

# Discovery of the Reward Function for Embodied Reinforcement Learning Agents

Corresponding Author: Professor Hai-Tao Zhang

**This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.**

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

Discovering reward functions, whether that be ones which encode a task given observed behaviors (as is typical in IRL), which is more efficiently optimized than a given (perhaps sparse) target reward function, or which encodes an unknown preference relation (as is often the case for RLHF) is a crucial and active area of research underlying reinforcement learning.

This paper approaches a version of this problem by alternately optimizing the estimated reward function with respect to how it affects policy learning and optimizing the policy through RL with respect to the estimated reward. This general pattern is fairly common throughout the field of IRL, with specific forms of each optimization problem varying greatly. The overall presentation of the work is excellent. However, I do have concerns about some of the language used which edges into making unsubstantiated claims as well as about the rigor and precision of the theoretical components of the work.

# General Comments / Questions

In what way do you view this work as being specific to, or even especially suited to “embodied” RL versus the general RL problem? The narrative of the work heavily emphasizes the embodied component, but the method itself does not appear to be influenced by embodiment.

There are a lot of statements made throughout the paper that I believe make unsubstantiated claims. Generally, I think these are simply instances of using overly enthusiastic language and could be easily edited to fix:

\* Second sentence in abstract: Although I agree with the underlying thesis expressed here, “universal” is a significant overstatement especially when paired with “necessary and sufficient”. A proper framing would be that this is a broadly shared or at least investigated hypothesis, but certainly not “universally recognized.”

\* Page 3, first line: “Consequently, the primate brain must embody diverse types of cognitive intelligence that facilitate optimal reward acquisition.” Why the primate brain specifically? Why “must” and “optimal”, as biological life may well have found heuristic or suboptimal approaches.

\* Page 3, near top: “This process, gives the robot agent capabilities that approximate and surpass those of natural intelligence.” Ideally soften the statement, but ideally would provide references for this claim.

\* “Powerful embodied RL agents could provide a solution to AGI.” Sure, maybe, but this is just speculation without any provided support.

\* “Furthermore, the fundamental principles underlying optimal reward functions and the tasks remain largely unexplored, often requiring a case-by-case approach.” There is a rich literature on reward design and related topics, far from an unexplored topic. However, I find this line more pernicious in that this paper does not compare with or even discuss connections to even a single other method for reward design/discovery. Thus, this line seems to attempt to justify this omission by claiming the non-existence of related work.

\* Page 5, Framework Overview: “This alternating optimization process is iteratively performed until convergence, at which

point the optimal reward function is discovered.” Care is needed here as you have neither proven that the process will converge nor that if it does so it must converge on the optimal reward function.

## # Empirical Results

Figure 4 does not seem to add much to understanding. It is referenced once, as showing the learned policies are more efficient / effective, but we already know that e.g. in Lunar Lander the agent does better due to looking at the learning curve. A much better use of this figure space would be investigating more detail on the discovered reward function itself.

As mentioned previously, I believe these results need to include comparisons with other reward discovery methods, ideally 2 or more with one more closely related to the approach taken here and one more different.

Despite this issue, the empirical results are the strongest part of the paper. There is a solid set of environments, many of them well studied, exhibiting a variety of properties and levels of difficulty. As well the authors do well at varying the baseline RL algorithm itself, which shows generality of the method. And finally of course, the empirical results do generally support this method as being effective at improving learning performance.

## # Methods and Theory

Definition 1 (proximal) is not defined precisely enough as hidden inside of ‘nearly equal’ there is both (1) a choice of tolerance  $\epsilon > 0$ , and (2) a choice of distance metric (e.g.  $L_{\infty}$ ). This definition is essentially saying that when the term ‘proximal’ is used it is used to refer to the identified equivalence classes on policies or reward functions in terms of expected returns. Noticing this, it is worth pointing out that the  $\sim$  is a more common choice for notation on equivalence classes than  $\backslash \text{approx}$ . Similarly, we might naturally consider  $V^{\pi}_{\pi^*} R$  or  $v^{\pi}_{\pi^*} R$  instead of  $U(\pi, M[R])$ .

I’ll use this more standard notation for ease of typing comments out.

Proposition 1. Given a reward function  $R$  and the optimal reward function  $R^*$ , if  $R$  is proximal to  $R^*$ , then policy  $\pi$  under  $R$  will be proximal to the optimal policy  $\pi^*$ .

This proposition also has immediate issues with precision. Breaking it down:

\*  $R^*$  is “optimal” with respect to what notion of optimality? It seems instead the term optimal is simply being used to identify a reward as the ‘target’ for reward discovery.

\*  $R$  is proximal to  $R^*$ , but under what policy? Given there are two policies mentioned, I’ll consider if the theory works for either one, but obviously this needs to be specified for this proposition to be complete.

\* “policy  $\pi$  under  $R$  will be proximal to the optimal policy  $\pi^*$ ”, this is not clearly phrased and can have many meanings. Are we saying the two policies are proximal wrt  $R$ ? Or are we saying that  $\pi$  is optimal wrt  $R$  and the two policies are proximal wrt  $R^*$ ?

Just after the statement of Proposition 1 there is a fairly strong claim given without proof: “if the policy  $\pi$  optimized under a specific reward function  $R$  is proximal to the optimal policy  $\pi^*$ , then the reward function  $R$  can be considered the optimal reward function  $R^*$ .”

Which reward function are the two policies proximal wrt? I think the one that makes most sense here is  $R$ , so going forward with that the claim is essentially:

$$\pi \sim \pi^* \text{ wrt } R \Rightarrow R \backslash \text{equiv } R^*$$

First, “can be considered” needs to be specified formally. I think a natural choice would be for this to mean  $R \sim R^* \text{ wrt } \pi^*$ , as this would be fairly nice. However, none of this is true which can be seen with a simple counter example:  $R = R_0 = 0$  (the constant zero reward function).

So, we choose  $R = R_0$ , and for any policy  $\pi$  we have  $\pi \sim \pi^* \text{ wrt } R$  because all policies have value zero under this reward and as a result all policies are optimal wrt  $R$ , thus  $\pi$  being optimized under  $R$  is satisfied as is the proximal property. However, under any reasonable notion of equivalence  $R_0$  should not be considered the optimal reward function  $R^*$ .

Now turning to regret and Definition 2: “A reward function  $R$  is defined as the optimal reward function  $R^*$  if it minimizes the expected discounted regret  $L(\pi, R)$  over the space of possible policies  $\pi$ , where the policy  $\pi$  is learned by an embodied RL agent interacting with the world model  $M[R]$ .” With  $L$  defined above with:

$$L(\pi, R) = V^{\pi^*}_{\pi^*} R - V^{\pi}_{\pi} R$$

Recall the issue raised previously,  $R_0$  perfectly achieves zero regret here. However, this isn’t the whole of the problem.

Notice that  $\pi^*$  is some policy unrelated to a particular choice of  $R$ . So, there will be arbitrarily many reward functions  $R$ , with corresponding 'optimized' policies  $\pi$  such that  $L(\pi, R) \ll 0$ , whereas for the desired reward  $R^*$  we have  $L(\pi, R^*) = 0$  (for  $\pi$  optimized for  $R = R^*$  as the definition calls for). This objective, or definition for the optimal reward problem, appears to be misspecified at a foundational level.

A small notational note: it might make sense to use  $L_{\pi^*}$  to indicate the dependence on the given optimal policy.

The formal statement of Lemma 1 contains much less information, far less than needed for a precise statement, than the informal description just below it. Additionally, this lemma is entirely unnecessary.  $\bar{R}$  is a fixed reward function,  $\pi^*$  is a fixed policy, therefore the value of  $\pi^*$  under  $\bar{R}$  will be fixed, which is all this lemma seems to be saying.

Moving on to the proof of Proposition 1.

The first line of the proof indicates that the intended meaning is  $R \sim R^*$  wrt  $\pi$ . It then goes on to claim that this implies

$$|v^{\pi}_R - v^{\pi}_{R^*}| \leq \delta$$

This should be in Definition 1. Equations 22 and 23 both assume a unique optimal policy which is not in general true, so the equals should be replaced with "in" as the argmax may return a set of optimal policies.

The inequalities in 24 are very strange. First, it seems to assume  $R \sim R^*$  wrt  $\pi_R$  and  $R \sim R^*$  wrt  $\pi^*$ . Second, if a suitably formal definition was used then the corresponding thresholds  $\delta > 0$ , would potentially be different for the two equivalence classes. Ignoring that for the moment, as of (24) what has been assumed without being clear about it is that  $R \sim R^*$  wrt both  $\pi_R$  and  $\pi^*$ .

The rest of the proof follows correctly, with one partial exception. Recall my concerns about precision of the definition of proximal. Here this comes up again because the result of the proof depends on this lack of precision. Taking  $\epsilon = 4\delta$  and saying this means that  $\pi \sim \pi^*$  wrt  $R$ , means that proximal has its meaning further eroded.

Since the definition of proximal you give, and actively rely on for your proof, allows for the distance between value functions to be arbitrarily large (as large as is needed for the narrative) it makes the result much less meaningful.

What do you \*actually\* care about saying here? For instance, why do we care if the discovered reward function is close (proximal) to the optimal one? Does this result support that goal?

To be clear, I think that the theory components can be fixed, but I am unsure if they will give interesting results. I strongly suspect that they can, but not without some work.

