Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# LLM-Driven Automatic Reward Optimization for Ambient Intelligence Robots with Curriculum Interference Auto-Correction

**Pengyun Bi**

pengyun_8@163.com

Shanxi Institute of Technology

**Ruixian Ren**

Shanxi Institute of Technology

**Zhidong Zhang**

Shanxi Institute of Technology

**Yongqiang He**

Shanxi Institute of Technology

**Article**

**Keywords:** Large Language Model, Reinforcement Learning, Reward Design, Curriculum Interference Detection, Ambient Intelligence, Robot Manipulation

**Posted Date:** November 16th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-11453076/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# LLM-Driven Automatic Reward Optimization for Ambient Intelligence Robots with Curriculum Interference Auto-Correction

Pengyun Bi[1*], Ruixian Ren[1], Zhidong Zhang[1], Yongqiang He[1]

[1*]Shanxi Institute of Technology, No. 1 Xueyuan Road, Development Zone, Yangquan, 045000, Shanxi, China.

*Corresponding author(s). E-mail(s): pengyun_8@163.com;

**Abstract**

Service robots in ambient intelligence (AmI) scenarios such as smart factories and home services must rapidly adapt to diverse manipulation tasks, yet traditional reinforcement learning (RL) reward design heavily relies on domain expertise, requiring days to weeks of manual tuning. Existing LLM-assisted reward design methods assume that generated rewards can be directly applied to training environments, overlooking potential incompatibility with built-in training mechanisms such as curriculum learning. To our knowledge, this is the first study to quantify the signal interference phenomenon between LLM-generated rewards and curriculum dynamic weight adjustments—experiments reveal that curriculum modules amplify penalty weights from an initial $\mathbf{-1 \times 10^{-4}}$ to $\mathbf{-0.1}$ (a factor of 1000×), overwhelming the positive reward signals—and it proposes an automatic detection and correction mechanism based on feedback ratio anomaly detection. A unified dual-paradigm framework adapting to both ManagerBasedRL and DirectRL paradigms on the IsaacLab platform, along with a humanized interaction paradigm for non-expert users, are also developed. Systematic experiments on five robot tasks of varying difficulty, including ablation studies, 8-sample random search baselines, and multi-seed statistical tests, demonstrate: on Franka Lift the success rate increases from 0% to 84%±2% and the mean reward from 0.90 to 84.41±5.50, where simply disabling the curriculum yields a roughly 5× increase in mean reward (0.90→4.47)—a necessary prerequisite for LLM reward optimization to take effect—and combining it with the LLM-designed multi-stage reward structure multiplies the improvement by a further 18.9×; under an equal-compute random-search comparison, LLM weight optimization achieves a 47% improvement over the best random configuration. Mean-reward ratios depend on the reward definition and should be read together with the scale-invariant success rate. The method reduces reward design time from expert-days to hours, validated on the commercial LLM DeepSeek-V4-Pro.

**Keywords:** Large Language Model, Reinforcement Learning, Reward Design, Curriculum Interference Detection, Ambient Intelligence, Robot Manipulation

## 1 Introduction

Deep reinforcement learning (RL) has achieved remarkable progress in robotic manipulation, yet reward function design remains a critical bottleneck for engineering deployment Knox et al. (2023). Early work on learning reward functions

from human preferences demonstrated the feasibility of data-driven reward acquisition Christiano et al. (2017), yet automated reward design for complex manipulation tasks remains an open challenge. In ambient intelligence (AmI) scenarios such as smart factories and home services Ahmad et al. (2023), robots must rapidly adapt to diverse manipulation tasks ranging from simple pick-and-place to precision assembly, each requiring a tailored reward function to guide policy learning. However, reward design heavily depends on expert experience: designers must deeply understand task physics, reasonably decompose subgoals, carefully craft reward terms, and iteratively tune weights and thresholds. The design cycle often spans days to weeks, severely limiting the large-scale deployment of RL in real-world AmI scenarios.

The rapid advancement of large language models (LLMs) has opened new avenues for automated reward design. Eureka Ma et al. (2024) first demonstrated GPT-4's potential in generating reward code to optimize RL training; Text2Reward Xie et al. (2024) and L2R Yu et al. (2023) further explored methods for converting natural language task descriptions into reward functions. However, existing work has three key limitations.

The first limitation—and the core focus of this paper—is the compatibility issue between LLM-generated rewards and built-in training mechanisms in complex RL platforms. Existing studies all assume that LLM-generated rewards can be directly applied to training environments, without considering that built-in mechanisms such as curriculum learning—a widely adopted strategy to stabilize RL training where recent works have begun leveraging LLMs to automate curriculum design for robotic tasks Ryu et al. (2025); Liang et al. (2024), and diffusion-based methods have also been explored for curriculum generation Sayar et al. (2024)—and adaptive weight adjustment may produce unintended interactions with externally generated rewards. We quantify this issue, to our knowledge for the first time, in experiments on the IsaacLab platform Mittal et al. (2024): in the Franka Lift task, the CurriculumCfg module dynamically amplified penalty weights set by the LLM from $-1 \times 10^{-4}$ to $-0.1$ (a $1000\times$ amplification), causing a well-designed multi-stage reward to completely fail. This finding reveals that when exogenously generated rewards are injected into modular RL frameworks with built-in training strategies, severe signal interference can occur—a risk that has not been systematically studied before.

