VIRAL: Vision-grounded Integration for Reward design And Learning

Valentin Cuzin-Rambaud, Emilien Komlenovic, Alexandre Faure, Bruno Yun
Universite Claude Bernard Lyon 1, CNRS, INSA Lyon, LIRIS, UMR5205, 69622 Villeurbanne, France

Challenge: The **alignment between humans and machines**; in reinforcement learning (RL), the reward function is particularly vulnerable to the **risks associated with poorly designed**.

Contribution

- **The VIRAL pipeline** to automatically design reward functions using a simple natural language prompt and/or annotated image.
- **A self-refinement process** for reward functions, augmented with human feedback or video description from an LVLM.
- **An empirical study** showing its various benefits for learning and user intent alignment.

Introduction

Recent advances [1] have leveraged OpenAI's GPT-4 model to generate code for reward functions, demonstrating competitive performance and improved adaptability across different environments.

We propose the VIRAL framework to design reward functions based on simple user prompts and/or annotated images..

VIRAL prioritizes the use of **open-source**, **efficient**, and **lightweight LLMs**.

VIRAL integrates **Large Vision Language Models (LVLMs)** [2] to process both text/images, enhancing its ability to interpret user intent.

VIRAL is the first reward shaping approach to incorporate **Video-LVLMs** [3] for describing the movements of objects within the scene, providing richer context for reward function generation.

Architecture

As **input**, we have (a_1) a textual environment description d_{env}, (a_2) an optional implementation of a success function f_{succ} provided as Python code, and (b) the goal prompt p_g (which can be multi-modal: text prompt, an annotated image img, or both).