(Remarks on code availability)

I have looked at the code fairly briefly. It is built on top of CleanRL to handle the agent-environment interaction and RL optimization components. The reward function learning code is undocumented but it is well formatted and easy to read and understand even without further documentation or comments. Overall, I would say that the code is well done because it is very easy to contrast with the baseline CleanRL implementation to see differences (which are almost entirely modular).

Yes, there is a README that should be sufficient for testing out the code. There is an odd pattern of 'typo' like misspellings of words, but done consistently (e.g. reward\_funciton, optimaze\_reward), which is worth fixing.

Reviewer #2

(Remarks to the Author)

> Summary

For embodied agents in Embodied AI (EAI), reinforcement learning (RL) is a promising method for realizing intelligent behavior to solve complex problems. However, RL suffers from the so-called reward-engineering problem, in which reward quality heavily depends on human expertise and is critical for improving agent performance. In this paper, the authors propose an optimal reward-function discovery method. Using a meta-gradient approach, they implement the reward discovery process as bilevel optimization: the upper level refines the reward function, and the lower level performs the traditional RL step. With the proposed method, agents discovered the optimal policy in fewer episodes, indicating high sample efficiency. The method worked well on both energy-management and UAV-control tasks. The authors also provide theoretical results for the proposed approach.

> What are the noteworthy results?

The authors propose a meta-gradient-based bilevel optimization method to optimize the reward function. While traditional approaches rely on inverse RL or active learning, the proposed method operates without them. It generates a more informative dense reward signal than the inefficient sparse rewards of manually designed functions, thereby accelerating the agents' learning speed and improving their performance.

> Will the work be of significance to the field and related fields? How does it compare to the established literature? If the work is not original, please provide relevant references.

Their proposed method is novel, but it closely resembles [14] or [39] (in the appendix). Although the implementation and design differ, the core idea remains very similar.

> Does the work support the conclusions and claims, or is additional evidence needed?

Considering the previous comment, comparisons with the methods in [14] or [39] would clarify that the authors' proposed approach is more effective than theirs; the work currently only compares against naive RL baselines. It would also be preferable to include visualizations of the learned reward to assess what reward was actually learned. While this may be difficult for high-dimensional control tasks, it could at least be done for classical control tasks.

> Are there any flaws in the data analysis, interpretation and conclusions? Do these prohibit publication or require revision?

In the energy-management task, the consumption reduction achieved by SAC and PPO appears smaller than that of DQN (Fig. 7(c)). In the UAV control task, Fig. 8(b) does not clearly demonstrate an acceleration in learning: while PPO shows noticeable improvement, the gains for TD3 and SAC are comparatively modest. Moreover, the advantage illustrated in Fig. 8(e) over Fig. 8(d) is not immediately apparent. It would be helpful to clarify the form of the reward function that was learned, as this would shed light on how the proposed method drives performance improvements. Finally, the t-SNE visualizations are somewhat challenging to interpret; additional explanation or annotation could improve their clarity.

> Is the methodology sound? Does the work meet the expected standards in your field?

The methodology is fundamentally sound; however, a more detailed comparison with state-of-the-art approaches would better highlight the advantages of the proposed method. I therefore recommend conducting additional, comprehensive experiments to this end.

> Is there enough detail provided in the methods for the work to be reproduced?

The methods as described are sufficient for reproduction; however, we will further strengthen the manuscript by including additional implementation details to support a more thorough evaluation.

(Remarks on code availability)

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

The author response and changes made address my two primary concerns with the work (limited empirical comparison with other methods and precision of, especially theoretical, claims). The updated language in the article, improved theory statements and inclusion of additional experimental work satisfies these and significantly strengthen the work.

My only remaining, perhaps pedantic, suggestion has to do with the framing of this work and its dependence on knowledge of the task relative to existing methods. The authors repeatedly distinguish their approach from others in terms of dependency on "external knowledge", e.g.

"In contrast, our approach aims to discover reward functions in a fully autonomous manner, guided solely by environmental interactions and generalization signals, without relying on predefined expert trajectories, heuristics, or task-specific priors."

This seems to claim that the proposed method does not depend on any prior information about the "task", which obviously is not correct. There is indeed information about the task, just as in other methods, but this method and others certainly differ in the type and form of that "external knowledge". The type of dependency in this article is most closely related to those such as LIRBO and SASR, making them great choices for comparison in the experiments.

I think it is important to be more clear that this method does indeed require some such knowledge, in what form, and then to contrast with other in these terms. The argument cannot be that this approach requires no knowledge of the task, instead of must be that the type of dependency and type of knowledge here is preferable (e.g. sparse reward or fitness vs. expert demonstrations). This detail is decipherable from the methods section, but in the main the claims do not match this detail, and clarity would be much improved on this point by simply stating the dependency and comparison upfront.

(Remarks on code availability)

Reviewer #2

(Remarks to the Author)

I have confirmed the authors' reply and have no further questions.

The authors have added comparative experiments with existing methods, and it is evident that the performance of the proposed method is comparable to or better than the existing ones.

The visualizations have also been improved, making it easier for readers to understand.

(Remarks on code availability)

**Open Access** This Peer Review File is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

In cases where reviewers are anonymous, credit should be given to 'Anonymous Referee' and the source.

The images or other third party material in this Peer Review File are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>

## Response to Reviewer #1 (Remarks to the Author):

Discovering reward functions, whether that be ones which encode a task given observed behaviors (as is typical in IRL), which is more efficiently optimized than a given (perhaps sparse) target reward function, or which encodes an unknown preference relation (as is often the case for RLHF) is a crucial and active area of research underlying reinforcement learning.

This paper approaches a version of this problem by alternately optimizing the estimated reward function with respect to how it affects policy learning and optimizing the policy through RL with respect to the estimated reward. This general pattern is fairly common throughout the field of IRL, with specific forms of each optimization problem varying greatly. The overall presentation of the work is excellent. However, I do have concerns about some of the language used which edges into making unsubstantiated claims as well as about the rigor and precision of the theoretical components of the work.

**Response:** Thanks for the reviewer's kind efforts to improve our manuscript. We truly appreciate the reviewer's overall positive feedback and constructive comments, which help us to refine the manuscript. In response, we have addressed all the issues raised and highlighted the changes in yellow in the revised version. We hope that the revised manuscript could meet reviewer's expectations for high publication criteria in *Nature Communications*. Below, we respond to the comments one-by-one.

### # General Comments / Questions

In what way do you view this work as being specific to, or even especially suited to “embodied” RL versus the general RL problem? The narrative of the work heavily emphasizes the embodied component, but the method itself does not appear to be influenced by embodiment.

**Response:** Thanks for the insightful comment. As is well known, *General RL* is a machine learning algorithm in which an agent seeks to make decisions by interacting with an uncertain environment, so as to maximize the cumulative rewards [1,2]. Therein, the term *Embodied RL* refers to the particular RL tasks during which agents physically interact with more complex high-dimensional environments. Agent actions are often governed by underlying dynamics and constraints [3,4]. For example, our case studies include a range of benchmark tasks widely recognized in the embodied RL community [5,6]. MuJoCo features articulated agents with joints that engage in physical interactions. Unmanned systems control tasks require that agents address the physical dynamics and spatial navigation. Agents are required to learn to control their bodies or movements in a physically plausible manner, and the present study is designed to discover reward functions that more effectively capture these complex relationships than do hand-crafted rewards.

In general, embodied RL is in fact a subset of general RL. Advances in embodied RL naturally contribute to the broader RL field. While our method is developed with embodied scenarios in mind, the insights and techniques will also inform and benefit the general RL community.

- [1] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. *Nature* 518(7540), 529–533 (2015).
- [2] Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., et al.: Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature* 575(7782), 350–354 (2019).
- [3] Howard, D., Eiben, A.E., Kennedy, D.F., Mouret, J.-B., Valencia, P., Winkler, D.: Evolving embodied intelligence from materials to machines. *Nature Machine Intelligence* 1(1), 12–19 (2019).
- [4] Bartolozzi, C., Indiveri, G., Donati, E.: Embodied neuromorphic intelligence. *Nature Communications* 13(1), 1024 (2022).
- [5] Nygaard, T.F., Martin, C.P., Torresen, J., Glette, K., Howard, D.: Real-world embodied ai through a morphologically adaptive quadruped robot. *Nature Machine Intelligence* 3(5), 410–419 (2021).
- [6] Gupta, A., Savarese, S., Ganguli, S., Fei-Fei, L.: Embodied intelligence via learning and evolution. *Nature Communications* 12(1), 5721 (2021).

There are a lot of statements made throughout the paper that I believe make unsubstantiated claims. Generally, I think these are simply instances of using overly enthusiastic language and could be easily edited to fix:

**Response:** Many thanks for highlighting the presence of unsubstantiated or overly enthusiastic claims in our original manuscript. We have reviewed the whole paper and identified all instances where the language may have appeared overly enthusiastic or insufficiently supported by evidence. Such statements have been revised to ensure a neutral, precise, and evidence-based tone throughout the text. In response to the reviewer’s concerns, we address the specific instances in detail below.

\* Second sentence in abstract: Although I agree with the underlying thesis expressed here, “universal” is a significant overstatement especially when paired with “necessary and sufficient”. A proper framing would be that this is a broadly shared or at least investigated hypothesis, but certainly not “universally recognized.”

**Response:** Thanks for the kind comment. We agree with the reviewer’s comment that use of “universally recognized” in conjunction with “necessary and sufficient” is an overstatement that could potentially mislead readers.

