2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Rewarding Change Beyond State: Directional VLM Rewards for Sample-Efficient Robot Reinforcement Learning*

Linus Lundgren[1,†], Wenhao Lu[2,†], Zhitao Liang[2], Ze Zhang[1], Karinne Ramirez-Amaro[2], and Emmanuel Dean[2]

***Abstract*— Sparse rewards are a persistent bottleneck for robotic manipulation with Reinforcement Learning (RL), primarily because RL agents must discover long-horizon, multi-step behaviors while receiving infrequent and weakly informative feedback. Recent work uses pre-trained Vision Language Models (VLMs) to provide dense per-step rewards, yet most approaches score only a single image against a goal text, ignoring whether the *recent change* actually moves the system toward success. We argue that this omission impairs exploration (e.g., goal-like detours, wrong-way progress, action aliasing) and propose to *make time explicit* in VLM rewards by adding a directional signal that evaluates short-horizon change. Concretely, we pair visual change over a few steps with a text description of the desired change, and finetune lightweight heads with RL; the resulting directional signal is combined with a standard positional signal into a single shaping reward. We evaluated our approach in six MetaWorld manipulation tasks with fixed goals. This directional shaping improves running average success at a fixed budget to 78.2%, versus 63.8% for the best-tuned positional baseline (improvements were observed in five of six tasks). Ablations identify key design choices for the proposed directional term to be effective and show its synergy with the positional term when supplying dense VLM rewards, demonstrating improved exploration and sample efficiency.**

I. INTRODUCTION

Robotic manipulation with Reinforcement Learning (RL) often features long horizons and sparse, binary feedback, which makes exploration and credit assignment notoriously sample-inefficient [1]. In settings where interactions are costly and slow, improving sample efficiency is essential for practical learning on real systems. To address this, recent works leverage pre-trained Vision Language Models (VLMs) [2] to provide dense rewards by aligning image observations to text goals, thereby supplying progress signals at every step rather than only at completion [3]–[6].

However, most VLM-as-reward methods evaluate only an *instantaneous* similarity between the current observation and a textual goal. This captures how "goal-like" a single frame appears but ignores the *temporal* structure that defines sequential decision making. In long-horizon manipulation, this omission can mislead exploration: visually goal-like detours can receive high scores despite being off a feasible path (e.g., the gripper near the handle but on the wrong side); visually salient wrong-way motions can spuriously increase similarity; and different actions may map to equally "goal-looking" next frames by appearance alone, aliasing action credit. These effects slow learning under sparse rewards and limit the benefits of dense shaping.

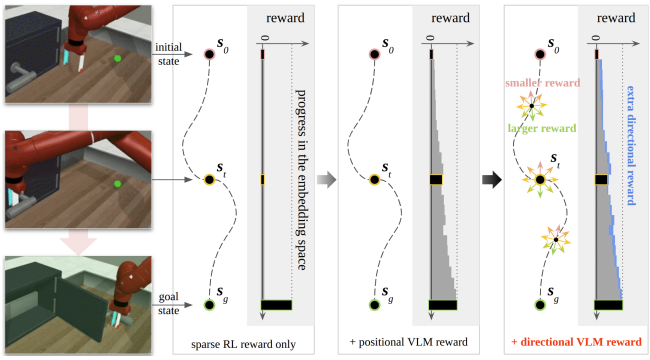

Fig. 1. The concept of the directional VLM reward for RL given the same trajectory. The original RL reward is sparse, which only gives a positive reward when the goal state is reached. The added positional VLM reward can provide extra dense per-step guidance, but scores only where the robot is. Our proposed directional VLM reward scores short-horizon what just changed, yielding further action–consequence feedback for RL training.

In this work, it is explored how to *make time explicit* in VLM rewards so that they better guide exploration. The key idea (cf. conceptual overview in Fig. 1) is to complement the standard positional score (how suitable the current state looks for the goal) with a directional score that evaluates the *observed change* over a short horizon. Concretely, we pair visual change across a few steps with a textual description of the desired change (goal vs. baseline). Lightweight heads on top of frozen VLM encoders are finetuned with a two-stage ranking schedule adapted from [6]; the directional and positional scores are then combined into a single shaping signal. Intuitively, the directional term provides sharper, action-consequence feedback and is simpler to optimize (local ranking at the same state), while the positional term supplies coarse progress. Empirically, on six MetaWorld manipulation tasks with fixed goals, the proposed directional shaping yields a marked +22.6% relative improvement in running-average success at a fixed interaction budget over a tuned positional baseline. Ablations show that including the current state together with the change is critical, and the directional term often provides the dominant guidance for exploration since removing the positional term changes

† Equal contribution.

*This work is supported by Chalmers AI Research Centre (CHAIR), Chalmers Gender Initiative for Excellence (Genie), and National Academic Infrastructure for Supercomputing in Sweden (NAISS).

[1]Faculty of Computer Science and Engineering, [2]Faculty of Electrical Engineering, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden. {linulun, wenhaol, zhitao, zhze, karinne, deane}@chalmers.se

performance negligibly. We further show that the directional signal is a practical diagnostic: sustained high segments emerge earlier and more frequently in successful episodes, resulting in a simple, interpretable proxy for learned, goal-directed motion.

