MARVL: MULTI-STAGE GUIDANCE FOR REINFORCEMENT LEARNING VIA VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Designing dense reward functions is pivotal for efficient Reinforcement Learning (RL). However, most dense rewards rely on manual engineering, which fundamentally limits the scalability and automation of reinforcement learning. While Vision-Language Models (VLMs) offer a promising path to reward design, naïve VLM rewards often misalign with task progress, struggle with spatial grounding, and show limited understanding of task semantics. To address these issues, we propose **MARVL**—**M**ulti-st**A**ge guidance for **R**einforcement learning via **V**ision-**L**anguage models. MARVL fine-tunes a VLM for spatial and semantic consistency and decomposes tasks into multi-stage subtasks with task direction projection for trajectory sensitivity. Empirically, MARVL significantly outperforms existing VLM-reward methods on the Meta-World benchmark, demonstrating superior sample efficiency and robustness on sparse-reward manipulation tasks.

1 INTRODUCTION

Reinforcement learning (RL) provides a general framework for sequential decision making (Levine et al., 2016; Kalashnikov et al., 2018; Gu et al., 2017; Han et al., 2023; Kroemer et al., 2021; Tang et al., 2025; Singh et al., 2022), yet reward specification remains a bottleneck (Yu et al., 2025; Ibrahim et al., 2024; Fu et al., 2024a; Kaelbling et al., 1996; Li, 2017; Eschmann, 2021; Lee et al., 2026). Vision-Language Models (VLMs) offer a promising path toward reward design by grounding rewards in natural language (Ma et al., 2024; Rocamonde et al., 2023; Venuto et al., 2024; Lee et al., 2026; Sontakke et al., 2023; Wang et al., 2024; Venkataraman et al., 2024). A widely adopted approach computes the cosine similarity between visual embeddings of environment observations and text embeddings of task instructions, using this score as a dense reward signal (Rocamonde et al., 2023; Adeniji et al., 2023; Baumli et al., 2023; Ma et al., 2023; Fu et al., 2024b). This strategy leverages the rich knowledge of pre-trained VLMs, but its effectiveness in guiding RL remains limited.

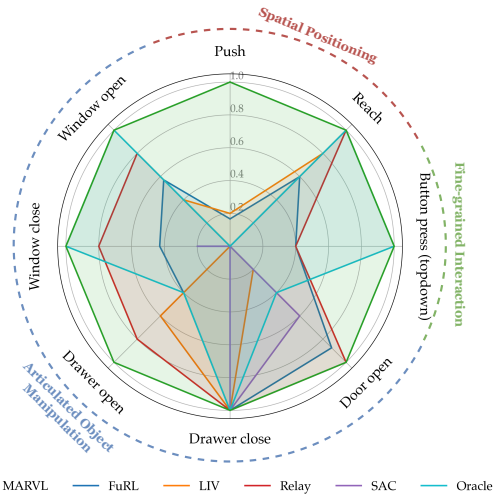

Figure 1: Performance across eight sparse-reward Meta-World tasks. MARVL achieves consistently strong and balanced performance across all skill categories, surpassing the Oracle reward on several tasks and outperforming prior VLM-based reward methods.

Despite the advancements introduced in these varied studies (Rocamonde et al., 2023; Adeniji et al., 2023; Baumli et al., 2023; Ma et al., 2023; Fu et al., 2024b; Huang et al., 2024), we find that VLM-based rewards continue to face three critical challenges: **(i) Weak Spatial Grounding**: VLMs struggle with cross-view consistency and are highly sensitive to camera perspectives, leading to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

misaligned rewards. **(ii) Lack of Progress Awareness**: embeddings are easily distracted by irrelevant visual factors, failing to reflect incremental progress and exposing agents to reward hacking. **(iii) Semantic Misalignment**: VLMs frequently misinterpret task instructions, leading to unreliable reward signals. These limitations suggest that current progress in VLM-guided RL may be less robust than initially believed.

To address these challenges, we propose **MARVL—M**ulti-st**A**ge guidance for **R**einforcement learning via **V**ision-**L**anguage models, a plug-and-play framework for RL algorithms. MARVL incorporates three key design components: (1) **Scene-View Decomposition**: Disentangles task semantics from viewpoint noise via targeted fine-tuning to restore *spatial grounding*; (2) **Task Direction Projection**: Projects embeddings onto instruction-defined vectors, transforming volatile similarities into monotonic signals for *progress awareness*; (3) **Confidence-Thresholded Shaping**: Gates rewards based on semantic confidence, filtering spurious matches to mitigate *semantic misalignment*.

Extensive experiments on the Meta-World benchmark demonstrate that MARVL substantially improves success rate and robustness over all VLM-reward baselines as shown in Figure 1. The primary contributions of this work are as follows:

- We analyze existing VLM reward paradigms and identify three key limitations: weak spatial grounding, lack of task progress sensitivity, and semantic misalignment.
- We propose **MARVL**, a plug-and-play framework that improves the quality and reliability of VLM rewards through targeted refinements.
- We demonstrate that MARVL consistently outperforms prior methods across diverse environments.

2 Related Works

2.1 Markov Decision Process and RL

Reinforcement learning (RL) provides a principled framework for sequential decision making under uncertainty, typically formalized as a Markov Decision Process (MDP) (Puterman, 2014; Sutton & Barto, 2018; Li, 2017). In an MDP $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, an agent learns a policy $\pi_\theta(a|s)$ to maximize the expected discounted return $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$ (Mnih et al., 2013). Despite strong empirical performance across a wide range of sequential decision-making problems, designing effective reward functions remains a central challenge, particularly when the environment provides only sparse or delayed supervision (Yu et al., 2025; Eschmann, 2021; Dewey, 2014).

2.2 Vision-Language Models (VLMs) and RL

Vision-language models (VLMs) learn joint visual-text representations and have shown strong zero-shot generalization across vision tasks (Radford et al., 2021; Alayrac et al., 2022; Jia et al., 2021; Li et al., 2022). Their semantic grounding makes them increasingly useful for RL, where they have been applied to reward specification and shaping (Mahmoudieh et al., 2022; Rocamonde et al., 2023; Adeniji et al., 2023; Baumli et al., 2023; Lubana et al., 2023; Dang et al., 2023; Huang et al., 2024; Fu et al., 2024b), task description and success verification (Du et al., 2023), and state representation learning (Chen et al., 2024). These works suggest that VLMs can serve as powerful priors for vision-based RL by providing richer representations and more flexible reward signals.

2.3 VLMs as Rewards

Leveraging Vision-Language Models (VLMs) as reward sources has become an active direction in RL (Mahmoudieh et al., 2022; Rocamonde et al., 2023; Adeniji et al., 2023; Baumli et al., 2023; Lubana et al., 2023; Dang et al., 2023; Nam et al., 2023; Fu et al., 2024b; Venuto et al., 2024; Lee et al., 2026; Sontakke et al., 2023; Wang et al., 2024). Existing methods either directly use VLM-based similarity between observations and task instructions as rewards (Mahmoudieh et al., 2022; Rocamonde et al., 2023; Adeniji et al., 2023), or use such signals as auxiliary supervision to complement environment rewards (Rocamonde et al., 2023; Baumli et al., 2023; Lubana et al., 2023; Dang et al., 2023; Fu et al., 2024b). This is particularly useful in sparse-reward settings, where the task reward is often binary, e.g., $r_t^{\text{task}} = \delta_{\text{success}}$, and provides limited guidance for exploration.

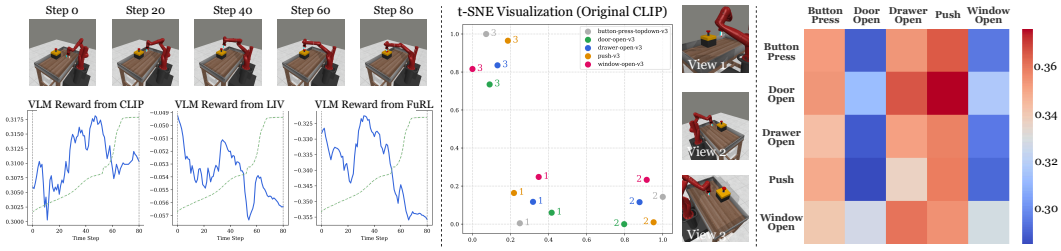

Figure 2: **Reward Misalignment in VLM-Based Methods. Left:** VLM reward signals along an oracle `button-press-topdown` trajectory. The green dashed curve denotes the environment-provided dense reward in Meta-World, whose scale differs from VLM rewards and is shown only to indicate the overall trend of task progress. **Middle**: t-SNE projection of image embeddings from five Meta-World tasks under three viewpoints, showing that embeddings cluster by viewpoint rather than by task identity. **Right**: Text–image similarity matrix for the instruction set (`button-press-topdown`, `door-open`, `drawer-open`, `push`, and `window-open`) in original CLIP, where rows correspond to image embeddings and columns correspond to text embeddings. Ideally, the diagonal should dominate, but weak diagonal structure and noisy off-diagonal activations indicate poor alignment between text and image embeddings.

A common formulation augments the task reward with a VLM-derived dense signal (Chen et al., 2024; Rocamonde et al., 2023; Fu et al., 2024b):

$$r_t = r_t^{\text{task}} + \rho \cdot r_t^{\text{vlm}}, \tag{1}$$

where ρ controls the contribution of the VLM reward. In many works, r_t^{vlm} is instantiated as a CLIP score, computed as the cosine similarity between the language-goal embedding and the current observation embedding:

$$r_t^{\text{vlm}} \triangleq r_t^{\text{CLIP}} = \frac{\langle e_l, e_{o_t} \rangle}{\|e_l\| \cdot \|e_{o_t}\|}, \tag{2}$$

where o_t is the observation, l is the language instruction, and e_l, e_{o_t} are their corresponding embeddings (Radford et al., 2021).

2.4 Challenges in VLM Rewards

Despite its simplicity, CLIP-based reward is often criticized for being fuzzy: prior work (Fu et al., 2024b; Lubana et al., 2023; Huang et al., 2024; Mahmoudieh et al., 2022; Rocamonde et al., 2023) shows that it can suffer from semantic misalignment and false positives, providing unreliable feedback for complex tasks. High similarity scores may be caused by spurious visual–text matches unrelated to task progress, thereby misleading exploration. Figure 2 left illustrates such noisy and uninformative reward dynamics.

Recent methods tackle this issue from two directions. BIMI (Huang et al., 2024) and VLM-RMs-GB (Rocamonde et al., 2023) suppress irrelevant noise, with the latter using Goal-Baseline Regularization to project out extraneous information; however, they still struggle with false positives or limited adaptability. In contrast, LIV (Ma et al., 2023) and FuRL (Fu et al., 2024b) fine-tune reward representations through value-oriented objectives or reward-space alignment. Yet, as shown in Figure 2, these approaches remain imperfect.

