{o_{t,i}^+\}_{i=1}^p$ are a set of p transformations that should produce similar or equal rewards (e.g., horizontal flips in CartPole). Contrastive examples $\{o_{t,j}^-\}_{j=1}^n$ are a set of n semantically distinct states that should yield different rewards. These can include hard negatives (e.g., vertical flips in CartPole) or softer negatives such as other time steps within the trajectory.

Given observation o_t, our *Contrastive Augmentation Loss (CAL)* enforces reward invariance to positive transformations while maintaining discriminative capacity:

$$L_{\text{CAL}} = \beta \,\text{std}\big(r(o_t), r(o_{t,1}^+), \ldots, r(o_{t,p}^+)\big) - (1-\beta)\,\text{std}\big(r(o_t), r(o_{t,1}^-), \ldots, r(o_{t,n}^-)\big) \tag{5}$$

where $r(o) = \cos(f_{\text{image}}(o), w)$, and $\beta \in [0, 1]$ is a tunable weighting term. The first term minimizes variance among positive augmentations, enforcing invariance to semantically equivalent transformations. The second term maximizes variance among contrastive examples, ensuring the model maintains the ability to distinguish states and prevents representational collapse where all states receive similar rewards. For all experiments, we utilize $\beta = 0.5$, providing equal weighting to both terms. Ablation studies regarding this value are detailed in Appendix B.3.

We find that most environments naturally contain states with identical reward values, for example when camera viewpoints change, when background noise is present, or when states are equally related to the goal along different dimensions, making our augmentation-based approach broadly applicable across diverse tasks.

3.1.2 Proportionality via Reward Lipschitz Regularization (RLR)

In goal-directed environments, rewards often change smoothly with states within a local region, reflecting the intuition that states closer to the goal tend to have higher rewards. We encourage this structural prior through *Reward Lipschitz Regularization (RLR)*, using a soft loss-based approach rather than the hard constraints typical of traditional Lipschitz methods [Gouk et al., 2020]. A visualization of the effects of this auxiliary loss is shown in Figure 4.

Given a window of W consecutive states $\{s_i\}_{i=1}^W$, we compute the scalar mean reward $\bar{r}$ and the element-wise mean state $\bar{s}$. We then define normalized rewards as $\tilde{r}_i = r(o_i)/\bar{r}$ and normalized states as $\tilde{s}_i = s_i/\bar{s}$, where the division is also applied element-wise. With this, we define the Lipschitz regularization loss as:

$$L_{\text{RLR}} = \sum_{i=1}^{W} \sum_{j \neq i} \left(\frac{\|\tilde{r}_i - \tilde{r}_j\|_2}{\|\tilde{s}_i - \tilde{s}_j\|_2 + \epsilon} - 1 \right)^2 \tag{6}$$

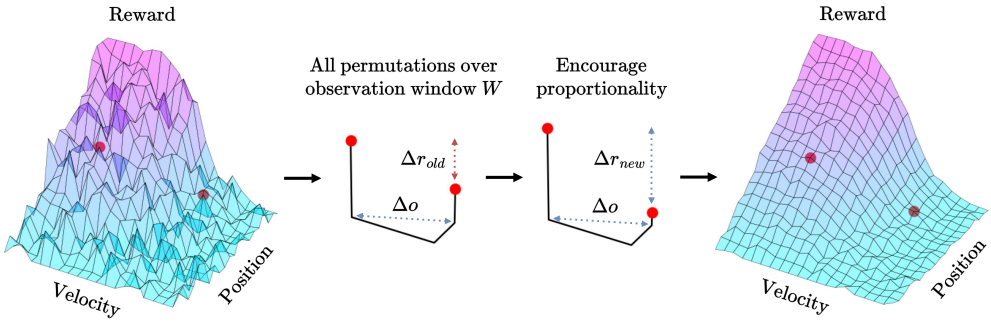

Figure 4: Reward Lipschitz Regularization (RLR), defined in (6), encourages proportional changes between observations and rewards, punishing large spikes and inconsistencies within the reward model. Visualization pertains to the MountainCar environment.

where $\epsilon > 0$ avoids division by zero. This loss penalizes deviations from proportionality between state distances and reward differences, encouraging the reward function to respect the underlying geometric structure of the state space. The window size W determines the temporal scope over which proportionality is enforced. Larger windows impose stronger structural priors by requiring consistency across longer temporal horizons. Ablation studies regarding this value are detailed in Appendix B.4.

Although accurately measuring distances between states remains an open research problem, we adopt the L2 distance because of its simplicity and broad applicability within state-based RL. This regularization is effective across diverse environments, particularly where proximity in the state space reflects progress toward goals. In settings where the underlying state is inaccessible, or in high-dimensional environments where noise dominates the state and the assumptions of RLR are difficult to satisfy, CAL is generally more effective.

4 Experiments

We conduct a variety of experiments to evaluate the efficacy of SAFT. Our evaluation tests whether imposing appropriate task structure on VLM reward models improves reward quality, policy learning, and alignment. We examine which structural priors match different environment types, whether online fine-tuning improves sample efficiency, how much human labeling effort is reduced relative to preference-based learning, and whether the resulting rewards align more closely with ground truth.

4.1 Matching structural priors to environments

We evaluate our method across four goal-based RL environments: CartPole and MountainCar from the classic control suite [Brockman et al., 2016], where VLM reward models have shown prior success, and modified versions of Reach (Franka) and ReposeCube (Allegro) from Isaac Lab [Mittal et al., 2023], representing realistic and high-dimensional robotic manipulation tasks. We highlight that the ReposeCube task, with its 72-dimensional observation space and 16-dimensional action space, represents a significant challenge on par with complex humanoid locomotion. All learning is conducted solely with the rewards provided by the VLM. We additionally note that we purposely omit the Humanoid task used in prior work [Rocamonde et al., 2024] as it lacks a well-defined ground-truth reward function. This mandates a reliance on subjective human-labeled success, which precludes a rigorous quantitative evaluation of policy performance.

The choice of auxiliary loss is determined by the structural properties available in each environment, as summarized in Table 1. We use RLR when local state distances provide meaningful information about reward differences, and CAL when the environment admits task-preserving transformations that should leave the reward unchanged. Under this criterion, CartPole supports both CAL and RLR, MountainCar supports only RLR, and Reach and ReposeCube support CAL through rotational invariances. Exact details behind these matchings are provided in Appendix A.4. These matchings are

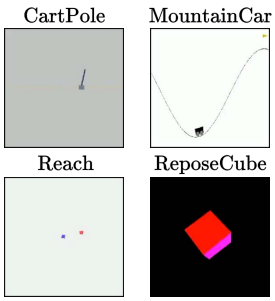

Figure 5: Visualizations of the four environments.

	Classic Control		*Isaac Lab*	
	CartPole	MountainCar	Reach	ReposeCube
CAL	✓	✗	✓	✓
RLR	✓	✓	✗	✗

Table 1: Applicability of Contrastive Augmentation Loss (CAL) and Reward Lipschitz Regularization (RLR) across all four environments.

not meant to exhaust the space of possible priors, but to test whether simple task-inherent structure can serve as useful supervision for VLM reward adaptation.

