Closing The Performance Gap Between Zero-shot And Post-trained Reward Models

Marius Memmel[1*] Ankit Goyal[2] Dieter Fox[1] Abhishek Gupta[1] Anqi Li[2] Fabio Ramos[2,3]

***Abstract*—Accurately estimating task progress and deriving robust reward functions from raw video are critical for advancing reinforcement learning (RL) and robotics. While recent Reward Foundation Models (RFMs) have shown promise by fine-tuning Vision-Language Models (VLMs) on robotic datasets, leveraging existing zero-shot VLMs for this task remains difficult due to a significant lack of calibration and a tendency for temporal hallucinations. In this work, we propose SCORE, a novel prompting framework that transforms progress prediction from a black-box logit extraction task into an explicit reasoning-in-language process.**

SCORE decomposes the problem into two stages: (1) grounded video description, which ensures the model focuses on task-relevant physical interactions, and (2) semantic progress reasoning, where the VLM jointly predicts a textual completion anchor and a calibrated numerical progress sequence. Our approach effectively closes the performance gap between zero-shot methods and state-of-the-art post-trained RFMs. In offline benchmarks, SCORE outperforms existing baselines in trajectory ranking and cross-task calibration. Furthermore, we demonstrate the real-world utility of SCORE by using it as a reward signal for Diffusion Steering RL (DSRL); our method enables a VLA policy to overcome strong initial biases, achieving a +90% success rate improvement over vanilla policies. Finally, we provide an empirical scaling analysis showing that progress prediction capabilities improve significantly with each new generation of frontier VLMs, positioning SCORE as a scalable, high-performance solution for zero-shot reward modeling.

***Index Terms*—component, formatting, style, styling, insert.**

I. Introduction

The ability to accurately predict task progress and learn robust reward functions is a fundamental challenge in modern reinforcement learning (RL) and robotics. Such capabilities are essential across a wide range of paradigms: in online RL, they alleviate the sample inefficiency caused by sparse rewards; in offline RL and filtered behavioral cloning (BC), they enable the selection of high-quality transitions from suboptimal data; and in model-based settings, they provide the necessary signal for world model imagination. While hand-defining dense reward functions is notoriously difficult and task-specific, it is especially challenging in unstructured real-world scenarios where these functions must be derived directly from raw video streams and high-level language instructions. To address this complexity, recent advancements have turned toward Reward Foundation Models (RFMs), which leverage the vast knowledge of Vision-Language Models (VLMs) by post-training them to predict temporal progress [1, 2, 3].

[1]University of Washington, [2]NVIDIA, [3]University of Sydney, *Work done during an internship at NVIDIA

Despite the success of post-trained RFMs, leveraging existing zero-shot VLMs for general progress prediction remains a significant challenge. We identify a critical *lack of calibration* as the primary bottleneck. For instance, methods like Generative Video Learning (GVL) [4] are designed to predict the ordering of shuffled image sequences. Because GVL maps any sequence to a normalized range of 0 to 1 regardless of the actual task outcome or optimality, it fails to distinguish between successful trajectories and failures. Similarly, while models like TOPReward [5] attempt to use the logits of a "True" token to estimate progress, we find these signals to be uncalibrated across different tasks due to varying logit magnitudes. Furthermore, these methods are prone to "hallucination" – often predicting high progress simply because task-relevant objects are present, even if the robot fails to manipulate them correctly.

In this work, we propose SCORE, a novel approach centered on the key insight that the explicit reasoning capabilities of VLMs can be leveraged to calibrate progress. Rather than treating the VLM as a black-box logit generator, we decompose the progress prediction process into two distinct stages: (1) **grounded video description** and (2) **reasoning about progress in language space**. By decoupling video understanding from progress evaluation, we significantly reduce hallucinations. The first stage ensures the model acknowledges the actual state of the environment, while the second stage allows the model to ground its progress estimate in absolute task success criteria, leading to superior calibration.

Our contributions are as follows:

- We present Semantic Calibration Of REwards (SCORE), a simple yet highly effective prompting scheme for VLMs that enables impressive zero-shot progress prediction capabilities, effectively closing the performance gap between explicitly post-trained RFMs and zero-shot methods.
- We demonstrate that SCORE is competitive with, and in several cases outperforms, state-of-the-art post-trained RBMs on standard offline benchmarks.
- We validate SCORE in real-world robotics experiments using Diffusion Steering RL (DSRL) [6] with a VLA policy (π_0 [7]). Our approach improves success rates by $+90\%$ over vanilla π_0 and $+95\%$ compared to steering with TOPReward, highlighting its utility as a robust real-world reward signal.
- We provide an empirical analysis of the trend in progress prediction capabilities, observing that these capabilities

scale significantly with newer generations of large-scale VLMs.

