Online Preference-based Reinforcement Learning with Self-augmented Feedback from Large Language Model

Songjun Tu
Institute of Automation, CASIA
Beijing, China
Pengcheng Laboratory
Shenzhen, China
School of Artificial Intelligence, UCAS
Beijing, China
tusongjun2023@ia.ac.cn

Jingbo Sun
Institute of Automation, CASIA
Beijing, China
Pengcheng Laboratory
Shenzhen, China
School of Artificial Intelligence, UCAS
Beijing, China
sunjingbo2022@ia.ac.cn

Qichao Zhang
Institute of Automation, CASIA
Beijing, China
School of Artificial Intelligence, UCAS
Beijing, China
zhangqichao2014@ia.ac.cn

Xiangyuan Lan
Pengcheng Laboratory
Shenzhen, China
Pazhou Laboratory (Huangpu)
Guangzhou, China
lanxy@pcl.ac.cn

Dongbin Zhao
Institute of Automation, CASIA
Beijing, China
School of Artificial Intelligence, UCAS
Beijing, China
dongbin.zhao@ia.ac.cn

ABSTRACT

Preference-based reinforcement learning (PbRL) provides a powerful paradigm to avoid meticulous reward engineering by learning rewards based on human preferences. However, real-time human feedback is hard to obtain in online tasks. Most work suppose there is a "scripted teacher" that utilizes privileged predefined reward to provide preference feedback. In this paper, we propose a RL Self-augmented Large Language Model Feedback (RL-SaLLM-F) technique that does not rely on privileged information for online PbRL. RL-SaLLM-F leverages the reflective and discriminative capabilities of LLM to generate self-augmented trajectories and provide preference labels for reward learning. First, we identify a failure issue in LLM-based preference discrimination, specifically "query ambiguity", in online PbRL. Then LLM is employed to provide preference labels and generate self-augmented imagined trajectories that better achieve the task goal, thereby enhancing the quality and efficiency of feedback. Additionally, a double-check mechanism is introduced to mitigate randomness in the preference labels, improving the reliability of LLM feedback. The experiment across multiple tasks in the MetaWorld benchmark demonstrates the specific contributions of each proposed module in RL-SaLLM-F, and shows that self-augmented LLM feedback can effectively replace the impractical "scripted teacher" feedback. In summary, RL-SaLLM-F introduces a new direction of feedback acquisition in online PbRL that does not rely on any online privileged information, offering an efficient and lightweight solution with LLM-driven feedback.[1]

[1]Corresponding author: Qichao Zhang (zhangqichao2014@ia.ac.cn) and Xiangyuan Lan (lanxy@pcl.ac.cn). Code Page: https://github.com/TU2021/RL-SaLLM-F

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

KEYWORDS

Online Preference-based Reinforcement Learning; Self-augmented LLM; LLM-driven Feedback; Query Ambiguity.

ACM Reference Format:
Songjun Tu, Jingbo Sun, Qichao Zhang, Xiangyuan Lan, and Dongbin Zhao. 2025. Online Preference-based Reinforcement Learning with Self-augmented Feedback from Large Language Model. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025,* IFAAMAS, 9 pages.

1 INTRODUCTION

Designing complex artificial reward functions is labor-intensive and time-consuming for reinforcement learning (RL) [8, 25]. Preference-based RL (PbRL) is considered a key paradigm to address this challenge by learning rewards based on human preference [5, 12].

However, the substantial human effort required to label a large number of preference queries significantly hinders its widespread application in real-world scenarios [13, 32]. Especially for online PbRL [20], obtaining real-time preference feedback necessitates immediate human interaction with the environment. Current online PbRL methods often assume a "scripted teacher" that provides real-time preference feedback by comparing handcrafted rewards of two trajectories from replay buffer [11, 12]. Unfortunately, relying on privileged rewards undermines the original intent of PbRL.

Large Pre-trained Models (LPMs), such as large language models (LLMs) [19] and vision-language models (VLMs) [38], equipped with extensive human prior knowledge, have recently gained significant attention. Recently, some studies have explored using LPMs instead of human supervision for reward design, including generating reward code [16, 31, 36] or calculating dense rewards directly based on the comparison of policy trajectories and targets [15, 22, 23]. Unfortunately, the first type of approach requires access to the simulation environment's code, and evaluating the reward code often involves multiple full RL training cycles, which is impractical for real-world applications. The other type of approach relies on comparing image-text similarities, but a single image may fail to reflect the underlying dynamic information described in the text, and reward variance is easily introduced by visual noise in the images [30]. Instead of designing reward functions or codes, PbRL learns rewards by comparing trajectory pairs. This approach requires only a single complete online training cycle and does not necessitate access to any low-level information, such as the environment's code [12, 30]. The learned downstream policies are guaranteed to have a suboptimal performance bound [42].

Instead of relying on "scripted teacher", some works [29, 30] try to obtain preference labels by querying LPMs with manually designed prompts, to train rewards and policies that align with human intentions. Although LPMs have substantial capabilities in analyzing trajectories, we find that they still struggle to distinguish the quality of suboptimal trajectories generated by the poor policy, especially during the early stages of training. The failure in preference discrimination impacts the learning of the reward model, further hindering performance. We refer to this phenomenon as "query ambiguity." Recently, LPMs have demonstrated reflective [18, 35, 41] and planning abilities in decision-making tasks, understanding the environment and predicting or planning future actions to achieve the task goal [3, 4, 21, 37, 39, 40]. These works inspire us with the following thought:

In addition to the discriminative capability, can LPMs leverage reflection to generate self-augmented trajectories that promoting efficient reward learning in online PbRL?

In this paper, we propose **R**einforcement **L**earning from **S**elf-**a**ugmented **LLM F**eedback (RL-SaLLM-F). This method seeks to mitigate query ambiguity in online PbRL and replaces "scripted teacher" with LLM-driven feedback. RL-SaLLM-F operates without relying on any predefined privileged rewards, thereby establishing a practical new paradigm for online PbRL. Specifically, state trajectories in the replay buffer are converted into text descriptions, and LLMs are queried to assign preference labels based on these trajectories. Furthermore, a second round of queries is performed to prompt the LLM to generate imagined trajectories aligned with task goals, serving as as new preference pairs for reward learning. In addition, we mitigate the randomness of LLM-based preference labels using a double-check mechanism that swaps the order of two input trajectories. Finally, we evaluate RL-SaLLM-F on multiple tasks in the Metaworld benchmark [34], it achieves comparable or better success rates than feedback from "scripted teacher" with privileged rewards. The core contributions in this paper are as follows:

- We identify an issue of query ambiguity in online PbRL with LLM feedback, and propose the RL-SaLLM-F method to mitigate the potential challenges.
- We leverage self-augmented LLM feedback to obtain preference labels efficiently and use a double-check mechanism to reduce randomness in the LLM-based labels.
- RL-SaLLM-F achieves comparable performance to that of "scripted teacher" with privileged reward information in the Metaworld benchmark, requiring only the lightweight and cost-effective GPT-4o-mini.
- The overall framework of RL-SaLLM-F does not rely on any predefined rewards or real-time human interaction, establishing a practical new paradigm for online PbRL.