The second limitation is that existing work has only been validated on simple Gymnasium/MuJoCo/Isaac Gym environments, without systematic evaluation on industrial-grade robot simulation platforms (such as IsaacLab) that support multiple RL paradigms (ManagerBasedRL, abbreviated MBRL; and DirectRL). The third limitation is the lack of interaction design optimization for non-expert user deployment in AmI scenarios.

This paper proposes an LLM-driven automatic reward function optimization method for robot RL. The core original contribution is the curriculum interference auto-detection and correction mechanism (Section 3.2): we first quantify the compatibility issue between LLM-generated rewards and built-in curriculum learning, and design an automatic diagnosis rule based on feedback ratio anomaly detection and an automatic correction scheme. Ablation experiments show that simply disabling curriculum learning yields a roughly $5\times$ base reward improvement ($0.90 \rightarrow 4.47$), a necessary prerequisite for LLM reward optimization to take effect, while the LLM-designed multi-stage reward structure contributes a further $18.9\times$ gain ($4.47 \rightarrow 84.41$); the two factors multiply to the total improvement of approximately $94\times$. Supporting contributions include a unified dual-paradigm adaptation framework (Section 3.3) and a humanized interaction paradigm for non-expert users (Section 3.4).

## 2 Related Work

### 2.1 Robot Learning in Ambient Intelligence

The AmI concept has been extensively studied in recent literature Ahmad et al. (2023), with the core philosophy of building transparent, non-intrusive intelligent services in everyday environments. Under this framework, robots as physical agents must autonomously perceive, decide, and act in dynamic, unstructured AmI spaces Ahmad et al. (2023). Ahmad et al. Ahmad et al. (2023) identified that a core bottleneck for

service robots in AmI scenarios is "how to enable non-professional users to rapidly configure and deploy robot behavior." Our method responds to this challenge by combining LLM natural language understanding with automated RL training.

### 2.2 Humanized Robot Programming

Liekens et al. Liekens et al. (2022) surveyed recent advances in natural language instruction for industrial robots, noting that current methods mainly address deterministic task sequences and lack effective support for RL training processes that require exploration and trial-and-error. Our work fills this gap by using LLMs to convert natural language task descriptions into reward function code, and then automatically discovering optimal policies through RL training, realizing a "describe-to-train" humanized robot programming paradigm.

### 2.3 LLM-Assisted Reinforcement Learning

LLM applications in RL have attracted significant recent attention Ma et al. (2024); Xie et al. (2024); Yu et al. (2023); Hu et al. (2024), spanning reward generation, task planning Huang et al. (2022), and language-guided policy learning Zhang et al. (2025). Eureka Ma et al. (2024) first demonstrated the potential of GPT-4 for reward code generation. Text2Reward Xie et al. (2024) used LLMs to directly convert natural language into executable reward functions. L2R Yu et al. (2023) introduced template constraints on LLM output formats. Hu et al. Hu et al. (2024) analyzed the capability boundaries of language models in reward design. Curriculum learning is a widely adopted strategy to stabilize RL training, and recent works have begun leveraging LLMs to automate curriculum design for robotic tasks Ryu et al. (2025); Liang et al. (2024). However, its compatibility with LLM-generated rewards has not been previously studied—this is precisely the core entry point of this paper.

Our work extends prior research in three directions: (1) we identify, to our knowledge for the first time, and systematically mitigate the compatibility issue between LLM-generated rewards and built-in curriculum learning—a dimension not addressed by existing work; (2) we expanded validation from classical control environments to the industrial-grade IsaacLab platform covering 5 robot tasks across two RL paradigms; (3) we designed a complete natural language interaction workflow for AmI scenarios, and validated the feasibility of using a commercial LLM (DeepSeek-V4-Pro) for this task.

### 2.4 IsaacLab Platform and RL Training Paradigms

IsaacLab Mittal et al. (2024) is NVIDIA's official unified modular robot learning framework, supporting both MBRL and DirectRL paradigms. The MBRL paradigm declares rewards, observations, and termination conditions through `@configclass` configuration classes, suitable for standard manipulation tasks (Cartpole, Reach, Lift, Drawer). Its built-in CurriculumCfg module can dynamically adjust reward weights during training—it is precisely this design that leads to interaction interference between LLM-generated rewards and built-in training strategies. The DirectRL paradigm computes rewards directly in environment code, typically without built-in curriculum learning modules. Consequently, our core interference detection mechanism primarily applies to MBRL tasks, while DirectRL tasks serve to validate the generality of the dual-paradigm framework.

