DREFUN-V: LLM for Design REward FUNction with Vision as grounding

I Pre-proposal context, positioning and objective(s)

a Context

This innovative project is a part of Bruno Yun's LLM course. Thanks to the effectiveness of LLMs and their widespread use, more and more research is combined with LLM. The RL (reinforcement-learning) domain is one of the most used for robotics/mechanics problems, and this area of research has also been affected by the growth of the LLM. One of the most distinctive feature of RL is the reward signal, which formalizes the agent's goal. It's a hard problem that requires an expert, who understands the environment, the possible actions, and the weights of each reward. This human design approach is difficult and time-consuming. Some approaches like inverse-RL or autoRL allow for finding better reward functions.

Recent approaches use LLM as a reward function designer, proving that the large knowledge base of LLM allows for writing better reward function than human. LLM can be zero-shot and few-shot, and comparing these two cases, I want to try an architecture with zero-shot for initial prompt, and few-shot for context prompts to reshape the function.

The use of multiple modalities of input to the LLM can benefit from a better comprehension of the meaning. The analysis of video/image inputs by LLM can now be achieved with decent accuracy. I'm thinking that the usage of a multi-modal LLM in case of designing reward function can potentially outperform actuals architectures.

a1 Positioning to state of the art

According to [7] this project is in the LLM4RL branch. Some previous work about NLP [2] using RNN or word encoding, have been demonstrating the efficiency of NLP in Atari environment for reward shaping (cut in sub-reward). One of the first paper about using LLM for reward [3] used GPT3 as the reward signal. they directly give to the LLM the environment and the action, and the LLM return a reward signal. This approach doesn't seam to be appropriate for complexes robotics tasks because it required a non-binary reward signal. One other paper [10] introduces the self-refinement loop. Given a zero-shot prompt including environment, task, observable states and rules, the LLM generates a reward function. Next, the reward function is evaluated with the success rate of the trained policy and objective metrics. Finally, they construct a feedback prompt based on results, then the LLM updates the reward function. This architecture was benchmarked in several robotics environments. A novel architecture was EUREKA [6], which shows impressive performance on many environments. The particularity of EUREKA was to give the environment source code as an initial prompt, with a specific task. They use GPT4 to generate some reward function, then analyse the policy feedback and regenerate an improved function. EUREKA permit to have direct human feedback in the loop, and they demonstrate the benefit of this intervention. The reader can worry about used of GPT4 and the biggest LLM, but recently, some light LLM around 7B parameters like qwen2.5 [8] have show similar performances. Note that our take is specialised to generated code, so using an adapted LLM seam to be appropriate. Text2Reward [11] is the latest model in this soat. This model is quite similar to EUREKA, but they bring some novel things. First they used a pythonic description of the environment as the input of GPT4, secondly the architecture is focused on human feedback for improving the reward code. Finally, they test on several MuJoCo environments and deploy a policy learn by Text2Reward on a real robot.

Have seen, the resulting performances during the training depend largely on the initial and refined prompt. The famous paper how introduce LLaVA [5] have shown impressive performance for analysis image task. More recently, the model video-LLaVA [4] are capable to describe short video.

Finally, two papers [1, 9] related to use VLM as the reward signal directly. In our case, we want to use VLM only for description task.