2 RELATED WORKS

2.1 Large Pre-trained Models as Rewards

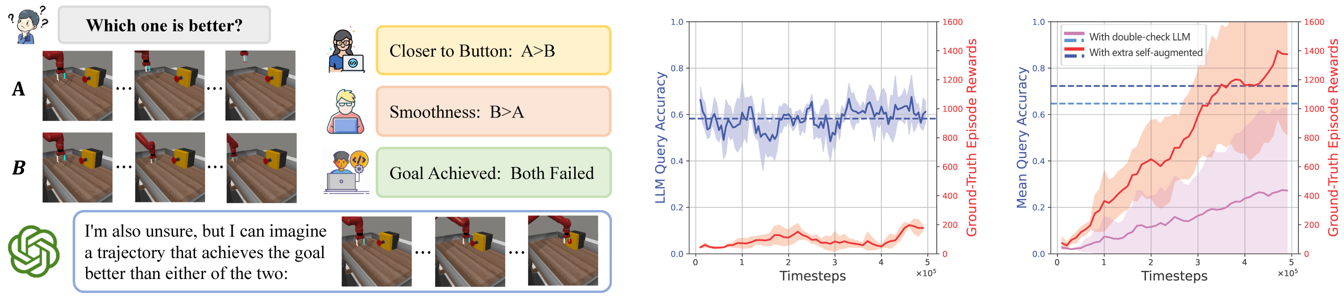
Applying RL in reward-free environments is challenging. Some studies assist in reward code design with the perceptual capabilities of LPMs [16, 27, 31, 33, 36]. A notable example is Eureka [16], which leverages GPT-4 [1] to evaluate environment and task information and generate reward code, followed by iterative updates using evolutionary algorithms. Building on this, [36] introduces a reflective mechanism to achieve further self-alignment. However, these methods assume that the environment code is accessible, and each evaluation of reward code requires a full RL training cycle, which is impractical for real-world deployment. Instead of designing reward code directly, we utilizes a LLM to provide preference labels for trajectory pairs from online replay buffer, relying on a more accessible textual description of the environment rather than full access to its code.

Another line of research obtains rewards from visual observations by VLMs [14, 15, 17, 22, 23]. For example, RoboCLIP [23] computes reward signals by comparing video representations of expert demonstrations with those of policy trajectories. Recently, FuRL [7] incorporates a learnable layer into CILP, and fine-tunes it to align with real tasks. Despite achieving impressive performances, these methods often need to fine-tune with expert data to reduce variance and noise in the reward [7, 23]. On the other hand, some works compare image-text similarities to obtain dense rewards, but a single image is insufficient to capture the required dynamic information [30]. For example, in robotic tasks like picking up an object, a single image cannot indicate if the arm is approaching or moving away, causing confusion reward learning. Different from the aforementioned methods, we abstracts observed trajectories into text and learns a reward model through pairwise trajectory comparisons. Our approach enhances the ability to understand trajectories, reduces query costs, and guarantees the convergence properties of downstream RL with preference-based reward modeling [42].

2.2 Preference-based Reinforcement Learning

A new paradigm for learning reward functions during interactions with the environment, known as online PbRL, leverages human feedback given in the form of trajectory preferences [5, 12]. Most of online PbRL research aims to address challenges such as preference noise [26], reward credit assignment [28], limited preference data [10] and efficient querying [20]. However, frequent querying of human preferences during online training is impractical. In previous online PbRL methods, a "scripted teacher" is assumed to provide preference labels by comparing privileged predefined rewards of two trajectories in the specific task [11].

This "scripted teacher" relying on privileged rewards serves as a compromise to investigate online PbRL algorithms as immediate human feedback is not available. In this work, we aim to replace the "scripted teacher" with a LLM to provide preference feedback. Recently, existing works such as RL-VLM-F [30] and PrefCLM [29] have explored using VLMs to evaluate the quality of trajectories for visual input tasks. However, repeated image pair queries [30] and the aggregation of multiple GPT4 feedback [29] make these methods costly. Additionally, we find samples with no significant preference difference in the replay buffer can lead to a potential risk



(a) The sampled trajectory pairs that the LLM failed to assess. (b) Training curves of LLM feedback. (c) Double-check & Extra self-augmentation.

Figure 1: Query ambiguities in online PbRL. (a) Two failed trajectories are provided, converted into text form, and input into the LLM, showing that the LLM struggles to evaluate such trajectories. (b) Training curves of PEBBLE with LLM feedback. The blue line represents the labeling accuracy of the LLM and the red line represents the predefined episode rewards. (c) Training curves of double-check mechanism and additional self-augmented LLM feedback.

of query ambiguity, which will further increase costs and reduce efficiency. In contrast, we use a self-augmented and double-checking feedback to improve the efficiency and reliability based on two-round preference queries, mastering robotic manipulation tasks with with cost-effective GPT-4o-mini [1].

3 PRELIMINARIES

3.1 Reinforcement Learning

RL is formulated as a Markov Decision Process (MDP) [24], which is characterized by the tuple $M = \langle S, A, P, r, \gamma \rangle$. S is the state space, A is the action space, $P : S \times A \times S \rightarrow \mathbb{R}$ is the transition probability distribution, $r : S \rightarrow \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. The objective of RL is to determine an optimal policy π that maximizes the expected cumulative reward: $\pi = \arg\max_\pi \mathbb{E}_{s_0,a_0,\ldots} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right]$.

We use Soft Actor-Critic (SAC) [9], an off-policy RL algorithm, as our low-level approach. Specifically, transitions from interactions with the environment are stored in the replay buffer. The actor-critic is then trained by sampling data from the buffer and maximizing the entropy of the stochastic policy.