In the original manuscript, we sought to state that reward-driven behavior is widely believed to play a crucial role in the various, complex cognitive capacities of embodied agents, including perception, imitation, and learning. And these studies [1-3] also suggest that reward maximization may serve as a principle for generating intelligent behavior.

[1] Silver, D., Singh, S., Precup, D., Sutton, R.S.: Reward is enough. *Artificial Intelligence* 299, 103535(2021).

[2] Davidson, G., Todd, G., Togelius, J., Gureckis, T.M., Lake, B.M.: Goals as reward-producing programs. *Nature Machine Intelligence* 7(2), 205–220 (2025).

[3] Kasdin, J., Duffy, A., Nadler, N. et al. Natural behaviour is learned through dopamine-mediated reinforcement. *Nature* 641, 699–706 (2025).

In the revised manuscript, we have rephrased this sentence to reflect more accurately the prevailing view in the literature, as shown below:

**In particular, in the contexts of cognitive science and embodied agents, reward-driven behavior has been widely regarded as important in terms of governing complex cognitive abilities such as perception, imitation, and learning.**

\* Page 3, first line: “Consequently, the primate brain must embody diverse types of cognitive intelligence that facilitate optimal reward acquisition.” Why the primate brain specifically? Why “must” and “optimal”, as biological life may well have found heuristic or suboptimal approaches.

**Response:** Thanks for the reviewer’s kind comments.

In response to “*Why the primate brain specifically?*”, as illustrated in Fig. 1, we used the primate brain (specifically, a monkey) as an example to highlight the significance of reward maximization during the development of natural intelligence. We fully agree with the reviewer that the principle of reward-driven behavior is not limited to primates, but is broadly applicable across various forms of biological life.

In response to “*Why “must” and “optimal”, as biological life may well have found heuristic or suboptimal approaches*”, we completely agree that biological systems often employ heuristic or sub-optimal policies to develop remarkable intelligence. Nevertheless, reward maximization remains a key driver of intelligent behavior development [1-6].

[1] Nygaard, T.F., Martin, C.P., Torresen, J., Glette, K., Howard, D.: Real-world embodied ai through a morphologically adaptive quadruped robot. *Nature Machine Intelligence* 3(5), 410–419 (2021).

[2] Gupta, A., Savarese, S., Ganguli, S., Fei-Fei, L.: Embodied intelligence via learning and evolution. *Nature Communications* 12(1), 5721 (2021).

[3] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. *Nature* 518(7540), 529–533 (2015).

[4] Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., et al.: Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature* 575(7782), 350–354 (2019).

[5] Silver, D., Singh, S., Precup, D., Sutton, R.S.: Reward is enough. *Artificial Intelligence* 299, 103535 (2021).

[6] Icarte, R.T., Klassen, T.Q., Valenzano, R., McIlraith, S.A.: Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research* 73, 173–208 (2022).

In the revised manuscript, according to the reviewer’s comment, we have modified the sentence to provide a more objective and nuanced description of how reward maximization fosters intelligence. The updated sentences are copied here for your convenience:

For example, as shown in Fig. 1, the monkey brain can be conceptualized as exhibiting a form of natural intelligence, and the behaviors thereof can be interpreted as following a policy that maximizes cumulative rewards (bananas). **Natural organisms tend to embrace various types of cognitive intelligence that improve life [21, 22].** Likewise, in robotic EAI systems, the reward maximization principle leads the robots to develop diverse intelligence-driven abilities [15].

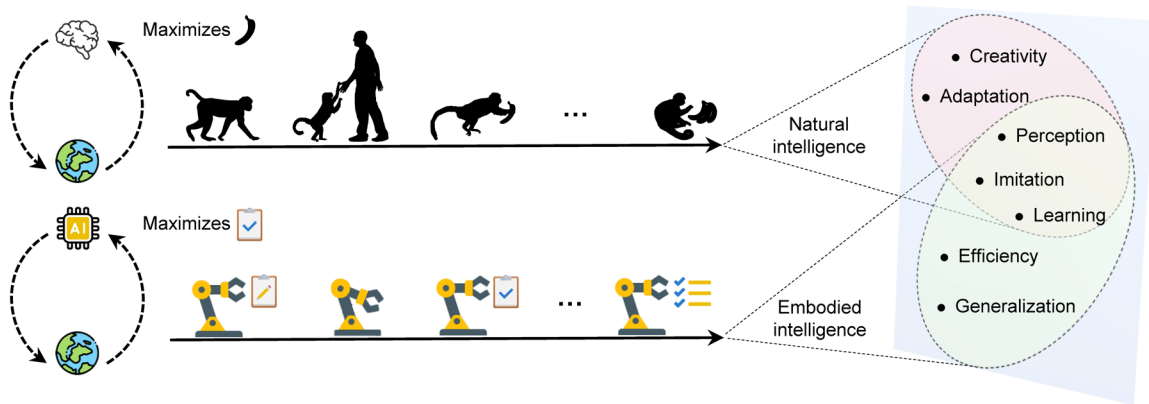


Fig. 1 The reward maximization principle of natural and embodied intelligence.

\* Page 3, near top: “This process, gives the robot agent capabilities that approximate and surpass those of natural intelligence.” Ideally soften the statement, but ideally would provide references for this claim.

**Response:** Thanks for the kind comment. Accordingly, we have revised the sentences to present a balanced and nuanced comparison between artificial and natural intelligence. Additionally, we have also added relevant references to substantiate the statement. The revised sentences are shown as below:

The physical structure of the robot functions as the execution system that receives orders from an embodied intelligent brain that optimizes task execution. **During this process, the robotic agent seeks to perform optimally and, at times, even demonstrates capabilities in certain areas that closely**

**resemble those of natural intelligence [11, 23, 24].** Therefore, the general objective of reward maximization includes many or all of the goals of intelligence [25–27].

[11] Meng, Y., Bing, Z., Yao, X., Chen, K., Huang, K., Gao, Y., Sun, F., Knoll, A.: Preserving and combining knowledge in robotic lifelong reinforcement learning. *Nature Machine Intelligence* 7(2), 256–269 (2025).

[23] Kim, T., Hong, I., Im, S., Rho, S., Kim, M., Roh, Y., Kim, C., Park, J., Lim, D., Lee, D., et al.: Wing-strain-based flight control of flapping-wing drones through reinforcement learning. *Nature Machine Intelligence* 6(9), 992–1005 (2024).

[24] Lee, J., Bjelonic, M., Reske, A., Wellhausen, L., Miki, T., Hutter, M.: Learning robust autonomous navigation and locomotion for wheeled-legged robots. *Science Robotics* 9(89), 9641 (2024).

\* “Powerful embodied RL agents could provide a solution to AGI.” Sure, maybe, but this is just speculation without any provided support.

**Response:** Thanks for the kind comment. According to the comment, we have modified this sentence to reflect more accurately the current state of research:

**Some researchers have suggested that powerful embodied RL agents could open a path toward AGI [21, 28].**

[21] Khalvati, K., Park, S.A., Dreher, J.-C., Rao, R.P.: A probabilistic model of social decision making based on reward maximization. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 2909–2917 (2016).

[28] Duñez-Guzmán, E.A., Sadedin, S., Wang, J.X., McKee, K.R., Leibo, J.Z.: A social path to human-like artificial intelligence. *Nature Machine Intelligence* 5(11), 1181–1188 (2023).

\* “Furthermore, the fundamental principles underlying optimal reward functions and the tasks remain largely unexplored, often requiring a case-by-case approach.” There is a rich literature on reward design and related topics, far from an unexplored topic. However, I find this line more pernicious in that this paper does not compare with or even discuss connections to even a single other method for reward design/discovery. Thus, this line seems to attempt to justify this omission by claiming the non-existence of related work.

**Response:** Thanks for the constructive comments.

The original sentence: “Furthermore, the fundamental principles underlying optimal reward functions and the tasks remain largely unexplored, often requiring a case-by-case approach” was intended to state that, while existing approaches seek to learn reward functions using external information or prior knowledge, relatively few works have addressed the general definition and/or formalization of the optimal reward function *per se* [1-3]. Most existing methods treat reward modeling as a supervised or inverse learning problem, in which the rewards are shaped [4-8] or inferred based on prior knowledge such as expert demonstrations [9-12] or human preference

labels [13-16]. In contrast, our approach aims to discover reward functions in a fully autonomous manner, guided solely by environmental interactions and generalization signals, without relying on predefined expert trajectories, heuristics, or task-specific priors. In the revised manuscript, we have added the relevant text that reflects more accurately the breadth of existing research. We have accordingly also updated the Introduction in the Main File and Supplementary Related Works to provide a clearer comparison between our approach and existing relevant methods, which are also copied here for your convenience:

#### Introduction

However, despite progress in constructing and exploring reward function structures from various perspectives, a widely applicable method of reward function design that is fully independent of external knowledge has yet to be found. **Furthermore, clarifying the formalization of the optimal reward function remains a challenging task.**