The contributions of this work are threefold:

1) We identify the temporality gap in VLM rewards for RL, and propose a generic yet simple method to extract short-horizon, change-aware (directional) signals from pretrained VLMs for use in online RL;
2) We empirically show that temporally informed VLM rewards improve exploration and sample efficiency on sparse-reward robotic benchmarks, compared to standard similarity-based VLM rewards;
3) We provide analyses and ablations that explain when and why these temporal (directional) rewards help.

II. RELATED WORK

VLMs as reward providers. A growing line of work uses pre-trained VLMs to turn image–text alignment into dense rewards for RL [3]–[5], [7]. Subsequent papers add lightweight projection heads on frozen encoders to better align with task semantics [6], [8]. Across both zero-shot and head-calibrated variants, the reward remains fundamentally positional: they score where the robot is, not what just changed. In contrast, we promote temporal directionality to a first-class VLM signal and train it online alongside a positional stream to yield sharper action–consequence feedback during exploration.

Temporal structure in shaping. Closer to our aim, several ideas inject limited temporal information into shaping. Goal–baseline regularization scores progress relative to a reference image or baseline phrase [4]; difference-based encodings contrast consecutive states or predicted vs. observed states to help the agent better understand dynamics [9], [10]. However, they do not turn short-horizon observed change into a first-class VLM reward trained online for exploration. Our approach differs by explicitly pairing a visual state and its short-horizon change with a text-side change description (goal minus baseline), which explicitly targets the exploration of failure modes that arise from single-frame scoring.

III. PRELIMINARIES

A. Markov Decision Process and Reinforcement Learning

Reinforcement learning is a machine learning method for sequential decision-making tasks, where an agent interacts with an environment to learn which actions are optimal to execute through experience. The RL problem is formalized as a Markov Decision Process (MDP) [11], expressed by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, \mathcal{R})$. This tuple defines a set of states $\mathcal{S}$, actions $\mathcal{A}$ and rewards $\mathcal{R}$, the dynamics $\mathcal{P}$ (state transition and reward probabilities), and the discount rate γ, which determines the present value of future rewards. The RL algorithm learns a policy π, which maps states to actions, by maximizing the accumulated reward over time.

B. Positional VLM Rewards in Reinforcement Learning

Using *zero-shot* VLMs as a reward source in RL is a popular emerging trend [3]–[5]. In robotic tasks, environment rewards are often very sparse, typically provided only at task completion (e.g., a successful open/close/push). Such VLM-generated rewards provide dense shaping: unlike sparse task rewards, dense rewards are available at every timestep and give continual learning signals that ease credit assignment over long horizons, thereby reducing sample complexity [1].

Concretely, at each environment step t we obtain an image o_t and compute an image embedding $s_t := \phi_I(o_t)$ using VLM image encoder ϕ_I, and we encode the task goal instruction l_g into a goal text embedding $g := \phi_L(l_g)$ using the language encoder ϕ_L. So the VLM similarity (used as the VLM reward) is

$$r^{\text{VLM}}(s_t) = \alpha\big(\phi_I(o_t),\, \phi_L(l_g)\big) = \alpha(s_t, g) = \frac{s_t \cdot g}{\|s_t\|_2 \cdot \|g\|_2}.$$

The scalar reward observed and used during RL training at timestep t is then

$$r_t \;=\; r_t^{\text{task}} \;+\; \rho \cdot r^{\text{VLM}}(s_t),$$

where r_t^{task} is the sparse task reward and ρ scales the VLM shaping term, giving per-step feedback that complements the sparse task signal. Note the RL agent itself typically observes proprioceptive state rather than raw images.

Zero-shot VLM rewards provide dense shaping but are often *fuzzy*, mainly due to their zero-shot nature [6], [12]: scores can be miscalibrated under domain shift between pre-trained VLM representations and downstream robot tasks (viewpoints, lighting, or distractors). As a representative instance, FuRL [6] keeps the VLM image and text encoders *frozen* and attaches MLP heads f_{W_I} and f_{W_L} to map embeddings into a task-adapted similarity space. These heads are finetuned online with RL using a ranking-based objective that collects and separates successful from failed episodes, referred to as positive and negative trajectories, and boosts rewards along successful ones. In this work, we adopted FuRL as a starting point: it is considered a strong, widely used baseline and convenient for our head training schedule in Sec. IV.

Importantly, both zero-shot variants and calibrated methods like FuRL still compute an instantaneous, single-frame similarity: they refine *where the robot is* (position) but do not directly score *what just changed* (direction). This positional nature motivates our method in Sec. IV, where we make time explicit in VLM rewards by adding a directional signal that evaluates short-horizon change and combines it with the positional term.