3.2 Learning Rewards from Human Feedback

Following previous studies [6], we consider state-only trajectories of length H composed of states and actions, defined as $\sigma = \{s_k, \ldots, s_{k+H}\}$. The goal is to align human preference y between pairs of trajectory segments σ^0 and σ^1, where y denotes a distribution indicating human preference, captured as $y \in \{1, 0, 0.5\}$. The preference label $y = 1$ indicates that σ^0 is preferred to σ^1, namely, $\sigma^0 \succ \sigma^1$, $y = 0$ indicates $\sigma^1 \succ \sigma^0$, and $y = 0.5$ indicates equal preference for both. The preference data are stored as triples, denoted as $\mathcal{D}$: (σ^0, σ^1, y). Then, the Bradley-Terry model [2] is employed to couple preferences with rewards. The preference predictor is defined as follows:

$$P_\psi[\sigma^1 \succ \sigma^0] = \frac{\exp\left(\sum_t \hat{r}_\psi(s_t^1)\right)}{\sum_{i\in\{0,1\}} \exp\left(\sum_t \hat{r}_\psi(s_t^i)\right)} \tag{1}$$

where $\hat{r}_\psi$ is the reward model to be trained, and ψ is its parameters. Subsequently, the reward function is optimized using the cross-entropy loss, incorporating the human predefined label y and the preference predictor P_ψ:

$$\begin{aligned}\mathcal{L}_{\text{CE}} = -\mathbb{E}_{(\sigma^0,\sigma^1,y)\sim\mathcal{D}} \Big\{ &(1-y) \log P_\psi[\sigma^0 \succ \sigma^1] \\ &+ y \log P_\psi[\sigma^1 \succ \sigma^0] \Big\}\end{aligned} \tag{2}$$

In online PbRL, we train an off-policy RL algorithm while periodically sampling trajectory pairs from the replay buffer $\mathcal{B}$ for preference queries, as done in PEBBLE [12]. As online human preference is often not available, the "scripted teacher" is used to provide preference labels y in most previous works [5, 12]. Specifically, it generates preferences based on predefined task reward r from the environment as follows: $y = i$, where $i = \arg\max_{i\in\{0,1\}} \sum_t r(s_t^i, a_t^i)$. These trajectory pairs, along with the preference labels, form the preference dataset $\mathcal{D}$. We train the reward model on $\mathcal{D}$, and subsequently relabel the data in $\mathcal{B}$ using this model.

4 QUERY AMBIGUITY IN ONLINE PBRL

In this section, we highlight a potential risk of query ambiguity in online PbRL. This risk stems from the inherent challenges due to the trajectories sampled from the replay buffer tend to be suboptimal and of insufficient quality. Figure 1a illustrates an example from the MetaWorld Button-Press task, with two suboptimal trajectories sampled from the replay buffer. Neither trajectory successfully achieves the target: trajectory A deviates upward with larger movement amplitude, while trajectory B moves to the left-rear with smaller movement amplitude. For such low-quality trajectories, human may be able to evaluate their quality from multiple perspectives (e.g., distance to the button, trajectory smoothness), but it is challenging to determine which trajectory is better in terms of achieving the target. The "scripted teacher" obtains labels from privileged task rewards by comparing the rewards of two trajectories and selecting the one with the higher cumulative rewards as the preferred trajectory, thereby avoiding this potential issue. However, when human or LLM provide feedback evaluations, the situation becomes significantly more complex. In Figure 1b, we

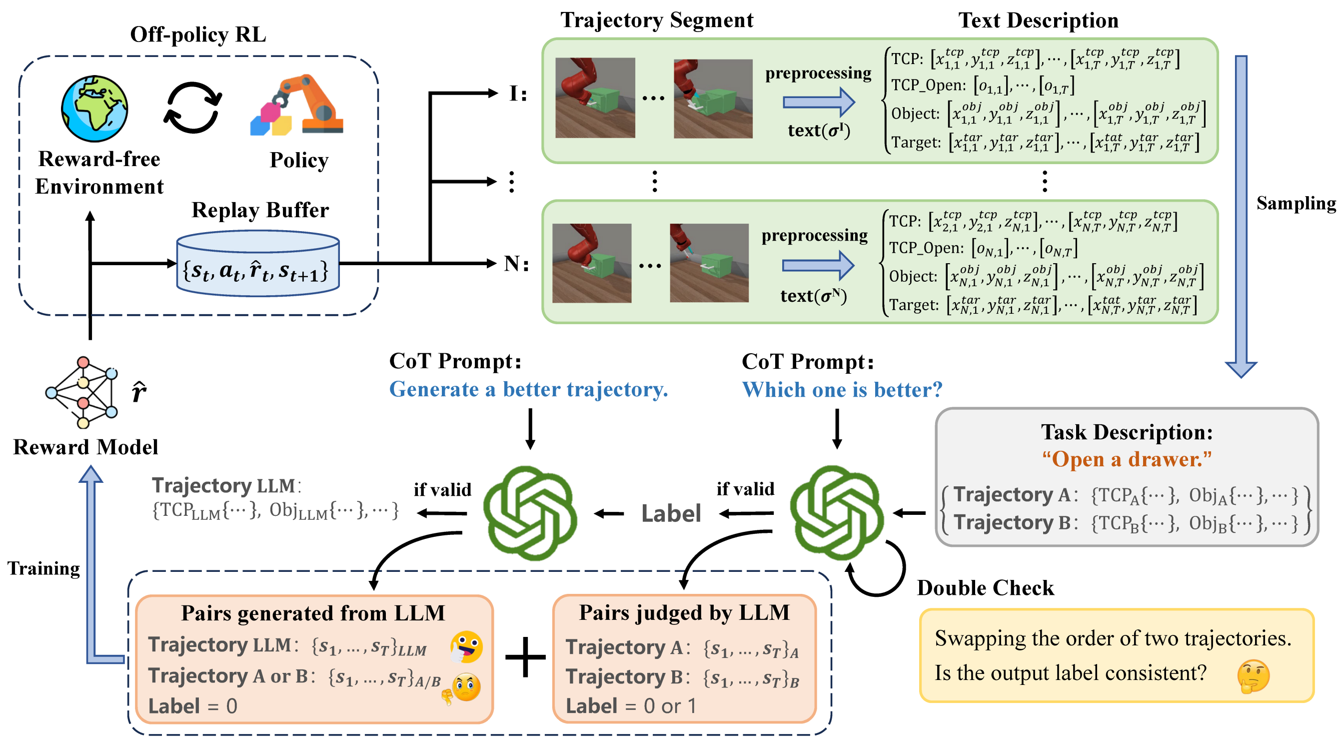


Figure 2: The overall framework of RL-SaLLM-F. First, trajectories are sampled from the replay buffer and converted into coordinate text descriptions. Next, the text representation of the trajectory pairs are selected and queried through the LLM twice with different orderings to obtain feedback labels. Subsequently, based on the sampled trajectories, the 'imagined' trajectories that better achieve the goal are generated by the LLM to train the reward model.

train PEBBLE [12] by directly replacing the "scripted teacher" with GPT-4o-mini as feedback, and record the label accuracy (compared to the "scripted teacher") and episode rewards. The policy performance remain poor throughout the training, and the accuracy of the LLM labels is consistently low, with an average accuracy of only 58.2%.