While these studies provide valuable insights, their analyses remain fragmented: some highlight noise and instability (Huang et al., 2024; Lubana et al., 2023; Rocamonde et al., 2023), others note misalignment (Fu et al., 2024b; Ma et al., 2023), but none offer a comprehensive perspective. In this work, we identify three fundamental challenges underlying VLM rewards—**weak spatial grounding**, **lack of task progress sensitivity**, and **semantic misalignment**—which we discuss systematically in this section.

Weak Spatial Grounding. Ideally, VLM-based rewards should capture task semantics invariant to changes in visual perspective (Ma et al., 2023; Pang et al., 2025; Liu et al., 2024; Zhou et al., 2023). However, our analysis shows that Meta-World embeddings cluster by *camera viewpoint* rather than *task identity* (Figure 2, middle), indicating that VLMs rely heavily on superficial visual cues.

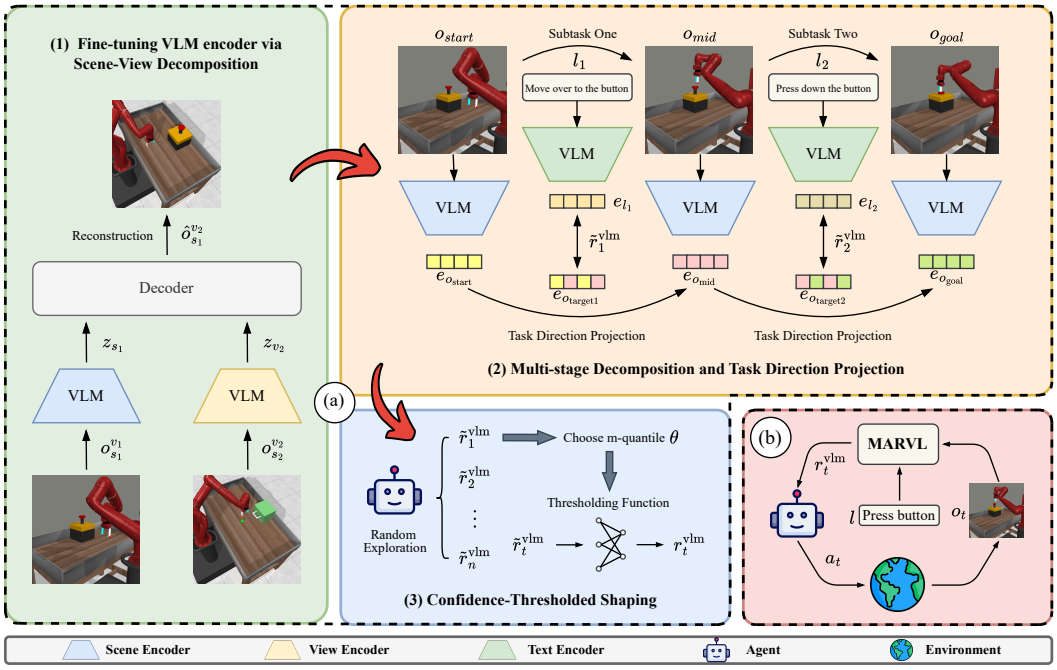

Figure 3: **Overview of MARVL.** (a) We first fine-tune the VLM encoder via Scene-View Decomposition, followed by Multi-stage Decomposition with Task Direction Projection and Confidence-Thresholded Shaping to derive calibrated VLM rewards. (b) MARVL could be easily integrated into the RL loop, where the agent interacts with the environment under dense VLM reward guidance.

Consequently, the resulting reward may respond to incidental viewpoint variation rather than changes in the task-relevant environment state.

Lack of Progress Awareness. Dense rewards should correlate positively with task progress (Fu et al., 2024b; Ma et al., 2023; Huang et al., 2024). Yet along an oracle trajectory, VLM rewards exhibit strong volatility and often oscillate or decrease even as the agent approaches the goal (Figure 2, left). This suggests poor sensitivity to fine-grained state changes, making the signal unreliable for policy learning.

Semantic Misalignment. VLM reward design assumes that instructions are well aligned with their corresponding observations (Fu et al., 2024b; Huang et al., 2024; Lubana et al., 2023; Sontakke et al., 2023; Ma et al., 2023). However, the similarity heatmap shows diffuse activations and a weak diagonal (Figure 2, right), indicating poor task separability. Consistently, adapted rewards such as LIV and FuRL can remain strictly negative (Figure 2, left), providing only weak ordinal cues rather than a calibrated measure of semantic correctness.

3 Methods

3.1 Fine-tuning VLM via Scene-View Decomposition

To mitigate the view-sensitivity of off-the-shelf VLMs, we employ a **Scene-View Decomposition** framework inspired by Pang et al. (2025). We leverage an automatically collected dataset of **500** multi-view Meta-World images to adapt the VLM's visual encoder toward robust spatial grounding.

Disentangled Architecture. As illustrated in Figure 3(a), we adopt a dual-encoder–decoder structure initialized from CLIP. Given an image x, the model produces two disentangled codes:

- **Scene representation** $z_s = E_s(x)$: captures view-invariant, task-critical semantics (e.g., robot and object states). E_s is retained as the backbone for our downstream reward model.

- **View representation** $z_v = E_v(x)$: captures nuisance variations like camera pose. This auxiliary branch is discarded after training.

To enforce disentanglement, we employ a reconstruction objective based on latent swapping. Given two observations $o_{s_1}^{v_1}$ and $o_{s_2}^{v_2}$ that differ in both scene and viewpoint, we combine the scene code of the first (z_{s_1}) with the view code of the second (z_{v_2}) to synthesize a cross-compositional reconstruction $\hat{o}_{s_1}^{v_2} = D(z_{s_1}, z_{v_2})$. This forces z_s to isolate spatial semantics from perspective shifts. The total loss is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_1 \mathcal{L}_{\text{shuffle}} + \lambda_2 \mathcal{L}_{\text{consistency}} + \lambda_3 \mathcal{L}_{\text{clip}}. \tag{3}$$

We provide full definitions, mathematical formulations, and fine-tuning hyperparameters for each loss term in Appendix C.

As shown in Figure 4, this process successfully realigns the embedding space: scene embeddings cluster by task identity rather than viewpoint, providing a stable, spatially grounded foundation for MARVL.

3.2 Task Direction Projection via Multi-Stage Decomposition

To bridge the gap between high-level language objectives and fine-grained state transitions, we adopt a Multi-stage Decomposition strategy (Figure 3(a)), which breaks the task into a sequence of subtasks $\{l_k\}$ with corresponding intermediate goal states. This decomposition makes progress evaluation stage-specific, allowing dense rewards to track task-aligned changes more reliably.

However, off-the-shelf VLM embeddings are high-dimensional and dominated by task-irrelevant variations, causing raw similarity-based rewards to fluctuate and violate monotonicity even within a single stage. To address this issue, we introduce **Task Direction Projection (TDP)**, a geometric mechanism that suppresses orthogonal noise while preserving task-aligned progress signals.

Task Direction Principle. Rather than assuming linear evolution of physical trajectories or embeddings, we rely on a weaker condition: for each subtask, there exists a task-aligned direction in the joint visual–semantic embedding space such that progress along this direction provides a monotonic proxy for task completion after noise attenuation. This direction is defined by the displacement between the start and goal embeddings and is used as a reference axis for filtering, not as a literal model of task execution.

Formally, given a task direction vector $\mathbf{d}$, we define a projection operator

$$\mathcal{P}_{\mathbf{d}}(\mathbf{x}) = \left(\alpha \frac{\mathbf{d}\mathbf{d}^\top}{\|\mathbf{d}\|^2} + (1-\alpha)\mathbf{I} \right) \mathbf{x}, \quad \alpha \in [0,1]. \tag{4}$$

This operator preserves the component of $\mathbf{x}$ aligned with $\mathbf{d}$ while attenuating orthogonal components by a factor $(1-\alpha)$. As shown in Appendix D, multi-stage decomposition confines each projection to a semantically coherent subtask, enabling anisotropic filtering to substantially improve the signal-to-noise ratio of the embedding trajectory and restore stage-wise reward monotonicity along the projected progress coordinate, without assuming linearity in the underlying physical dynamics.

Text Projection. We isolate the dynamic "action" semantics from static context. Given a subtask l_k (e.g., "robot press button") and a baseline l_b (e.g., "robot arm"), we define the text direction $\mathbf{d}_{\text{text}} = e_{l_k} - e_{l_b}$. The refined target $e'_{l_k} = \mathcal{P}_{\mathbf{d}_{\text{text}}}(e_{l_k})$ thus focuses strictly on the manipulation intent.

Image Projection. Given the start observation o_{start} and the subtask goal observation o_{goal}, we define the visual task direction $\mathbf{d}_{\text{img}} = e_{o_{\text{goal}}} - e_{o_{\text{start}}}$. The projected embedding of the current observation o_t is defined as:

$$e'_{o_t} = \mathcal{P}_{\mathbf{d}_{\text{img}}}(e_{o_t} - e_{o_{\text{start}}}) + e_{o_{\text{start}}}. \tag{5}$$

This formulation grounds the reward in a projected progress coordinate defined by the start and goal embeddings, suppressing background variations while preserving task-aligned semantic change.

Reward and Transition. The dense reward is computed as the cosine similarity between the projected embeddings: $\tilde{r}_t = \cos(e'_{o_t}, e'_{l_k})$. Stage transitions are automatically managed by monitoring the similarity between the current observation and the intermediate target embedding. Once this similarity surpasses a predefined threshold, the system recognizes the subtask as complete and advances the agent to the next stage. In our experiments, intermediate target images are manually collected;

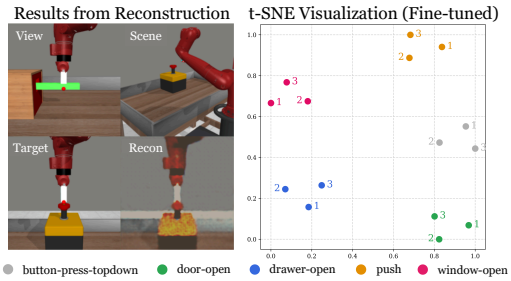

Figure 4: **Effectiveness of Scene-View Decomposition. Left:** Reconstructions retain critical spatial structure (e.g., arm pose) despite high-frequency pixel abstraction, verifying the encoder's geometric grounding. **Right:** t-SNE visualization confirms that fine-tuned embeddings cluster by *task identity* rather than camera viewpoint, demonstrating effective disentanglement of semantics from nuisance factors.