## 3 Method

### 3.1 Overall Framework

The proposed method consists of four modules forming a closed loop, as illustrated in Fig. 1: (1) **Humanized Input Module**: users describe tasks in natural language. (2) **LLM Reward Generation Module**: task descriptions are encoded into layered Prompts, and the LLM API (this paper uses the commercial LLM DeepSeek-V4-Pro, API model identifier: `deepseek-v4-pro`) is called to generate RewardsCfg configuration code or DirectRL `reward_scale` parameter schemes. (3) **Automatic Injection and Training Module**: regex-based matching achieves automatic configuration class code replacement or parameter-level setting, after which RSL-RL PPO training is executed and performance metrics are collected. (4) **Feedback and Diagnosis Module**: reward component distributions from

**Table 1** Core differences from representative LLM+RL methods

| Method | Platform | Paradigms | Feedback | Curriculum | Fault Diag. | Non-Expert UI |
|---|---|---|---|---|---|---|
| Eureka | Isaac Gym | Code-based | Free-text | N/A | None | No |
| Text2Reward | MuJoCo/Gym | Config-based | Single-round | N/A | None | Partial |
| L2R | ManiSkill2 | Template-based | No iteration | N/A | None | No |
| Ours | IsaacLab | Dual-paradigm | Structured | Auto-fix | Yes | Yes |

training results are analyzed, and the curriculum interference auto-detection rules (Section 3.2) diagnose potential interference sources, generating structured feedback Prompts to drive the next LLM iteration.

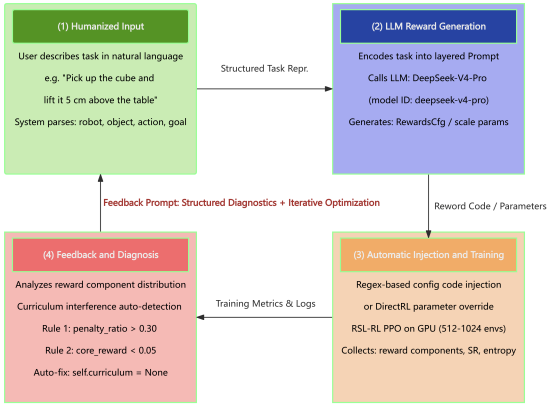



**Fig. 1** Overall framework of the proposed method.

## 3.2 Core Innovation: Curriculum Interference Auto-Detection and Correction

In industrial-grade robot RL platforms, curriculum learning mechanisms are commonly built in to enhance training stability, dynamically adjusting penalty weights during training. Reward signal failure and misalignment are long-standing challenges in RL Knox et al. (2023), and LLM-generated rewards face similar compatibility issues with built-in training mechanisms. In this phenomenon, the agent does not exploit reward loopholes opportunistically (as in traditional reward hacking Skalse et al. (2022)), but rather the environment's built-in training mechanisms themselves disrupt the expected structure of the reward signal. Curriculum learning is a widely adopted strategy, and recent works have begun leveraging LLMs to automate curriculum design Ryu et al. (2025); Liang et al. (2024). However, its compatibility with LLM-generated rewards has not been previously examined.

This paper takes the Franka Lift task as a representative case to first reveal and quantify this phenomenon, as shown in Fig. 2. In the first round, the LLM generated a well-structured 8-term multi-stage reward (Fig. 2a), yet training performance was worse than the official baseline (mean_reward 0.47 vs. baseline 0.90). Stepwise analysis revealed that the CurriculumCfg module dynamically amplified the penalty weights from the LLM-set value of $-1 \times 10^{-4}$ to $-0.1$ (a $1000\times$ weight amplification), overwhelming the positive reward signals (Fig. 2b). After disabling the curriculum module, positive reward signals immediately recovered to normal gradient levels (Fig. 2c), confirming that curriculum interference is the root cause of the failure.

Based on this discovery, we designed an automatic detection and correction mechanism. The detection algorithm is based on the IsaacLab platform's reward naming conventions and configuration class architecture: core positive reward terms are pre-defined per task type (e.g., `lift_object` for grasping tasks, `open_drawer` for drawer tasks), penalty terms are identified via `rate/vel/penalty` keyword matching, and the correction operation is implemented by disabling the curriculum module (`self.curriculum = None`) in the configuration class.

The core idea of the algorithm: by monitoring the relative proportion of positive vs. negative reward components and the absolute activation levels of key positive terms (line 5: core reward term absolute value below 0.05), the system automatically determines whether curriculum dynamic weight adjustment has produced negative interference. When both conditions are simultaneously met (penalty ratio exceeding 30% AND core positive reward activation below 0.05), the correction operation is automatically triggered.