How can we improve policy learning in online PbRL? An intuitive approach is to enhance the reliability of the labels. In Section 5.2, we reduce the randomness of the labels through a double-check mechanism, which further improve the accuracy of the LLM queries. In Figure 1c, we plot the training curves and average label accuracy (pink and cyan lines) after incorporating the double-check mechanism. The label accuracy increase from 58.2% to 64.8%, which accompanies an improvement in policy performance. However, it still demonstrates low sample efficiency and fails to achieve the task goal.

Another idea is, can we use additional high-quality data to train the reward model, thereby driving both reward and policy learning? In Section 5.3, we propose a self-augmented LLM feedback method, which generates high-quality additional trajectory pairs to facilitate the learning of the reward model. Similarly, we plot the corresponding training curves and average accuracy (red and blue lines) in Figure 1c. Note that we do not enhance the discriminative ability of LLM by prompt engineering; instead, we simply use the higher-quality data generated by the LLM to train the reward model. As the training policy gradually approaches optimality, the average label accuracy further increases from 64.8% to 72.3%.

Based on these observations, we think self-augmented LLM feedback is effective for mitigating the risk of query ambiguity in online PbRL. This approach not only directly improves the discriminative reliability of the LLM but also further drives reward learning through self-augmented data. In turn, the highly discriminative reward function encourages the policy to sample diverse, high-quality data, enriching the LLM's query diversity and indirectly enhancing its discriminative accuracy, forming a positive feedback loop in the training process.

5 RL-SALLM-F

5.1 The Overall Framework

In this section, we propose **R**einforcement **L**earning from **S**elf-**a**ugmented **LLM F**eedback (RL-SaLLM-F).

The overall training process is shown in Figure 2, following PEBBLE[12], with an off-policy SAC agent as the policy, consisting of three steps:

- **Step 1 (Unsupervised Pre-training)**: In the early stage of training, to encourage exploration and increase trajectory diversity, an intrinsic reward $r^{\text{int}}(\mathbf{s}_t) = \log(\|\mathbf{s}_t - \mathbf{s}_t^k\|)$ is used for pre-training downstream policy π_ϕ, here s_t^k is the k-th nearest neighbor of s_t.

- **Step 2 (Label Querying and Reward Learning)**: Sampling candidate trajectory pairs and querying labels from the LLM (see Section 5.2), followed by generating self-augmented imagined trajectory to train the reward model $\hat{r}_\psi$ (see Section 5.3).
- **Step 3 (Policy Learning)**: Relabeling the replay buffer $\mathcal{B}$ with $\hat{r}_\psi$ and updating the policy π_ϕ and Q-function Q_θ using SAC.
- **Repeat Step 2 and Step 3**.

5.2 Query Feedback Labels Judged by LLM

First, we sample a trajectory pair $\{\sigma^0, \sigma^1\}$ from the replay buffer $\mathcal{B}$, convert it into textual representations $\{\text{text}(\sigma^0), \text{text}(\sigma^1)\}$, and query the LLM with task-directed Chain-of-Thought (CoT) prompt for preference labels:

$$y = \text{LLM}\left(\text{Task, Traj 0} = \text{text}(\sigma^0), \text{Traj 1} = \text{text}(\sigma^1)\right) \quad (3)$$

To improve query accuracy, we use a double-check mechanism to mitigate the randomness in the labels. Specifically, we reverse the order of the input trajectories and query the LLM again:

$$y_{inv} = \text{LLM}\left(\text{Task, Traj 0} = \text{text}(\sigma^1), \text{Traj 1} = \text{text}(\sigma^0)\right) \quad (4)$$

If the values of y and y_{inv} indicate the same preference feedback is labeled, namely $y = 0, y_{inv} = 1$ or $y = 1, y_{inv} = 0$, we consider the feedback label to be valid. Otherwise, we consider this trajectory pair to be indistinguishable for the LLM, and discard it. When the label is valid, the trajectory pair triple (σ^0, σ^1, y) is stored into the preference dataset $\mathcal{D}$.

Remark 1. *An alternative is to consider the LLM's evaluation of the trajectory pair as equivalent, retaining the pair and setting the label as $y = 0.5$, treating it as soft label in reward learning. We compare this approach and find that, during early training, the LLM struggle to distinguish the quality of sampled trajectories, resulting in numerous hallucinated labels. Including such trajectory pairs is detrimental to reward model learning, so we conclude that discarding them leads to more efficient reward training.*

5.3 Self-augmented Feedback Generated by LLM

Since the differences between trajectory pairs sampled from the replay buffer are small, particularly during early training, we leverage the reflective and planning abilities of LLM to generate an imagined trajectory that is more goal-directed to accelerate reward model training. Specifically, we prompt the LLM to generate a textual trajectory $\text{text}(\sigma^{LLM})$ that outperform the better trajectory $\sigma^{0/1}$ of the current trajectory pair $\{\sigma^0, \sigma^1\}$, while ensuring that it shares the same initial state as the input:

$$\text{text}(\sigma^{LLM}) = \text{LLM}\left(\text{Generate Prompt, Traj} = \text{text}(\sigma^{0/1})\right) \quad (5)$$

Next, the algorithm checks whether σ^{LLM} has the same step length and data format as the sampled trajectory segments. Once the generated trajectory passes the format check, it is converted into a state-based trajectory σ^{LLM} and the imagined pair triple $(\sigma^{LLM}, \sigma^{0/1}, y = 0)$ (i.e., self-augmented feedback) is stored into $\mathcal{D}$.

Remark 2. *The imagined trajectories do not have to adhere to physical constraints, as they serve solely for preference comparisons to train the reward model, rather than for extracting policies or constructing a world model. We assume the reward model is Markovian, implying that the immediate reward relies exclusively on the current state. We find that once the LLM understands the task goal, the generated trajectories, even if not physically feasible, are still beneficial for training the reward model efficiently. This insight enables us to use a lightweight LLM with simple prompts for trajectory generation without incorporating numerous executable trajectory constraints.*

After several rounds of querying and generating, $\mathcal{D}$ contains trajectory pairs sampled from the replay buffer $\mathcal{B}$ with LLM-judged labels, and self-augmented trajectory pairs and labels generated by the LLM. These labeled trajectory pairs will be used to train the reward model by minimizing Equation 2.