IV. TEMPORAL VLM SHAPING

Prior VLM-as-reward methods [3]–[6], [8] compute an instantaneous similarity at a single frame $s_t = \phi_I(o_t)$ against a goal text $g = \phi_L(l_g)$. This ignores whether the recent change actually moves the system closer to success or not. In long-horizon manipulation with sparse reward (r_t^{task}), this

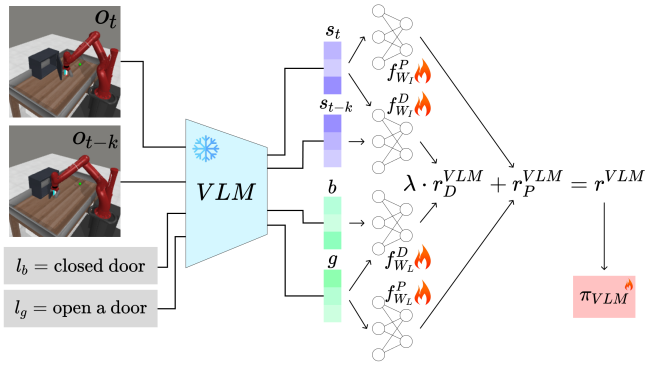

Fig. 2. Proposed integration of directional VLM reward (FuD-RL). Frozen VLM encoders map images and texts (goal l_g, baseline l_b) to embeddings; a positional stream ($f^P_\cdot$) scores the current state, while a directional stream ($f^D_\cdot$) scores short-horizon change against a desired text-side change. The two signals are fine-tuned jointly and combined to shape the RL reward.

omission hurts exploration and credit assignment in at least three concrete ways:

1) Goal-like detours. States can be locally similar to the goal yet lie off any feasible path (e.g., a gripper visually near the handle but on the wrong side); a positional score may be high for such states.

2) Wrong-way progress. Small but correct motions may barely change positional similarity (cosine ramp); conversely, visually salient but wrong motions may spuriously increase similarity.

3) Ambiguous credit. From the same o_t, two actions can lead to equally "goal-looking" next states by image features, but only one aligns with the task's kinematics. Without judging the direction of change, dense shaping is aliased across actions.

We argue that efficiency improves when we also provide the value of the observed change. The purpose of both VLM shaping is to guide the agent toward a goal state, yet a purely positional reward is agnostic to the direction in which the agent is moving. As illustrated in Fig. 1, the positional reward at s_t is invariant to preceding states, whereas the directional stream can explicitly inform the agent that it is moving away/towards since it accounts for this direction.

A. Temporally Informed VLM Rewards

We keep the standard positional VLM reward but add a directional term that scores the observed change. To score change, we form a short-horizon difference $\Delta s_t = \mathrm{norm}(s_t - s_{t-k})$ (offset $k \geq 1$) and concatenate with the position, $x_t = s_t \oplus \Delta s_t$. This pairing grounds directional movement in spatial context and interprets "direction" as a conceptual change vector that approximates the intended change. We ℓ_2-normalize both s_t and Δs_t to avoid magnitude bias, ensuring the scale between position and direction is equivalent. The shaped reward used by RL is

$$r_t \;=\; r_t^{\text{task}} \;+\; \rho\left(r_P^{\text{VLM}}(s_t) \;+\; \lambda\, r_D^{\text{VLM}}(x_t)\right),$$

where $\rho \geq 0$ sets the VLM shaping strength and $\lambda \geq 0$ balances directional vs. positional. For brevity, we refer to this positional+directional shaping as *FuD-RL* (Fuzzy Directional rewards-aided Reinforcement Learning), a FuRL-style setup augmented with a directional stream. As an overview, Fig. 2 shows the two-stream design.

a) Positional reward (r_P^{VLM}): Following FuRL [6], the positional signal is

$$r_P^{\text{VLM}}(s_t) \;=\; \alpha\Big(f^P_{W_I}(s_t),\; f^P_{W_L}(g)\Big),$$

which evaluates the current position s_t w.r.t. the goal text g. It remains a single-frame (positional) score.

b) Directional reward (r_D^{VLM}): On the text side, we encode the desired semantic change:

$$d = g - b, \tag{1}$$

where $g = \phi_L(l_g)$ encodes the goal and $b = \phi_L(l_b)$ encodes a baseline that captures the initial state (e.g., "closed door"). Rewarding similarity with d encourages movements that increase similarity to g while decreasing similarity to b. The directional reward is

$$r_D^{\text{VLM}}(x_t) \;=\; \alpha\Big(f^D_{W_I}(x_t),\; f^D_{W_L}(d)\Big). \tag{2}$$

B. Unified Finetuning for r_P^{VLM} and r_D^{VLM}

We used the FuRL's two-stage head-tuning schedule to reduce fuzziness in both reward functions, and applied the same ranking logic to both the positional heads $f^P_{W_I}, f^P_{W_L}$. Let $s_g = \phi_I(o_g)$ be an image embedding of a success state, and let $\ell_\delta(\cdot,\cdot)$ denote a margin-ranking loss with $\delta > 0$:

$$\ell_\delta(s^p, s^n) = \max\big(0,\; r_P^{\text{VLM}}(s^n) - r_P^{\text{VLM}}(s^p) + \delta\big),$$

so that s^p is trained to outscore s^n [1]. Superscripts p/n denote samples from positive/negative trajectories.

a) Stage 1 (negative samples only): When only negative trajectories are available, the positional heads minimize

$$\begin{aligned}\mathcal{L}^P_{\text{neg}} &= \ell_\delta\big(s^n_i,\; s^n_j\big), \\ &\text{s.t.}\;\; \|s^n_i - s_g\|_2 \leq \|s^n_j - s_g\|_2\end{aligned} \tag{3}$$

which encourages higher scores for negatives closer to success. For the directional heads, we contrast the observed negative change Δs^n_i with a *goal-ward proxy* $\hat{\Delta} s^n_i$:

$$\begin{aligned}&\Delta s^n_i = \mathrm{norm}(s^n_i - s^n_{i-k}), \quad \hat{\Delta} s^n_i = \mathrm{norm}(s_g - s^n_i), \\ &\mathcal{L}^D_{\text{neg}} = r_D^{\text{VLM}}(s^n_i \oplus \Delta s^n_i) - r_D^{\text{VLM}}(s^n_i \oplus \hat{\Delta} s^n_i)\end{aligned} \tag{4}$$