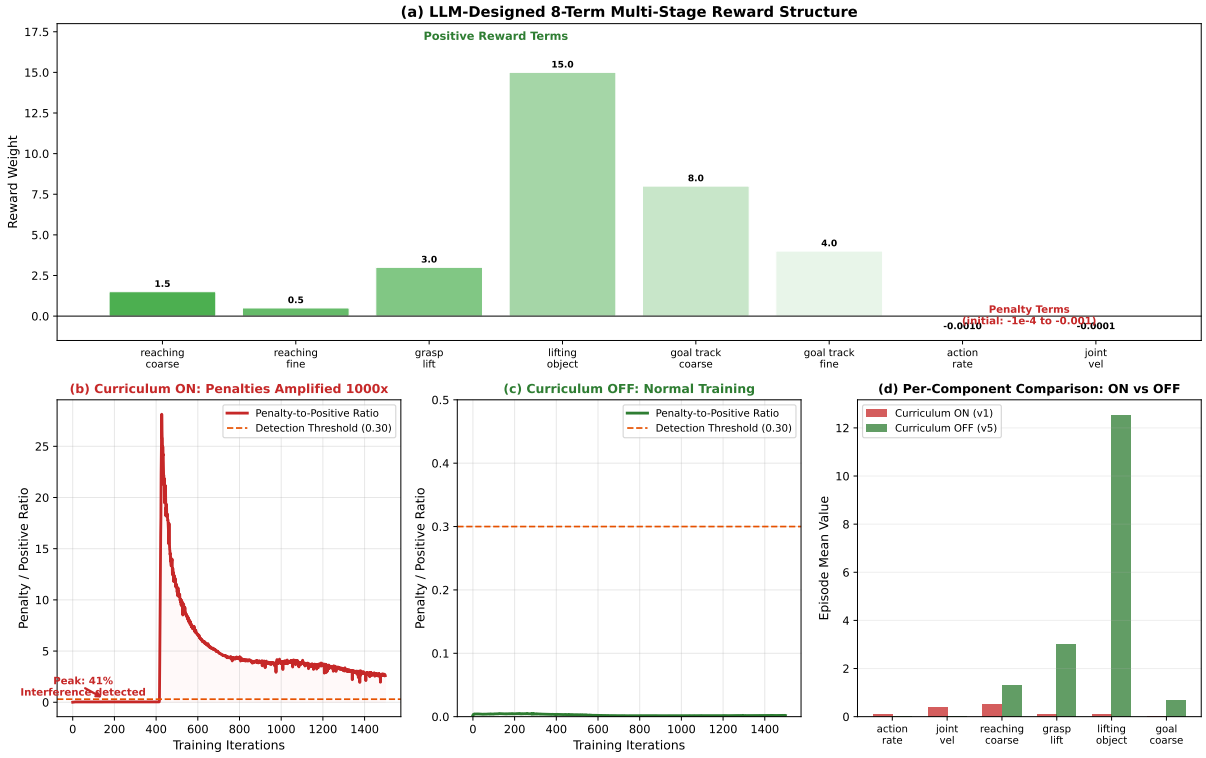



**Fig. 2** Curriculum interference phenomenon on Franka Lift.

The detection and disabling steps are fully automatic. However, initial weight magnitude calibration and training hyper-parameter decisions (such as threshold selection and iteration count extension) still require human-assisted judgment across multiple iteration rounds in the current stage. Fully end-to-end unattended automation remains an important direction for future work.

Additionally, the root cause of the v2 gradient explosion (NaN) merits analysis: the LLM excessively amplified positive weights (3×), causing the reward value scale to become too large and policy gradient variance to explode. This phenomenon precisely corroborates the necessity of multi-round feedback iteration—single-round LLM generation has inherent limitations in controlling weight magnitudes, and training feedback is needed for correction. Future work could add weight range constraints in the Prompt (e.g., "all absolute weight values must not exceed 20") to reduce such failure risks.

## 3.3 Supporting Contribution: Unified Dual-Paradigm Adaptation Framework

This paper designs two adaptation interfaces: (1) **MBRL Interface**—the LLM generates complete RewardsCfg class code, and the injector uses regex matching to locate and replace the target configuration, simultaneously integrating the curriculum interference detection logic from Section 3.2. (2) **DirectRL Interface**—the LLM generates `reward_scale` parameter adjustment suggestions, completing parameter settings via attribute assignment after environment initialization, with zero intrusion into the underlying environment code. The DirectRL task (Cabinet Direct) is used to validate the generality of the dual-paradigm framework; its performance improvement comes from scale parameter adjustment and does not involve curriculum interference repair. (3) A unified closed-loop scheduler automatically selects the code injection or parameter adjustment channel based on the task's RL paradigm.

**Algorithm 1** Curriculum Interference Auto-Detection and Correction

**Require:** Reward component dictionary $R = \{name : mean_value\}$ from training logs, environment config file path $cfg_path$

**Ensure:** Interference verdict + correction operation

$positive_sum = sum(|v| \text{ for } name, v \in R \text{ if } name \text{ does not contain } rate/vel/penalty)$

$penalty_sum = sum(|v| \text{ for } name, v \in R \text{ if } name \text{ contains } rate/vel/penalty)$

$penalty_ratio = penalty_sum / \max(positive_sum, 10^{-8})$

$core_active = \{name : |v| \text{ for } name, v \in R \text{ if } name \in task_specific_rewards\}$ ▷ platform-adapted

$low_active = [name \text{ for } name, val \in core_active \text{ if } val < 0.05]$

**if** $penalty_ratio > 0.30$ **and** $|low_active| > 0$ **then**

    Append `self.curriculum = None` in `__post_init__()` of $cfg_path$ ▷ platform-adapted

    **return** (True, "Curriculum interference detected and corrected")

**else**

    **return** (False, "No interference detected")

**end if**

### 3.4 Supporting Contribution: Humanized Interaction for Non-Expert Users

The interaction workflow is designed in three layers: (1) **Base Layer**—users input in natural language, e.g., "Have the Franka arm pick up the black cube on the table and lift it to 5 cm above the surface." (2) **Feedback Layer**—after each training round, progress is reported in natural language, e.g., "Round 1 complete: the robot can now steadily approach the cube, but grasping has not yet formed." (3) **Decision Layer**—users can inspect the LLM-generated reward code and training curves at any round and intervene with adjustments. We validated the commercial LLM DeepSeek-V4-Pro on this task, confirming that it successfully generates usable reward configurations.

## 4 Experiments

### 4.1 Experimental Setup

**Software**: IsaacLab 3.0 + NVIDIA Isaac Sim 5.1, RSL-RL PPO. **LLM backend**: DeepSeek-V4-Pro. All experimental code will be open-sourced upon acceptance.