6 EXPERIMENTS

6.1 Setup

We evaluate RL-SaLLM-F on multiple robotic manipulation tasks in the MetaWorld [34] benchmark. The states include the coordinates of the Tool Center Point (TCP), the extent of TCP's opening, the coordinates of the manipulated object and the target position. All of these states can be easily converted into text-based string representations. We conduct experiments on eight tasks:

- ***Button Press***: Press a button;
- ***Drawer Open***: Open a drawer;
- ***Drawer Close***: Push and close a drawer;
- ***Door Open***: Open a door with a revolving joint;
- ***Door Unlock***: Rotate the lock and unlock the door;
- ***Window Open***: Push and open a window;
- ***Handle Pull***: Pull a handle up;
- ***Reach***: Reach a goal position.

To be deemed successful, the agent must achieve the task target within a limited number of steps and maintain it until the end of the episode. All the positions are randomized at each initialization. In the following sections, we aim to address five key questions:

(1) Can RL-SaLLM-F master robot control without any privileged predefined rewards or human feedback? (Section 6.2)
(2) Does each component of RL-SaLLM-F contribute to performance improvement? (Section 6.3)
(3) Does the learned reward function align effectively with task progress? (Section 6.4)
(4) How accurately does RL-SaLLM-F assess the quality of trajectories, and how high is the quality of the LLM-based generated trajectories? (Section 6.5 and 6.6)
(5) How do larger LLM impact performance? (Section 6.7)

6.2 Comprehensive Performance Comparison

To evaluate whether RL-SaLLM-F can master robotic manipulation tasks without relying on prior reward information, we compare it against PEBBLE[12] and SAC. PEBBLE serves as a popular approach for evaluation in online PbRL and relies on the "scripted teacher" feedback based on task rewards. SAC directly leverages task rewards for policy learning, representing the refer upper bound of online RL performance.

Remark 3. *At the outset, it should be clarified that our goal is not to surpass these methods, as they both rely on privileged information from the environment, whereas RL-SaLLM-F operates without such*

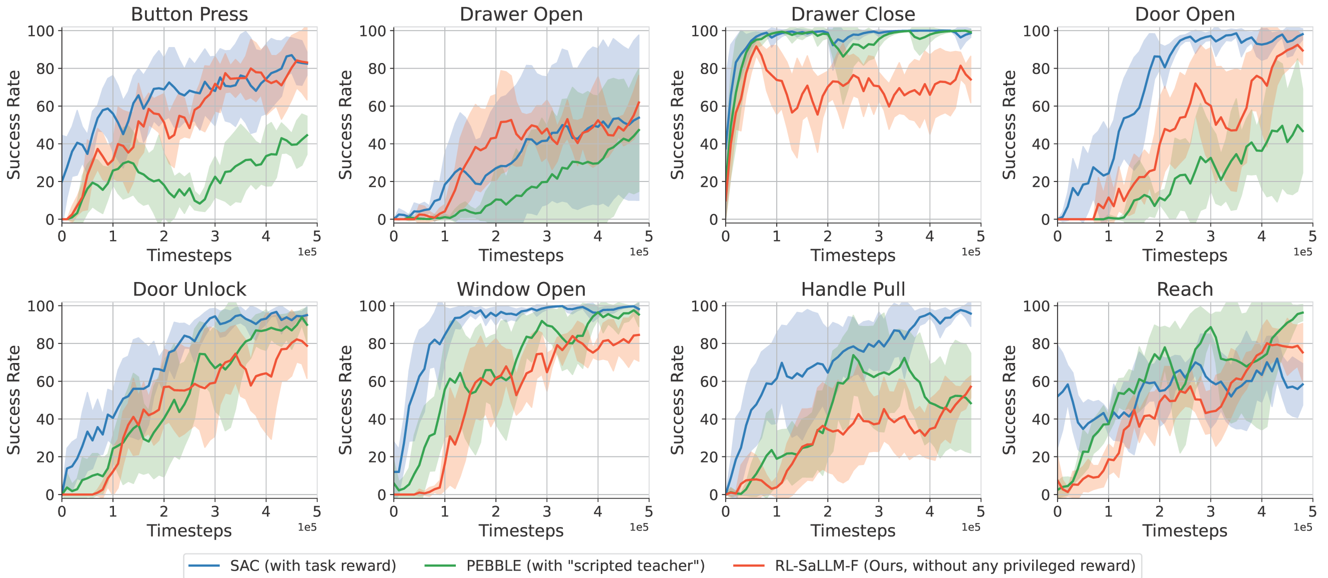


Figure 3: Learning curves of all compared methods on 8 tasks. Results are averaged over 5 seeds, and shaded regions represent standard error. RL-SaLLM-F masters robotic manipulation without any online privileged reward, performing on par with PEBBLE, which uses 'scripted teacher' feedback, and even SAC with predefined reward functions in partial tasks.

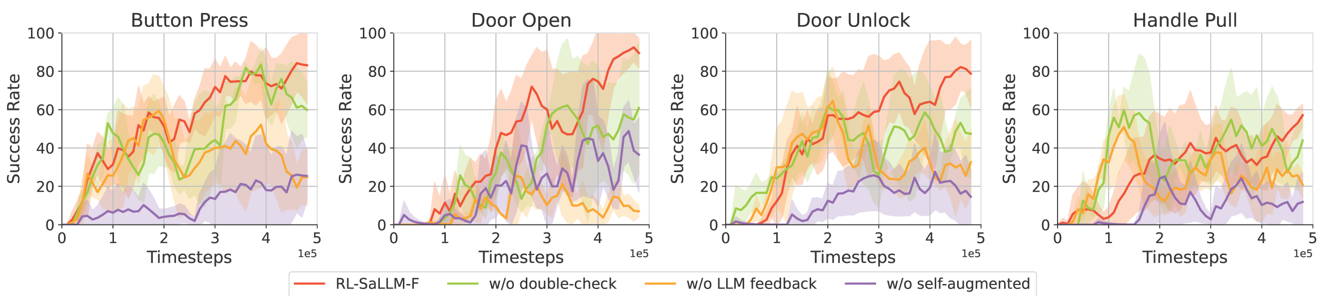


Figure 4: Learning curves of the ablation study. When any component of RL-SaLLM-F is removed, the performance decreases. Specifically, the absence of self-augmented feedback leads to notably poor success rate.

assumptions. Due to the lack of baseline comparisons under the same settings, we consider the comparison in this section as reference only.

All methods share the same hyperparameters, including the reward model structure, query number and feedback segment length. The total query budget for the entire training process is 2000, with each session of concentrated queries consisting of 20 queries, each involving a trajectory segment length of 10. Due to the cost and time constraints of queries, RL-SaLLM-F uses GPT-4o-mini-2024-07-18 as the LLM feedback. Each task is trained with 5e5 environment steps,and the learning curves are shown in Figure 3.