This pushes the model to prefer goalward changes over the actually observed (unsuccessful) ones at the same s^n_i.

b) Stage 2 (positive samples available).: Once successful trajectories appear, we add monotonic and separation constraints for the positional heads:

$$\mathcal{L}^P_{\text{pos}} = \ell_\delta\big(s^p_i,\; s^p_{i-n}\big), \tag{5}$$

$$\mathcal{L}^P_{\text{pos-neg}} = \ell_\delta\big(s^p_i,\; s^n_j\big). \tag{6}$$

[1]This follows from minimizing the ranking loss, which increases $r_P^{\text{VLM}}(s^p)$ and decreases $r_P^{\text{VLM}}(s^n)$ until $r_P^{\text{VLM}}(s^p) \geq r_P^{\text{VLM}}(s^n) + \delta$.

Algorithm 1 FuD-RL (Stage 2); additions colored blue.

Input: Pretrained VLM ϕ_I, ϕ_L, goal image o_g, task goal l_g, baseline l_b, shared replay $\mathcal{D}_{\text{shared}}$, episodes N.
Output: Trained VLM agent π_{VLM}.
Init: $s_g \leftarrow \phi_I(o_g)$, $g \leftarrow \phi_L(l_g)$, $b \leftarrow \phi_L(l_b)$, heads $f^P_{W_I}, f^P_{W_L}, f^D_{W_I}, f^D_{W_L}$, policy π_{VLM}.
for $i = 1$ **to** N **do**
  Unroll π_{VLM} and update $f^P_{W_I}$ and $f^P_{W_L}$ via Eqs. 5, 6.
  Update directional heads via Eqs. 4, 7.
  $r^{\text{VLM}} \leftarrow r_P^{\text{VLM}} + \lambda\, r_D^{\text{VLM}}$
  Update π_{VLM} using $r_{\text{task}} + \rho\, r^{\text{VLM}}$.
end for
return π_{VLM}

For the directional heads, the first stage objective Eq. (4) is complemented by:

$$\Delta s_i^p = \text{norm}(s_i^p - s_{i-k}^p), \quad \overline{\Delta} s_i^p = \text{norm}(s_{i-k}^p - s_i^p),$$
$$\mathcal{L}^D_{\text{pos}} = r_D^{\text{VLM}}(s_i^p \oplus \overline{\Delta} s_i^p) - r_D^{\text{VLM}}(s_i^p \oplus \Delta s_i^p). \tag{7}$$

encouraging the observed goalward change and penalizing its reverse.

c) Overall objective and policy update: We sum the active losses (Stage 1 or Stage 2) for both streams,

$$\mathcal{L} \;=\; \mathcal{L}^P + \mathcal{L}^D,$$

update the heads $\{f^P_{W_I}, f^P_{W_L}, f^D_{W_I}, f^D_{W_L}\}$, compute

$$r_P^{\text{VLM}}(s_t) = \alpha\big(f^P_{W_I}(s_t), f^P_{W_L}(g)\big),$$
$$r_D^{\text{VLM}}(x_t) = \alpha\big(f^D_{W_I}(x_t), f^D_{W_L}(d)\big), \tag{8}$$

and use the shaped reward

$$r_t \;=\; r_t^{task} + \rho\left(r_P^{\text{VLM}}(s_t) + \lambda\, r_D^{\text{VLM}}(x_t)\right)$$

to train the RL agent (e.g., SAC [13]) off-policy. A detailed second-stage training procedure can be found in Algorithm 1.

V. EXPERIMENTS

A. Environments

We evaluate on six MetaWorld [14] manipulation tasks as listed in Table I. Each task has fixed-goal and variable-goal versions. We adopt the fixed-goal setting to isolate the effect of reward shaping on sample efficiency without conflating it with goal-distribution-induced variance. In particular, fixed goals reduce reward-noise stemming from visual/pose diversity, making changes in performance more attributable to the shaping signal (positional vs. directional). Note that our primary metric (defined below) evaluates efficiency at a fixed interaction budget and is agnostic to whether goals vary; extending to varying goals is deferred to future work to specifically probe generalization of the learned reward heads.

B. Baselines

We included two baselines for comparison to put the performance of our temporal rewards in context. The first is standard Soft Actor-Critic (SAC) [13], with no reward shaping. The second is the original FuRL [6] using the positional VLM reward $\alpha(f^P_{W_I}(s_t), f^P_{W_L}(g))$ with tuned ρ. Ours (FuD-RL) leverages both positional and directional rewards.