Training hyper-parameters are shown in Table 2. Unless otherwise noted, all training runs use 3 random seeds (42/123/456), with results reported as mean ± standard deviation. The random search baseline uses an equal computational budget constraint (300 iterations × 512 environments per sample, 8 total samples).

Success criteria: (i) Cartpole: pole angle within $\pm 12^\circ$ and cart within $\pm 2.4$ m for a full 300-step episode. (ii) Franka Reach: end-effector reaches within 0.05 m of the target and maintains stability. (iii) Franka Lift: cube is stably grasped with its bottom exceeding 0.03 m above the table surface and maintained for over 0.5 s (approximately 30 environment steps); success_rate is the proportion of successes over 100 test episodes. (iv) Franka Open Drawer: drawer joint displacement exceeds 0.30 m. (v) Franka Cabinet Direct: drawer joint displacement exceeds 0.30 m.

### 4.2 Task Selection

### 4.3 Main Experimental Results

*Notes*: (1) Franka Lift SR=84% is the mean over 100 test episodes across 3 seeds. (2) Cabinet Direct belongs to the DirectRL paradigm and does not involve curriculum interference repair; its optimization validates the generality of the dual-paradigm framework, with performance improvement coming from `reward_scale` parameter adjustment. (3) The Cabinet Direct baseline is near-complete failure, so its improvement is reported conceptually rather than as a multiplier (marked $^\dagger$); it reflects a transition from non-functional to functional behaviour, not an absolute superiority claim. (4) Mean-reward ratios (e.g. the 94× figure for Franka Lift) compare means computed under different reward definitions; they quantify the scale of the observed improvement within this setup and are not directly comparable across tasks or reward terms. The scale-invariant success rate is the primary evidence. (5) For the Cabinet Direct

**Table 2** PPO training hyper-parameters for each task

| Task | PPO Params (steps/lr/entropy) | Num Envs | Max Iters | Network |
|---|---|---|---|---|
| Cartpole | 16 / $10^{-3}$ / 0.005 | 1024 | 300 | [32,32] |
| Franka Reach | 24 / $10^{-3}$ / 0.001 | 1024 | 300 | [64,64] |
| Franka Lift | 24 / $10^{-4}$ / 0.006 | 1024 | 800–1500 | [256,128,64] |
| Franka Drawer | 96 / $5 \times 10^{-4}$ / $10^{-3}$ | 512 | 400 | [256,128,64] |
| Franka Cabinet Dir. | 16 / $5 \times 10^{-4}$ / 0.005 | 512 | 800 | [256,128,64] |

**Table 3** Task overview (* marks the core analysis task involving curriculum interference detection)

| Task | Task ID | Category | Difficulty | Obs/Act | Paradigm | Baseline |
|---|---|---|---|---|---|---|
| Cartpole | Isaac-Cartpole-v0 | Classic Control | Easy | 4D/1D | MBRL | Converged |
| Franka Reach | Isaac-Reach-Franka-v0 | End-Effector | Easy | 23D/9D | MBRL | Converged |
| Franka Lift* | Isaac-Lift-Cube-Franka-v0 | Grasp & Lift | Hard | ∼16D/9D | MBRL | Failed |
| Franka Drawer | Isaac-Open-Drawer-Franka-v0 | Drawer Opening | Hard | ∼25D/9D | MBRL | Converged |
| Franka Cab. Dir. | Isaac-Franka-Cabinet-Direct-v0 | Cabinet Drawer | Very Hard | ∼40D/14D | DirectRL | Failed |

task, `open` denotes the weighted drawer-opening reward term (the reward coefficient multiplied by the drawer joint displacement, averaged over test episodes), not an opening angle; the corresponding success criterion is the drawer joint displacement of 0.30 m defined in Section 4.1.

The Franka Lift training curves (Fig. 3) clearly show the effect: the LLM-optimized policy (blue) begins to significantly surpass the baseline (red) at approximately 600 iterations, with the shaded band indicating mean±std across 3 seeds. This validates the synergistic effect of the LLM-designed multi-stage reward structure combined with curriculum interference repair.

## 4.4 Core Ablation: Independent Contribution of the Curriculum Interference Detection Mechanism

**Key findings**: (1) Comparing A and B: merely disabling curriculum raised reward from 0.90 to 4.47 (5.0×), but was insufficient for policy convergence. (2) Comparing C and A: the LLM-designed structure under curriculum suppression performed even worse than the official config (0.47 < 0.90). (3) Comparing F and D: even with v5 optimized weights (action_rate=−0.001, joint_vel=−0.0001), curriculum still suppressed reward from 84.41 down to 29.5—a performance loss of approximately 65%. This indicates that curriculum systematically inhibits positive reward signals regardless of weight quality. (4) Config F reveals that curriculum interference is independent of weight quality—even an already-optimized reward structure still suffers severe degradation under curriculum activation. Core quantitative comparisons are based on configs A/B/D/E.

*Notes*: Config C serves as a qualitative diagnostic experiment to verify the existence of the "LLM-optimized structure deteriorates under curriculum interference" phenomenon. Config E's detector triggered curriculum disabling after v1 training; subsequent versions share the same configuration as D. Its core contribution lies in eliminating the need for manual investigation. Config F (29.5) is an independent experiment with v5 weights + curriculum (2 seeds), conducted specifically to verify curriculum interference on already-optimized weights.