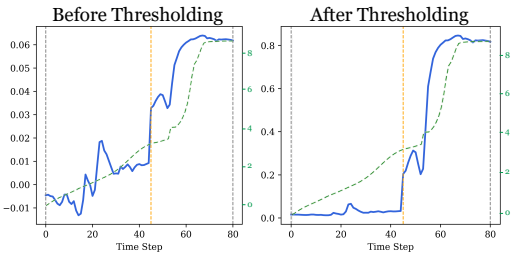

Figure 5: The raw MARVL reward (left) exhibits noise and spurious positive activations at early steps, while the thresholded reward (right) suppresses these false positives and amplifies high-confidence signals, yielding a cleaner and more reliable reward signal under MARVL. The orange dashed line indicates the stage transition point. The green dashed curve shows the environment-provided dense reward.

however, Appendix B.3 demonstrates that they can be generated fully automatically using LLMs and image generation models.

3.3 Confidence-Thresholded Shaping

In VLM-based rewards, irrelevant actions often produce small but consistent positive scores due to generic semantic similarity. As a result, agents can accumulate these trivial signals instead of making genuine task progress.

To rectify this, we introduce **Confidence-Thresholded Shaping (CTS)**. We first quantify the environment's background noise by collecting scores from the first 10k steps, setting the threshold θ as the m-th quantile. We then apply a soft gating function:

$$r_t^{\text{vlm}} = \sigma\big(\kappa(\tilde{r}_t^{\text{vlm}} - \theta)\big). \tag{6}$$

This mechanism serves a **dual purpose**: it effectively **zeros out low-magnitude noise** to prevent spurious accumulation, while simultaneously **amplifying high-confidence matches** via the steepness parameter κ. As illustrated in Figure 5, this sharpens the reward gradient for semantically relevant states, blocking trivial solutions while providing a decisive, high-contrast signal once the agent breaks through the noise floor. We further explore more advanced filtering schemes—such as adaptive EMA with hysteresis and Kalman filtering—which are discussed in Appendix E.3.

4 Experiments

In this section, we investigate the following questions: (1) How does MARVL perform compared to existing VLM-reward baselines? (2) Are the main components of MARVL effective individually? (3) Can MARVL generalize across different RL backbones? (4) Does MARVL transfer well to new environments and camera settings?

4.1 Experiment Setup

We use the **Meta-World** benchmark (Yu et al., 2020) as our primary testbed for evaluating VLM-based reward design in sparse-reward visual RL. We select eight environments spanning diverse behaviors, including pushing, button pressing, door opening, and drawer manipulation. Meta-World is widely used in prior work on VLM-based rewards and visual manipulation, enabling fair comparison with existing methods, and its tasks are semantically well-defined and exhibit clear notions of incremental progress, making them particularly suitable for evaluating language-guided reward design. To assess cross-domain generalization, we further evaluate MARVL on the **Panda-Gym** benchmark

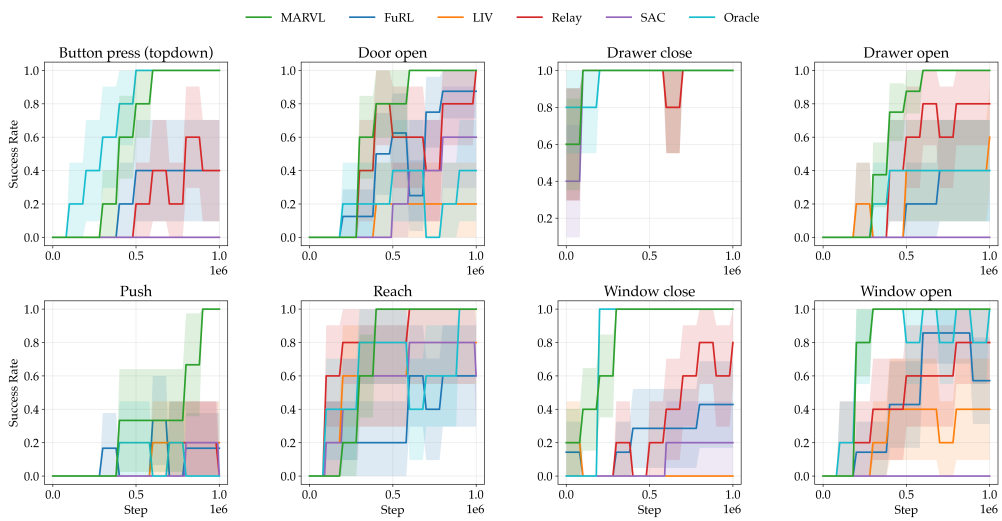

Figure 6: Learning curves on eight Meta-World tasks. We report the mean success rate across 5 random seeds. MARVL (ours) is compared against state-of-the-art VLM-reward baselines (FuRL, LIV, Relay) and an Oracle agent trained with the environment's standard hand-crafted dense reward using internal states. MARVL consistently outperforms prior VLM methods and achieves performance parity with the privileged Oracle, demonstrating that learned visual rewards can effectively match the quality of engineered dense rewards.

(Gallouédec et al., 2021), which differs from Meta-World in both robot embodiment and visual appearance.

4.2 Baselines

To validate the effectiveness of MARVL, we compare it against the following baselines. All methods use the same state-based SAC backbone (Haarnoja et al., 2018) and differ only in how rewards are computed. The hyperparameters for the SAC are attached in Appendix G.

1. SAC (Haarnoja et al., 2018). A standard SAC agent using only the sparse binary task reward r_t^{task}. This represents the lower bound of performance given no additional shaping signals.

2. LIV (Ma et al., 2023). A state-based SAC agent trained with task reward and dense LIV reward as in Eqn. 1.

3. FuRL (Fu et al., 2024b). A recent VLM-based reward method that leverages two key mechanisms: lightweight reward alignment and relay RL, where a VLM-guided agent and an auxiliary SAC agent take turns acting in the environment and share a replay buffer so that the SAC agent can help the VLM agent escape local minima induced by fuzzy VLM rewards.

4. Relay (Lan et al., 2023). A simplified variant of FuRL that keeps only the relay RL mechanism used in FuRL.

5. MARVL. Our full method including Scene-View Decomposition, Task Direction Projection, and confidence-thresholded reward shaping.

4.3 Main Results on Meta-World Tasks

Figure 6 summarizes the performance of MARVL across eight sparse-reward tasks, compared against prior VLM-based methods and an Oracle baseline. Here, the Oracle corresponds to an agent using the dense rewards provided by Meta-World, which are manually designed from environment states. Across all tasks, MARVL consistently achieves higher success rates and faster convergence than existing VLM-based rewards. While methods such as FuRL and Relay improve exploration over

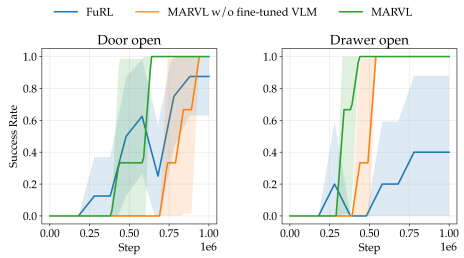

Figure 7: **Impact of Scene-View Decomposition**

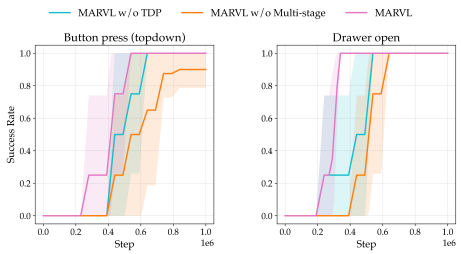

Figure 8: **Impact of TDP and Multi-Stage Decomposition**

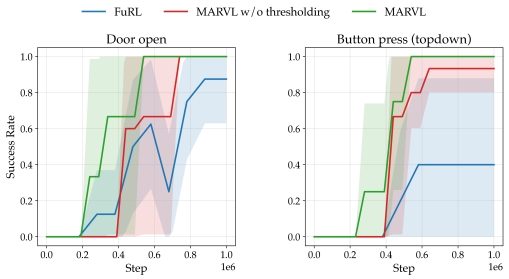

Figure 9: **Impact of Confidence-Thresholded Shaping**

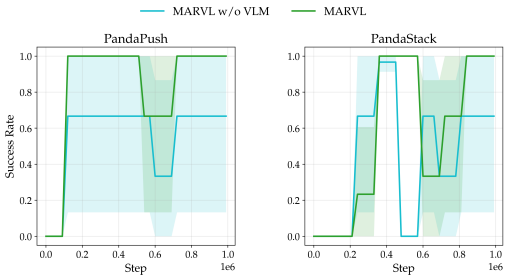

Figure 10: **Generalization to Novel Environments**

standard RL, they remain sensitive to noisy and unstructured visual signals, often converging to suboptimal plateaus.

We further compare MARVL against the Meta-World dense Oracle rewards. Despite relying solely on learned visual–semantic signals, MARVL matches the Oracle's performance on tasks such as Button Press and Window Close, and exceeds it in exploration-heavy environments, including Push and Door Open. This indicates that semantically grounded rewards can provide even more effective guidance than handcrafted dense rewards in complex, sparse-reward settings.

4.4 Ablation Studies

In this section, we conduct a comprehensive analysis to investigate the internal mechanisms and external robustness of MARVL.

Impact of Scene-View Decomposition. Removing Scene-View Decomposition and relying on raw, pre-trained VLM encoders consistently degrades performance (Figure 7), indicating that separating view-dependent variation from scene-level semantics improves reward stability. Further analysis is provided in Appendix E.2.

Impact of Task Direction Projection and Multi-Stage Decomposition. Ablating either Task Direction Projection (TDP) or Multi-Stage Decomposition preserves final success but significantly slows convergence (Figure 8), highlighting MARVL's improved sample efficiency.

Impact of Confidence-Thresholded Shaping. Removing Confidence-Thresholded Shaping and directly using raw VLM scores degrades both sample efficiency and training stability (Figure 9). Thresholding accelerates convergence in Door Open and prevents performance collapse in Button Press, underscoring its role in suppressing spurious VLM signals.

Transfer Across Simulated Environments. We evaluate MARVL on Panda-Gym to examine whether the learned reward representation transfers across simulated environments with different visual appearances, simulator configurations, and agent embodiments. As shown in Figure 10, MARVL maintains high success rates without target-domain adaptation, whereas the non-fine-tuned VLM degrades substantially. This indicates that fine-tuning improves transferable spatial grounding rather than merely fitting Meta-World-specific visual statistics.