II. Methodology

A. Problem Formulation

Given a video sequence $V = \{v_1, v_2, \ldots, v_T\}$ of length T and a natural language instruction L describing a specific task, the goal of progress estimation is to predict a corresponding sequence of progress values $P = \{p_1, p_2, \ldots, p_T\}$. Each scalar value $p_t \in [0, 1]$ represents the estimated task completion progress at timestep t. The overarching objective is to learn a mapping function capable of accurately deriving P from the multi-modal inputs V and L. Depending on the chosen architecture, this mapping can be defined as a function $P = f_\theta(V, L)$ and is commonly modeled through a VLM. Prior works [1, 2, 5] model $p_t = f_\theta(\{v_0, \ldots, v_t\}, L)$ and compute P by iteratively querying f_θ. While this is feasible for open source models where we have access to computing progress in batches, ideally, we'd want to infer the entire progress sequence P directly.

To address these limitations, we propose to decompose f_θ into 1) grounded video description and 2) reasoning about progress in language space.

B. Grounded Video Description

To ensure the model focuses on relevant information in the video, we first ground the video in a discrete set of objects. Given a language instruction L and a video sequence $V = \{v_1, v_2, \ldots, v_T\}$, the VLM identifies a set of task-relevant objects O:

$$O = f_\theta(V, L) \tag{1}$$

This set O represents the intersection of entities mentioned in L and those physically present in V. Conditioned on these objects, the model generates a sequence of natural language descriptions $D = \{d_1, d_2, \ldots, d_T\}$. Each d_t describes the state of the objects in O for the corresponding frame v_t:

$$d_t = f_\theta(v_t, O, L) \tag{2}$$

By bottlenecking the perception through O, we ensure the descriptions capture the physical state changes of the environment necessary for progress estimation.

C. Progress Reasoning in Language Space

We treat progress estimation as a purely linguistic reasoning task to mitigate the hallucinations common in end-to-end multi-modal regression. Rather than predicting progress frame-by-frame, the model processes the entire sequence of descriptions D to maintain temporal consistency. Specifically, the model jointly predicts a "completion state" s^*—a textual anchor defining 100% task success—and the full progress sequence $P = \{p_1, p_2, \ldots, p_T\}$:

$$(P, s^*) = f_\theta(D, L) \tag{3}$$

By predicting P and s^* together, the model enforces a global semantic calibration, evaluating each description d_t against the instruction L and the intended goal state s^* to assign consistent numerical values.

In practice, we use a single VLM f_θ for the entire pipeline. The transitions between object extraction, frame description, and joint progress reasoning are managed through natural language prompting, steering the model to output structured JSON strings for each stage of the inference flow.

III. Experimental Evaluation

Our evaluation aims to address the following questions: (Q1) How does SCORE compare quantitatively to post-trained and zero-shot methods? (Q2) How does SCORE compare to other zero-shot methods in enabling downstream real-world reinforcement learning? (Q3) How does SCORE's performance scale with the underlying VLM?

We evaluate SCORE against the strongest performing approaches, spanning both explicitly post-trained and zero-shot reward modeling to provide a comprehensive analysis of the current landscape.

post-trained Baselines are explicitly trained on robotics datasets to predict progress.

- **RoboReward-4B** [2] fine-tunes a Qwen-3-VL [8] model to predict discrete end-of-episode progress scores ranging from 1 to 5. It is trained on large-scale robotic datasets [9, 10]. To address the lack of failure examples in the original datasets, it augments the data by generating counterfactual language instructions via zero-shot VLMs to simulate failed or mismatched trajectories.
- **Robometer** [1] scales reward modeling by fine-tuning a Qwen-3-VL backbone to predict continuous progress values in $[0, 1]$. It is trained on RBM-1M, a curated dataset of over one million trajectories that explicitly includes substantial suboptimal and failure data. The model utilizes a multi-objective training framework to simultaneously predict absolute progress, binary success, and relative preference comparisons between trajectories.

Zero-shot Baselines leverage the pre-existing world knowledge of off-the-shelf VLMs without any task-specific or robotics-specific fine-tuning.