Fig. 4 presents the ablation results as a bar chart. The mean reward contrast across configurations A–E visually confirms: disabling curriculum alone (A→B) yields a 5× gain; the LLM reward structure under curriculum (C) performs worse than official; and the full combination of LLM structure with curriculum disabled (D/E) achieves the 94× improvement. Expressed as a product, the two effects account for the total gain: 5.0× (A→B) × 18.9× (B→D) ≈ 94×.

## 4.5 Baseline Comparison: LLM vs. Random Search

Using the LLM-designed multi-stage reward structure (8 terms), 8 independent weight configurations were randomly sampled and trained (300

**Table 4** Main experimental results (3 seeds mean±std)

| Task | Baseline | LLM-Optimized | Gain | Remarks |
|---|---|---|---|---|
| Cartpole | SR=100%, R=4.95 | SR=100%, R=4.95 | — | LLM confirmed optimal |
| Franka Reach | SR=100%, R=−0.06 | SR=100%, R=−0.06 | — | LLM confirmed optimal |
| Franka Lift | SR=0%, R=0.90 | SR=84%, R=84.41±5.5 | ≈94× | Interference detection + multi-round LLM |
| Franka Drawer | SR=100%, R=98.99 | SR=100%, R=98.99 | — | LLM confirmed optimal |
| Franka Cab. Dir. | open=0.0007, R=1384 | open=6.09, R=2027 | n/a† | DirectRL scale adj. |

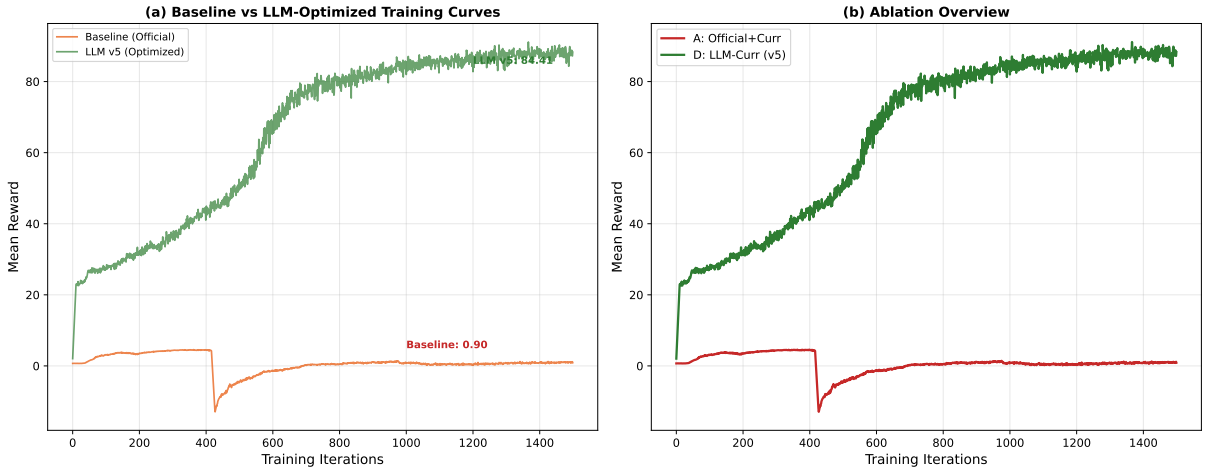



**Fig. 3** Franka Lift training curves (3 seeds, mean±std shaded).

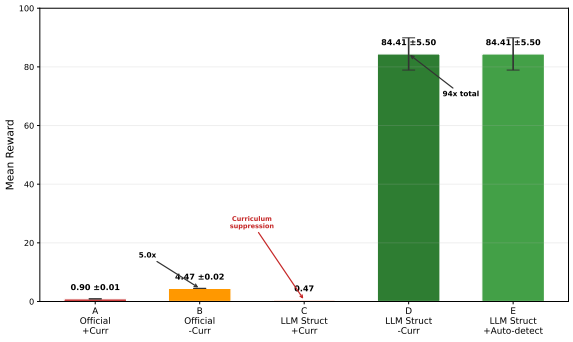



**Fig. 4** Ablation experiment results (configs A–E, Franka Lift).

iterations × 512 environments per sample), compared against LLM multi-round iterative weight optimization.

The LLM-designed multi-stage reward structure has independent value, as shown in Fig. 5: even with randomized weights, the best random sample (57.37) far exceeds the official baseline (0.90). Subfigure (a) compares 8 random weight samples against the LLM-optimized result (dashed green) and random search mean (dashed orange); subfigure (b) summarizes best random, average random, LLM-optimized, and official baseline. LLM weight optimization on the same structure provides an additional improvement of approximately 47% (84.41/57.37 ≈ 1.47).

## 4.6 Franka Lift Task Iteration Process

*Note*: The v2 NaN gradient explosion originated from the LLM excessively amplifying positive weights, causing reward value scale to become too large. This corroborates the necessity of multi-round feedback iteration. v4 was an intermediate hyper-parameter tuning version (grid search over thresholds 0.015→0.02→0.015, and iteration counts 500→800→1000); the core optimization effects are concentrated in the v5 version.