Overall, RL-SaLLM-F achieve comparable or even superior performance compared to PEBBLE. Notably, in the *Button Press* and *Drawer Open* tasks, the success rate of RL-SaLLM-F is on par with that of SAC with real-time task rewards. This indicates that SaLLM effectively implement feedback and leverage improved imagine trajectories to drive more efficient and superior reward function training. Specifically, in the Button Press task, RL-SaLLM-F generated significantly higher-quality imagined trajectories (see Section 6.6 for detail analysis). Meanwhile, we observe that imperfect LLM feedback labels led to RL-SaLLM-F exhibiting lower training efficiency on some tasks compared to PEBBLE with "scripted teacher" feedback, see Section 6.5 for further analysis on label accuracy.

6.3 Ablation Study

To investigate the role of each component in RL-SaLLM-F, we conduct ablation studies. Specifically, we remove different modules from the algorithm the training curves of 4 chosen tasks are shown in the Figure 4. In the legend, 'w/o double-check' indicates executing sampling feedback and self-augmented feedback without checking the feedback labels; 'w/o LLM feedback' indicates executing only LLM self-augmented feedback without sampling feedback;

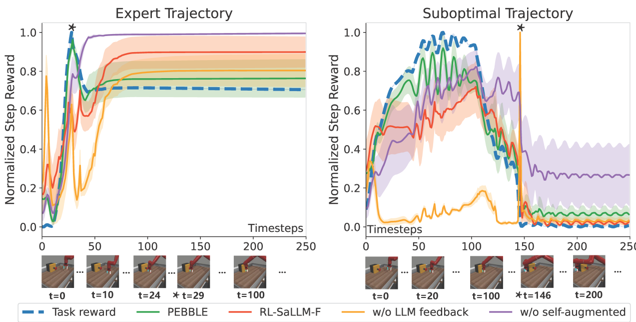


Figure 5: Normalized step rewards for the expert and suboptimal trajectories in the *Button Press* task. The rewards of RL-SaLLM-F show better alignment with the predefined task reward than the ablation variants, particularly in the suboptimal trajectory.

and 'w/o self-augmented' indicates executing only LLM sampling feedback (with double-check) without self-augmented feedback.

As Figure 4 shows, when any component of RL-SaLLM-F is removed, the performance decreases. Specifically, the absence of self-augmented feedback leads to notably poor success rate. On the other hand, only self-augmented feedback may lead to rapid performance gains in early training but often causes instability later on. The double-check mechanism serves as a valuable enhancement to the algorithm. When incorporating self-augmented feedback, the double-check mechanism further stabilizes performance by increasing the accuracy of sampled trajectory labels.

6.4 Analysis of The Learned Reward Model

In addition to evaluating the quality of the learned policy by comparing the performance, a remaining problem is, whether the learned reward model aligns with task progress? To answer this question, we compare the reward models learned by PEBBLE and RL-SaLLM-F with different ablation versions. Specifically, we choose an expert trajectory and a suboptimal trajectory in the *Button Press* task, then label the rewards for these two trajectories. In the expert trajectory, the robotic arm moves directly to the button, presses it, and then stays still. In the suboptimal trajectory, the robotic arm moves for a while, presses the button, and then moves away from the button center, causing it to spring back. The normalized step rewards of each method are shown in the Figure 5.

We observe that the rewards of PEBBLE exhibit a very similar trend to the predefined task reward, which aligns with our intuition, as PEBBLE's reward model is trained relying on the task reward

Table 1: Label accuracy and discard rate comparison for RL-SaLLM-F and the ablation variants.

	Button Press	*Door Open*	*Door Unlock*	*Handle Pull*
RL-SaLLM-F	**72.30%**	**65.21%**	**71.66%**	**67.73%**
w/o double-check	63.06%	60.07%	66.29%	61.13%
w/o LLM feedback	62.77%	59.62%	59.64%	57.65%
w/o self-augmented	64.79%	60.60%	66.25%	65.89%
Discard Rate	36.70%	37.89%	34.14%	37.42%

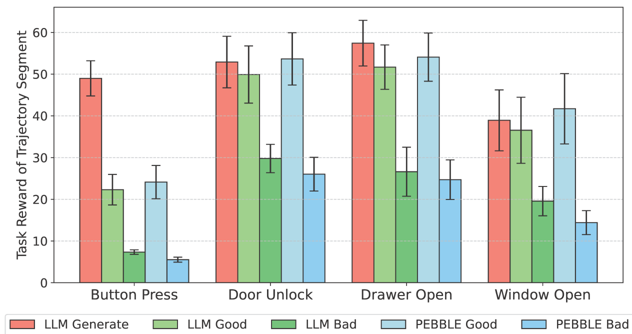


Figure 6: Comparison of task rewards for trajectories generated by the RL-SaLLM-F and evaluated by LLM and "scripted teacher". The red bars represent trajectories generated by the LLM. The others represent trajectories sampled from the environment during training, with the green bars judged by the LLM and the blue bars evaluated by the "scripted teacher".

indirectly. The rewards of RL-SaLLM-F show changes that are more aligned with the task reward compared to the two ablation variants, particularly in the suboptimal trajectory. The 'w/o LLM feedback' variant unexpectedly shows a reward peak at $t = 146$, while the 'w/o self-augmented' variant maintains a high reward value from $t = 130$ to $t = 145$. This anomaly may be due to insufficient fine-grained understanding by the LLM, leading to distort generated trajectories and labeling errors.

Remark 4. *Interestingly, although the rewards of RL-SaLLM-F appear to have a larger discrepancy from the predefined rewards compared to PEBBLE, RL-SaLLM-F achieves higher task success rates. We suspect this may be due to the predefined reward being less effective than the goal-based evaluation reward, or the trajectory augmentation in RL-SaLLM-F contributing to greater stability in reward model training, leading to better policy improvement.*

6.5 Feedback Labels Quality Evaluation

We further compare the quality of feedback labels to investigate the performance of our method in labeling. We extract all sampled trajectory pairs from the training process and analyze the accuracy of the LLM in judging these trajectory pairs (compared to the "scripted teacher"). The average results of five training seeds are presented in Table 1. Apart from the header, the first four rows show the query label accuracy for RL-SaLLM-F and its ablation variants, while the last row indicates the proportion of query trajectory pairs discarded due to the the double-check mechanism in RL-SaLLM-F. For the 'w/o LLM feedback' variant, we still query trajectory preferences during training but does not perform self-checking of the LLM labels.