4.2 Policy performance

We first evaluate the sample efficiency of SAFT compared to baseline methods without fine-tuning, as well as goal-baseline regularization [Rocamonde et al., 2024] from prior work. We omit direct comparisons to RoboCLIP [Sontakke et al., 2023] and FuRL [Fu et al., 2024] as they operate under fundamentally different assumptions: RoboCLIP requires video-based temporal information for trajectory-level rewards, while FuRL relies on privileged ground-truth sparse reward signals for fine-tuning. Consequently, Rocamonde et al. [2024] remains the only methodologically consistent baseline for our unsupervised, image-based setting.

We emphasize that our primary evaluation metric is the *relative* improvement yielded by SAFT over a base model, rather than absolute task performance. As SAFT is designed to refine existing reward signals, its utility naturally depends on the quality of the underlying VLM, a relationship we analyze across a broader spectrum in Section 4.4.

However, to capture the nuances of online training, such as convergence speed, stability, and sample efficiency, that scalar summary metrics cannot convey, we focus our main policy analysis on a representative regime where the base VLM is functional but imperfect. We initialize the VLM at a performance threshold where it can partially solve the task, allowing us to explicitly demonstrate how SAFT accelerates learning and stabilizes the reward landscape compared to the base model. This initialization is performed through pretraining using the ground-truth reward as detailed in Appendix A.3. This controlled setting isolates the benefits of our proposed method while also allowing us to use a computationally efficient 86 million parameter ViT-B-16 model [Ilharco et al., 2021] that achieves performance comparable to larger models without the associated computational overhead. Critically, this initialization serves only to establish a consistent starting point for evaluation. We measure the *marginal utility* of our structural priors relative to the equally pretrained base model, with no ground-truth information available to the SAFT agent during the fine-tuning phase.

Our fine-tuning procedure uses LoRA adapters with 180K parameters, representing 0.2% of the total model parameters, consistent with standard practice. We evaluate performance across five seeds.

Figure 6 shows that SAFT consistently outperforms the baselines across environments by correcting reward inconsistencies and producing smoother reward landscapes. When the base VLM fails to reach ground-truth performance, SAFT often achieves a higher final reward by improving shaping in high-reward regions, helping the policy identify task-success states. Overall, SAFT substantially narrows the gap between VLM-derived and ground-truth rewards.

4.3 Reward model alignment

Next, we conduct offline evaluation to assess how well our fine-tuned VLMs align with ground-truth rewards compared to their unmodified counterparts. We employ the Equivalent-Policy Invariant Comparison (EPIC) [Gleave et al., 2021] distance as our evaluation metric, which measures differences between reward functions by comparing their canonical forms while removing the effects of reward shaping and scaling. This metric captures only the differences that affect optimal behavior, making it

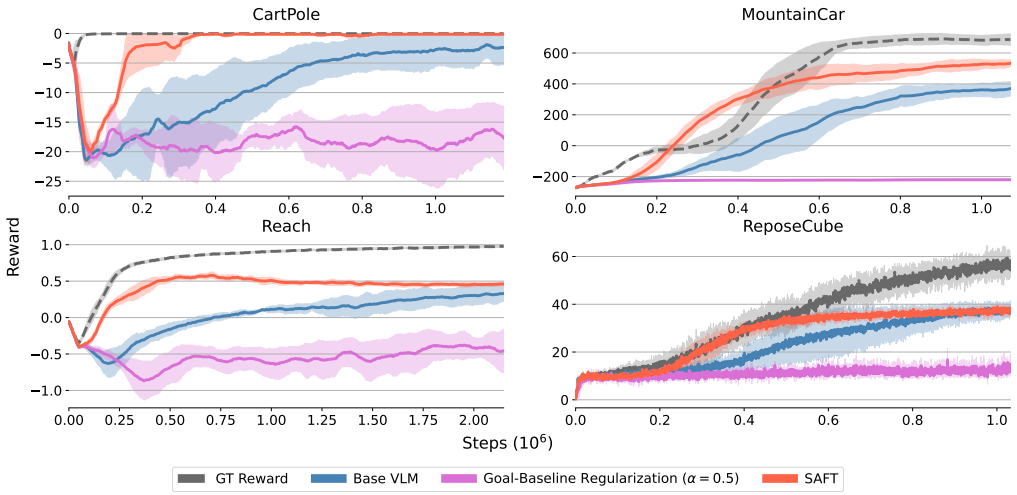

Figure 6: Policy learning curves comparing SAFT against the base VLM, goal-baseline regularization [Rocamonde et al., 2024], and Ground-Truth (GT) rewards across four environments. SAFT demonstrates superior sample efficiency and convergence speed, substantially reducing the performance gap to GT rewards. Shaded regions indicate standard deviation across five seeds.

Environment	EPIC Distance ↓	
	Before SAFT	After SAFT
Classic Control		
CartPole	0.4306	**0.2782** $\pm$ 0.0205
MountainCar	0.3106	**0.2330** $\pm$ 0.0074
Isaac Lab		
Reach	0.3443	**0.3129** $\pm$ 0.0069
ReposeCube	0.2963	**0.2605** $\pm$ 0.0098

Table 2: EPIC distance between VLM reward models and the ground-truth reward. SAFT reduces EPIC distance, indicating closer alignment to the ground truth. Post-fine-tuning results report mean and one standard deviation across five seeds.

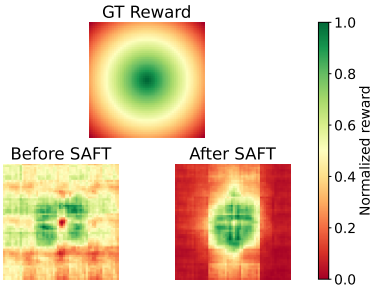

Figure 7: Reach reward landscapes predicted by the VLM with and without SAFT, showing improved alignment to the ground truth under SAFT.

ideally suited for our analysis. For this analysis, we employ the base and fine-tuned VLM checkpoints from Section 4.2.

Table 2 presents the EPIC distance measurements on the different environments. Across all environments, fine-tuned models demonstrate substantially lower EPIC distances compared to their non-fine-tuned counterparts, indicating closer alignment with ground truth reward functions. These alignment improvements suggest that structural fine-tuning improves the reward model itself, rather than merely producing policies that exploit incidental shaping effects. Figure 7 shows VLM-predicted rewards before and after applying SAFT. Further results using additional VLM backbones are provided in Appendix B.7.

4.4 Human labeling efficiency

Finally, we evaluate SAFT against preference-based reinforcement learning (PbRL), one of the most commonly used approaches for learning reward models when manually specified rewards are unavailable or unreliable [Christiano et al., 2017]. We implement a PbRL baseline under the same setup as SAFT, freezing the base VLM and updating only LoRA adapters in the image encoder. After each rollout, binary preference queries are generated and the reward model is updated online using the Bradley-Terry objective from Section 2.1. We then measure how many preference queries are required for PbRL to match the performance achieved by SAFT. Implementation details are provided in Appendix A.5.