**Table 5** Ablation study of curriculum interference detection (Franka Lift, 3 seeds)

| Config | Mean Reward | grasp_lift/lifting | SR% | Description |
|---|---|---|---|---|
| A (Official+curr) | 0.90±0.01 | —/0.12±0.01 | 0% | Official config, complete failure |
| B (Official−curr) | 4.47±0.02 | —/∼0.50 | 0% | Curriculum disabled only, limited |
| C (LLM struct+curr) | 0.47 | 0.12/0.10 | 0% | LLM structure suppressed by curriculum |
| D (LLM struct−curr) | 84.41±5.50 | 3.0/11.68±1.22 | 84%±2% | LLM structure + curriculum disabled |
| E (LLM+Auto-detect) | 84.41±5.50 | 3.0/11.68±1.22 | 84%±2% | Detector auto-triggered correction |
| F (LLM opt wts+curr) | 29.5 (2 seeds) | 2.96/0.14 | 0% | v5 weights still suppressed by curr. |

**Table 6** LLM weight optimization vs. random search (same 8-term reward structure, equal computational budget)

| Method | Best Reward | Avg Reward | Min Reward | Valid/Total |
|---|---|---|---|---|
| Random Search (LLM struct) | 57.37 | 43.77 | 24.48 | 8/8 |
| LLM Iter. Optim. (LLM struct) | 84.41±5.50 | — | — | 3/3 |

# 5 Discussion

## 5.1 Generalizability of the Curriculum Interference Detection Framework

The contribution of the proposed interference detection framework extends beyond the performance improvement of a single task. This mechanism reveals a category of risk not previously systematically studied: when exogenously generated rewards are injected into modular RL frameworks containing built-in training strategies, unintended interactive interference may arise. The core idea of our detection framework—automatic monitoring based on the relative proportion of positive/negative reward components and the activation status of key terms—is not limited to curriculum learning interference. It can be extended to compatibility detection between LLM-generated rewards and other built-in mechanisms such as domain randomization scheduling, adaptive annealing, and reward dynamic scaling, providing a general risk-screening approach for automated reward design in complex RL platforms. The current thresholds (30% and 5%) are empirical values from IsaacLab experiments; upgrading these thresholds from fixed values to adaptive decision rules is an important direction for future improvement.

## 5.2 Implications for Humanized Robot Deployment

Taking smart factory flexible production lines as a scenario, when the production line needs to switch products, operators can describe new process requirements in natural language, and the system automatically completes the entire process from reward design to policy training. The presence of the automatic interference detection mechanism makes this workflow more robust—non-expert users need not understand the technical pitfalls of RL training; the system automatically identifies and corrects them.

## 5.3 Limitations and Future Work

(1) **Automation boundary**: interference detection and disabling can be fully automated, but weight magnitude calibration and hyperparameter decisions still require human assistance. Fully end-to-end unattended automation is an important future direction. (2) **Threshold generalizability**: the current 30%/5% thresholds are empirical values; cross-platform validity requires verification, and upgrading to adaptive decision rules is a key improvement direction. (3) **Cross-platform generalization**: currently only validated on IsaacLab; evaluation on MuJoCo and other platforms is needed. (4) **Visual RL extension**: only low-dimensional state observations have been validated; extension to visual observation-based RL tasks is needed. (5) **Sim-to-real validation**: experiments were conducted entirely in simulation; transfer to physical robots requires verification. (6) **Open-source model exploration**: future work will validate the applicability of truly open-source models such as DeepSeek-V3-Lite on this task. (7) **Scope of the claims**: all results are obtained in a single simulator (IsaacLab 3.0 / Isaac Sim 5.1) with a single