#### Supplementary Related Works

##### Meta Learning in Reinforcement Learning

Meta learning methods have been widely applied to RL tasks [23–28] including hyperparameter optimization [25, 29], model learning [24, 30], and algorithm generalization [28]. In this paper, we focus on a meta learning framework specific to reward function learning [31, 32]. One approach used to reward meta learning involves optimizing the reward functions across diverse task distributions for adaptation to new tasks [33, 34]. A promising alternative is to establish a bilevel optimization framework based on meta learning. Following the theory proposed in [21], researchers in [14] posited that reward learning could be framed as a bilevel optimization problem, where policy optimization in an inner loop shapes the rewards and the reward function parameters are optimized in the outer loop. Similarly, [2] viewed reward learning as a bilevel optimization framework but focused on how the optimized reward function is integrated within the framework. Our approach aligns with this idea. The framework has been expanded in various ways. For example, [35–37] incorporated it into multi-agent RL, allowing each agent to learn a personal reward function. Currently, the key challenges posed by the use of bilevel framework, to optimize reward functions, include effective definition of the upper-level optimization objectives. Usually, the definition is the same as that of the optimal reward function, as explored in [38], which sought to optimize imperfect rewards. Meanwhile, [39] and [5] focused on alignment of reward functions with human preferences. **Our approach differs fundamentally from previous methods in terms of how the upper-level optimization objective is defined within the bilevel framework. Prior works often rely on indirect or heuristic formulations, but our method introduces a direct and principled objective. Specifically, we define the optimal reward function by explicitly minimizing policy regret. The policy performance of the agent is the metric used for reward optimization, which effectively resolves the long-standing challenge of specifying the upper-level optimization goal. Thus, our approach establishes a clear and theoretically grounded connection between reward function learning and policy optimization, setting it apart from existing methods.**

## Reward Function Design Methods

In the early stages of RL, researchers relied primarily on manual, trial-and-error adjustments to refine the reward functions [40]. However, this approach is time-consuming and often culminates in reward functions that inadvertently guide agents to act contrary to the human intent [22], associated with policy deviations. Therefore, many methods have been developed to optimize reward learning. One widely studied approach is reward shaping [2, 41–43], which involves the creation of auxiliary rewards that supplement the environmental reward function and ensure policy invariance [44]. However, the definition of shaping functions and the attainment of policy invariance have long posed challenges [45]. Another popular approach is inverse RL (IRL), initially proposed by [46], which is aimed at deriving a reward function from expert demonstrations, assuming that such demonstrations represent an optimal policy. Recent advances in deep learning (DL) have extended the performance of IRL [47, 48], suggesting that an adequate number of expert demonstrations can indeed produce an optimal reward function. However, in many cases, the aim of RL is not only to replicate human performance but also to exceed human capabilities, particularly in tasks that are beyond human reach [49, 50]. This limitation constrains the applicability of IRL, although some researchers have sought to surpass expert performance by incorporating additional information and using non-expert demonstrations [51, 52]. Recently, an active reward learning approach has emerged, inspired by the training paradigms of LLMs [53, 54], with its origins in robotics [55]. Active reward learning is aimed at constructing reward functions that align closely with human preferences by modeling those preferences [56–58], even though some researchers have explored whether it is possible to completely bypass reward learning [59–61]. Nonetheless, the capture of sufficiently rich human feedback remains a significant challenge [62]. Other reward learning methods have leveraged characteristics of the agent or environmental [15, 63] to design reward functions, such as encouraging curiosity-driven exploration in maze tasks [16, 64]. **All cited methods extensively explored the automatic design or optimization of reward functions. However, such approaches rely heavily on priors. For example, IRL requires high-quality expert demonstrations and ARL requires substantial human preference data. In contrast, our work directly optimizes the reward function using the interactions of the agent during RL, leveraging agent experience (policy meta-gradients). There is essentially no need for prior knowledge or external supervision. By continuously refining the reward function based on the experience accumulated by the agent, our approach renders reward function discovery more autonomous and adaptive. This is particularly helpful when expert demonstrations and/or human feedback are scarce or unavailable.**

For the comment of “*There is a rich literature on reward design and related topics, far from an unexplored topic. However, I find this line more pernicious in that this paper does not compare with or even discuss connections to even a single other method for reward design/discovery. Thus, this line seems to attempt to justify this omission by claiming the non-existence of related work*”, we fully agree with the reviewer’s comment that reward design [3, 17-21] and related topics [4-16] are well-established research areas. In the revised manuscript, we have included more existing reward function design methods [8,17,21] into baseline methods to fully compare the performance.

Specifically, [17] formulated intrinsic reward learning as a bilevel optimization problem and leveraged hyperparameter optimization and conditional layer normalization during learning of the

intrinsic rewards. [8] presented an efficient adaptive reward shaping mechanism, using the success rate as the shaping reward. Kernel density estimation (KDE) and random Fourier features (RFFs) were employed to derive a beta distribution from which the success rate was sampled. Additionally, we also selected [21] as a comparison method, which also emphasized the important role of reward maximization in RL. However, unlike other reward function design method, [21] directly maximized the highest value of the reward obtained, not the cumulative return of RL.

[1] Zhang, H., Parkes, D.C., Chen, Y.: Policy teaching through reward function learning. In: Proceedings of the 10th ACM Conference on Electronic Commerce, pp. 295–304 (2009).

[2] Singh, S., Lewis, R.L., Barto, A.G.: Where do rewards come from. In: Proceedings of the Annual Conference of the Cognitive Science Society, pp. 2601–2606 (2009).

[3] Grunitzki, R., Silva, B.C., Bazzan, A.L.: A flexible approach for designing optimal reward functions. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, pp. 1559–1561 (2017).

[4] Hu, Y., Wang, W., Jia, H., Wang, Y., Chen, Y., Hao, J., Wu, F., Fan, C.: Learning to utilize shaping rewards: a new approach of reward shaping. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, pp. 15931–15941 (2020).

[5] Devidze, R., Kamalaruban, P., Singla, A.: Exploration-guided reward shaping for reinforcement learning under sparse rewards. In: Proceedings of the 36th International Conference on Neural Information Processing Systems, pp. 5829–5842 (2022).

[6] Chen, E., Hong, Z.-W., Pajarinen, J., Agrawal, P.: Redeeming intrinsic rewards via constrained optimization. In: Proceedings of the 36th International Conference on Neural Information Processing Systems, pp. 4996–5008 (2022).

[7] Lidayan, A., Dennis, M., Russell, S.: Bamdp shaping: a unified framework for intrinsic motivation and reward shaping. preprint arXiv:2409.05358 (2024).

[8] Ma, H., Luo, Z., Vo, T.V., Sima, K., Leong, T.-Y.: Highly efficient self-adaptive reward shaping for reinforcement learning. In: Proceedings of the 13th International Conference on Learning Representations (2025).

[9] Ng, A.Y., Russell, S.J.: Algorithms for inverse reinforcement learning. In: Proceedings of the 17th International Conference on Machine Learning, pp. 663–670 (2000).

[10] Balakrishnan, S., Nguyen, Q.P., Low, B.K.H., Soh, H.: Efficient exploration of reward functions in inverse reinforcement learning via bayesian optimization. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, pp. 4187–4198 (2020).

[11] Zeng, S., Liu, Y., Rangwala, H., Karypis, G., Hong, M., Fakoor, R.: From demonstrations to rewards: Alignment without explicit human preferences. preprint arXiv:2503.13538 (2025).

[12] Chen, L., Paleja, R., Gombolay, M.: Learning from suboptimal demonstration via self-supervised reward regression. In: Conference on Robot Learning, pp. 1262–1277 (2021).

- [13] Christiano, P.F., Leike, J., Brown, T.B., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 4302–4310 (2017).
- [14] Choi, H., Jung, S., Ahn, H., Moon, T.: Listwise reward estimation for offline preference-based reinforcement learning. In: Proceedings of the 41st International Conference on Machine Learning, pp.8651–8671 (2024).
- [15] Lu, Y., Wang, Q., Cao, H., Xu, X., Zhang, M.: Smoothed preference optimization via renoise inversion for aligning diffusion models with varied human preferences. In: Proceedings of the 42nd International Conference on Machine Learning (2025).
- [16] Bıyık, E., Losey, D.P., Palan, M., Landolfi, N.C., Shevchuk, G., Sadigh, D.: Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences. *The International Journal of Robotics Research* 41(1), 45–67 (2022).
- [17] Stadie, B., Zhang, L., Ba, J.: Learning intrinsic rewards as a bi-level optimization problem. In: Conference on Uncertainty in Artificial Intelligence, pp. 111 – 120 (2020).
- [18] Devidze, R., Radanovic, G., Kamalaruban, P., Singla, A.: Explicable reward design for reinforcement learning agents. In: Proceedings of the 35th International Conference on Neural Information Processing Systems, pp. 20118–20131 (2021).
- [19] Booth, S., Knox, W.B., Shah, J., Niekum, S., Stone, P., Allievi, A.: The perils of trial-and-error reward design: misdesign through overfitting and invalid task specifications. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 5920–5929 (2023).
- [20] Knox, W.B., Allievi, A., Banzhaf, H., Schmitt, F., Stone, P.: Reward (mis) design for autonomous driving. *Artificial Intelligence* 316, 103829 (2023)2182–2191 (2023).
- [21] Vevurko, G., Böhmer, W., Weerdt, M.: To the max: reinventing reward in reinforcement learning. In: Proceedings of the 41st International Conference on Machine Learning, pp. 49455 – 49470 (2024).