TABLE I
TTFS (EPISODES) UNDER FURL POSITIONAL WITH ρ=0.05 (FIXED GOALS). HIGH VARIANCE HIGHLIGHTS STAGE-1 STOCHASTICITY.

drawer-open	door-open	window-open
379 ± 214	420 ± 335	171 ± 129
window-close	**push**	**button-press-topdown**
214 ± 23	349 ± 172	559 ± 268

TABLE II
FURL POSITIONAL ρ-SWEEP (STAGE-2). ENTRIES ARE AVG. RAS (%), AVERAGED OVER 6 TASKS. WE SELECT ρ=1.0.

ρ	0.05	0.25	1.00	2.00	4.00
Avg. RAS	44.5	58.6	**63.8** ✓	63.7	62.0

C. Evaluation Metrics

a) Running Average Success at Fixed Budget (RAS): To quantify sample efficiency, we report the running average success rate computed at a fixed timestep budget, varying by task difficulty. Let $z_i \in \{0, 1\}$ denote the success indicator of episode i and w a window size (in episodes). The running average at episode index e is:

$$\text{RAS}(e; w) \;=\; \frac{1}{w} \sum_{i=e-w+1}^{e} z_i,$$

clipped at $i \geq 1$. We evaluate RAS by taking the RAS measured in the episode that ends closest to the budget steps. Unless noted, we use w=150 episodes (500 environment steps each) and 4 seeds per task; we report mean across seeds where illustrative.

b) Time to First Success (TTFS).: To analyze Stage 1 variability, we measure the number of episodes until the first success:

$$\text{TTFS} \;=\; \min\{i \mid z_i = 1\},$$

aggregated as mean $\pm$ std across seeds (Table I).

All runs use a horizon of 1,000 episodes and compute scoring changes with a fixed offset $k = 5$ (smaller k evaluates actions over shorter spans and a broader sweep is left to future work). Directional settings sweep $\lambda \in \{30, 80, 150\}$ with ρ=0.05 (to match FuRL [6]). Positional baselines sweep $\rho \in \{0.05, 0.25, 1.0, 2.0, 4.0\}$. The VLM is LIV [15], as in FuRL; all experiments ran on 4 NVIDIA Tesla T4 GPUs.

D. Results & Analysis

1) Stage-1 variability motivates Stage-2-focused evaluation.: As shown in Table I, TTFS under FuRL positional with ρ=0.05 (fixed goals, 4 seeds) exhibits high variance; some runs fail to achieve a single success within 1,000 episodes, as shown in Fig. 3. This makes the length of Stage 1 the dominant factor for apparent convergence.

To decouple this confound and evaluate shaping quality directly, we also adopt a Stage-2-only protocol by preloading

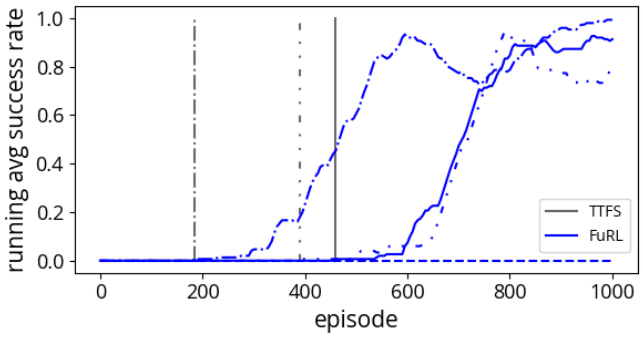

Fig. 3. Running average success rate (blue lines) and TTFS (gray vertical lines), for 4 different seeds, fixed goals and 500k steps. It demonstrates high variation in TTFS, with one run failing to achieve even a single success.

TABLE III

DIRECTIONAL REWARD (FUD-RL) λ-SWEEP AT ρ=0.05. ALSO SHOWN: SAC (NO SHAPING) AND FURL@ρ=1.0. ENTRIES ARE RAS (IN %).

Method	drawer open	door open	window open	window close	push	button-press topdown	Avg.
SAC	53.5	35.2	39.7	32.8	28.0	56.2	40.9
FuRL@ρ=1.0	56.2	83.0	54.8	42.7	**56.5**	89.3	63.8
$\lambda = 30$	**97.8**	93.2	76.0	**77.7**	43.5	76.5	77.4
$\lambda = 80$	87.0	**96.2**	73.0	76.2	46.7	**90.3**	**78.2**
$\lambda = 150$	88.5	81.8	**77.5**	65.3	51.2	78.2	73.8

a small set (50) of positive trajectories (from a separate run) at training start so that ranking over positives is immediately active. All RAS values in Tables II-VII use this Stage-2 evaluation, with the same set of preloaded trajectories for each task.

2) Directional shaping (FuD-RL) improves RAS.: For the directional setting, we keep ρ=0.05 (matching the FuRL [6]) and sweep $\lambda \in \{30, 80, 150\}$; results are in Table III. For comparison, the table also reports SAC (no shaping) and the best FuRL baseline (with ρ=1.0, from Table II). We find relatively high λ values perform best, which intuitively compensates for the small ρ assigned to the VLM term. Overall, *directional shaping significantly improves sample efficiency*: average RAS rises to **78.2%** vs. **63.8%** for FuRL with ρ=1.0, with push as the lone exception. This suggests task-dependent benefits of using directional rewards. Fig. 4 further provides a qualitative view: gains in sample efficiency are pronounced in door-open, smaller in button-press-topdown. However, by and large, our results suggest that directional rewards offer sharper guidance than purely positional shaping.