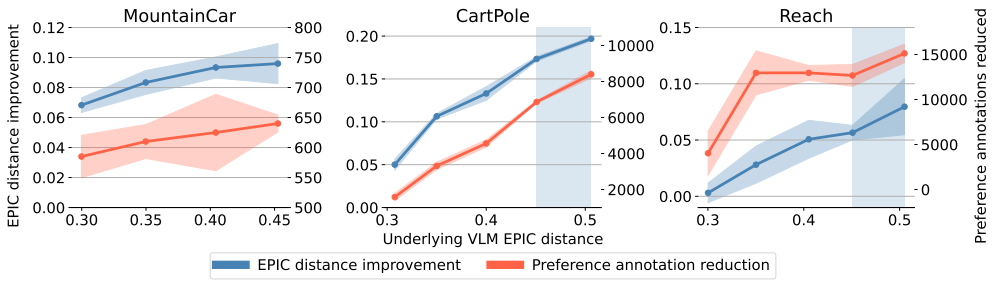

Figure 8: Utility of SAFT in relation to base VLM reward quality, averaged over two seeds. As the underlying VLM degrades, SAFT becomes increasingly useful. The critical regime, highlighted in blue, is where the base VLM fails yet SAFT enables successful training.

	Classic Control		*Isaac Lab*	
	CartPole	MountainCar	Reach	ReposeCube
Binary Comparisons Saved ↑	$5\,608 \pm 309$	670 ± 27	$13\,516 \pm 1\,682$	$8\,755 \pm 1\,711$

Table 3: Number of binary preference comparisons eliminated by using SAFT instead of human feedback, rounded to the closest integer. SAFT substantially reduces required human effort. Values show mean and one standard deviation across five seeds.

Table 3 reports the number of preference queries required by PbRL to match SAFT performance. Across environments, SAFT replaces several hundred to several thousands of binary comparisons with a one-time specification of structural priors. This both reduces human labeling effort and avoids the accumulated noise and inconsistency of online human feedback.

As stated in Section 4.2, the effectiveness of our method depends on the initial quality of the underlying VLM. An oracle VLM that perfectly models the ground-truth reward would render our improvements less impactful. To characterize this relationship, we evaluate binary feedback savings as a function of underlying VLM strength by applying SAFT to VLMs with varying levels of pretraining, as presented in Figure 8.

As the VLM reward model deviates from the ground truth, SAFT becomes increasingly valuable by correcting inconsistencies and reducing noise in the reward signal. We observe that SAFT consistently improves performance across varying VLM strengths, lowering EPIC distance and effectively substituting for costly online human labeling. Notably, for most environments, there exists a critical regime in which the base VLM alone cannot solve the task due to noisy scene understanding, yet SAFT enables successful learning by mitigating this noise. Smaller variants within the same model family can fall into this regime and fail where larger counterparts succeed. Our method bridges this gap (Appendix B.1). Past a certain point, when the VLM is fundamentally misaligned with the task, our method cannot enable policy convergence, as the auxiliary losses operate without ground-truth supervision and are intended to improve consistency and reduce noise in the reward signal, but cannot compensate for a reward model that lacks any ability to interpret the scene.

5 Related work

Self-supervised auxiliary losses for reinforcement learning. Self-supervised auxiliary losses have long improved generalization and policy performance in RL [Jaderberg et al., 2016, Shelhamer et al., 2017, Yarats et al., 2020]. Image augmentations are especially effective for vision-based policies [Laskin et al., 2020, Yarats et al., 2021], and contrastive learning methods such as CURL have also proven effective [Srinivas et al., 2020].

Another line of work has explored constraining networks to be Lipschitz continuous [Asadi et al., 2018, Scaman and Virmaux, 2019, Gouk et al., 2020]. We take a similar approach but regularize the reward function rather than enforcing hard constraints and use L2 distances between states as a baseline despite the existence of more sophisticated metrics [Jiang and Qin, 2020, Myers et al., 2025].

So far, these auxiliary losses have been applied to policy networks. To the best of our knowledge, we are the first to adapt them for online fine-tuning of VLM reward models, combining Lipschitz regularization and visual augmentations to improve reward quality.

Large pretrained models as reinforcement learning reward functions. RLHF [Christiano et al., 2017] aligns policies with human intent by training reward models from preference data, but collecting such data is costly. Recent work instead uses foundation models for automated reward specification. Early methods had LLMs generate reward signals and refine them with policy feedback [Kwon et al., 2023, Song et al., 2023], while newer ones generate reward code [Xie et al., 2024, Ma et al., 2024, Li et al., 2025]. However, the unimodality of LLMs limits these approaches to relying on the indirectness of intermediary textual representations.

Multimodal VLMs offer a promising alternative. Initial studies used them as success detectors [Cui et al., 2022, Du et al., 2023] or for generating preference labels in PbRL [Wang et al., 2024], though these approaches reduce information to binary signals or require auxiliary models. Others trained policies using embedding similarity [Mahmoudieh et al., 2022], but required large offline datasets.

Most closely related, Rocamonde et al. [2024] and Sontakke et al. [2023] use VLMs directly as reward models, with the former introducing goal-baseline regularization and the latter applying video-based models for motion goals. Fu et al. [2024] further show that fine-tuning with sparse ground-truth rewards improves dense reward quality. Our work builds on these directions, but argues that task structure can serve as a direct source of supervision for improving VLM reward models, without requiring ground-truth rewards or preference labels during online fine-tuning.

6 Limitations and future work

Our primary limitation is that the auxiliary loss, the augmentations for CAL, and the window size for RLR are chosen on an environment by environment basis, which reduces generality. However, this situation is not unique to our work. Early visual RL began with seemingly ad hoc augmentation techniques [Sadeghi and Levine, 2017, Lee et al., 2020, Cobbe et al., 2019], and over time, work such as RAD [Laskin et al., 2020] revealed randomized crops as a strategy that generalized well.

In the same spirit, we introduce ground-truth-free auxiliary losses for fine-tuning VLM reward models, playing a role for reward learning analogous to early augmentation studies for policy learning. SAFT shows that simple structural priors improve reward alignment, increase sample efficiency, and reduce human supervision. We do not claim a general purpose solution, but we view SAFT as a step toward more automated and broadly applicable approaches.

A natural next step is to replace manual choices with methods that automatically discover and combine structural priors within a unified framework that subsumes our current objectives. The text encoder remains frozen and underused, so future work could fine-tune it with contrastive goal descriptions and their textual negations to sharpen task understanding. As tasks grow in complexity, hierarchical scene decomposition and compositional augmentation may be required, for example in manipulation tasks that demand both rotation invariant grasping and position sensitive placement.