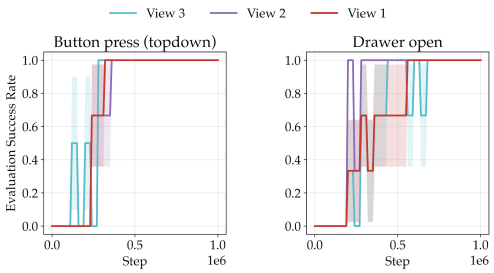

Figure 11: **Generalization to Camera Settings**

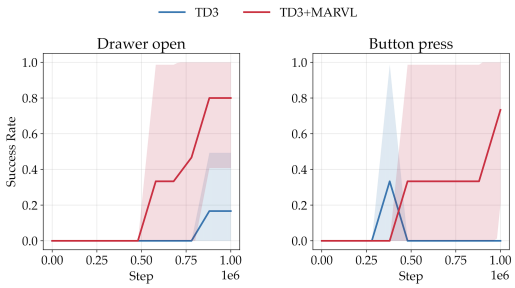

Figure 12: **Generalization across RL Backbones**

Robustness to Diverse Camera Configurations. We assess robustness to viewpoint changes by evaluating MARVL under three distinct camera configurations. As shown in Figure 11, performance remains consistent across views for both Button Press and Drawer Open tasks, with all settings converging to high success rates despite minor variance during early training. This suggests that MARVL is invariant to viewpoint-specific visual cues.

Generalization across RL Backbones. To test algorithmic robustness, we replace the SAC backbone with TD3 (Fujimoto et al., 2018) while keeping the reward shaping unchanged. As shown in Figure 12, MARVL maintains strong sample efficiency and substantially outperforms sparse-reward baselines, indicating that MARVL functions as an algorithm-agnostic reward interface across off-policy RL methods.

5 Conclusions

This work shows that VLM-based rewards become more effective when their representations are explicitly grounded and structured. By integrating Scene-View Decomposition, Task Direction Projection, and Confidence-Thresholded Shaping, MARVL converts raw VLM similarities into stable, progress-sensitive rewards, improving sample efficiency and robustness across visual RL tasks. These results suggest that prior limitations arise less from insufficient semantic knowledge than from insufficient structural alignment.

MARVL still has limitations. It relies on lightweight VLM fine-tuning and is currently validated only on simulation-based visual control benchmarks, making broader real-world evaluation an important future direction. Scaling to longer-horizon and more compositionally complex tasks is another promising avenue. Overall, MARVL shows that modest adaptation and principled reward structuring can substantially improve VLM-based reinforcement learning.

AI use statement

In this work, we did not use generative AI tools for any of the tasks that require disclosure under the ICLR 2027 AI Policy for Authors: no generative AI was used to develop the theoretical framework or formulate mathematical claims, to write or verify proofs, to propose or refine hypotheses, to design the methodology or experiments, to implement the method, to process data, or to interpret results. All research ideas, methods, experiments, analyses and conclusions are the work of the authors. The only generative models involved in the research itself are the language model (Qwen-3-Flash) and the image-editing model (Qwen-ImageEdit-Plus) that constitute the automated sub-goal generation pipeline evaluated in Appendix B.3; they are components of the proposed method, are fully described there, and were not used to assist in producing this manuscript.

Generative AI tools (large language models) were used only for two tasks with recommended disclosure: (i) editing author-written text to improve grammar, readability and consistency of terminology, and (ii) assisting with LaTeX formatting, such as adjusting figure layouts and reference formatting. These tools did not introduce new content, claims, results or citations. We have reviewed all AI-assisted edits to confirm that they preserve the intended technical meaning of the original text. We

take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Ademi Adeniji, Amber Xie, Carmelo Sferrazza, Younggyo Seo, Stephen James, and Pieter Abbeel. Language reward modulation for pretraining reinforcement learning. *arXiv preprint arXiv:2308.12270*, 2023.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Kate Baumli, Satinder Baveja, Feryal Behbahani, Harris Chan, Gheorghe Comanici, Sebastian Flennerhag, Maxime Gazeau, Kristian Holsheimer, Dan Horgan, Michael Laskin, et al. Vision-language models as a source of rewards. *arXiv preprint arXiv:2312.09187*, 2023.

William Chen, Oier Mees, Aviral Kumar, and Sergey Levine. Vision-language models provide promptable representations for reinforcement learning. *arXiv preprint arXiv:2402.02651*, 2024.

Xuzhe Dang, Stefan Edelkamp, and Nicolas Ribault. Clip-motion: Learning reward functions for robotic actions using consecutive observations. *arXiv preprint arXiv:2311.03485*, 2023.

Daniel Dewey. Reinforcement learning and the reward engineering principle. In *AAAI spring symposia*, 2014.

Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando De Freitas, and Serkan Cabi. Vision-language models as success detectors. *arXiv preprint arXiv:2303.07280*, 2023.

Jonas Eschmann. Reward function design in reinforcement learning. *Reinforcement learning algorithms: Analysis and Applications*, pp. 25–33, 2021.

Yuwei Fu, Haichao Zhang, Di Wu, Wei Xu, and Benoit Boulet. Robot policy learning with temporal optimal transport reward. *Advances in Neural Information Processing Systems*, 37:122078–122103, 2024a.

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

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Quentin Gallouédec, Nicolas Cazin, Emmanuel Dellandréa, and Liming Chen. panda-gym: Open-source goal-conditioned environments for robotic learning. *arXiv preprint arXiv:2106.13687*, 2021.

Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3389–3396. IEEE, 2017.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Dong Han, Beni Mulyana, Vladimir Stankovic, and Samuel Cheng. A survey on deep reinforcement learning algorithms for robotic manipulation. *Sensors*, 23(7):3762, 2023.

Sukai Huang, Shu-Wei Liu, Nir Lipovetzky, and Trevor Cohn. The dark side of rich rewards: Understanding and mitigating noise in vlm rewards. *arXiv preprint arXiv:2409.15922*, 2024.

Sinan Ibrahim, Mostafa Mostafa, Ali Jnadi, Hadi Salloum, and Pavel Osinenko. Comprehensive overview of reward engineering and shaping in advancing reinforcement learning applications. *IEEE Access*, 2024.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pp. 651–673. PMLR, 2018.

Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *Journal of machine learning research*, 22(30):1–82, 2021.

Li-Cheng Lan, Huan Zhang, and Cho-Jui Hsieh. Can agents run relay race with strangers? generalization of rl to out-of-distribution trajectories. *arXiv preprint arXiv:2304.13424*, 2023.

Tony Lee, Andrew Wagenmaker, Karl Pertsch, Percy Liang, Sergey Levine, and Chelsea Finn. Roboreward: General-purpose vision-language reward models for robotics. *arXiv preprint arXiv:2601.00675*, 2026.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

Fanfan Liu, Feng Yan, Liming Zheng, Chengjian Feng, Yiyang Huang, and Lin Ma. Robouniview: Visual-language model with unified view representation for robotic manipulation. *arXiv preprint arXiv:2406.18977*, 2024.

Ekdeep Singh Lubana, Johann Brehmer, Pim De Haan, and Taco Cohen. Fomo rewards: Can we cast foundation models as reward functions? *arXiv preprint arXiv:2312.03881*, 2023.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning*, pp. 23301–23320. PMLR, 2023.

Yecheng Jason Ma, Joey Hejna, Chuyuan Fu, Dhruv Shah, Jacky Liang, Zhuo Xu, Sean Kirmani, Peng Xu, Danny Driess, Ted Xiao, et al. Vision language models are in-context value learners. In *The Thirteenth International Conference on Learning Representations*, 2024.

Parsa Mahmoudieh, Deepak Pathak, and Trevor Darrell. Zero-shot reward specification via grounded natural language. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14743–14752. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/mahmoudieh22a.html`.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Taewook Nam, Juyong Lee, Jesse Zhang, Sung Ju Hwang, Joseph J Lim, and Karl Pertsch. Lift: Unsupervised reinforcement learning with foundation models as teachers. *arXiv preprint arXiv:2312.08958*, 2023.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Jing-Cheng Pang, Nan Tang, Kaiyuan Li, Yuting Tang, Xin-Qiang Cai, Zhen-Yu Zhang, Gang Niu, Masashi Sugiyama, and Yang Yu. Learning view-invariant world models for visual robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vJwjWyt4Ed.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. *arXiv preprint arXiv:2310.12921*, 2023.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

Bharat Singh, Rajesh Kumar, and Vinay Pratap Singh. Reinforcement learning in robotic applications: a comprehensive survey. *Artificial Intelligence Review*, 55(2):945–990, 2022.

Sumedh Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies. *Advances in Neural Information Processing Systems*, 36:55681–55693, 2023.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. *Annual Review of Control, Robotics, and Autonomous Systems*, 8(1):153–188, 2025.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Sreyas Venkataraman, Yufei Wang, Ziyu Wang, Navin Sriram Ravie, Zackory Erickson, and David Held. Real-world offline reinforcement learning from vision language model feedback. *arXiv preprint arXiv:2411.05273*, 2024.

David Venuto, Sami Nur Islam, Martin Klissarov, Doina Precup, Sherry Yang, and Ankit Anand. Code as reward: Empowering reinforcement learning with vlms. *arXiv preprint arXiv:2402.04764*, 2024.

Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. *arXiv preprint arXiv:2402.03681*, 2024.

Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Rui Yu, Shenghua Wan, Yucen Wang, Chen-Xiao Gao, Le Gan, Zongzhang Zhang, and De-Chuan Zhan. Reward models in deep reinforcement learning: A survey. *arXiv preprint arXiv:2506.15421*, 2025.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Yiyang Zhou, Qinghai Zheng, Shunshun Bai, and Jihua Zhu. Semantically consistent multi-view representation learning. *Knowledge-Based Systems*, 278:110899, 2023.

A Broader Impact and Ethical Considerations

Societal Impact and Democratization. MARVL significantly lowers the technical barrier for robotic programming by replacing complex reward engineering with natural language instructions. Such democratization facilitates the deployment of intelligent assistants in unstructured environments (e.g., household service, elderly care), potentially alleviating labor shortages in critical sectors. However, as with broadly applicable advancements in automation, widespread adoption necessitates ongoing discourse regarding workforce adaptation to mitigate potential displacement in manual labor markets.

Safety and Ethical Considerations. The safety of our system is intrinsically bound to the reliability of the underlying Vision-Language Models (VLMs). Since MARVL leverages pre-trained encoders, it may inherit perceptual hallucinations where the model is affected by yield false-positive success signals, posing risks in safety-critical physical environments. Consequently, real-world deployment must be accompanied by robust safety constraints and human-in-the-loop verification. Furthermore, foundation models often contain inherent biases regarding cultural contexts or object typologies, these must be rigorously audited to prevent performance disparities across diverse visual settings.

B Experimental Setup

B.1 Meta-World and Panda-Gym Benchmark

We employ two established robotic manipulation and reinforcement learning benchmarks: **Meta-World** (Yu et al., 2020) and **Panda-Gym** (Gallouédec et al., 2021) to evaluate the generalizability of MARVL across diverse task types and embodiments. **Meta-World**, built on MuJoCo (Todorov et al., 2012), features a 7-DoF Sawyer arm with a 4-dimensional continuous action space. We select 8 representative tasks (hidden-goal) to investigate complex manipulation capabilities, specifically focusing on spatial positioning, fine-grained interaction and articulated object manipulation. Furthermore, **Panda-Gym**, based on PyBullet, employs the Franka Emika Panda robot (7-DoF), which presents significant differences in kinematics, geometrical appearance, and rendering style compared to the Sawyer arm used in Meta-World. We use it to assess robustness against visual domain shifts and novel embodiments.