The experimental results and comparative analysis are provided in the Results of the Main File. The detailed implementation of the state-of-the-art methods is described in the Supplementary Information. The corresponding contents are copied here for your convenience:

## Results

Furthermore, **the reward functions discovered by the proposed approach are compared to the baseline rewards, including sparse reward signals, manually designed reward functions, and rewards from three state-of-the-start reward learning methods, including LIRBO [49], Tomax [30], SASR [53].** The comparison serves to illustrate further the relationship between the reward functions and the tasks, as well as how these functions help embodied RL agents to achieve better performance. **The experimental setups for both the numerical experiments and the real-world tasks, and the details on how the comparative methods are implemented, are described in the Supplementary Experimental Details.**

### Case Study of Sparse-Reward Tasks

**Fig. 3(c-d) shows the average episodic rewards of embodied RL agents when using our proposed approach compared to those of agents that receive sparse reward signals and reward signals from baseline methods. Agents under the proposed reward function outperform the baseline agents in terms of both convergence speed and learning stability.** When receiving sparse reward signals, the agent finds it challenging to learn the optimal policy and achieve fast convergence because of the limited number of samples that can be effectively utilized, resulting in low sample efficiency. This low efficiency requires the agent to engage in extensive trial-and-error, thereby slowing the convergence rate. **Reward function design methods have relieved this challenge. LIRBO replaces sparse rewards with intrinsic reward signals, and SASR uses a shaping auxiliary function to the same end. Tomax directly learns historical maximum rewards but sometimes does not perform well, as the initial random exploration in sparse-reward tasks does not deliver an adequate reward to the agent.** In contrast, the proposed approach delivers dense reward signals directly, thereby improving the sample efficiency.

Fig. 3(e) shows the average episodic returns and standard errors throughout the entire learning process. **The PPO and DQN agents that employ our proposed approach comprehensively outperform the agents under baseline methods. Additionally, as shown in Fig. 3(c-e), our approach enhances learning stability and the convergence speed.**

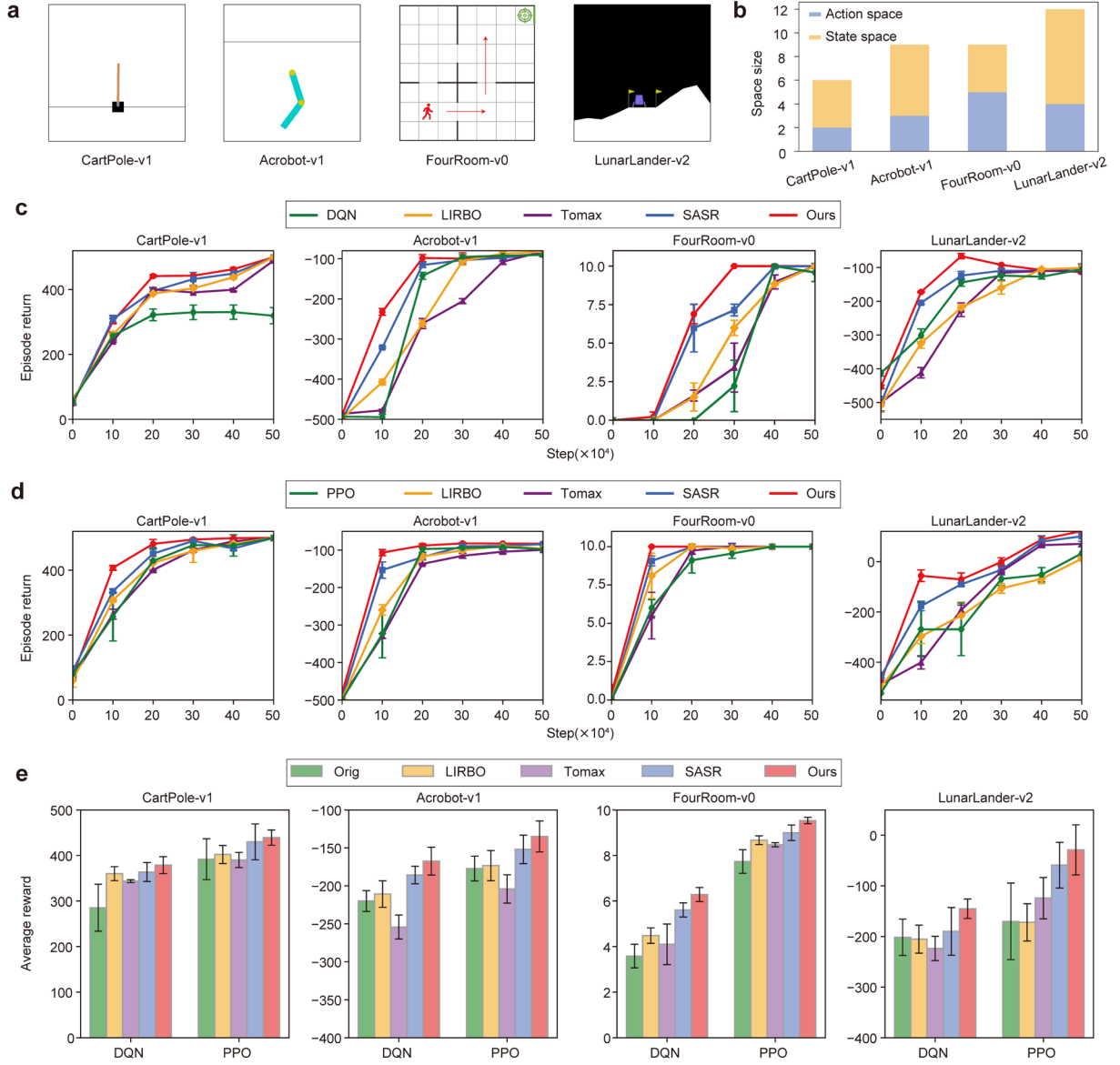


Fig. 3 Evaluation results for sparse-reward tasks. **a** The environments used in the sparse-reward task, including *CartPole-v1*, *Acrobot-v1*, *FourRoom-v0* and *LunarLander-v2*. **b** The sizes of the observation and action spaces of the four tasks, illustrating the varying complexity levels. **c** The reward curves created during the learning processes of the DQN agent with different reward functions on different tasks. The solid lines show the average rewards and the error bars indicate the standard deviations across five trials using different random seeds. **d** The reward curves created during the learning processes of PPO agent with different reward functions on different tasks. **e** The average rewards obtained by the embodied RL agent during all episodes, with the standard deviations of the rewards received when different algorithms performed each task.

### Case Study of High-Dimensional Control Tasks

Fig. 5(c-d) shows the cumulative reward curves of embodied RL agents after each episode across the four tasks, with the error bars indicating the variabilities across the five parallel experiments. The proposed approach exhibits superior performance, even outperforming the reward functions of OpenAI designed by humans. **Compared to other baseline methods, our approach is more effective when PPO and SAC agents are employed. The convergence speed is faster and the episodic returns are higher. SASR and LIRBO perform similarly, but SASR achieves better when a PPO agent is applied because SASR samples the success rate of historical data.** Fig. 5(c-d) also reveals that the proposed approach accelerates agent convergence and improves the cumulative expected returns, particularly when the number of agent joints increases. Fig. 5(e) presents the means and variance of the cumulative returns given during learning and compares the rewards of the proposed approach with those afforded by the baseline rewards. It is clear that, for both the SAC and PPO agents, the proposed approach not only outperforms the baseline but also reduces risk induced by randomness.