7 Conclusion

In this work, we introduce Structure-Aware Fine-Tuning (SAFT), a simple method for adapting frozen vision-language models whose purpose it is to provide rewards for reinforcement learning. SAFT incorporates inductive bias through simple structural priors, applied via LoRA adapters to reshape and denoise the reward signal. This yields smoother and more consistent reward landscapes, faster policy convergence, closer alignment with the ground truth task reward, and the ability for smaller VLMs to solve tasks that otherwise require larger models. Our experiments across both classic control and robotic manipulation environments suggest that VLM reward failures are not always caused by semantic misunderstanding alone, but can also arise from structural brittleness in the reward landscape. By showing that such brittleness can be mitigated without ground-truth supervision, SAFT supports a broader paradigm in which task structure serves as a scalable source of supervision for practical, fully text-conditioned reinforcement learning.

References

Youssef Abdelkareem, Shady Shehata, and Fakhri Karray. Advances in preference-based reinforcement learning: A review. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, page 2527–2532. IEEE, October 2022. doi: 10.1109/smc53654.2022.9945333. URL http://dx.doi.org/10.1109/SMC53654.2022.9945333.

Kavosh Asadi, Dipendra Misra, and Michael L. Littman. Lipschitz continuity in model-based reinforcement learning, 2018. URL https://arxiv.org/abs/1804.07193.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016. URL https://arxiv.org/abs/1606.01540.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2017. URL https://arxiv.org/abs/1706.03741.

Yung-Sung Chuang, Yang Li, Dong Wang, Ching-Feng Yeh, Kehan Lyu, Ramya Raghavendra, James Glass, Lifei Huang, Jason Weston, Luke Zettlemoyer, Xinlei Chen, Zhuang Liu, Saining Xie, Wen tau Yih, Shang-Wen Li, and Hu Xu. Meta CLIP 2: A worldwide scaling recipe. 2025.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning, 2019. URL https://arxiv.org/abs/1812.02341.

Yuchen Cui, Scott Niekum, Abhinav Gupta, Vikash Kumar, and Aravind Rajeswaran. Can foundation models perform zero-shot task specification for robot manipulation?, 2022. URL https://arxiv.org/abs/2204.11134.

Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando de Freitas, and Serkan Cabi. Vision-language models as success detectors, 2023. URL https://arxiv.org/abs/2303.07280.

Yuwei Fu, Haichao Zhang, Di Wu, Wei Xu, and Benoit Boulet. Furl: Visual-language models as fuzzy rewards for reinforcement learning, 2024. URL https://arxiv.org/abs/2406.00645.

Adam Gleave, Michael Dennis, Shane Legg, Stuart Russell, and Jan Leike. Quantifying differences in reward functions, 2021. URL https://arxiv.org/abs/2006.13900.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J. Cree. Regularisation of neural networks by enforcing lipschitz continuity, 2020. URL https://arxiv.org/abs/1804.04368.

Joey Hejna and Dorsa Sadigh. Few-shot preference learning for human-in-the-loop rl, 2022. URL https://arxiv.org/abs/2212.03363.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. URL https://doi.org/10.5281/zenodo.5143773.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks, 2016. URL https://arxiv.org/abs/1611.05397.

Kai Jiang and XiaoLong Qin. Reinforcement learning with goal-distance gradient, 2020. URL https://arxiv.org/abs/2001.00127.

Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=10uNUgI5Kl.

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data, 2020. URL https://arxiv.org/abs/2004.14990.

Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgcvJBFvB.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies, 2016. URL https://arxiv.org/abs/1504.00702.

Pengyi Li, Jianye HAO, Hongyao Tang, Yifu Yuan, Jinbin Qiao, Zibin Dong, and YAN ZHENG. R*: Efficient reward design via reward structure evolution and parameter alignment optimization with large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=qZMLrURRr9.

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. URL https://arxiv.org/abs/2310.12931.

Parsa Mahmoudieh, Sayna Ebrahimi, Deepak Pathak, and Trevor Darrell. Zero-shot reward specification via grounded natural language, 2022. URL https://openreview.net/forum?id=zRb7IWkTZAU.

Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi: 10.1109/LRA.2023.3270034.

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making, 2025. URL https://arxiv.org/abs/2406.17098.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 278–287, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606122.

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, 2021. URL https://arxiv.org/abs/2103.00020.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

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. URL https://openreview.net/forum?id=N0I2RtD8je.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 91–100. PMLR, 2022. URL https://proceedings.mlr.press/v164/rudin22a.html.

Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image, 2017. URL https://arxiv.org/abs/1611.04201.

Kevin Scaman and Aladin Virmaux. Lipschitz regularity of deep neural networks: analysis and efficient estimation, 2019. URL https://arxiv.org/abs/1805.10965.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. URL https://arxiv.org/abs/2210.08402.

Evan Shelhamer, Parsa Mahmoudieh, Max Argus, and Trevor Darrell. Loss is its own reward: Self-supervision for reinforcement learning, 2017. URL https://arxiv.org/abs/1612.07307.

Jiayang Song, Zhehua Zhou, Jiawei Liu, Chunrong Fang, Zhan Shu, and Lei Ma. Self-refined large language model as automated reward function designer for deep reinforcement learning in robotics, 2023. URL https://arxiv.org/abs/2309.06687.

Sumedh A Sontakke, Jesse Zhang, Sébastien M. R. Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies, 2023. URL https://arxiv.org/abs/2310.07899.

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning, 2020. URL https://arxiv.org/abs/2004.04136.

Leitian Tao, Ilia Kulikov, Swarnadeep Saha, Tianlu Wang, Jing Xu, Sharon Li, Jason E Weston, and Ping Yu. Hybrid reinforcement: When reward is sparse, it's better to be dense, 2025. URL https://arxiv.org/abs/2510.07242.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. URL https://arxiv.org/abs/2502.14786.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning, Nov 2019. ISSN 1476-4687. URL https://doi.org/10.1038/s41586-019-1724-z.

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, 2024. URL https://arxiv.org/abs/2402.03681.

Christian Wirth, Johannes Fürnkranz, and Gerhard Neumann. Model-free preference-based reinforcement learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2222–2228. AAAI Press, 2016.

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. URL https://openreview.net/forum?id=tUM39YTRxH.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images, 2020. URL https://arxiv.org/abs/1910.01741.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning, 2021. URL https://arxiv.org/abs/2107.09645.

A Implementation details

A.1 Reinforcement learning setup

For the MountainCar environment, we use the Stable-Baselines3 [Raffin et al., 2021] PPO implementation with default hyperparameters. For CartPole, Reach, and ReposeCube, we employ the RSL-RL [Rudin et al., 2022] PPO implementation, also with default hyperparameters. In all environments, the agent receives task reward exclusively from the VLM-based reward model without access to the ground-truth reward. Exact parameter values can be found in Table S4.

Table S4: Default reinforcement learning parameters from the Stable-Baselines3 [Raffin et al., 2021] and RSL-RL [Rudin et al., 2022] PPO implementations, used without modification in our experiments.