B.2 Hyperparameters

Table 1 outlines the key hyperparameters for the backbone algorithm, MARVL, the baseline methods followed from prior work, Table 2 details the specific settings for VLM fine-tuning. To ensure a fair comparison, We adopt the FuRL codebase `https://github.com/fuyw/FuRL` and adhere to the original baseline implementations and hyperparameters. Our code will be released upon acceptance.

B.3 Multi-Stage Decomposition and Automation

Figure 15 illustrates the multi-stage text descriptions (with "robot arm" as the baseline text) and the corresponding target images used in our experiments. While these intermediate goals can be specified manually for controlled comparison, the proposed multi-stage decomposition does not rely on human supervision. As shown in Figure 13, the entire pipeline can be fully automated using an AI-based generation process, without requiring expert demonstrations or manually predefined task stages. Specifically, we employ Qwen-3-Flash to rewrite a task instruction into sequential sub-goals, and Qwen-ImageEdit-Plus to synthesize the corresponding visual targets.

Beyond the Button Press task shown in Figure 13, we further evaluate the fully automated pipeline across a broader set of Meta-World environments. The success rate curves are summarized in Figure 14, where MARVL with AI-labeled intermediate goals consistently achieves competitive performance across diverse manipulation tasks. These results indicate that the proposed automation strategy generalizes beyond a single task and can serve as a viable substitute for manually specified sub-goals in practice.

Importantly, this decomposition introduces structural guidance rather than additional semantic supervision: consistent with the basic VLM reward approach, both the human-labeled and AI-labeled

variants are derived from the same original task instruction, differing only in how the instruction is restructured into intermediate goals. Counter-intuitively, our experimental results show that the AI-labeled pipeline achieves even faster convergence and superior asymptotic performance compared to the human-labeled baseline constructed from oracle simulation frames. We hypothesize that this advantage arises because AI-generated goal images are synthesized directly from the textual sub-goals, and are therefore naturally aligned with the language instructions by construction, avoiding the potential semantic ambiguity or incidental visual details present in human-selected frames.

Notably, we observe that the AI-labeled variant exhibits higher variance in training dynamics compared to its human-labeled counterpart. We attribute this increased fluctuation primarily to the quality and consistency of the generated goal images: unlike oracle frames sampled from the simulator, AI-synthesized images may occasionally contain visual artifacts or subtle geometric inconsistencies, which introduce noise into the reward signal. Despite this variability, the overall performance remains robust, suggesting that MARVL is resilient to moderate imperfections in intermediate goal specification.

Despite these promising results, we emphasize that the effectiveness of the automated pipeline is ultimately bounded by the quality of the generative models. Severe visual hallucinations or physically implausible generations could mislead the reward signal and degrade policy learning. Ensuring the physical plausibility and controllability of AI-generated intermediate goals thus remains an important direction for future work.

B.4 Computational Resources

All RL experiments (including baselines and MARVL) were conducted on a workstation equipped with NVIDIA GeForce RTX 3090 GPU and Dual Intel Xeon Silver 4210R CPUs (totaling 20 physical cores and 40 threads) , while the VLM fine-tuning was performed on NVIDIA A100 GPU with 28 CPU cores and 256GB RAM. Regarding runtime, the Scene-View Decomposition fine-tuning requires approximately 2 hours as a one-time cost, MARVL averages 1 hour per task (1M environment steps).

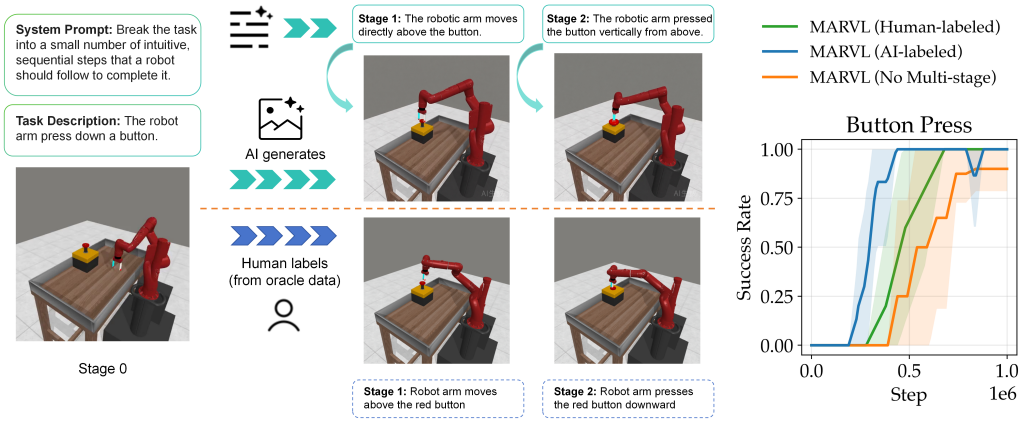

Figure 13: **Automated Sub-goal Generation Pipeline and Performance Comparison. Left:** The AI pipeline utilizes an LLM to decompose task instructions into textual sub-stages, which are then visualized by an image generation model to serve as intermediate visual goals. **Right:** Success rates on the Button Press task demonstrate that MARVL with AI-labeled goals (blue) achieves comparable or even superior performance to human-labeled goals (green), verifying the scalability and effectiveness of our generative pipeline.

C Fine-tuning Implementation Details

C.1 Data Collection

To construct multi-view paired data for scene–view disentanglement, we uniformly sample frames from random trajectories across tasks at different time steps. Using the MuJoCo interface, we

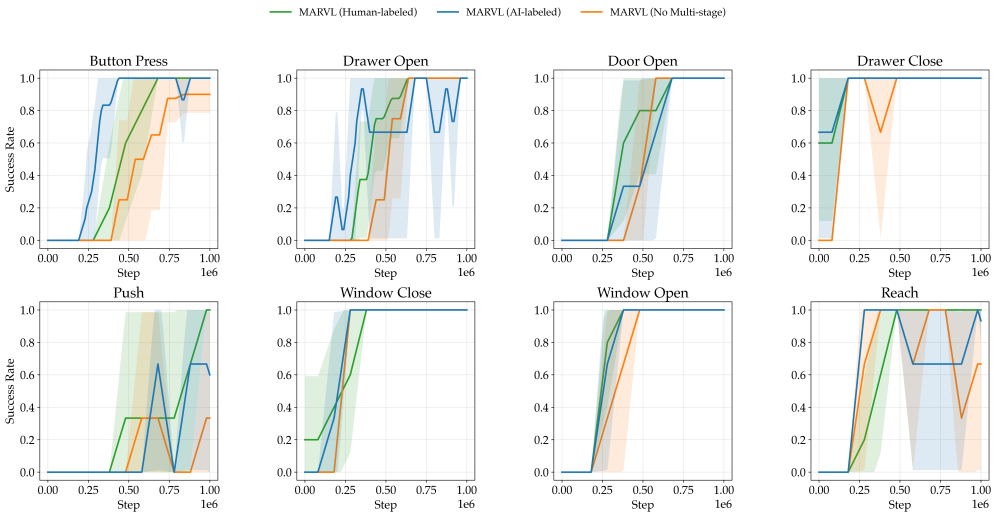

Figure 14: **Further Experiments on the Effectiveness of the Automated Sub-goal Generation Pipeline.**

programmatically perturb camera parameters in Meta-World to render each sampled scene from multiple viewpoints, while automatically recording the corresponding scene identity (Scene ID) and camera configuration (View ID). Importantly, this procedure yields a lightweight dataset of **500 images**, which are randomly shuffled and recombined to form training pairs during fine-tuning. No additional interaction, expert demonstrations, or target-domain data are required. Figure 16 illustrates representative multi-view examples from the collected dataset.

C.2 Architecture

The Scene Encoder E_s, View Encoder E_v, and Text Encoder are initialized with pre-trained CLIP ViT-B/32 (Radford et al., 2021) weights from laion2b_s34b_b79k (Schuhmann et al., 2022). They accept normalized 224×224 RGB images or task text descriptions as input, yielding 512-dimensional feature vectors. The Decoder is a lightweight network (approx. 10% of encoder parameters) based on transposed convolutions. It concatenates scene and view embeddings, followed by a projection layer and five transposed convolution blocks with Batch Normalization and CBAM (Woo et al., 2018) attention to reconstruct 224×224 RGB images. Detailed layer specifications are provided in Table 2.

C.3 Process and Loss function

The fine-tuning process proceeds in three substages. **Stage 1:** Both encoders remain frozen, and the decoder is optimized solely via $\mathcal{L}_{\text{recon}}$. **Stage 2:** We unfreeze the last four Transformer blocks of the ViT backbone and train using the full loss function. **Stage 3:** We progressively increase λ_{clip} to ensure encoders retains alignment with the original CLIP text embedding space. The Text Encoder remains frozen during the fine-tuning process.

We provide a detailed formulation of the fine-tuning loss function below. Given a batch of sampled pairs of scenes $\{s_i, s_j\}$ and viewpoints $\{v_m, v_n\}$, let $o_{s_i}^{v_m}$ denote an observation image of Scene i captured from View m. The scene embedding is denoted as $z_s^{(i,m)} = E_s(o_{s_i}^{v_m})$ and the view embedding is denoted as $z_v^{(i,m)} = E_v(o_{s_i}^{v_m})$. The Decoder $D(z_s, z_v)$ reconstructs an image $\hat{o}_{s_i}^{v_k}$ from these embeddings.