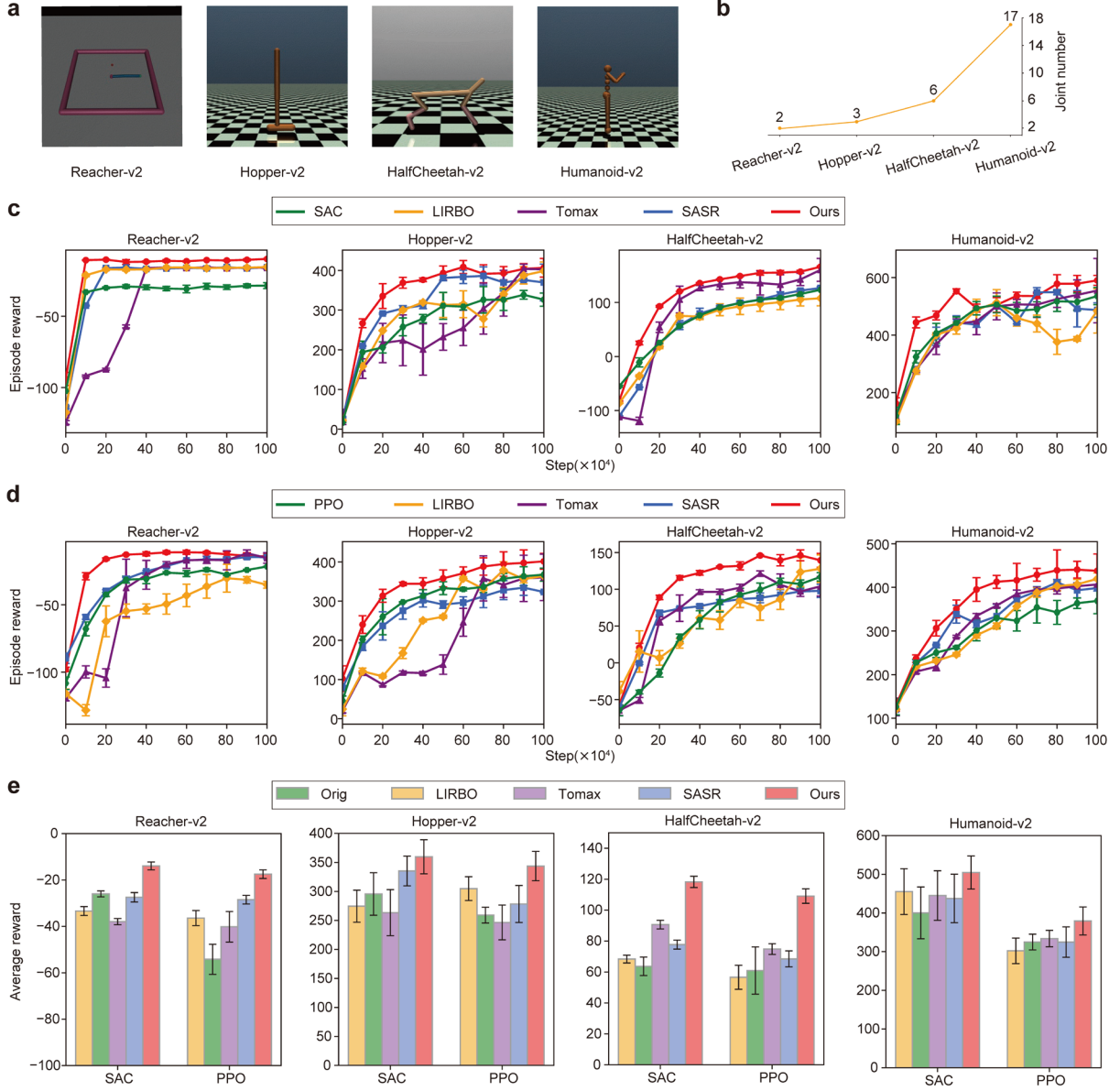


Fig. 5 Evaluation results for high-dimensional control tasks. a The high-dimensional control tasks, including *Reacher-v2*, *HalfCheetah-v2*, *Hopper-v2* and *Humanoid-v2*. b The number of controllable joints increases from *Reacher-v2* to *Humanoid-v2*, reflecting the increasing complexity of the tasks. c The reward curves during the learning processes of the SAC agent across the four tasks. The solid lines show the average episode rewards and the error bars indicate the standard deviations over five trials with different random seeds. d The reward curves during the learning processes of the PPO agent across the four tasks. e The average rewards with their standard deviations for different RL agents and different tasks. The results highlight the improved performance of the proposed approach across all tasks.

## Case Study of Real-World Tasks

### Case Study of Energy Management Task

Fig. 7(b) shows the rate of reduction in energy consumption when applying baseline RL and RL guided by the reward design approaches in data centers over a 1-year period. The numerical values are the percentage reduction rates. Fig. 7(c) illustrates the total energy consumption over 3, 6, 9, and 12 months for each RL agent. Although all RL methods reduce energy consumption, the reward function discovery approach further reduces consumption, particularly over longer times. Notably, as RL algorithms become more advanced, from the DQN to the more sophisticated SAC, the energy reduction rate increases (from 21% for DQN to 52% for SAC). The superiority of the proposed approach is more obvious when compared with the baseline models. With our method, all RL agents (including DQN, PPO, and SAC) achieve energy reductions that exceed 60%. These results highlight the generality and effectiveness of our approach for RL agents involved in energy management and other tasks. Therefore, the proposed optimal reward function discovery approach can be applied to RL agents operating in data centers and elsewhere.

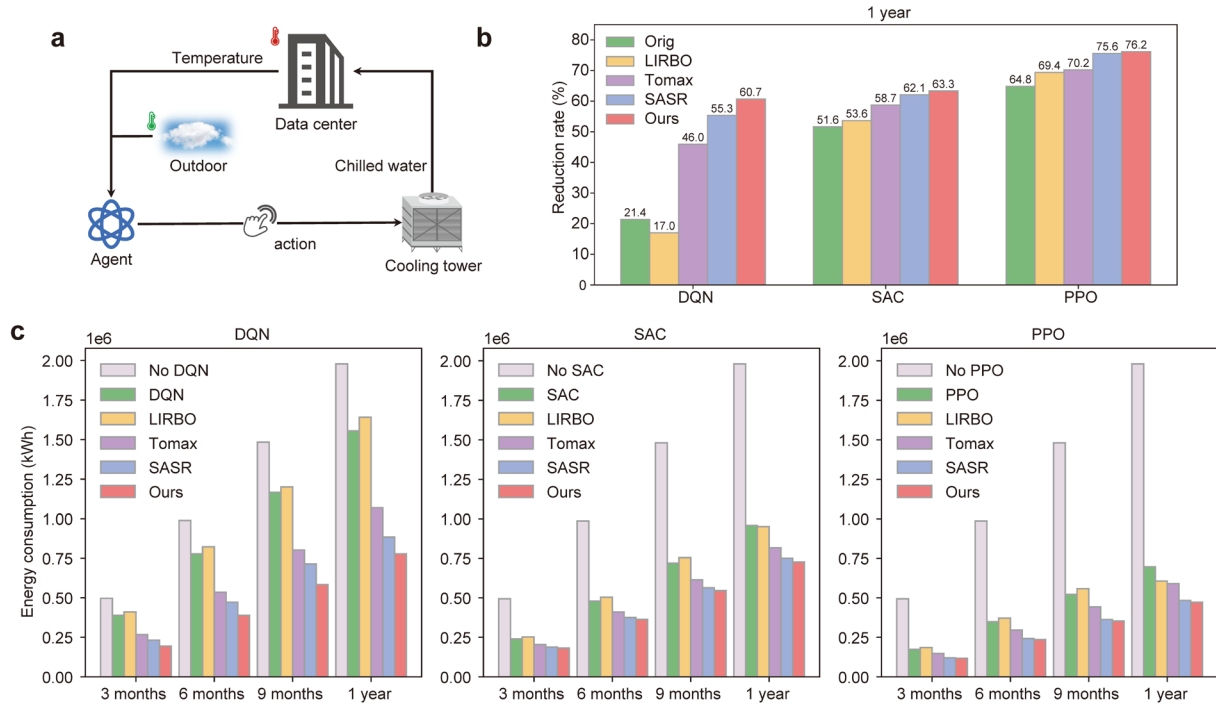


Fig. 7 Evaluation results for the energy management task. a Schematic of the data center cooling task, where an RL agent optimizes cooling system based on temperature and silicon data. b The 1-year reductions in energy consumption afforded by different reward design methods for different RL algorithms (DQN, SAC, and PPO). The agents that receive rewards from the discovered reward functions perform significantly better than others. c Energy consumption over 3, 6, 9, and 12 months. The RL approach using the discovered optimal reward function reduce energy consumption more effectively than do the comparative methods.

## Case Study of Unmanned Systems Control Task

Fig. 8(b) shows the reward curves when agent learning is guided by the reward function of the present work, reward function constructed on the basis of dynamic principles, and other reward design methods. Across the three different RL agents, the proposed approach significantly accelerates learning. For the PPO agent, our approach significantly improves both learning speed and the final performance, enabling the agent to successfully accomplish the task, whereas the hand-designed and other comparison methods fail to do so. For the SAC and TD3 agents, the improvement in learning speed is less pronounced, but our method performs comparable performance and ensures stable learning. These results demonstrate that the proposed reward function discovery approach is particularly useful when traditional rewards are insufficient, consistently ensuring robust performance across different RL algorithms.

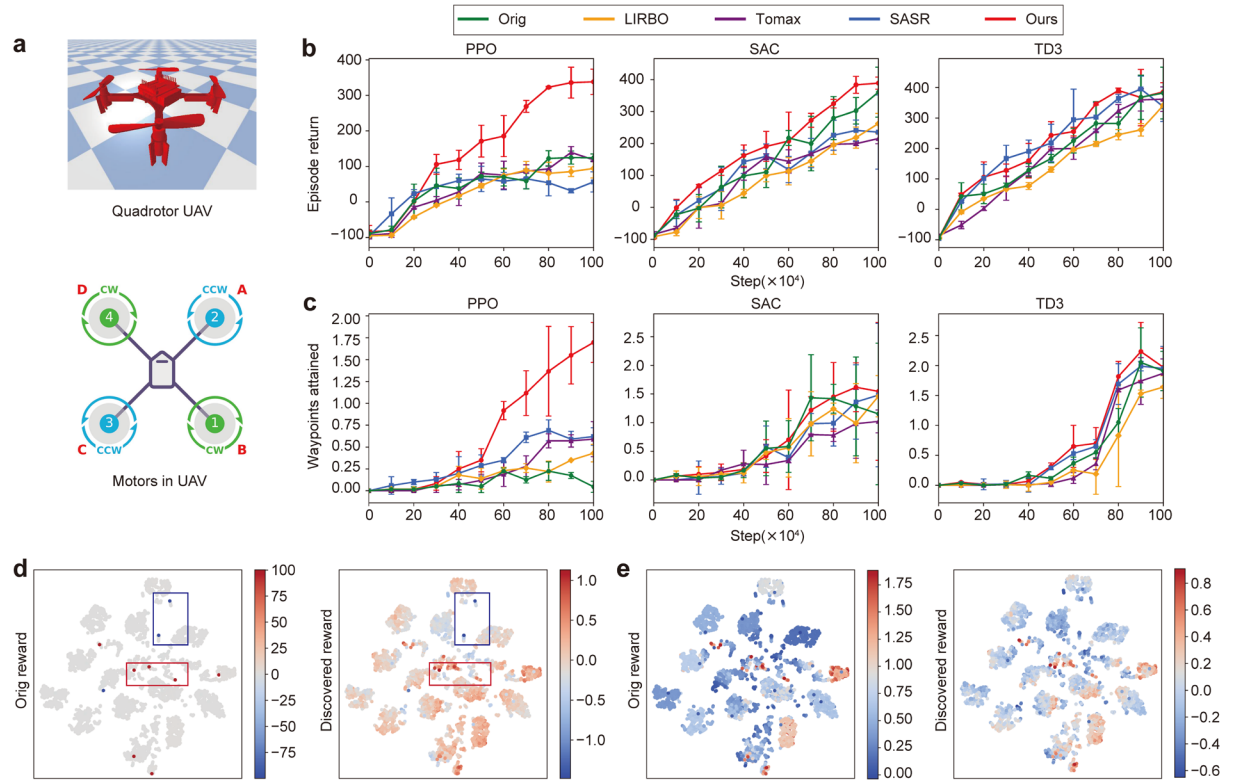


Fig. 8 Evaluation results for unmanned systems control task. a The Quadrotor UAV and the powered motors in a world model. b Learning curves showing reward progression during UAV trajectory tracking using the PPO, SAC, and TD3 agents. The proposed approach is associated with faster convergence and better performance. c The numbers of target trajectory points attained by the UAV during learning. d The state space visualizations of the UAV task, showing the reward distributions of the original reward function derived using dynamic principles and the discovered reward function. The discovered reward function successfully captures critical states. e The state space visualizations after removal of certain states associated with extremely high or low rewards, such as those