Component	Stable-Baselines3	RSL-RL
Actor Network	MLP $[64, 64]$	MLP $[256, 256, 256]$
Critic Network	MLP $[64, 64]$	MLP $[256, 256, 256]$
Epochs	10	1
Batch Size	64	`rollout_steps`
Learning Rate	$3 * 10^{-4}$	10^{-3}
Gamma (γ)	0.99	0.998
GAE Lambda (λ)	0.95	0.95
Clip Range	$[-0.2, 0.2]$	$[-0.2, 0.2]$
Entropy Coefficient	0	0
Optimizer	`Adam`	`Adam`

A.2 Vision-language model selection and LoRA finetuning

Following prior work, we adopt the ViT-B-16 backbone from the OpenCLIP [Ilharco et al., 2021] collection, pretrained on the LAION-2B dataset [Schuhmann et al., 2022], as our base vision-language model. The vision encoder is fine-tuned once per rollout using LoRA adapters [Hu et al., 2021], while the text encoder and all non-LoRA parameters of the vision backbone remain frozen throughout all experiments. The VLM reward serves as a drop-in replacement for the ground-truth reward, with the textual goal description kept fixed throughout training. Image observations are encoded by the vision backbone at every step and compared against this fixed goal embedding to produce rewards. VLM and LoRA configuration details are reported in Table S6.

A.3 Vision-language model pretraining

Table S5: Ground-truth reward definitions for the four environments. CartPole penalizes the pole's angular velocity ω_{pole}. Reach penalizes the Euclidean distance between the end-effector $\boldsymbol{x}$ and the goal $\boldsymbol{x}_{\text{goal}}$. MountainCar rewards the car's horizontal position x, and punishes large actions a. Finally, ReposeCube rewards the agent based on the angular error $\Delta\theta$ between the cube's current and goal quaternions.

Environment	Term	Weight
CartPole	$-\omega_{\text{pole}}^2$	1.0
Reach	$-\|\boldsymbol{x} - \boldsymbol{x}_{\text{goal}}\|_2$	0.2
	$1 - \tanh\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_{\text{goal}}\|_2}{\sigma}\right)$	0.1
MountainCar	$x + 0.3$	1.0
	$-a^2$	0.1
ReposeCube	$(\Delta\theta + 0.1)^{-1}$	1.0

To rigorously evaluate SAFT, we must isolate the method's structural contributions from the stochastic quality of the underlying Vision-Language Model. SAFT is designed to refine and structurally align *existing* reward signals. Therefore, its utility is naturally a function of the base model's capability. A

base model that outputs pure noise cannot be meaningfully aligned, whereas a model that perfectly reflects the ground truth requires no refinement.

To evaluate SAFT on different points of this spectrum, as presented in Figure 8, we employ supervised pretraining as a controlled initialization protocol. This allows us to smoothly interpolate between these extremes, ranging from random initialization to near-oracle performance, and demonstrate that SAFT provides consistent relative improvements regardless of the base model's starting quality. This strategy simulates deploying off-the-shelf VLMs across the capability spectrum, treating larger models as stronger initializations, while offering a smooth performance gradient for more precise evaluation than discrete model comparisons.

Pretraining methodology To achieve these targeted initialization states, we pretrain the LoRA adapters by regressing the VLM reward predictions toward the ground-truth task rewards, as listed in Table S5. We employ Mean Squared Error (MSE) as the loss function:

$$L_{\text{pretrain}} = \|r_{\text{VLM}}(o_t) - r_{\text{scaled}}(s_t)\|^2 \tag{S7}$$

where $r_{\text{VLM}}(o_t)$ is the reward output of the VLM, given an observation o_t at time t. Since this reward stems from calculating cosine similarity, and cosine similarity is strictly bounded to $[-1, 1]$, attempting to regress to arbitrary, unconstrained ground-truth values would prevent the VLM from learning the correct reward structure. To align the targets with the VLM's output space, we linearly scale the ground-truth reward $r_{\text{gt}}(s_t)$, given a state s at time t, to the $[-1, 1]$ range:

$$r_{\text{scaled}}(s_t) = 2 \cdot \frac{r_{\text{gt}}(s_t) - R_{\min}}{R_{\max} - R_{\min}} - 1 \tag{S8}$$

where $R_{\min}$ and $R_{\max}$ represent the empirical minimum and maximum rewards of the environment. During pretraining, we apply this loss after every step.

We select regression over a PbRL-style pretraining objective for this phase because regression allows for linear, low-variance control over the reward structure (measured via EPIC distance). In contrast, PbRL pretraining tends to yield sparser and noisier rewards [Christiano et al., 2017, Tao et al., 2025], making it difficult to target specific capabilities for controlled evaluation. Furthermore, we posit that direct regression serves as the most faithful proxy for the behavior of future general-purpose VLMs, as it promotes a representation that remains unbiased toward any specific component of the reward structure. However, to demonstrate generality, we also provide an evaluation of SAFT applied to a PbRL-pretrained baseline in Appendix B.6.

Selection of operating point for main results While Figure 8 demonstrates SAFT's utility across a range of model strengths, the detailed learning curves in the main text (Figure 6, Table 2, Table 3) utilize a specific initialization threshold where the base VLM is capable of partially solving the task. We identified this operating point empirically by rolling out the policy from different VLM pretraining checkpoints and visually checking when the policy showed learning and roughly solved the task. We deliberately selected this functional regime as relying solely on off-the-shelf models that often fail completely, as seen in the ViT-B-16 analysis in Figure S9, would reduce our evaluation to a binary success or failure outcome. By contrast, evaluating on a functional but imperfect model allows us to demonstrate that SAFT accelerates learning and denoises the reward landscape even when the base model is not catastrophic.

The exact pretraining budget for the base VLM in the main results of Sections 4.2 and 4.3 differs by environment. CartPole uses 30 rollouts, resulting in $30 \times 32 = 960$ individual updates, or in other words 960 epochs of batch size 1. Reach uses $48 \times 27 = 1296$. MountainCar uses 100×1, because a full rollout would be excessive. ReposeCube uses $24 \times 180 = 4320$.

Strict separation of phases We emphasize that this access to ground-truth rewards is strictly limited to the *pretraining phase* to establish experimental starting conditions. Once the SAFT fine-tuning phase begins, all access to ground truth is revoked, and the model relies exclusively on the self-supervised auxiliary losses (L_{CAL} and L_{RLR}). Thus, no ground-truth information leaks into the SAFT update process.

Off-the-shelf baselines Finally, to complement the pretrained experiments, we provide training curves comparing SAFT against an off-the-shelf ViT-B-16 model without any pretraining for all four environments in Figure S9. These results confirm that even in regimes where the base model fails to converge, SAFT is useful, sometimes even being the deciding factor that recovers a solvable policy.

Table S6: Vision-language model and LoRA training parameters used across all experiments.