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_{\text{shuffle}}\mathcal{L}_{\text{shuffle}} + \lambda_{\text{consistency}}\mathcal{L}_{\text{consistency}} + \lambda_{\text{clip}}\mathcal{L}_{\text{clip}}$$

Reconstruction Loss $\mathcal{L}_{\text{recon}}$. To ensure feature completeness, we require the decoder D to reconstruct the original image $o_{s_i}^{v_m}$ by combining the scene embedding $z_{s_i}^{(i,n)}$ (different view n) with the view

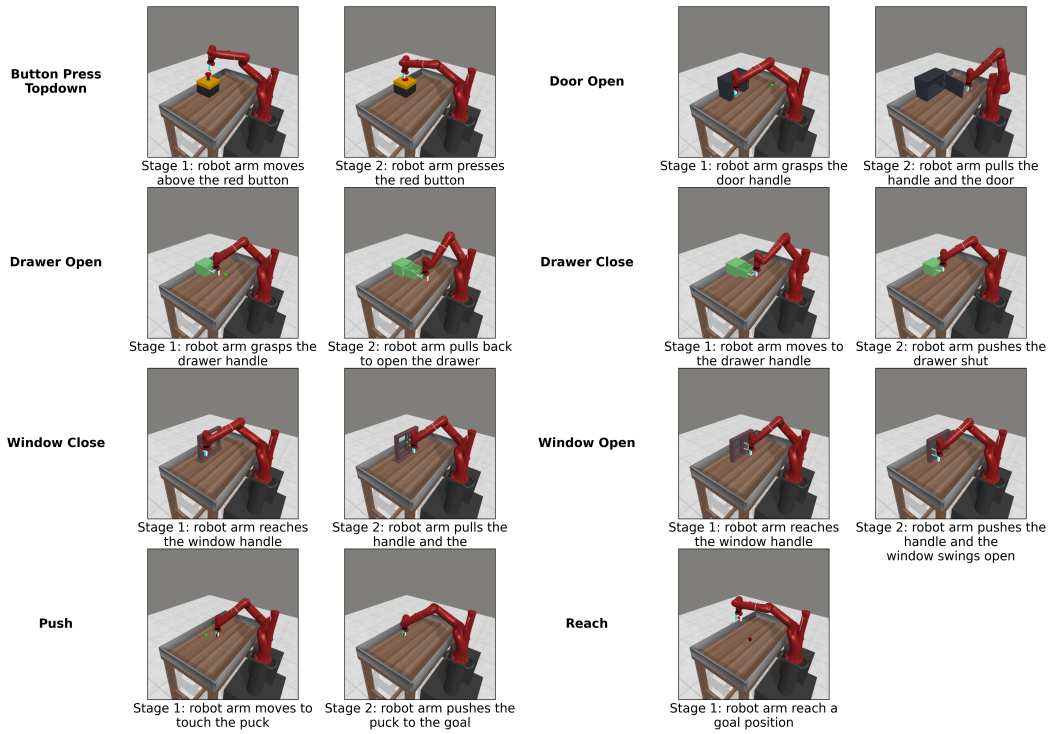

Figure 15: **Multi-Stage Target Image and Text Instruction**

embedding $z_{v_m}^{(j,m)}$ (different scene j), thereby forcing the model to learn scene-view decomposition by cross-reconstruction rather than self-reconstruction.

$$\hat{o}_{s_i}^{v_m} = D(z_{s_i}^{(i,n)}, z_{v_m}^{(j,m)})$$

$$\mathcal{L}_{\text{recon}} = \|\hat{o} - o\|_1 + \lambda_{\text{lpips}} \mathcal{L}_{\text{LPIPS}}(\hat{o}, o)$$

where $\mathcal{L}_{\text{LPIPS}}$ denotes the Learned Perceptual Image Patch Similarity metric (Zhang et al., 2018), employed to capture high-frequency perceptual details.

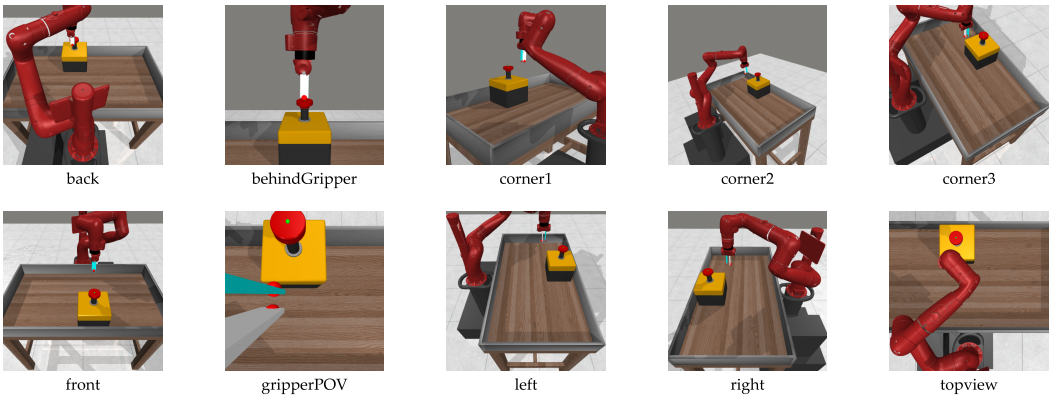

Figure 16: **Multi-View Robotic Arm Image Examples in Button-Press Task**

Disentanglement Shuffle Loss $\mathcal{L}_{\text{shuffle}}$. This loss explicitly enforces orthogonality between E_s and E_v via latent variable swapping:

- **Scene Invariance:** Swapping scene embeddings between different views of the same scene ($z_{s_i}^{(i,m)} \leftrightarrow z_{s_i}^{(i,n)}$) should yield invariant reconstructions, as the view code remains fixed.
- **View Invariance:** Swapping view embeddings between different scenes under the same viewpoint ($z_{v_m}^{(i,m)} \leftrightarrow z_{v_m}^{(j,m)}$) should similarly preserve the reconstruction, assuming the view code captures only perspective information.

Define $\hat{o}_{\text{base}} = D(z_{s_i}^{(i,m)}, z_{v_m}^{(i,m)})$ as the baseline reconstruction. The shuffle loss is defined as:

$$\mathcal{L}_{\text{shuffle}} = \underbrace{\|D(z_{s_i}^{(i,n)}, z_{v_m}^{(i,m)}) - \hat{o}_{\text{base}}\|_2^2}_{\text{Scene Swap}} + \underbrace{\|D(z_{s_i}^{(i,m)}, z_{v_m}^{(j,m)}) - \hat{o}_{\text{base}}\|_2^2}_{\text{View Swap}}$$

Notably, we align the reconstruction $\hat{o}_{\text{base}}$ rather than the ground-truth image $o_{s_i}^{v_m}$. This design choice isolates the disentanglement error from the decoder's inherent reconstruction imperfections.

Feature Consistency Loss $\mathcal{L}_{\text{consistency}}$. We impose direct constraints within the feature space to ensure that scene embeddings are insensitive to viewpoint shifts and view embeddings are insensitive to scene content variations:

$$\mathcal{L}_{\text{consistency}} = \underbrace{\left(1 - \cos(z_{s_i}^{(i,m)}, z_{s_i}^{(i,n)})\right)}_{\text{Scene Consistency}} + \underbrace{\left(1 - \cos(z_{v_m}^{(i,m)}, z_{v_m}^{(j,m)})\right)}_{\text{View Consistency}}$$

Semantic Preservation Loss $\mathcal{L}_{\text{clip}}$. To prevent catastrophic forgetting and preserve the pre-trained semantic alignment of the VLM, we employ an InfoNCE (Oord et al., 2018) contrastive loss. This objective aligns the fine-tuned scene embedding z_s with its corresponding textual instruction embedding t_i while separating it from irrelevant texts:

$$\mathcal{L}_{\text{clip}} = -\log \frac{\exp(z_{s_i} \cdot t_i/\tau)}{\sum_k \exp(z_{s_i} \cdot t_k/\tau)}$$

Collectively, these terms establish a synergistic optimization landscape: while $\mathcal{L}_{\text{recon}}$ ensures the latent representations retain sufficient information for physical state recovery, $\mathcal{L}_{\text{shuffle}}$ and $\mathcal{L}_{\text{consistency}}$ act as structural regularizers that aggressively penalize the leakage of view nuisance factors into the scene embedding. Finally, $\mathcal{L}_{\text{clip}}$ serves as a semantic anchor, preventing the fine-tuned manifold from drifting away from the pre-trained linguistic alignment essential for instruction following.

D Theoretical Analysis of Task Direction Projection

This section provides a rigorous theoretical analysis of the **Task Direction Projection (TDP)** mechanism introduced in Section 3.2. We model the embedding geometry to demonstrate that the proposed projection operator significantly enhances the Signal-to-Noise Ratio (SNR) and restores the monotonicity of the reward signal with respect to task progress. The mathematical symbols used in the following text follow the usage in the main body: $\mathbf{d}_{\text{img}} = e_{o_{\text{goal}}} - e_{o_{\text{start}}}$ denotes the visual task direction and $\mathcal{P}_{\mathbf{d}_{\text{img}}}$ the projection operator defined in Eq. (4).

Assumption 1 (Monotone projected coordinate)**.** Define the scalar projected coordinate

$$c_t = \frac{\langle e_{o_t} - e_{o_{\text{start}}}, \mathbf{d}_{\text{img}} \rangle}{\|\mathbf{d}_{\text{img}}\|^2}.$$

We posit that within a sub-stage, c_t serves as a monotonic proxy for the latent task progress λ_t in expectation. Formally, there exists an increasing function h such that

$$\mathbb{E}[c_t \mid \lambda_t] = h(\lambda_t), \qquad h'(\cdot) > 0.$$

Remark. It is worth noting that for complex tasks, relying solely on a single terminal image and task description often fails to accurately characterize progress, potentially violating this monotonicity assumption (e.g., necessary intermediate states may appear visually or semantically distant from the final goal). To address this limitation and ensure the validity of Assumption 1, we decompose the global task into sequential sub-stages.

Theorem 1 (SNR gain of TDP)**.** *For any observation embedding e_{o_t}, the projection operator $\mathcal{P}_{\mathbf{d}_{img}}$ preserves the task-aligned component while linearly attenuating the orthogonal nuisance component by a factor of $(1-\alpha)$. Consequently, this mechanism enhances the Signal-to-Noise Ratio (SNR) of the embedding by a factor of $(1-\alpha)^{-2}$.*

Proof. Let the relative displacement be $\Delta e_t = e_{o_t} - e_{o_{\text{start}}}$. We decompose Δe_t into a component along $\mathbf{d}_{\text{img}}$ and an orthogonal residual noise ϵ_t:

$$\Delta e_t = c_t \mathbf{d}_{\text{img}} + \epsilon_t, \quad \text{where } c_t = \frac{\langle \Delta e_t, \mathbf{d}_{\text{img}} \rangle}{\|\mathbf{d}_{\text{img}}\|_2^2} \text{ and } \epsilon_t \perp \mathbf{d}_{\text{img}}.$$

Writing the projection operator as

$$\mathcal{P}_{\mathbf{d}}(\mathbf{x}) = (\alpha \Pi_{\mathbf{d}} + (1-\alpha)\mathbf{I})\,\mathbf{x}, \quad \text{where } \Pi_{\mathbf{d}} = \frac{\mathbf{d}\mathbf{d}^\top}{\|\mathbf{d}\|_2^2},$$

and using the properties $\Pi_{\mathbf{d}_{\text{img}}}(\mathbf{d}_{\text{img}}) = \mathbf{d}_{\text{img}}$ and $\Pi_{\mathbf{d}_{\text{img}}}(\epsilon_t) = 0$, we obtain

$$\begin{aligned} e'_{o_t} &= \mathcal{P}_{\mathbf{d}_{\text{img}}}(c_t \mathbf{d}_{\text{img}} + \epsilon_t) + e_{o_{\text{start}}} \\ &= \left[\alpha \Pi_{\mathbf{d}_{\text{img}}}(c_t \mathbf{d}_{\text{img}} + \epsilon_t) + (1-\alpha)(c_t \mathbf{d}_{\text{img}} + \epsilon_t)\right] + e_{o_{\text{start}}} \\ &= [\alpha c_t \mathbf{d}_{\text{img}} + (1-\alpha) c_t \mathbf{d}_{\text{img}} + (1-\alpha)\epsilon_t] + e_{o_{\text{start}}} \\ &= e_{o_{\text{start}}} + c_t \mathbf{d}_{\text{img}} + (1-\alpha)\epsilon_t. \end{aligned}$$