As shown in the Table 1, RL-SaLLM-F achieve the highest label accuracy among all variants. The absence of the double-check mechanism lead to a decrease in accuracy, indicating that self-checking can reduce label randomness. Furthermore, compared to the 'w/o self-augmented' variant, the label accuracy of RL-SaLLM-F is still higher, suggesting that self-augmentation improves reward and policy performance, leading to greater diversity in sampled trajectories. The diversity of trajectories helps the LLM make more

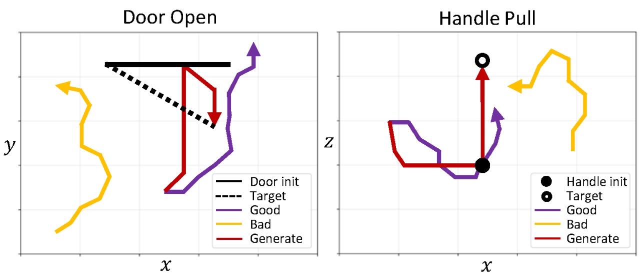


Figure 7: Two examples of 2D projections of trajectories generated by the LLM. Compared to the sampled trajectories, the generated trajectories successfully achieve the task goals.

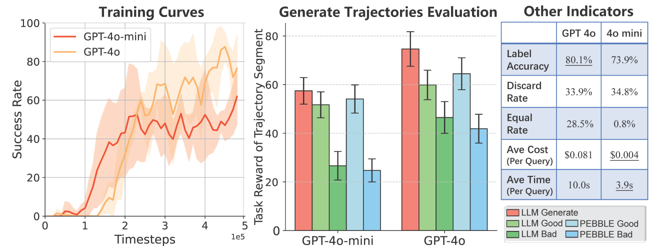


Figure 8: Comparison of RL-SaLLM-F performance with GPT-4o and GPT-4o-mini. Albeit the larger LLM delivers superior results, it comes with increased resource consumption.

accurate preference judgments, consistent with our intuition in Section 4. However, even the best RL-SaLLM-F method achieves only around 70% label accuracy. This could be due to the bias of the "scripted teacher", which provides binary reward labels based solely on absolute reward comparisons. Such labels may not align with human or LLM intentions for trajectories with similar performance.

6.6 Generated Trajectories Quality Evaluation

Furthermore, we study the quality of the trajectories generated by the LLM. As analyzed in the previous section, we extract the better and worse trajectories from the pairs buffer during RL-SaLLM-F training, along with the imagined trajectories generated by the LLM, and calculate the average rewards and variance across different seeds using the predefined task reward function. Then, we evaluate the same trajectories using the labels of "scripted teacher" and record the average rewards, as shown in Figure 7. Only tasks related to the reward function and state are considered, as the generated trajectories do not include action information.

Comparing the rewards of the LLM-generated trajectories with those it judged as better, we find that RL-SaLLM-F can generate higher-quality trajectories than those it initially preferred, which in turn benefits reward learning. When comparing the LLM-generated trajectories with those evaluated by the "scripted teacher", we find that the LLM-generated trajectories perform comparably to, or even better than, those deemed preferred by the ”scripted teacher“. Moreover, a positive correlation is observed between the quality of generated trajectories and task performance, particularly in the *Button Press* and *Drawer Open* tasks, where RL-SaLLM-F outperforms PEBBLE. The average reward of the generated trajectories is also higher than that of the trajectories deemed superior by PEBBLE.

Finally, we provide two visualization examples of 2D projections of trajectories generated by the LLM in Figure 7. In the *Door Open* task, compared to the sampled trajectories, the generated one first moves directly towards the door and then pulls it open to the target position. In the *Handle Pull* task, the generated trajectory first locates the handle and then pulls it upward to the target position. The examples demonstrate a thorough understanding of task goals and trajectory information by the LLM.

6.7 Impact of LLM Scale

An intuitive question is: would scaling up the LLM boost the performance of RL-SaLLM-F? The answer is certainly yes. To clearly showcase this improvement, we repeat the experimental analysis from the previous sections on the *Drawer Open* task with GPT-4o as the LLM, and conduct a comprehensive performance evaluation, as shown in Figure 8. In addition to the familiar metrics, we define an additional one: "Equal Rate", which is the proportion of queries where the LLM assigns equal preferences to two trajectories.

In terms of policy performance, trajectory quality, and preference label accuracy, GPT-4o significantly outperforms its smaller counterpart, GPT-4o-mini. Additionally, GPT-4o tends to provide more labels with $y = 0.5$, suggesting a more cautious evaluation, and avoiding making arbitrary judgments when the trajectories are similar. However, querying GPT-4o is expensive: about 20 times the price of GPT-4o-mini for the same number of tokens. Therefore, we still believe that combining more affordable LLMs offers a highly cost-effective solution.

7 CONCLUSIONS

In this work, we introduce RL-SaLLM-F, a novel technique that leverages self-augmented feedback from LLMs for online PbRL. By abstracting state trajectories into textual descriptions and utilizing LLMs to generate self-augmented imagined trajectories and provide preference labels, RL-SaLLM-F successfully addresses the limitations of relying on online privileged rewards or real-time human feedback. The experimental analysis from various perspectives have demonstrated the effectiveness of the proposed method.

Additionally, we find that the accuracy of the LLM labels in trajectory evaluation remains limited, and further improving this accuracy is essential for more efficient task training. Meanwhile, our method may not directly handle image inputs, future work could explore obtaining precise object coordinates through methods such as camera coordinate calibration to align with our algorithmic framework, or applying VLMs and diffusion models for discrimination and generation of image trajectories.

ACKNOWLEDGEMENTS

This work is supported by the National Key Research and Development Program of China under Grants 2022YFA1004000, the CAS for Grand Challenges under Grants 104GJHZ2022013GC, the National Natural Science Foundation of China under Grants 62173325 and 62402252, and Pengcheng Laboratory Project PCL2024Y02.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika* 39, 3/4 (1952), 324–345.

[3] Yaran Chen, Wenbo Cui, Yuanwen Chen, Mining Tan, Xinyao Zhang, Dongbin Zhao, and He Wang. 2025. RoboGPT: an LLM-based embodied long-term decision making agent for instruction following tasks. *IEEE Transactions on Cognitive and Developmental Systems* (2025).

[4] Yuanwen Chen, Xinyao Zhang, Yaran Chen, Dongbin Zhao, Yunzhen Zhao, Zhe Zhao, and Pengfei Hu. 2024. Common Sense Language-Guided Exploration and Hierarchical Dense Perception for Instruction Following Embodied Agents. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1–6.