This increase in efficiency can be attributed to the nature of the directional reward: it evaluates observed change, so its feedback is tightly coupled to the actions the agent is attempting to optimize. The respective finetuning of the positional and directional rewards can also explain it. In positive rollouts, the positional value of s_t depends on its location (neighborhood) along the whole trajectory, whereas the directional value of $s_t - s_{t-k}$ depends only on directions previously observed at the same s_t. This locality simplifies

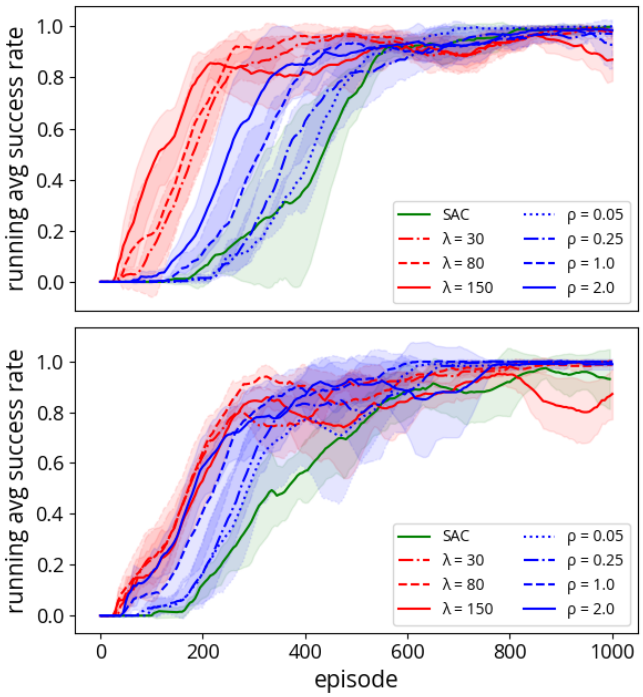

Fig. 4. Running-average success rate (150-episode window) for door-open (top) and button-press-topdown (bottom). Red lines: using directional rewards (FuD-RL) with ρ=0.05 and λ=30, 80, 150. Blue lines: using FuRL positional-only rewards with ρ=0.05, 0.25, 1.0, 2.0. Green: SAC baseline for reference.

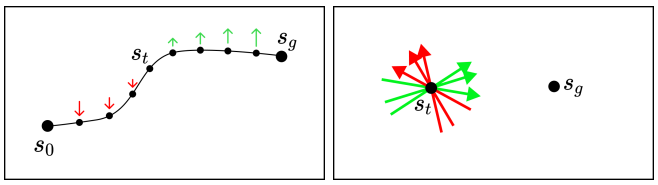

Fig. 5. Optimization contrast. Left: positional reward is trained to be monotone along positive trajectories, requiring comparisons across many states; optimization enforces a gradual decrease when moving backward from s_t (red arrows) and a gradual increase when moving forward (green arrows). Right: directional reward learns to score the local change at the same s_t, needing only to rank locally good (green) vs. bad (red) directions.

learning and provides crisper, action–consequence credit.

This contrast is illustrated in Fig. 5. To learn the positional value of a state s_t from a positive trajectory (left), fine-tuning repeatedly enforces pairwise inequalities of the form $r_P^{\text{VLM}}(s_t) > r_P^{\text{VLM}}(s_{t-k})$ (Eq. 5) across all other states in the trajectory. Thus, updates that raise r_P locally can disturb ordering elsewhere; many passes are needed before the entire trajectory settles into a consistent monotone profile. Credit is also indirect: the signal on s_t depends on its relative placement among many other states, which slows convergence as the policy distribution keeps shifting during training.

However, for the directional value of $s_t - s_{t-k}$, only previous directions taken in s_t need to be considered, and whether they led to success or failure, as illustrated in Fig. 5 (right). This makes the optimization simpler since there is no relative reward specification, and more localized as the reward is only dependent on directions observed in s_t.

TABLE IV

GENERIC VS. TASK-SPECIFIC l_b WITH ρ=0.05, λ=80 (ENTRIES ARE RAS, IN %).

Task	Specific	Generic
drawer-open	87.0	**89.1**
door-open	**96.2**	94.2
window-open	73.0	**78.2**
window-close	**76.2**	71.8
push	46.7	**60.3**
button-press-topdown	**90.3**	81.3
Average	78.2	**79.2**

TABLE V

ABLATION OF POSITIONAL TERM r_P^{VLM} WITH ρ=0.05, λ=80, GENERIC l_b (ENTRIES ARE RAS, IN %).

Task	with r_P^{VLM}	w/o r_P^{VLM}
drawer-open	**89.1**	77.7
door-open	94.2	**96.0**
window-open	78.2	**81.2**
window-close	71.8	**76.5**
push	**60.3**	51.3
button-press-topdown	81.3	**85.7**
Average	**79.2**	78.1

TABLE VI

INCREASING POSITIONAL WEIGHT WHILE KEEPING DIRECTIONAL MAGNITUDE FIXED ($\rho\cdot\lambda$=4.0). BASELINE l_b IS GENERIC; ENTRIES ARE RAS (IN %).