This result demonstrates that the signal component $c_t \mathbf{d}_{\text{img}}$ remains invariant, while the noise vector is scaled by $(1-\alpha)$. Defining the SNR as the ratio of signal energy to noise energy, the raw SNR is proportional to $1/\mathbb{E}[\|\epsilon_t\|^2]$. In the projected space, the noise energy becomes $\mathbb{E}[\|(1-\alpha)\epsilon_t\|^2] = (1-\alpha)^2 \mathbb{E}[\|\epsilon_t\|^2]$. Thus, the SNR gain is given by the ratio $\frac{1}{(1-\alpha)^2}$, which grows significantly as $\alpha \to 1$. □

Remark. It is well-established that CLIP embeddings are L_2-normalized during training, constraining them to a unit hypersphere $\mathbb{S}^{d-1} = \{e \in \mathbb{R}^d \mid \|e\|_2 = 1\}$. While the TDP operator $\mathcal{P}_{\mathbf{d}_{\text{img}}}$ is a linear map in $\mathbb{R}^d$ that may move the resulting embedding e'_{o_t} off the hypersphere ($\|e'_{o_t}\|_2 \neq 1$), the TDP mechanism remains geometrically consistent with the VLM's inductive bias for two reasons:

1. **Scale Invariance of Reward:** The cosine similarity $\cos(A, B) = \frac{\langle A, B \rangle}{\|A\|\|B\|}$ is a homogeneous function of degree zero. Formally, for any scalar $s > 0$, $\cos(sA, B) = \cos(A, B)$. Therefore, the norm-changing effect of the projection does not shift the reward value; only the directional alignment is preserved and denoised.

2. **Tangent Space Approximation:** In high-dimensional manifolds, for a localized sub-stage, the geodesic distance on the hypersphere is well-approximated by the Euclidean distance in the tangent space at $e_{o_{\text{start}}}$. TDP can be viewed as performing directional filtering in this local tangent space before the implicit re-normalization performed by the cosine similarity.

This ensures that TDP effectively filters noise in the embedding space while respecting the spherical topology of the VLM's representation.

Theorem 2 (Monotonicity of the TDP reward)**.** *While the raw cosine similarity is susceptible to non-monotonicity induced by fluctuations of the noise norm, under the condition that the visual task direction aligns sufficiently with the textual instruction in the embedding space, TDP ensures that the reward $\tilde{r}$ has a positive gradient with respect to the projected coordinate c_t, which serves as a monotonic proxy for the latent task progress λ_t (Assumption 1).*

Proof. We decompose the observation embedding into a deterministic signal and random noise:

$$e'_{o_t} = \underbrace{e_{o_{\text{start}}} + c_t \mathbf{d}_{\text{img}}}_{S(c_t)} + \underbrace{(1-\alpha)\epsilon_t}_{N_t},$$

where $S(c_t)$ is the signal vector evolving with the projected coordinate c_t, and N_t is the residual noise. In the raw setting ($\alpha = 0$), $N_t = \epsilon_t$; in the TDP setting ($\alpha \approx 1$), $N_t \to 0$. The reward function can be written as

$$\tilde{r}_t(c_t, N_t) = \cos(e'_{o_t}, e'_l) = \frac{\langle S(c_t) + N_t, e'_l \rangle}{\|S(c_t) + N_t\|\,\|e'_l\|}.$$

To rigorously analyze how noise disrupts the monotonicity of the reward signal, we consider the total change $d\tilde{r}_t$ with respect to time step t. When the agent makes infinitesimal task progress $dc_t > 0$, it is accompanied by a random perturbation of the noise vector dN_t:

$$d\tilde{r}_t = \underbrace{\frac{\partial \tilde{r}_t}{\partial c_t} dc_t}_{\text{Signal Gain}} + \underbrace{\langle \nabla_{N_t} \tilde{r}_t, dN_t \rangle}_{\text{Noise Interference}},$$

where

$$\begin{aligned}
\nabla_{N_t} \tilde{r}_t = \nabla_{e'_{o_t}} \tilde{r}_t &= \frac{\|e'_{o_t}\| \|e'_l\| \cdot e'_l - \langle e'_{o_t}, e'_l \rangle \cdot \|e'_l\| \frac{e'_{o_t}}{\|e'_{o_t}\|}}{\|e'_{o_t}\|^2 \|e'_l\|^2} \\
&= \frac{1}{\|e'_{o_t}\|} \left(\frac{e'_l}{\|e'_l\|} - \tilde{r}_t \frac{e'_{o_t}}{\|e'_{o_t}\|} \right) \\
&= \frac{1}{\|e'_{o_t}\|} (\hat{e}_l - \tilde{r}_t \hat{e}_{o_t}).
\end{aligned}$$

Consider a common worst-case scenario: noise magnitude expansion. Assume that at a certain moment the direction of the noise change aligns with the noise vector itself, i.e., dN_t is parallel to N_t (e.g., background clutter increases), denoted as $dN_t = \gamma \hat{N}_t$ with $\gamma > 0$. In this case, the noise interference term becomes

$$\text{Noise Interference} = \langle \nabla_{N_t} \tilde{r}_t, dN_t \rangle = \frac{\gamma}{\|e'_{o_t}\|} \left(\langle \hat{e}_l, \hat{N}_t \rangle - \tilde{r}_t \langle \hat{e}_{o_t}, \hat{N}_t \rangle \right).$$

In high-dimensional semantic spaces, the random noise vector N_t is likely orthogonal to the fixed task goal e'_l, so $\langle \hat{e}_l, \hat{N}_t \rangle \approx 0$. For the second term $\langle \hat{e}_{o_t}, \hat{N}_t \rangle$, expanding $\hat{e}_{o_t} = \frac{S(c_t) + N_t}{\|e'_{o_t}\|}$, we get

$$\langle \hat{e}_{o_t}, \hat{N}_t \rangle = \frac{\langle S(c_t), N_t \rangle + \|N_t\|^2}{\|e'_{o_t}\| \|N_t\|}.$$

Since the signal is orthogonal to the noise ($\langle S(c_t), N_t \rangle \approx 0$), this term approximates to $\frac{\|N_t\|}{\|e'_{o_t}\|}$, which is strictly positive. Hence

$$\text{Noise Interference} \approx \frac{\gamma}{\|e'_{o_t}\|} \left(0 - \tilde{r}_t \frac{\|N_t\|}{\|e'_{o_t}\|} \right) = -\gamma \tilde{r}_t \frac{\|N_t\|}{\|e'_{o_t}\|^2}.$$

When the noise magnitude increases ($\gamma > 0$) and is sufficiently large, the negative noise interference term dominates the positive signal gain term:

$$\left| -\gamma \tilde{r}_t \frac{\|N_t\|}{\|e'_{o_t}\|^2} \right| > \left| \frac{\partial \tilde{r}_t}{\partial c_t} dc_t \right| \implies d\tilde{r}_t < 0.$$

This derivation shows that even if the agent approaches the goal in the high-dimensional semantic space ($dc_t > 0$), the raw reward signal can decrease. This is due to the reward function's sensitivity to the full noise vector N_t in the denominator, where the negative gradient component (stemming from the $-\tilde{r}_t \hat{e}_{o_t}$ term) overwhelms the marginal signal improvement. This explains the severe oscillations and lack of monotonicity observed in raw VLM rewards during experiments.

In the TDP setting ($\alpha \to 1$), we have $\|N_t\| \to 0$ and $\mathrm{d}\|N_t\| \to 0$. Consequently, the reward function degenerates to a deterministic form:

$$\lim_{\alpha \to 1} \tilde{r}_t = \frac{\langle S(c_t), e'_l \rangle}{\|S(c_t)\| \|e'_l\|} = \frac{\langle e_{o_{\text{start}}} + c_t \mathbf{d}_{\text{img}}, e'_l \rangle}{\|e_{o_{\text{start}}} + c_t \mathbf{d}_{\text{img}}\| \, \|e'_l\|}.$$

Note that $\frac{\partial S(c_t)}{\partial c_t} = \mathbf{d}_{\text{img}}$; hence

$$\begin{aligned}
\frac{\partial \tilde{r}}{\partial c_t} &= \frac{1}{\|e'_l\|} \frac{\partial}{\partial c_t} \left(\langle S(c_t), e'_l \rangle \|S(c_t)\|^{-1} \right) \\
&= \frac{1}{\|e'_l\|} \left(\langle \mathbf{d}_{\text{img}}, e'_l \rangle \|S(c_t)\|^{-1} - \langle S(c_t), e'_l \rangle \|S(c_t)\|^{-2} \frac{\partial \|S(c_t)\|}{\partial c_t} \right).
\end{aligned}$$

The derivative of the norm is $\frac{\partial \|S(c_t)\|}{\partial c_t} = \frac{\langle S(c_t), \mathbf{d}_{\text{img}} \rangle}{\|S(c_t)\|}$. Substituting this back:

$$\frac{\partial \tilde{r}}{\partial c_t} = \frac{1}{\|e_l'\| \|S(c_t)\|^3} \left[\langle \mathbf{d}_{\text{img}}, e_l' \rangle \|S(c_t)\|^2 - \langle S(c_t), e_l' \rangle \langle S(c_t), \mathbf{d}_{\text{img}} \rangle \right].$$

Normalizing the terms by $\|\mathbf{d}_{\text{img}}\| \|e_l'\| \|S(c_t)\|^2$, we rewrite the positivity condition in its angular form:

$$\frac{\langle \mathbf{d}_{\text{img}}, e_l' \rangle}{\|\mathbf{d}_{\text{img}}\| \|e_l'\|} > \frac{\langle S(c_t), e_l' \rangle}{\|S(c_t)\| \|e_l'\|} \cdot \frac{\langle S(c_t), \mathbf{d}_{\text{img}} \rangle}{\|S(c_t)\| \|\mathbf{d}_{\text{img}}\|}$$
$$\implies \quad \cos(\mathbf{d}_{\text{img}}, e_l') > \cos(S(c_t), e_l') \cdot \cos(S(c_t), \mathbf{d}_{\text{img}}).$$

This condition requires that the visual task direction $\mathbf{d}_{\text{img}}$ aligns with the textual target e_l'. Since $e_l' = \mathcal{P}_{\mathbf{d}_{\text{text}}}(e_l)$ is projected onto the task direction, while $S(c_t) = e_{o_{\text{start}}} + c_t \mathbf{d}_{\text{img}}$ denotes the ideal observation embedding obtained by shifting $e_{o_{\text{start}}}$ along the task direction, $S(c_t)$ is in general not collinear with $\mathbf{d}_{\text{img}}$. For a well-aligned VLM, the left-hand side therefore dominates the right-hand-side product of intermediate alignments, and the positive-gradient property is maintained. Coupled with Assumption 1, this restores the critical condition of monotonicity between the reward $\tilde{r}$ and the latent task progress λ_t, effectively isolating the trend from background noise and theoretically validating the consistency of TDP rewards. The empirical validation in Appendix E.1 further corroborates Assumption 1 and the implications of Theorem 2. □

Note that while our theoretical analysis assumes the orthogonal component ϵ_t is pure noise, in practice the true latent manifold may exhibit curvature. Therefore, retaining a small portion of the raw embedding (i.e., $\alpha < 1$) helps preserve local geometric structures that are not captured by the global linear projection. Relevant empirical studies are provided in Appendix E.4.