Component	Configuration
Vision-Language Model	
Backbone	ViT-B-16
Parameters	86M
Source	OpenCLIP, trained on LAION-2B
LoRA Adapters	
Parameters	180k
Target Modules	Feedforward Projection (`c_fc`), Output Projection (`c_proj`)
Rank	2
Scaling Factor	32
Dropout	0.1
Bias	None
Optimizer	`Adam`
Learning Rate	10^{-6}

A.4 Environment-specific configurations and auxiliary loss selection

We keep training parameters consistent across environments, introducing only minimal task-specific modifications. Experiments on CartPole, Reach, and ReposeCube are conducted within Isaac Lab [Mittal et al., 2023], while MountainCar is run in Gymnasium [Brockman et al., 2016].

The choice of auxiliary loss depends on the inherent structure of each environment. We select between CAL and RLR based on which structural properties can be exploited for reward learning, as summarized in Table 1. RLR is used when local state distances provide meaningful information about reward differences, while CAL is used when task-preserving transformations should leave the reward unchanged.

CartPole CartPole exhibits both strong state-reward distance correlation and left-right symmetry, since poles tilted $\pm\theta$ degrees left or right from vertical should receive equal reward. Therefore, CartPole supports both RLR and CAL. In the main SAFT experiments, we use CAL, with horizontal flips as positive samples and vertical flips as hard negatives. For RLR experiments, we use a window size of 32.

MountainCar MountainCar maintains a state-reward distance correlation suitable for RLR, but lacks exploitable symmetries. Since the ground-truth reward is defined by distance from the right side of the screen, no two distinct states yield equal rewards, and hence no two observations can serve as positive augmentation pairs. We therefore apply only RLR, using a window size of 500. We also add an action penalty to stabilize learning and better distinguish the optimal policy from alternatives.

Reach In the Isaac Reach task, the camera is mounted above the end effector, showing both current and target markers and centering the view so only relative goal motion is visible. The task is modified to require only position alignment between the end effector and the goal, removing the orientation constraint. Rewards are computed from a top-down projection of proximity rather than full 3D distance, reflecting the limited depth perception of current VLMs. A visual tracker of the target and end effector positions is rendered to the agent.

The circular symmetry around the target, where states at fixed radius have equal reward, violates the assumptions of RLR but provides rotational invariances exploitable by CAL. Reach therefore uses CAL, with 8 rotated positive samples at $\pm 10^\circ, \pm 20^\circ, \pm 30^\circ$, and $\pm\ 40^\circ$, along with a soft negative sampled from the same rollout.

ReposeCube For ReposeCube, instead of matching a held cube to a goal cube, the task is reformulated to require orienting the cube such that a specific side faces upward, with the cube itself rendered in view. The environment exhibits rotational symmetries similar to Isaac Reach: multiple cube

orientations yield equivalent angular distances to the goal, making RLR inappropriate but providing natural invariances for CAL.

ReposeCube therefore uses CAL, with 6 rotated positive samples at $\pm 10°, \pm 20°$, and $\pm 30°$, coupled with a hard negative rendered from the opposing $180°$ viewpoint.

Together, these environments cover several structural regimes for VLM reward models, including sensitivity to asymmetric dynamics in CartPole, unique states without valid augmentation pairs in MountainCar, rotational invariances that break distance-based rewards in Reach, and high-dimensional manipulation with non-trivial equivalences across cube orientations in ReposeCube.

After each rollout, we compute one auxiliary loss, either L_{RLR} or L_{CAL}, depending on the environment, and backpropagate it once. Gradients update only the LoRA adapters in the otherwise frozen vision encoder. To avoid leaking ground-truth information, we disable early stopping in all environments. Depending on task difficulty, we adjust the rollout length and the number of parallel environments. The exact parameters, CAL augmentations, and goal and baseline descriptions used in goal-baseline regularization [Rocamonde et al., 2024] are provided in Table S7.

Table S7: Environment-specific parameters and augmentations used in all experiments. The labels *soft* and *hard* indicate whether the negative corresponds to an imperfect but plausible alternative observation or to a strict contradiction of the current state.

	Environments	
	CartPole	MountainCar
Rollout Steps	32	1 000
Parallel Environments	64	9
Auxiliary Loss	CAL	RLR
CAL Positive Sample	Horizontal Flip	N/A
CAL Negative Sample	Vertical Flip (*hard*)	N/A
RLR Window Size	32	500
Baseline Prompt	"pole and cart"	"a car in the mountain"
Goal Prompt	"pole vertically upright on top of the cart"	"a car at the peak of the mountain, next to the yellow flag"
	Reach	ReposeCube
Rollout Steps	48	24
Parallel Environments	64	1 024
Auxiliary Loss	CAL	CAL
CAL Positive Sample	Rotation ($\pm 10/20/30/40°$)	Rotation ($\pm 10/20/30°$)
CAL Negative Sample	Past Observation (*soft*)	Opposite Orientation (*hard*)
RLR Window Size	N/A	N/A
Baseline Prompt	"a red and a blue object"	"a cube with colors"
Goal Prompt	"the red and blue object are in the same place"	"only the pink side of the cube is visible"

A.5 PbRL baseline

To benchmark human labeling efficiency, as done in Section 4.4, we implement a standard Preference-based Reinforcement Learning (PbRL) baseline. Instead of real-time human feedback, we employ a synthetic oracle derived from the ground-truth reward functions defined in Table S5. We generate query pairs by sequentially pairing observations (o_t, o_{t+1}) from the rollout buffer as they are collected. The oracle provides a binary preference $y \in \{0, 1\}$ indicating which observation corresponds to a higher ground-truth reward. The reward model is updated online by minimizing the negative log-likelihood under the Bradley-Terry model (Equation 1). Consistent with the SAFT setup, we freeze the VLM backbone and only update the LoRA adapters described in Table S6.

A.6 EPIC distance calculation

EPIC distance, as defined by Gleave et al. [2021], is evaluated over a distribution of observations and their true rewards. To prevent skew within this distribution and improve the validity of the evaluation, we collect half of the observations using a random policy, and the other half from a policy trained using the true reward. In total, $2\,000$ observations are collected for evaluation. For ReposeCube, we directly render the cube, enabling us to synthetically generate $10\,000$ random rotations for EPIC distance evaluation.

A.7 Training infrastructure and runtime

All experiments were conducted using two NVIDIA A100 GPUs with 80GB memory each. The additional fine-tuning performed by SAFT introduces a wall-clock overhead compared to the base VLM reward model, as LoRA updates are performed online during policy training.

We report the corresponding training times for the main experiments in Section 4.2 in Table S8. These results provide a direct comparison between the base reward model and SAFT across all environments.

Table S8: Wall-clock training time for the base VLM reward model and SAFT across environments from Section 4.2.

Method	CartPole	MountainCar	Reach	ReposeCube
Base	2 h	1 h	10 h	29 h
SAFT	4 h	2 h	22 h	63 h

B Additional results

B.1 SAFT enables off-the-shelf use of smaller models

As discussed in Section 4.4, there exists a regime in which the underlying VLM alone cannot solve the task but succeeds when using SAFT.