Task	$\rho = 0.05$	$\rho = 1.0$	$\rho = 2.0$
drawer-open	89.1	**91.8**	83.7
door-open	94.2	**96.2**	95.0
window-open	78.2	74.5	**78.7**
window-close	71.8	**78.5**	76.2
push	**60.3**	54.2	48.0
button-press-topdown	81.3	84.7	**86.2**
Average	79.2	**80.0**	77.9

TABLE VII

DIRECTIONAL INPUT ABLATION: CONCATENATED $s_t \oplus \Delta s_t$ (λ=80) VS. DIRECTION-ONLY Δs_t (BEST AT λ=30). ENTRIES ARE RAS (IN %).

Task	Concat (λ=80)	Dir-only (λ=30)
drawer-open	**87.0**	30.0
door-open	**96.2**	90.3
window-open	**73.0**	68.5
window-close	**76.2**	59.3
push	**46.7**	39.2
button-press-topdown	**90.3**	80.2
Average	**78.2**	61.2

E. Ablation Studies

1) Task-specific vs. generic baseline (l_b): In Table IV we compare task-specific baselines (e.g., "closed door" for the door-opening task) with a single, scene-level baseline ("robot arm beside table") reused across all tasks. Protocol: Stage-2 evaluation for FuD-RL, ρ=0.05, λ=80, 4 seeds, fixed goals. The generic baseline slightly improves RAS across all tasks (**79.2%** vs. **78.2%**), though effects vary by task. Implication: with a strong directional signal, baseline choice is secondary; a generic l_b is a safe, reusable default.

2) Is the positional term still necessary?: We ablate r_P^{VLM} at ρ=0.05, λ=80, generic l_b (cf. Table V). The average change is ≤ 0.1 points (**79.2** $\rightarrow$ **78.1**), indicating positional reward has little to no effect on the performance. Two factors likely explain this: (i) r_P^{VLM} is underweighted relative to r_D^{VLM} at this scaling; (ii) $r_D^{\text{VLM}}(s_t \oplus \Delta s_t)$ already embeds position (s_t) while explicitly evaluating action consequences (Δs_t), which in turn accelerates exploration (cf. Fig. 4). We concluded that under a small ρ, directional shaping carries most of the useful signal.

3) Trade-off: increasing positional weight at fixed directional magnitude.: To probe complementarity between positional and directional shaping, we fix the directional magnitude ($\rho\cdot\lambda$=4.0) and shift the relative weighting toward the positional term by increasing ρ, i.e. $(\rho, \lambda) \in \{(0.05, 80), (1.0, 4), (2.0, 2)\}$ (cf. Table VI). With a very small positional term relative to the directional one, $(0.05, 80)$ already performs well (avg. **79.2**%). Increasing the positional weight to $(1.0, 4)$ yields a small but consistent gain (avg. **80.0**%), whereas pushing it further to $(2.0, 2)$ reduces efficiency (avg. **77.9**%). These results suggest that both positional and directional rewards add value to RL agents, with a clear preference for the directional signal as the primary driver of sample efficiency.

4) Does directional reward need position?: We compare concatenated input $s_t \oplus \Delta s_t$ vs. direction-only Δs_t (cf. Table VII). Concatenation improves the average RAS (**78.2%** vs. **61.2%**), indicating that the value of change is position-dependent, which is intuitive since where a change occurs matters for its effectiveness.

F. Visualization of Both VLM Rewards

Fig. 6 plots positional and directional rewards along one positive and one negative door-open trajectory. The positional curve ramps smoothly on the positive trajectory as expected and stays low after an early failed attempt in the negative trajectory, consistent with a progress signal over states. In contrast, the directional curve behaves like an indicator: it assigns high values to beneficial local changes and low values otherwise. This matches our training objectives: positional finetuning uses trajectory-wise monotonic ranking, whereas directional finetuning emphasizes locally good vs. bad changes at the same position.

(ii) *Data separability:* after the first success, policies tend to replay similar positive trajectories, yielding high homogeneity among positives and high variance among negatives; qualitative rollouts show near-repetition of suboptimal "quirks" within a run. (iii) *Model capacity:* the two-MLP projection heads provide ample capacity to implement near-threshold detectors. Net effect: positive state–direction pairs become easy to recognize and the directional reward saturates near its extremes; the positional reward shows a related

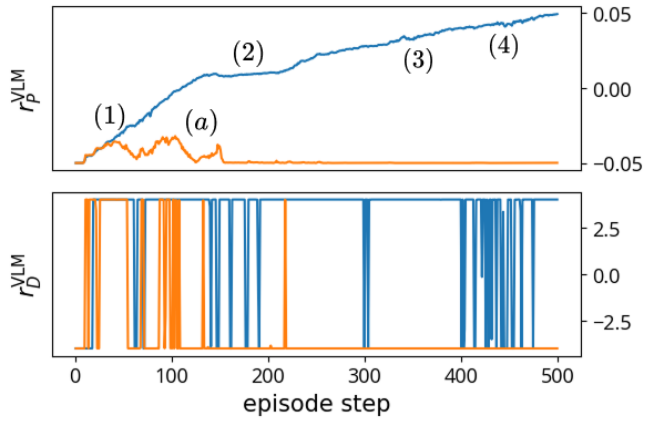

Fig. 6. **Positional vs. directional rewards on door-open** (ρ=0.05, λ=80). Blue: positive trajectory; orange: negative trajectory. Markers: (1) grasp the handle & begin opening; (2) brief struggle/plateau (reflected by dense down/up in r_D); (3) success; (4) slight backward shift while holding; (a) negative: reach the handle, briefly touch, then depart and never return. The near-binary r_D pattern provides a training-time diagnostic: long, contiguous high segments in r_D indicate consistent, goal-directed movements, while frequent up/down oscillations point to ineffective, non-progressing behavior.

tendency but is tempered by the global ranking objective over full trajectories.