**involving collisions or attainment of waypoints. The reward distribution resembles those of reward functions derived using dynamic principles.**

\* Page 5, Framework Overview: “This alternating optimization process is iteratively performed until convergence, at which point the optimal reward function is discovered.” Care is needed here as you have neither proven that the process will converge nor that if it does so it must converge on the optimal reward function.

**Response:** Thanks for the constructive comment. This work is aimed at discovering the reward functions that are tailored to RL agents during the learning process. Theorems 2 and 3 show the gradient existence of the proposed approach, and the parameter updating process is based on the gradient decent method. While it is indeed difficult to guarantee theoretically convergence to the optimal reward function in all possible environments for all agents, our framework specifically and iteratively improves the reward function and policy. The optimization objective is to minimize policy regret, thus to maximize cumulative returns. Each iteration brings the discovered reward function closer to that which is optimal. In all case studies, we observe that this consistently improve both the learned policy and the reward function, ultimately achieving optimal or near-optimal performance in our experimental settings. Such empirical results strongly support the effectiveness of our approach. In response to the comment, we have revised the relevant statement to provide a more accurate description. The updated sentences are shown as below:

After the reward function is updated in the upper-level, the embodied RL agent resumes its interaction with the world model, receiving new reward signals. **This alternating optimization is iteratively performed to improve both the policy and the reward function.**

## # Empirical Results

Figure 4 does not seem to add much to understanding. It is referenced once, as showing the learned policies are more efficient / effective, but we already know that e.g. in Lunar Lander the agent does better due to looking at the learning curve. A much better use of this figure space would be investigating more detail on the discovered reward function itself.

**Response:** Thanks for the helpful comment. In the original manuscript, Fig. 4 is intended to illustrate the behaviors of agents under different reward functions by visualizing their interaction trajectories. Each subplot features two rows of trajectories, where the first row is that for the agent receiving sparse reward signals and the second row corresponds to the agent learned with the discovered reward function. These results suggest that agents given discovered reward function achieve their objectives more efficiently and effectively. We agree with reviewer's comment that the efficiency and effectiveness of the learned policies are already clearly demonstrated in Fig. 3 through the learning curves. Therefore, in the revised manuscript, we have moved Fig. 4 and its related descriptions from Main File to Supplementary Information.

In addition, based on the reviewer's comment, we now offer a more detailed visualization and analysis of the discovered reward function in the revised manuscript. We used a classic control task (*Acrobot-v1* in Fig. 3(a)) with a discrete action space that allows clear visualization of the reward function. The agent in the *Acrobot-v1* system consists of two links connected linearly to form a chain, with one end of the chain fixed. The joint between the two links is actuated. The goal is to apply torques on the actuated joint to swing the free end of the linear chain above a given height while starting from the initial state of hanging downwards. The state space observed by the agents includes the sines and cosines of the two rotational joint angles and their angular velocities:  $(\cos(\theta_1), \sin(\theta_1), \cos(\theta_2), \sin(\theta_2), \dot{\theta}_1, \dot{\theta}_2)$ . The action space is discrete, deterministic, and represents the torque applied on the actuated joint between the two links.

In the revised manuscript, Fig. 4(e-h) illustrates the reward surfaces of the sparse reward function and the discovered reward function for each discrete action of the *Acrobot-v1* task, visualized as a function of the two joint angles  $\theta_1$  and  $\theta_2$ , when the angular velocities  $\dot{\theta}_1$  and  $\dot{\theta}_2$  are 0. The Supplementary Figs. 2-4 shows the distribution of the reward functions at different angular velocities. Each surface illustrates the reward values of our reward function for a specific action: application of -1 torque (action 0), 0 torque (action 1), and 1 torque (action 2) to the actuated joint. The results reveal that our reward function assigns higher values to actions 0 and 2 (-1 and 1 torque, respectively) when the joint angles are away from the downward resting position, encouraging the agent to apply torque when momentum generation is favorable. In contrast, action 1 (0 torque) receives relatively higher rewards near the neutral region, reflecting a learned preference for stabilizing when the system is aligned. This structured reward landscape aligns well with the task objective of swinging the outer link above a target height and demonstrates that the reward function of this work captures meaningful agent control behavior consistent with the dynamics of the *Acrobot-v1* system. The corresponding analyses are copied here for your convenience:

**Fig. 4(e-h) illustrates the reward surfaces of the sparse reward function and the discovered reward function for each discrete action in *Acrobot-v1* task. The reward surfaces are visualized as functions of the two rotational joint angles,  $\theta_1$  and  $\theta_2$ , when their angular velocities  $\dot{\theta}_1$  and  $\dot{\theta}_2$  are 0. Fig. 4(e) shows that the sparse reward function assigns a reward of -1 for all state and action. Fig. 4(f-h) shows the rewards assigned by our reward function for specific actions: application of -1 torque (action 0), 0 torque (action 1), or 1 torque (action 2) to the actuated joint. The results show that the reward function of this work assigns higher values to actions involving -1 and 1 torque when the joint angles deviate from the downward resting position. This encourages the agent to apply torque in situations when momentum generation can be effectively created. Conversely, the action corresponding to 0 torque receives higher rewards when the agent is near the neutral region, indicating a learned preference for stabilization or energy conservation when the system is aligned. This structured reward landscape is consistent with the task objective of swinging the outer link above a target height, demonstrating that the reward function of this work successfully captures meaningful control behaviors aligned with the dynamics of the *Acrobot-v1* task. Visualizations of the reward functions at different angular velocities can be found in the Supplementary Figs. 2-4.**

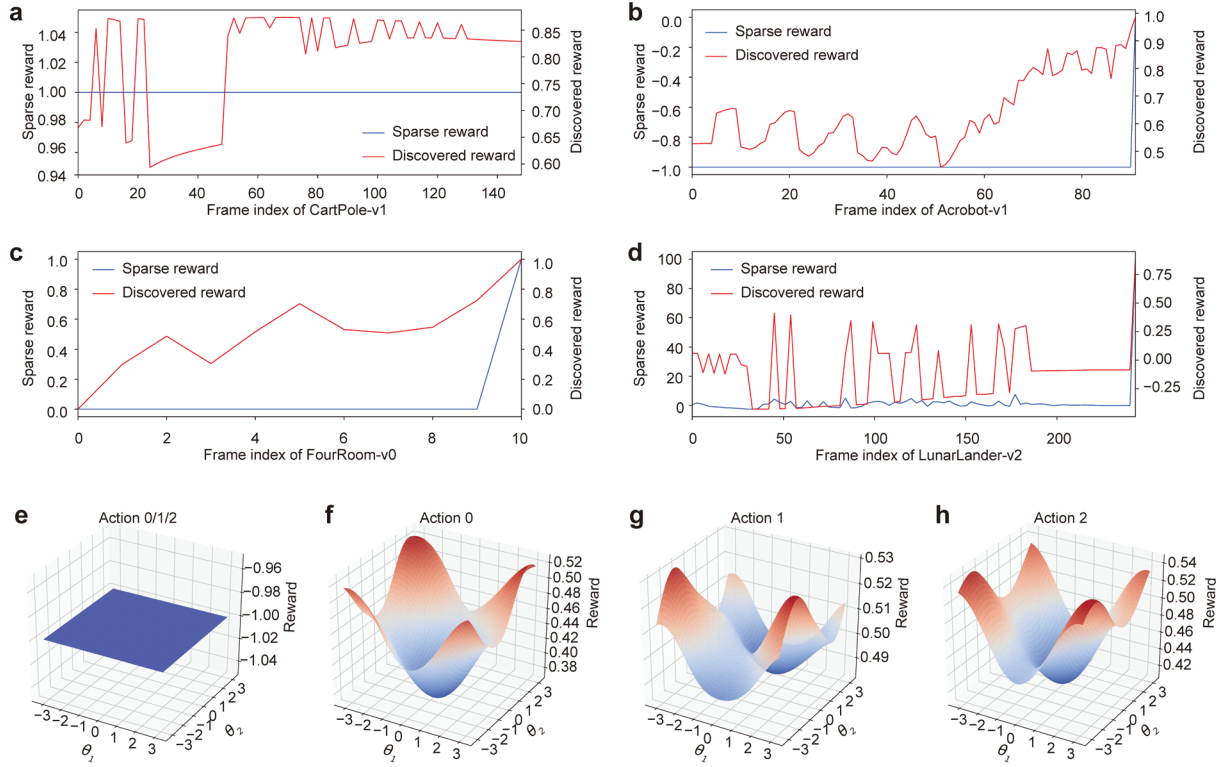


**Fig. 4 Reward visualization.** a Reward signals during a single interaction episode of the *CartPole-v1* task. The sparse reward signals (blue lines) and the reward signals discovered in this work (red lines) are shown. b Reward signals during a single interaction episode of the *Acrobot-v1* task, showing the difference between the sparse and discovered reward signals. c Reward signals during a single interaction episode of the *FourRoom-v0* task, illustrating the increased signal density achieved in this work. d Reward signals during a single interaction episode of the *LunarLander-v2* task. e The surface of the sparse reward function. For any state and action, the sparse reward function always delivers a reward signal of -1. f The reward surface for action 0 (-1 torque) of *Acrobot-v1* as a function of the joint angles  $\theta_1$  and  $\theta_2$  with angular velocities fixed at 0. g The reward surface for action 1 (0 torque). h The reward surface for action 2 (1 torque).