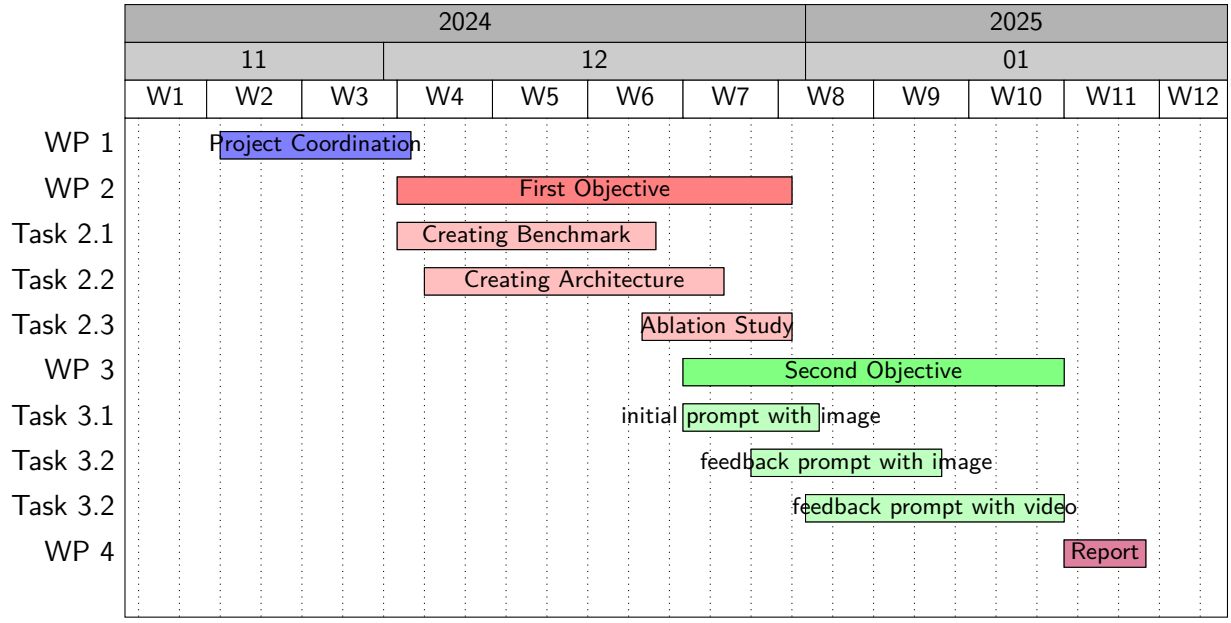
b Project's objectives and Methodology

b1 Objective 1: DREFUN architecture build and benchmark

This first objective is to create an architecture, based principally on EUREKA and Text2Reward [6, 11]. This architecture uses at first qwen2.5-coder [8] in this light version. DREFUN attempts to be capable of generating some rewards functions via LLM. It used self-refined loop by comparing policy founded with objective metrics and success rate, and can be augmented by human feedback. A Benchmark need to be created, on several tasks and environments from robotics in 3D and 2D simulations. The choice of the learning method is free between PPO, Reinforce and Direct Search, but that needed to be fixed earlier. Based on this benchmark, we're going to make an ablation study, focused on prompt engineering.

b2 Objective 2: DREFUN-V with Vision as grounding

The Second objective is to integrate LLaVA and video-LLaVA to our architecture. First we're going to use LLaVA with an initial prompt how included an image of the environment. It can be the goal image, or a random image just for enhancing the description. Next, LLaVA going to be used for self-refined loop, giving a prompt with image how show what is wrong or need to be upgraded. Finally, the integration of video-LLaVA as description of an episode for automatic analyse in the self-refined loop. Keep in mind that the objective 2 is going to be benchmarked during the development and be compared with the first objective.



References

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

[2] Prasoon Goyal, Scott Niekum, and Raymond J Mooney. "Using natural language for reward shaping in reinforcement learning". In: *arXiv preprint arXiv:1903.02020* (2019).

[3] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. "Reward design with language models". In: *arXiv preprint arXiv:2303.00001* (2023).

[4] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. "Video-llava: Learning united visual representation by alignment before projection". In: *arXiv preprint arXiv:2311.10122* (2023).

[5] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. "Visual instruction tuning". In: *Advances in neural information processing systems* 36 (2024).

[6] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. "Eureka: Human-level reward design via coding large language models". In: *arXiv preprint arXiv:2310.12931* (2023).

[7] Moschoula Pternea, Prerna Singh, Abir Chakraborty, Yagna Oruganti, Mirco Milletari, Sayli Bapat, and Kebei Jiang. "The RL/LLM Taxonomy Tree: Reviewing Synergies Between Reinforcement Learning and Large Language Models". In: *arXiv preprint arXiv:2402.01874* (2024).

[8] QwenLM. *Qwen2.5 Coder Family*. https://qwenlm.github.io/blog/qwen2.5-coder-family/. Accessed: 2024-12-01.

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

[10] Jiayang Song, Zhehua Zhou, Jiawei Liu, Chunrong Fang, Zhan Shu, and Lei Ma. "Self-refined large language model as automated reward function designer for deep reinforcement learning in robotics". In: *arXiv preprint arXiv:2309.06687* (2023).

[11] Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. "Text2Reward: Reward Shaping with Language Models for Reinforcement Learning". In: *The Twelfth International Conference on Learning Representations*. 2024.