- **GVL** [4] prompts a zero-shot VLM with temporally shuffled frames from a video sequence to predict task progress for the subsampled frames. This shuffling mechanism is used to mitigate the VLM's inherent bias toward time continuity, forcing the model to evaluate the intrinsic progress of each frame independently based on its content rather than its chronological order.
- **TOPReward** [5] prompts a zero-shot VLM with a full video sequence alongside a language instruction. Rather than relying on direct text outputs for numerical progress, which are prone to numerical misrepresentation, it asks the model if the instruction matches the video execution and extracts task progress directly from the internal token logits of the "True" token.

	Pretrained Baselines		Zero-shot Baselines		SCORE		
RBM-EVAL-OOD	**RoboReward-4B**	**Robometer-4B**	**GVL**	**TOPReward**	**Gemini 3 Flash**	**Gemini Robotics-ER 1.6**	**GPT-5.4**
(a) Kendall τ (avg) ↑	0.536	0.427	0.217	0.149	0.560	**0.594**	<u>0.542</u>
(b) Kendall τ (last) ↑	0.536	**0.648**	0.273	0.202	0.553	<u>0.592</u>	0.581

TABLE I: **Evaluation on the `RBM-EVAL-OOD` benchmark [1]:** Trajectory ranking comparison across Baselines and models. SCORE achieves competitive performance (Gemini 3 Flash) and even outperforms post-trained baselines (Gemini Robotics-ER 1.6, GPT-5.4). Best values in each row are highlighted in **bold**, runner-ups are <u>underlined</u>.

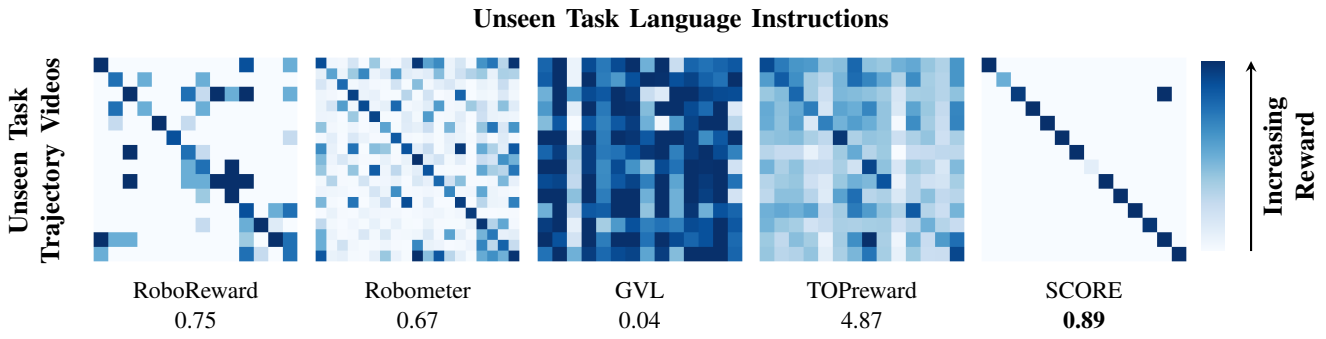

Fig. 1: **Video-Language Reward Confusion Matrix.** To evaluate performance, we calculated rewards for every possible pairing of demonstration videos and language descriptions from the `RBM-EVAL-OOD` benchmark. SCORE generates a distinctly diagonal-heavy confusion matrix, demonstrating superior alignment between new demonstrations and their corresponding instructions. Additionally, we report the column-normalized diagonal mean, which quantifies the proportion of the model's total reward assigned to correctly matched pairs.

A. (Q1) How does SCORE compare quantitatively to post-trained and zero-shot methods?

We report offline results on the `RBM-EVAL-OOD` benchmark [1], as this dataset remains unseen by all tested methods and utilizes more descriptive language instructions compared to its `RBM-EVAL-ID` counterpart. While prior work [1] typically reports the Kendall τ_a based on the final progress value of a trajectory (final), we argue that reporting the average τ_a over the entire trajectory (avg) provides a more robust metric. This approach better captures the robot's performance throughout the execution rather than relying on the final state, which can be inherently noisy.