Since r_D^{VLM} behaves like an indicator, simply tracking its up/down pattern over time provides a training-time diagnostic of policy quality. Concretely, the fraction (or moving window) of steps with high directional reward rises as the agent learns coherent motions; frequent, continued "upper lines" suggest the agent is consistently making good changes, while pervasive "lower lines" indicate non-progress behaviors (e.g., struggle after grasp). In Fig. 6, directional rewarding yields 443 high steps (positive) vs. 58 (negative). The main takeaway is r_D^{VLM} doubles as a compact, interpretable proxy for how well the agent is currently moving, even before consistent task success.

VI. CONCLUSION

We argued that treating VLM rewards as purely positional overlooks the signal most relevant to exploration: what just changed. By making time explicit and incorporating temporal information in reward shaping that scores short-horizon change, we provide sharper feedback during exploration. Empirically, the inclusion of a temporally informed directional signal consistently improves running-average success at a fixed interaction budget compared to purely positional shaping, even when positional rewards are excluded entirely. Nevertheless, for optimal performance, a moderate mix of positional and directional signals is preferable.

Limitations include our focus on fixed-goal robotic tasks, so extending to varying-goal settings would probe robustness under goal-distribution–induced variance (e.g., changing object poses and viewpoints). Another limitation is task dependency: as environment dynamics become overly complex, the benefit of directional rewards diminishes. In the push task, where a puck moves freely on a table, gains over FuRL are marginal, whereas tasks with more constrained object motion (e.g., drawer-open) show clearer advantages; in very simple tasks (button-press-topdown), FuRL can match FuD-RL. Thus, directional rewards are most beneficial at moderate complexity, where recognizing and reproducing meaningful local changes (e.g., grasping a handle) aids learning without being confounded by excessive dynamical variability. Future work can also include learning change (as input to directional term) with short-horizon sequence encoders (e.g., RNN/Transformer), beyond simple differencing; and reshaping the directional output beyond its near-binary behavior: e.g., modeling graded confidence so that the reward reflects not only "good vs. bad" changes but also how certain the model is. We hope these results encourage the community to reward change, not just state, when using VLMs for robotic RL.

REFERENCES

[1] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *Journal of Machine Learning Research*, vol. 22, no. 30, pp. 1–82, 2021.

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 8748–8763.

[3] K. Baumli, S. P. Singh, F. Behbahani, H. Chan, G. Comanici, S. Flennerhag, M. Gazeau, K. Holsheimer, D. Horgan, M. Laskin *et al.*, "Vision-language models as a source of rewards," in *Second Agent Learning in Open-Endedness Workshop, NeurIPS*, 2023.

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

[5] Z. Huang, Z. Sheng, Y. Qu, J. You, and S. Chen, "VLM-RL: A unified vision language models and reinforcement learning framework for safe autonomous driving," 2024. [Online]. Available: https://arxiv.org/abs/2412.15544

[6] Y. Fu, H. Zhang, D. Wu, W. Xu, and B. Boulet, "FuRL: Visual-language models as fuzzy rewards for reinforcement learning," in *International Conference on Machine Learning (ICML)*. PMLR, 2024, pp. 14 256–14 274.

[7] S. Schoepp, M. Jafaripour, Y. Cao, T. Yang, F. Abdollahi, S. Golestan, Z. Sufiyan, O. R. Zaiane, and M. E. Taylor, "The evolving landscape of llm-and vlm-integrated reinforcement learning," *arXiv preprint arXiv:2502.15214*, 2025.

[8] K. Wang, Y. Zhao, Y. He, S. Dai, N. Zhang, and M. Yang, "Guiding reinforcement learning with shaping rewards provided by the vision–language model," *Engineering Applications of Artificial Intelligence*, vol. 155, p. 111004, 2025.

[9] R. Zheng, X. Wang, Y. Sun, S. Ma, J. Zhao, H. Xu, H. Daumé III, and F. Huang, "TACO: Temporal latent action-driven contrastive loss for visual reinforcement learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023, pp. 48 203–48 225.

[10] D. Kimura, S. Chaudhury, R. Tachibana, and S. Dasgupta, "Internal model from observations for reward shaping," in *Adaptive Learning Agents Workshop, Federated AI Meeting (FAIM)*, 2018.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[12] S. Huang, S.-W. Liu, N. Lipovetzky, and T. Cohn, "The dark side of rich rewards: Understanding and mitigating noise in VLM rewards," *arXiv preprint arXiv:2409.15922*, 2024.

[13] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning (ICML)*. PMLR, 2018, pp. 1861–1870.

[14] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning (CoRL)*. PMLR, 2020, pp. 1094–1100.

[15] Y. J. Ma, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman, "Liv: Language-image representations and rewards for robotic control," in *International Conference on Machine Learning*. PMLR, 2023, pp. 23 301–23 320.