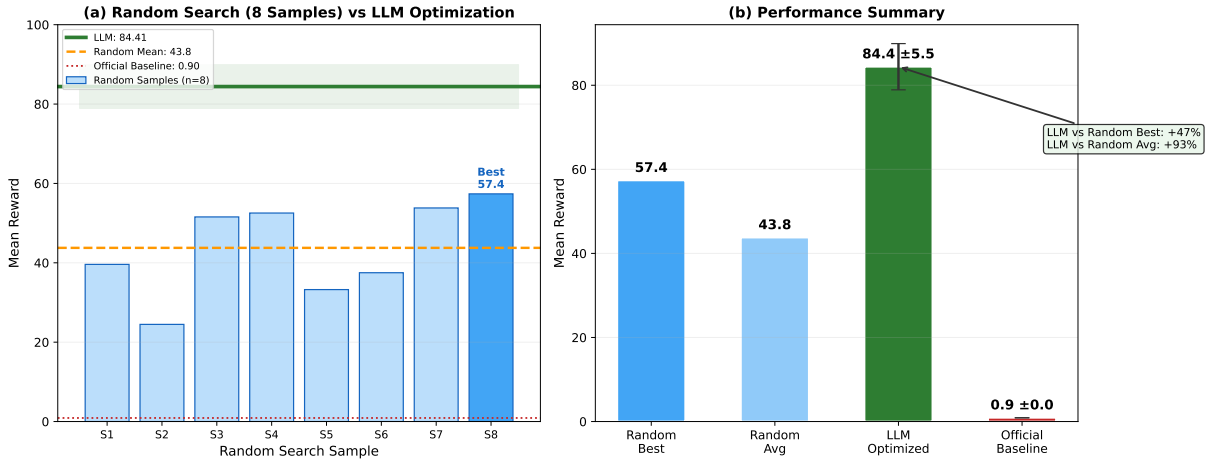



**Fig. 5** Random search baseline comparison.

**Table 7** Franka Lift iteration process

| Round | Action | Mean Reward | grasp_lift | lifting | Auto-detect? | Human? |
|---|---|---|---|---|---|---|
| Baseline | Official 6-term + curr. | 0.90 | N/A | 0.12 | — | — |
| LLM v1 | LLM: 8-term multi-stage | 0.47 | — | — | Triggered | No |
| LLM v2 | 3× weight amp. | NaN | — | — | — | No |
| v3 | Ctrl weight magnitude | 26.89 | 3.0 | 0.10 | — | Yes |
| v5 | Lower threshold, 1500 it. | 84.41±5.5 | 3.0 | 11.68±1.22 | — | Yes |

commercial LLM backend, three random seeds per configuration (two for config F and one for the qualitative diagnostic config C), and detector thresholds (30%/5%) tuned on the five tasks studied here. The reported gains therefore characterize this setting and are not established as general scaling laws for LLM-generated rewards.

# 6 Conclusion

The core original contribution of this paper is, to our knowledge, the first quantification and systematic characterization of the signal interference problem between LLM-generated rewards and built-in curriculum learning, along with the design of an automatic detection and correction mechanism. Supporting contributions include a unified dual-paradigm adaptation framework for IsaacLab and a humanized interaction paradigm for non-expert users. Systematic experiments on five robot tasks demonstrate: on Franka Lift the success rate rises from 0% to 84%, while simply disabling curriculum learning yields a roughly 5× increase in mean reward and the LLM-designed reward structure contributes a further 18.9×, giving a total mean-reward improvement of approximately 94×; LLM optimization achieves a 47% gain over the best random search configuration. The method reduces reward design time from expert-days to hours, and the interference detection framework can be generalized to compatibility detection with other built-in training mechanisms, providing a general risk-screening approach for automated reward design in complex RL platforms.


# Acknowledgments

This research received no external funding support.


# Declarations

## Funding


The authors did not receive support from any organization for the submitted work.


## Conflict of Interest

The authors declare no conflict of interest related to this work.

## Data Availability

Experimental data, including complete training logs and model checkpoints, are available upon reasonable request to the corresponding author.

## Code Availability

The core code of this method (including the curriculum interference detector, dual-paradigm adaptation framework, and experimental scripts) will be open-sourced via GitHub upon acceptance.

## Ethics Approval

This research does not involve human or animal subjects and does not require ethics approval.

# References


Ahmad MS, Salloum SA, Hishan SS, et al (2023) Human–robot interaction in ambient intelligence: a systematic review and future directions. Journal of Ambient Intelligence and Humanized Computing 14(12):16897–16921

Christiano PF, Leike J, Brown TB, et al (2017) Deep reinforcement learning from human preferences. In: Advances in Neural Information Processing Systems 30 (NeurIPS)

Hu Y, Xie A, Abbeel P, et al (2024) Reward design with language models. In: Proceedings of the 12th International Conference on Learning Representations (ICLR), Vienna

Huang W, Abbeel P, Pathak D, et al (2022) Language models as zero-shot planners: extracting actionable knowledge for embodied agents. In: Proceedings of the 39th International Conference on Machine Learning (ICML). PMLR, pp 9118–9147

Knox WB, Allievi A, Banzhaf H, et al (2023) Reward (mis)design for autonomous driving. Artificial Intelligence 316:103858

Liang W, Wang S, Wang HJ, et al (2024) Environment curriculum generation via large language models. In: Proceedings of the 8th Conference on Robot Learning (CoRL), Munich

Liekens A, De Laet T, Bruyninckx H (2022) Natural language instruction for industrial robot programming: a survey. Robotics and Computer-Integrated Manufacturing 77:102356

Ma YJ, Liang W, Wang G, et al (2024) Eureka: human-level reward design via coding large language models. In: Proceedings of the 12th International Conference on Learning Representations (ICLR), Vienna

Mittal M, Yu C, Li S, et al (2024) Isaac Lab: a unified modular framework for robot learning. arXiv preprint arXiv:2404.15243

Ryu K, Liao Q, Li Z, et al (2025) CurricuLLM: automatic task curricula design for learning complex robot skills using large language models. In: 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, pp 4470–4477

Sayar E, Iacca G, Oguz OS, et al (2024) Diffusion-based curriculum reinforcement learning. In: Advances in Neural Information Processing Systems 37 (NeurIPS), pp 97587–97617

Skalse J, Abate A, Gustafsson J, et al (2022) A taxonomy of reward hacking. arXiv preprint arXiv:2209.13085

Xie T, Zhao S, Wu CH, et al (2024) Text2Reward: automated dense reward function generation for reinforcement learning. In: Proceedings of the 12th International Conference on Learning Representations (ICLR), Vienna

Yu W, Gileadi N, Fu C, et al (2023) Language to rewards for robotic skill synthesis. In: Proceedings of the 7th Conference on Robot Learning (CoRL), Atlanta

Zhang J, Shen J, Zhang J, et al (2025) ReWind: language-guided rewards teach robot policies without new demonstrations. arXiv preprint arXiv:2505.10911