Figure S9 illustrates the effects of SAFT when applied to an off-the-shelf ViT-B-16 model [Ilharco et al., 2021]. We can observe that for the CartPole environment, the underlying model fails to converge unless SAFT is applied. This result highlights that enforcing the relative structure within the reward model via SAFT is not only beneficial for improving sample efficiency in stronger models (Figure 6), but can also determine whether the training converges at all when using weaker ones.

B.2 Necessity of fine-tuning

As fine-tuning the VLM incurs computational overhead, we first ask whether it offers benefits beyond a simple smoothing strategy that does not require any backpropagation. We compare SAFT, using CAL, to a naive baseline that averages the VLM's predicted rewards across positive augmentations without any fine-tuning. Figure S10 shows that fine-tuning consistently outperforms reward averaging, indicating that negative augmentations provide essential signal and that fine-tuning lets the model correct rewards for nearby states within the state space. This suggests that the gains come from learning rather than mere smoothing, and that the additional compute is justified.

B.3 Weighting of CAL terms

The CAL objective has two components, a positive consistency term that minimizes variation in rewards across positive transformations, and a negative separation term that maximizes variation against negative examples. To assess the contribution of each term, we ablate the weighting parameter β in Equation 5 by comparing our default $\beta = 0.5$ against $\beta \in \{1.0, 0.75, 0.25, 0.0\}$. The results presented in Figure S11 show that both terms contribute, with the negative separation term preventing representational collapse and the positive consistency term providing additional gains.

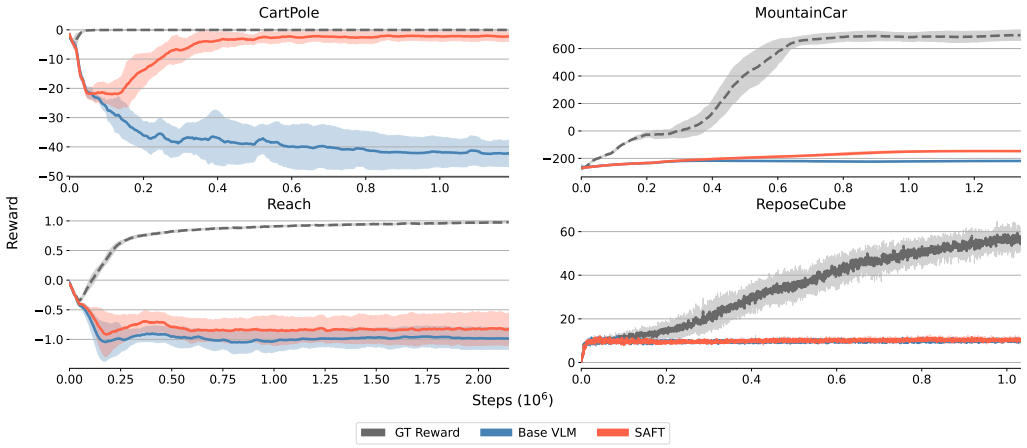

Figure S9: Results are averaged over five seeds for all environments, except ReposeCube which uses three. The off-the-shelf ViT-B-16 model fails to converge without SAFT, which shows that SAFT can be essential for convergence in smaller models.

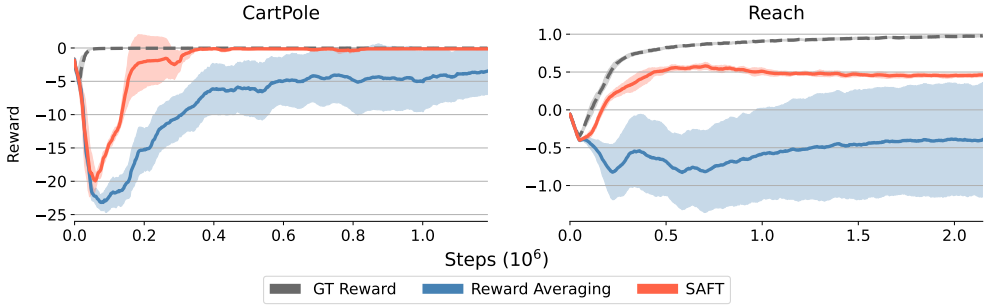

Figure S10: Comparison of SAFT with naive reward averaging over positive augmentations. Fine-tuning consistently outperforms averaging, indicating that simple smoothing falls short. Results presented over five seeds.

B.4 RLR window size

The window size W modulates the RLR temporal consistency prior, directly scaling the loss strength. We evaluate the sensitivity to this parameter by comparing our default settings against reduced window sizes: $W \in \{16, 8, 4, 2\}$ for CartPole (default $W = 32$) and $W \in \{250, 50, 10, 2\}$ for MountainCar (default $W = 500$).

Figure S12 demonstrates that larger window sizes improve performance in both environments. This suggests that the temporal consistency prior, which relates observation-space distances to reward-space distances, remains robust over long horizons.

B.5 Simultaneous usage of CAL and RLR

The CartPole environment is unique in that it supports both RLR and CAL (Table 1). This lets us compare them directly and test whether their effects stack. Figure S13 shows that CAL outperforms RLR, likely because its positive and negative transformations are both precise and unique. Notably, although both methods are helpful on their own, their combination yields no further improvement.

B.6 Pretraining using preference-based reinforcement learning

As an alternative to the regression loss in Equation S7, we employ PbRL pretraining for cases where the initial VLM quality precludes solving the task. This approach requires no ground-truth rewards yet effectively primes the model for SAFT, as shown in Figure S14. We utilize the standard Bradley-Terry formulation of PbRL from Equations 1 and 2, performing updates after each rollout.

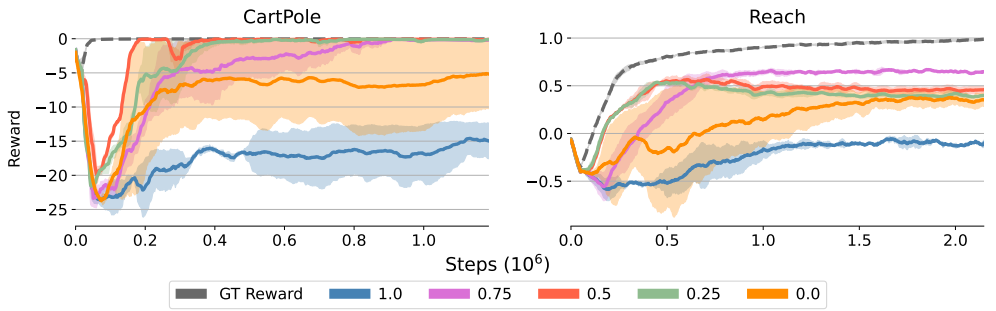

Figure S11: Ablation of the CAL weighting term β between the positive-consistency and negative-separation terms, averaged over two seeds. Results indicate complementary contributions, with performance degrading when either component is downweighted.