The results in Tab. I demonstrate that SCORE effectively closes the performance gap between zero-shot and post-trained methods, outperforming existing baselines such as RoboReward and Robometer when paired with the most recent VLMs (Gemini Robotics-ER 1.6 or GPT-5.4). Notably, SCORE succeeds where other zero-shot approaches fail due to architectural constraints: GVL lacks the ability to perform cross-trajectory comparisons by design, and TOPReward suffers from a lack of absolute calibration, rendering it unable to provide consistent rewards across different trajectories.

The primary advantage of SCORE lies in its explicit reasoning within the language space using a VLM, which leverages the model's inherent semantic understanding. This is evidenced by the model's ability to consistently assign zero progress to mismatched task-instruction pairs while accurately predicting progress for correct matches shown in Fig. 1. While post-trained models achieve a solid distinction between aligned and misaligned video-language pairs, the zero-shot methods cannot distinguish between them.

Minor discrepancies in the SCORE results are typically attributed to two factors: (1) rare VLM hallucinations regarding task descriptions not present in the video (dark outlier in the top right), and (2) specific tasks like "fold the towel," where the initial state is already partially folded, leading the VLM to occasionally interpret further folding as a reversal of progress (bright spots on the diagonal).

B. (Q2) How does SCORE compare to other zero-shot methods in enabling downstream real-world reinforcement learning?

To demonstrate the utility of the progress signals generated by SCORE as downstream reward functions, we conducted real-world experiments fine-tuning a VLA base policy π_0 using the Diffusion Steering Reinforcement Learning (DSRL) algorithm. The task involves "putting the fruit in the bowl." Initially, π_0 exhibits a strong bias, attempting to place an ice cream cone into the bowl 90% of the time. The objective of the DSRL fine-tuning is to redirect the policy to interact with a banana instead. Results are shown in Fig. 3.

SCORE provides a highly discriminative reward signal, assigning zero reward to trajectories involving the ice cream cone while providing a dense, informative reward for the successful manipulation of the banana. This leads to a 100% success rate

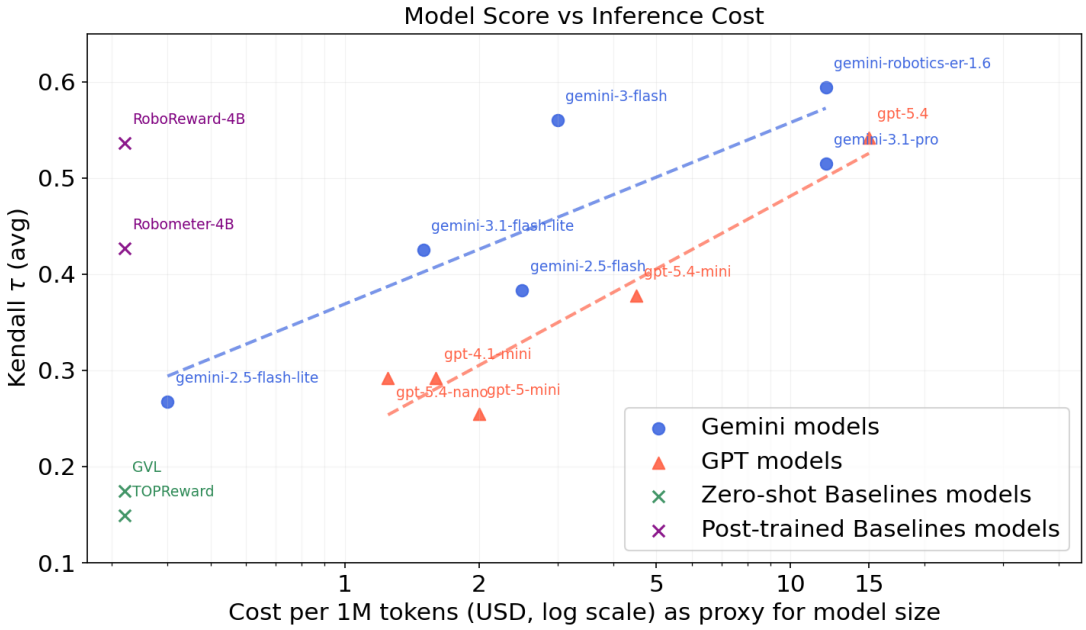

Fig. 2: **Scaling of progress prediction performance with different base models.** SCORE scales with the underlying base model, outperforming zero-shot and post-trained reward models.