[5] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems* 30 (2017).

[6] Zibin Dong, Yifu Yuan, HAO Jianye, Fei Ni, Yao Mu, YAN ZHENG, Yujing Hu, Tangjie Lv, Changjie Fan, and Zhipeng Hu. 2024. AlignDiff: Aligning Diverse Human Preferences via Behavior-Customisable Diffusion Model. In *The Twelfth International Conference on Learning Representations*.

[7] Yuwei Fu, Haichao Zhang, Di Wu, Wei Xu, and Benoit Boulet. 2024. FuRL: Visual-Language Models as Fuzzy Rewards for Reinforcement Learning. In *Forty-first International Conference on Machine Learning*.

[8] Abhishek Gupta, Aldo Pacchiano, Yuexiang Zhai, Sham Kakade, and Sergey Levine. 2022. Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. *Advances in Neural Information Processing Systems* 35 (2022), 15281–15295.

[9] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. PMLR, 1861–1870.

[10] Xiao Hu, Jianxiong Li, Xianyuan Zhan, Qing-Shan Jia, and Ya-Qin Zhang. 2024. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*.

[11] K Lee, L Smith, A Dragan, and P Abbeel. 2021. B-Pref: Benchmarking Preference-Based Reinforcement Learning. *Advances in Neural Information Processing Systems* (2021).

[12] Kimin Lee, Laura M Smith, and Pieter Abbeel. 2021. PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. In *International Conference on Machine Learning*. PMLR, 6152–6163.

[13] Xinran Liang, Katherine Shu, Kimin Lee, and Pieter Abbeel. 2022. Reward Uncertainty for Exploration in Preference-based Reinforcement Learning. In *10th International Conference on Learning Representations, ICLR 2022*. International Conference on Learning Representations.

[14] Calvin Luo, Mandy He, Zilai Zeng, and Chen Sun. 2024. Text-Aware Diffusion for Policy Learning. *arXiv preprint arXiv:2407.01903* (2024).

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

[16] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024. Eureka: Human-Level Reward Design via Coding Large Language Models. In *The Twelfth International Conference on Learning Representations*.

[17] Parsa Mahmoudieh, Deepak Pathak, and Trevor Darrell. 2022. Zero-shot reward specification via grounded natural language. In *International Conference on Machine Learning*. PMLR, 14743–14752.

[18] Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. 2024. Llm critics help catch llm bugs. *arXiv preprint arXiv:2407.00215* (2024).

[19] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196* (2024).

[20] Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. 2022. SURF: Semi-supervised Reward Learning with Data Augmentation for Feedback-efficient Preference-based Reinforcement Learning. In *International Conference on Learning Representations*.

[21] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*. 1–22.

[22] Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. 2024. Vision-Language Models are Zero-Shot Reward Models for Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*.

[23] Sumedh Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. 2024. Roboclip: One demonstration is enough to learn robot policies. *Advances in Neural Information Processing Systems* 36 (2024).

[24] Richard S Sutton. 2018. Reinforcement learning: An introduction. *A Bradford Book* (2018).

[25] Zhentao Tang, Yuanheng Zhu, Dongbin Zhao, and Simon M Lucas. 2023. Enhanced rolling horizon evolution algorithm with opponent model learning: results for the fighting game AI competition. *IEEE Transactions on Games* 15, 1 (2023), 5–15.

[26] Songjun Tu, Jingbo Sun, Qichao Zhang, Yaocheng Zhang, Jia Liu, Ke Chen, and Dongbin Zhao. 2025. In-Dataset Trajectory Return Regularization for Offline Preference-based Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

[27] David Venuto, Mohammad Sami Nur Islam, Martin Klissarov, Doina Precup, Sherry Yang, and Ankit Anand. 2024. Code as Reward: Empowering Reinforcement Learning with VLMs. In *Forty-first International Conference on Machine Learning*.

[28] Mudit Verma and Katherine Metcalf. 2024. Hindsight PRIORs for Reward Learning from Human Preferences. In *The Twelfth International Conference on Learning Representations*.

[29] Ruiqi Wang, Dezhong Zhao, Ziqin Yuan, Ike Obi, and Byung-Cheol Min. 2024. Prefclm: Enhancing preference-based reinforcement learning with crowdsourced large language models. *arXiv preprint arXiv:2407.08213* (2024).

[30] Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. 2024. RL-VLM-F: Reinforcement Learning from Vision Language Foundation Model Feedback. In *Forty-first International Conference on Machine Learning*.

[31] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. 2024. RoboGen: Towards Unleashing Infinite Data for Automated Robot Learning via Generative Simulation. In *Forty-first International Conference on Machine Learning*.

[32] Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. 2017. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research* 18, 136 (2017), 1–46.

[33] Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. 2024. Text2reward: Automated dense reward function generation for reinforcement learning. In *The Twelfth International Conference on Learning Representations*.

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

[35] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. 2024. Self-Rewarding Language Models. In *Forty-first International Conference on Machine Learning*.

[36] Yuwei Zeng, Yao Mu, and Lin Shao. 2024. Learning Reward for Robot Skills Using Large Language Models via Self-Alignment. In *Forty-first International Conference on Machine Learning*.

[37] Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, et al. 2024. Fine-Tuning Large Vision-Language Models as Decision-Making Agents via Reinforcement Learning. *arXiv preprint arXiv:2405.10292* (2024).

[38] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[39] Zhiyuan Zhang, Caixin Guo, Qichao Zhang, Xiaoxu Wu, Guogang Liao, Yongkang Wang, Xingxing Wang, and Dongbin Zhao. 2024. LLM-AC: large language models enhanced actor-critic for recommendation systems. In *The International Conference on Neural Information Processing (ICONIP)*.

[40] Yupeng Zheng, Zhongpu Xia, Qichao Zhang, Teng Zhang, Ben Lu, Xiaochuang Huo, Chao Han, Yixian Li, Mengjie Yu, Bu Jin, et al. 2024. Preliminary Investigation into Data Scaling Laws for Imitation Learning-Based End-to-End Autonomous Driving. *arXiv preprint arXiv:2412.02689* (2024).

[41] Yupeng Zheng, Zebin Xing, Qichao Zhang, Bu Jin, Pengfei Li, Yuhang Zheng, Zhongpu Xia, Kun Zhan, Xianpeng Lang, Yaran Chen, et al. 2024. PlanAgent: A Multi-modal Large Language Agent for Closed-loop Vehicle Motion Planning. *arXiv preprint arXiv:2406.01587* (2024).

[42] Banghua Zhu, Michael Jordan, and Jiantao Jiao. 2023. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *International Conference on Machine Learning*. PMLR, 43037–43067.