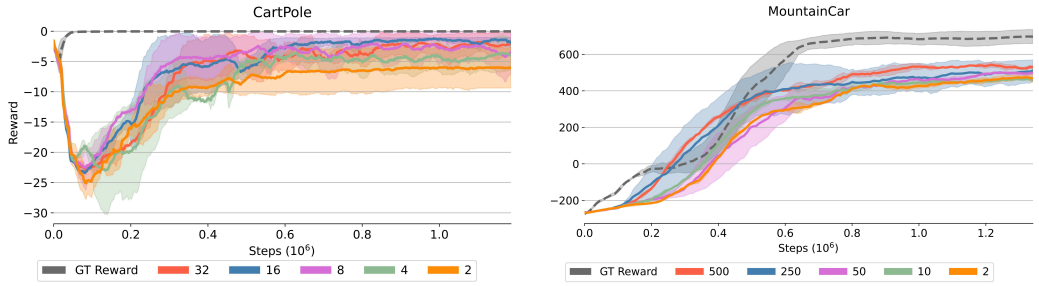

Figure S12: Ablation of the RLR window size W determining the strength of the temporal consistency prior, averaged over two seeds. Results indicate that for both environments, larger window sizes tend to fare better.

We note that the observed gains are smaller than those in Sections 4.2 and 4.3. This is consistent with Section 4.4, as the pretraining phase was longer and yielded a more capable base model.

B.7 Additional backbone experiments

Although there is no clear theoretical reason for SAFT to behave differently across architectures, as it operates via LoRA updates on the image encoder and focuses on improving reward structure rather than model-specific components, we additionally evaluate it using similarly-sized SigLIP2 [Tschannen et al., 2025] and MetaCLIP [Chuang et al., 2025] backbones (`siglip2-base-patch16-224`, `metaclip-2-worldwide-s16-384`).

Training is performed under the exact same setup as the main experiments in Section 4.2, including identical environment configurations, reward formulation, and LoRA parameterization (fine-tuning approximately 0.2% of total parameters), with no additional tuning or modifications.

We report results on CartPole, as this environment allows direct comparison of both CAL and RLR within a single setting. Across two seeds, Figure S15 and Figure S16 show that SAFT consistently improves performance over both the base model and goal-baseline regularization. Both CAL and RLR variants yield clear gains, confirming that the improvements from SAFT are not specific to a particular backbone.

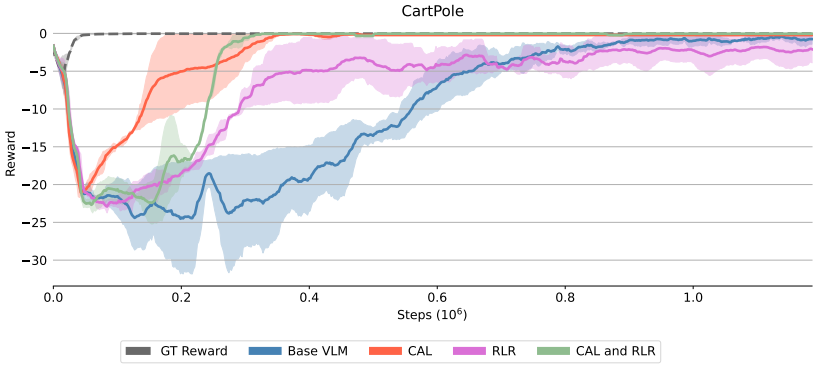

Figure S13: CartPole results comparing CAL, RLR, and their combination, averaged over two seeds. CAL outperforms RLR, and while each method is effective individually, their combination provides no additional gain.

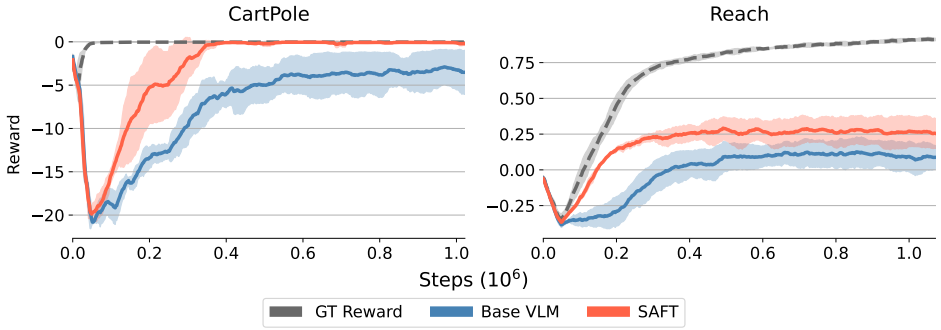

Figure S14: Application of SAFT to a model pretrained using a PbRL objective, averaged over five seeds. Results demonstrate that SAFT also yields significant performance gains following PbRL pretraining.

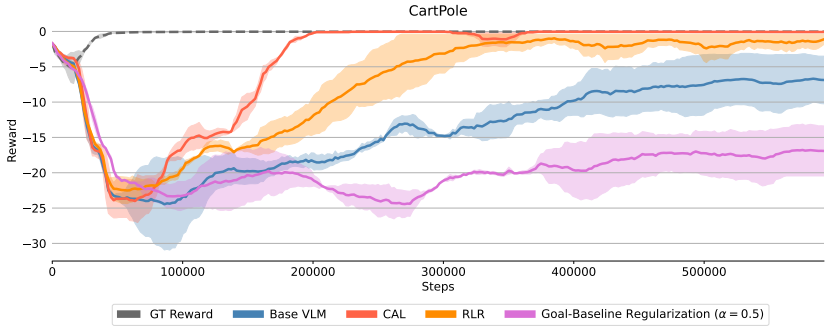

Figure S15: Application of SAFT to a similarly-sized SigLIP2 [Tschannen et al., 2025] backbone, pretrained under the same conditions as Section 4.2, averaged over two seeds. Results demonstrate that SAFT consistently yields significant performance gains across backbones.

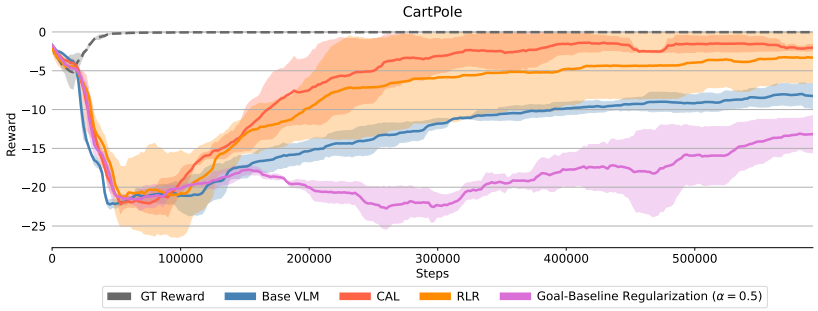

Figure S16: Application of SAFT to a similarly-sized MetaCLIP [Chuang et al., 2025] backbone, pretrained under the same conditions as Section 4.2, averaged over two seeds. Results demonstrate that SAFT consistently yields significant performance gains across backbones.