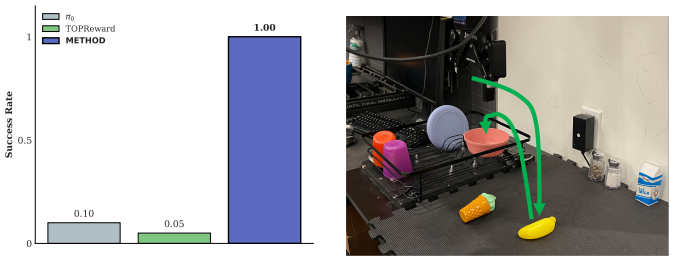

Fig. 3: Results for real-world DSRL experiments. While baseline methods struggle with the precision required for the task, SCORE achieves a perfect success rate (20/20).

post-tuning. In contrast, TOPReward assigns non-zero rewards to both successful and failure trajectories. Although interactions with the banana receive slightly higher rewards, the algorithm tends to latch onto the more frequent ice cream cone interactions, eventually reducing banana interaction success to only 10%. To maintain high-speed inference in these real-world settings, we utilize object caching to minimize VLM latency. Using Gemini-3-flash with low reasoning effort and object caching reduces VLM latency, allowing a single video to be annotated in 10–20 seconds. By running this process asynchronously, video annotation imposes no delay on RL training.

C. (Q3) How does SCORE's performance scale with the underlying VLM?

Finally, we investigate whether the ability to predict progress from video is an emergent property of modern VLMs and how it scales with model capability. In Fig. 2, we plot trajectory ranking performance as the Kendall τ_a (avg) against the cost per million tokens, using cost as a proxy for overall model capability.

Across both the GPT and Gemini model families, we observe a clear upward trend: as the underlying model capability increases, so does the trajectory ranking accuracy. These results suggest that progress and reward modeling are likely to improve further with future generations of VLMs. Consequently, SCORE serves not only as a high-performing current solution but as a robust baseline that will naturally scale with the evolution of foundational vision-language models.

REFERENCES

[1] A. Liang, Y. Korkmaz, J. Zhang, M. Hwang, A. Anwar, S. Kaushik, A. Shah, A. S. Huang, L. Zettlemoyer, D. Fox *et al.*, "Robometer: Scaling general-purpose robotic reward models via trajectory comparisons," *arXiv preprint arXiv:2603.02115*, 2026.

[2] T. Lee, A. Wagenmaker, K. Pertsch, P. Liang, S. Levine, and C. Finn, "Roboreward: General-purpose vision-language reward models for robotics," *arXiv preprint arXiv:2601.00675*, 2026.

[3] H. Tan, S. Chen, Y. Xu, Z. Wang, Y. Ji, C. Chi, Y. Lyu, Z. Zhao, X. Chen, P. Co *et al.*, "Robo-dopamine: General process reward modeling for high-precision robotic manipulation," *arXiv preprint arXiv:2512.23703*, 2025.

[4] Y. J. Ma, J. Hejna, C. Fu, D. Shah, J. Liang, Z. Xu, S. Kirmani, P. Xu, D. Driess, T. Xiao *et al.*, "Vision language models are in-context value learners," in *The Thirteenth International Conference on Learning Representations*, 2024.

[5] S. Chen, C. Harrison, Y.-C. Lee, A. J. Yang, Z. Ren, L. J. Ratliff, J. Duan, D. Fox, and R. Krishna, "Topreward: Token probabilities as hidden zero-shot rewards for robotics," *arXiv preprint arXiv:2602.19313*, 2026.

[6] A. Wagenmaker, M. Nakamoto, Y. Zhang, S. Park, W. Yagoub, A. Nagabandi, A. Gupta, and S. Levine, "Steering your diffusion policy with latent space reinforcement learning," *arXiv preprint arXiv:2506.15799*, 2025.

[7] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "

$$\backslash$$

pi_0 : $Avision - language - actionflowmodelforgeneralrobotcontrol,''$ *arXiv preprint arXiv:2410.24164*, 2024.

[8] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge *et al.*, "Qwen3-vl technical report," *arXiv preprint arXiv:2511.21631*, 2025.

[9] O.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, vol. 1, no. 2, 2023.

[10] P. Atreya, K. Pertsch, T. Lee, M. J. Kim, A. Jain, A. Kuramshin, C. Eppner, C. Neary, E. Hu, F. Ramos *et al.*, "Roboarena: Distributed real-world evaluation of generalist robot policies," *arXiv preprint arXiv:2506.18123*, 2025.