The VIRAL pipeline

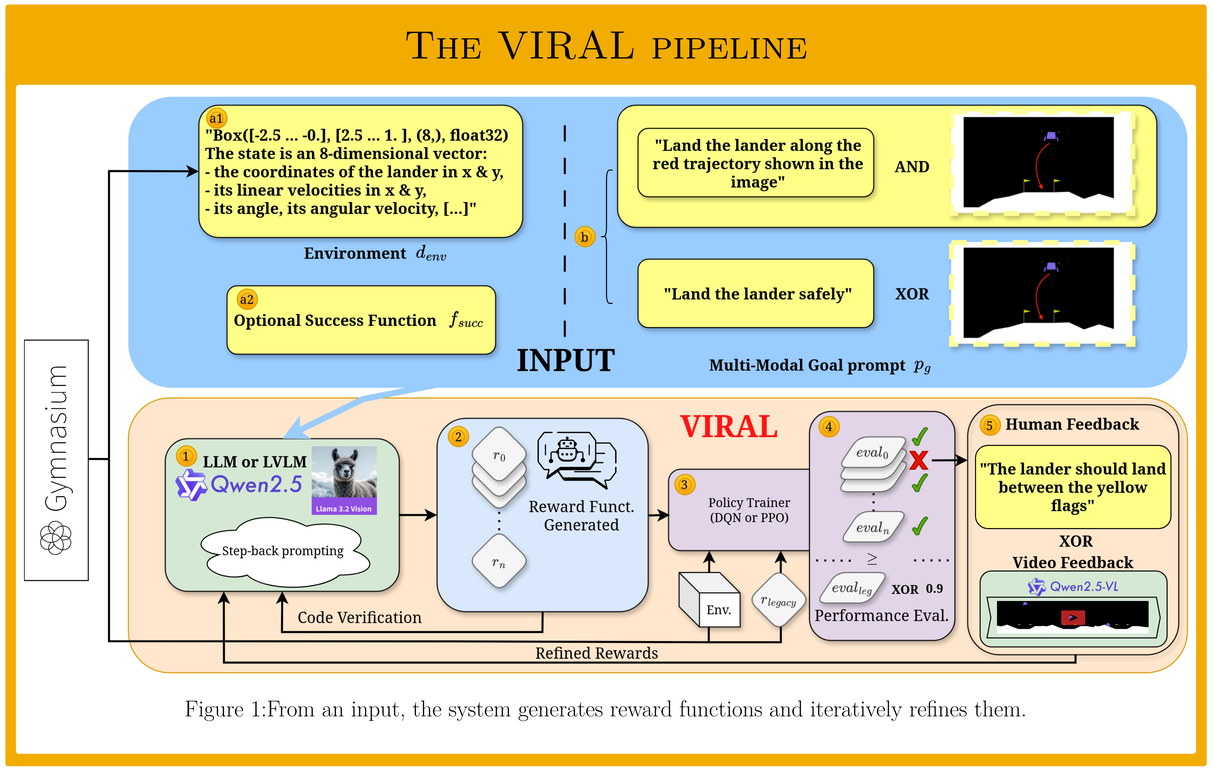

Figure 1:From an input, the system generates reward functions and iteratively refines them.

1. **Step-back prompting** creates a first reasoning about a problem at a higher level before generating a detailed response
2. We use **collaboration** between two LLMs (critic and coder), the critic generates the step-back prompt, which is given to the coder LLM
3. **An RL algorithm** is used to maximize the expected return of the reward function generated.
4. During **testing**, VIRAL compares the success rate of the policy against a user-defined threshold. If it is not acceptable, it goes on the refinement loop.
5. The Feedback can be made by a human or a Video-LLM, which **analyzes an episode of the trained policy**. It helps the refinement process.

Results

We conducted a study (see Fig. 2) with 25 annotators to determine whether **the learned behavior aligns with the goal prompt provided (text, image, or both)** by annotating videos (10 distinct videos per modality/environment).

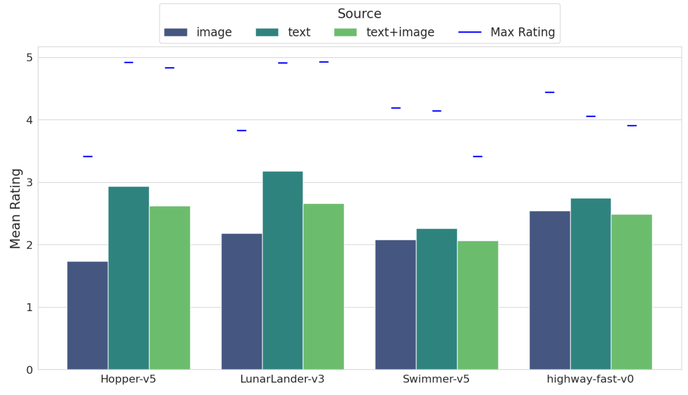

Figure 2:Semantic alignment Evaluation

The relative performance of each modality varied with the environment, which implies that the usefulness of multimodal or visual instructions may depend heavily on the task's nature.

Since **the main goal is to discover the optimal reward function across multiple generations**, in Table 1 we report the maximum of average rating on 10 distinct videos per modality and environment.

Environment	Image p_g	Textual p_g	Image+Textual p_g
Hopper-v5	3.41±1.00	**4.92±0.28**	4.83±0.39
LunarLander-v3	3.83±1.26	4.92±0.28	**4.93±0.26**
Swimmer-v5	**4.19±0.75**	4.14±0.95	3.41±1.33
Highway-fast-v0	**4.44±1.29**	4.06±1.03	3.91±1.38

Table 1:Maximum selection of average rating per video.

Image or text prompts win variably by environment: textual best for Hopper, image+text for LunarLander, image-only for Swimmer and Highway.

No single modality dominates overall.

Conclusion

Our results showed that agents learned new behaviors from simple sketches, showing our approach's robustness. Integrating Video-LVLM or users' feedback improved alignment with intended objectives.

GitHub Link to code

Link to video

References

[1] Yecheng Jason Ma, William Liang, and al.
Eureka: Human-level reward design via coding large language models, 2023.

[2] Meta.
Llama-3.2-11b-vision-instruct, 2024.

[3] Shuai Bai, Keqin Chen, and al.
Qwen2.5-vl technical report, 2025.

Contact Information

- Web: https://viral-ucbl1.github.io/
- Email: valentin.cuzin-rambaud@univ-lyon1.fr