E Additional Results

E.1 Empirical Validation of Task Direction Projection

Figure 17 compares the raw CLIP reward against the TDP-processed reward on oracle trajectories. The raw reward (orange) exhibits significant volatility and non-monotonicity, often decreasing as the agent approaches the goal. In contrast, the TDP reward (blue) demonstrates a monotonic progression aligned with the ground-truth task execution, empirically validating the noise suppression analysis in Theorems 1 and 2.

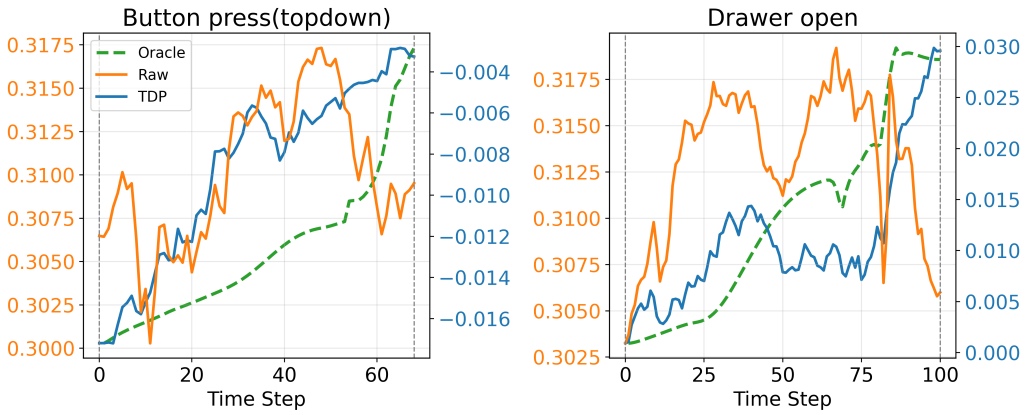

Figure 17: **Comparison of Reward Signal Quality on Oracle Trajectories.**

E.2 Robustness of Scene Encoder to Viewpoint Shifts

We evaluate reward consistency by randomly switching camera viewpoints every 5 steps along an oracle trajectory. As shown in Figure 18, the raw VLM reward fluctuates drastically with viewpoint

changes. However, by effectively filtering out irrelevant view information, the reward curve derived from our fine-tuned Scene Encoder remains consistent despite these abrupt visual transitions, recovers signals that faithfully reflects the true task progress. This invariance is particularly critical for real-world robotic deployment, where reward systems must remain robust to unexpected camera jitter or calibration errors.

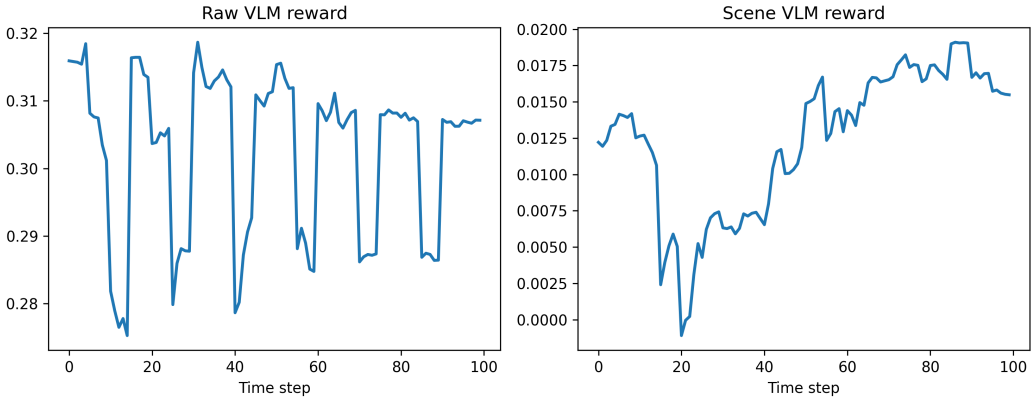

Figure 18: **Robustness to Camera Viewpoint Shifts**

E.3 Analysis of Confidence-Thresholded Shaping

In addition to the confidence-thresholded shaping (CTS) mechanism used throughout the main experiments, we investigate two alternative strategies for suppressing noise in VLM-based rewards: exponential moving average (EMA) smoothing and Kalman filtering. Both methods aim to stabilize reward signals by attenuating high-frequency fluctuations, but differ in their inductive biases and responsiveness to sudden semantic alignment.

As shown in Figure 19, all three filtering strategies effectively suppress spurious reward activations and ultimately enable successful learning. EMA filtering improves stability by smoothing short-term fluctuations, but this temporal averaging can delay the reward's response to abrupt progress signals in certain tasks. Kalman filtering enforces stronger temporal coherence, yet exhibits higher sensitivity to its modeling assumptions and hyperparameters, leading to less consistent convergence behavior across tasks. In contrast, CTS applies a simple, distribution-aware gating mechanism that preserves sharp reward transitions once semantic confidence surpasses the noise floor, yielding more uniform convergence dynamics. Given its favorable balance between robustness, responsiveness, and implementation simplicity, we adopt CTS as the default choice in MARVL.

E.4 Effect of the Mixing Parameter α in TDP

We study the effect of the mixing parameter α in Task Direction Projection (TDP), which controls the interpolation between the raw visual embedding and the task-aligned projected component. Figure 20 reports the success rate on the Drawer Open task for different values of α. Overall, intermediate values of α yield more reliable learning dynamics, while extreme settings tend to degrade stability. In particular, $\alpha = 0$ (no projection) results in slower convergence and reduced final performance, indicating that unstructured similarity signals are insufficient for consistent progress estimation. Conversely, approaching $\alpha = 1$ risks oversimplifying the state representation. While this maximizes noise suppression, it simultaneously eliminates fine-grained state information present in the orthogonal components. These results suggest that a moderate balance between raw observations and task-aligned projection is important for stable training, and we therefore fix α to a constant intermediate value ($\alpha = 0.8$) in all experiments.

F Hyperparameters

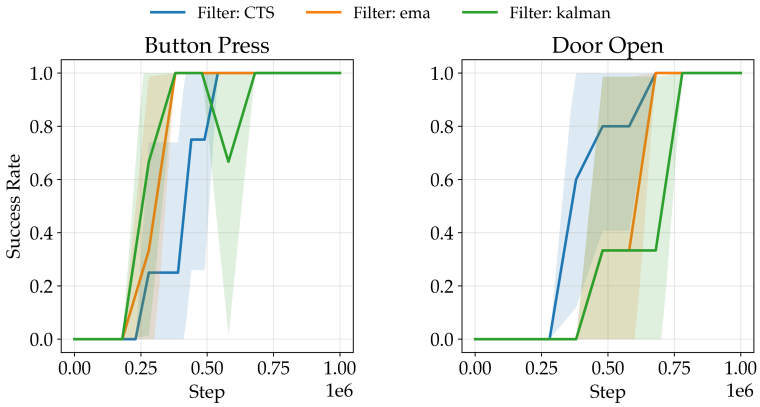

Figure 19: **Comparison of Different Confidence-Based Reward Filtering Strategies.**

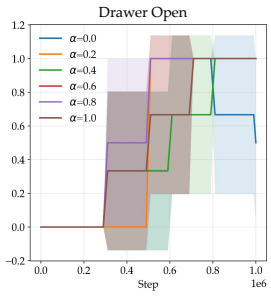

Figure 20: **Effect of the Mixing Parameter α in TDP.**

G HYPERPARAMETERS

Table 1: Hyperparameters for MARVL and baselines

Parameter	Value
Common hyperparameters (SAC)	
VLM reward weight ρ	0.05
Adam learning rate	3×10^{-4}
Batch size	256
Target network τ	0.005
Discount factor γ	0.99
Hidden layer dimensions	(256, 256)
Initial random exploration steps	10^4
Actor target entropy	$-\dim(\mathcal{A})/2$
Actor/Critic min std scale	1×10^{-3}
Action repeat	10
MARVL	
TDP projection blend α	0.8
Quantile threshold m	0.97
Logistic steepness parameter κ	100
Stage transition similarity threshold	0.997
Stage transition patience	4
FuRL	
FuRL cosine margin	0.25
FuRL L2 distance margin	0.25
FuRL embedding buffer size	2×10^4
FuRL window gap k	10
FuRL reward projection network	(256, 64)
TD3	
TD3 policy noise	0.2
TD3 noise clip	0.5
TD3 policy update frequency	2
TD3 exploration noise	0.1
Relay	
Relay success threshold	2500
Relay exploration noise	0.2
LIV	
LIV learning rate	1×10^{-5}
LIV weight decay	1×10^{-3}
LIV discount factor	0.98

Table 2: Fine-tuning hyperparameters

Parameter	Value
Architecture	
Image size	224
Patch size	14
ViT model	ViT-B/32
Pre-trained weights	laion2b_s34b_b79k
Encoder output dimension	512
Decoder input dimension	1024
Conv channels	[256, 128, 64, 32]
Conv kernel size	3
Upsampling mode	bilinear
Dropout rate	0.1
Training	
Batch size	32
Learning rate	1×10^{-4}
Total epochs	90
Stage-1 / Stage-2 / Stage-3 epochs	30 / 30 / 30
Optimizer	
Weight decay	0.01
Label smoothing	0.1
Gradient accumulation steps	2
Max gradient norm	1.0
Scheduler type	cosine
Warmup epochs	5
Minimum learning rate	1×10^{-6}
Loss weight configuration	
LPIPS weight λ_{lpips}	0.05
Shuffle coefficient λ_{shuffle}	1.0
Scene consistency coefficient	1.0
View consistency coefficient	0.25
Scene contrastive coefficient	1.0
View contrastive coefficient	0.25
CLIP coefficient λ_{clip}	[0, 0.5, 1]
Temperature τ	0.07