As mentioned previously, I believe these results need to include comparisons with other reward discovery methods, ideally 2 or more with one more closely related to the approach taken here and one more different.

**Response:** Thanks for the constructive comment. In the original manuscript, to verify the effectiveness of the proposed approach, two existing baseline rewards, including sparse reward signals and manually well-designed reward functions are selected to compare the performance with respect to the episode reward and convergence speed. These methods are commonly used for comparison with existing studies.

In the revised manuscript, based on the reviewer’s comment, we also added another three state-of-the-art reward discovery methods (two more closely related to the approach taken this work and one more different) to further compare the performance. This is similar to the previous comment, we have clarified the connections and distinctions between our approach and existing reward discovery methods in the Supplementary Information. We have conducted an extensive literature review and selected several recent reward function design methods as baselines to provide a more comprehensive evaluation of our proposed approach in the Main File. The corresponding comparison results are copied here again for your convenience:

## Results

Furthermore, **the reward functions discovered by the proposed approach are compared to the baseline rewards, including sparse reward signals, manually designed reward functions, and rewards from three state-of-the-start reward learning methods, including LIRBO [49], Tomax [30], SASR [53].** The comparison serves to illustrate further the relationship between the reward functions and the tasks, as well as how these functions help embodied RL agents to achieve better performance. **The experimental setups for both the numerical experiments and the real-world tasks, and the details on how the comparative methods are implemented, are described in the Supplementary Experimental Details.**

### Case Study of Sparse-Reward Tasks

**Fig. 3(c-d) shows the average episodic rewards of embodied RL agents when using our proposed approach compared to those of agents that receive sparse reward signals and reward signals from baseline methods. Agents under the proposed reward function outperform the baseline agents in terms of both convergence speed and learning stability.** When receiving sparse reward signals, the agent finds it challenging to learn the optimal policy and achieve fast convergence because of the limited number of samples that can be effectively utilized, resulting in low sample efficiency. This low efficiency requires the agent to engage in extensive trial-and-error, thereby slowing the convergence rate. **Reward function design methods have relieved this challenge. LIRBO replaces sparse rewards with intrinsic reward signals, and SASR uses a shaping auxiliary function to the same end. Tomax directly learns historical maximum rewards but sometimes does not perform well, as the initial random exploration in sparse-reward tasks does not deliver an adequate reward to the agent.** In contrast, the proposed approach delivers dense reward signals directly, thereby improving the sample efficiency.

Fig. 3(e) shows the average episodic returns and standard errors throughout the entire learning process. **The PPO and DQN agents that employ our proposed approach comprehensively outperform the agents under baseline methods. Additionally, as shown in Fig. 3(c-e), our approach enhances learning stability and the convergence speed.**

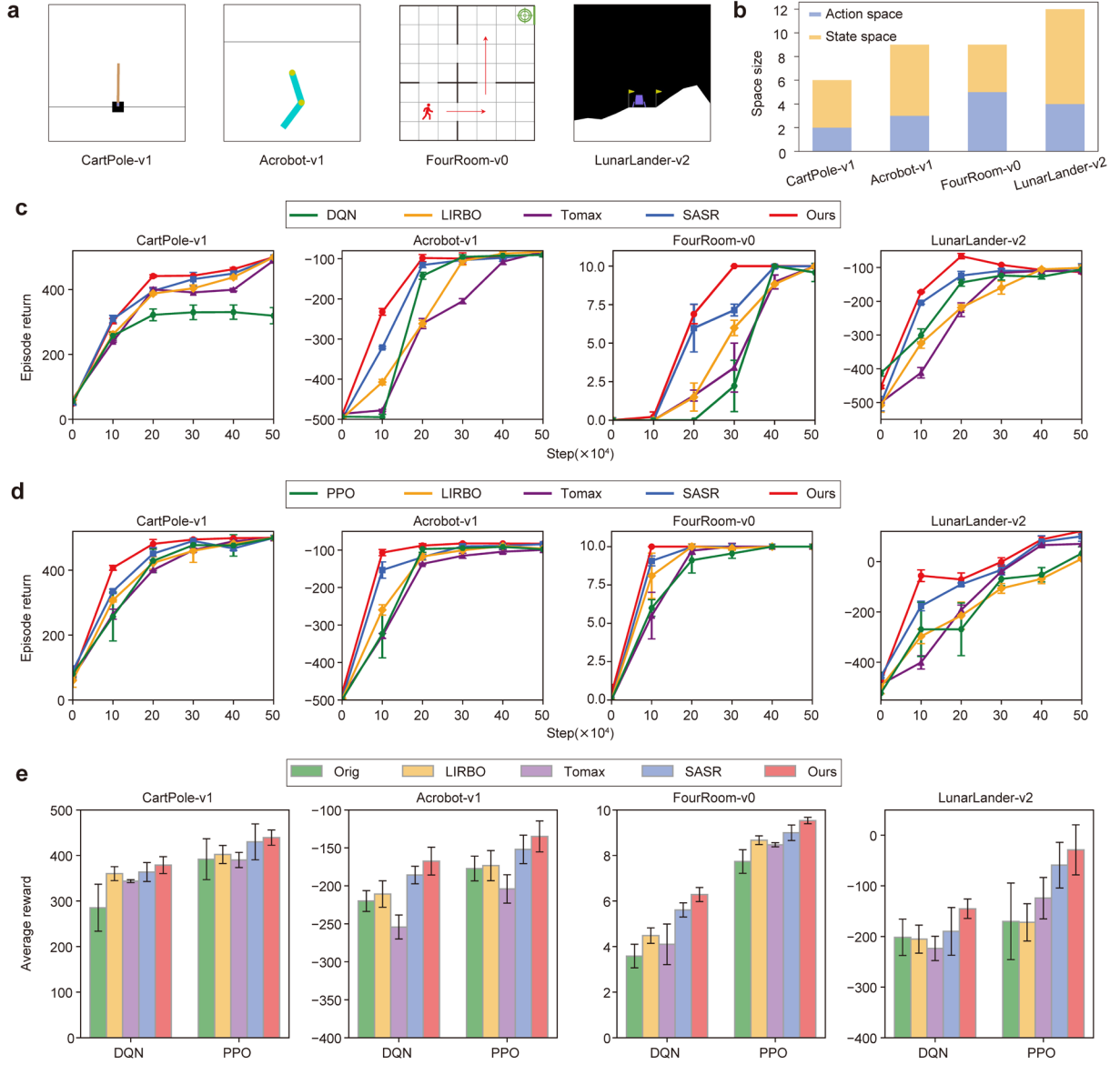


Fig. 3 Evaluation results for sparse-reward tasks. **a** The environments used in the sparse-reward task, including *CartPole-v1*, *Acrobot-v1*, *FourRoom-v0* and *LunarLander-v2*. **b** The sizes of the observation and action spaces of the four tasks, illustrating the varying complexity levels. **c** The reward curves created during the learning processes of the DQN agent with different reward functions on different tasks. The solid lines show the average rewards and the error bars indicate the standard deviations across five trials using different random seeds. **d** The reward curves created during the learning processes of PPO agent with different reward functions on different tasks. **e** The average rewards obtained by the embodied RL agent during all episodes, with the standard deviations of the rewards received when different algorithms performed each task.

### Case Study of High-Dimensional Control Tasks

Fig. 5(c-d) shows the cumulative reward curves of embodied RL agents after each episode across the four tasks, with the error bars indicating the variabilities across the five parallel experiments. The proposed approach exhibits superior performance, even outperforming the reward functions of OpenAI designed by humans. **Compared to other baseline methods, our approach is more effective when PPO and SAC agents are employed. The convergence speed is faster and the episodic returns are higher. SASR and LIRBO perform similarly, but SASR achieves better when a PPO agent is applied because SASR samples the success rate of historical data.** Fig. 5(c-d) also reveals that the proposed approach accelerates agent convergence and improves the cumulative expected returns, particularly when the number of agent joints increases. Fig. 5(e) presents the means and variance of the cumulative returns given during learning and compares the rewards of the proposed approach with those afforded by the baseline rewards. It is clear that, for both the SAC and PPO agents, the proposed approach not only outperforms the baseline but also reduces risk induced by randomness.

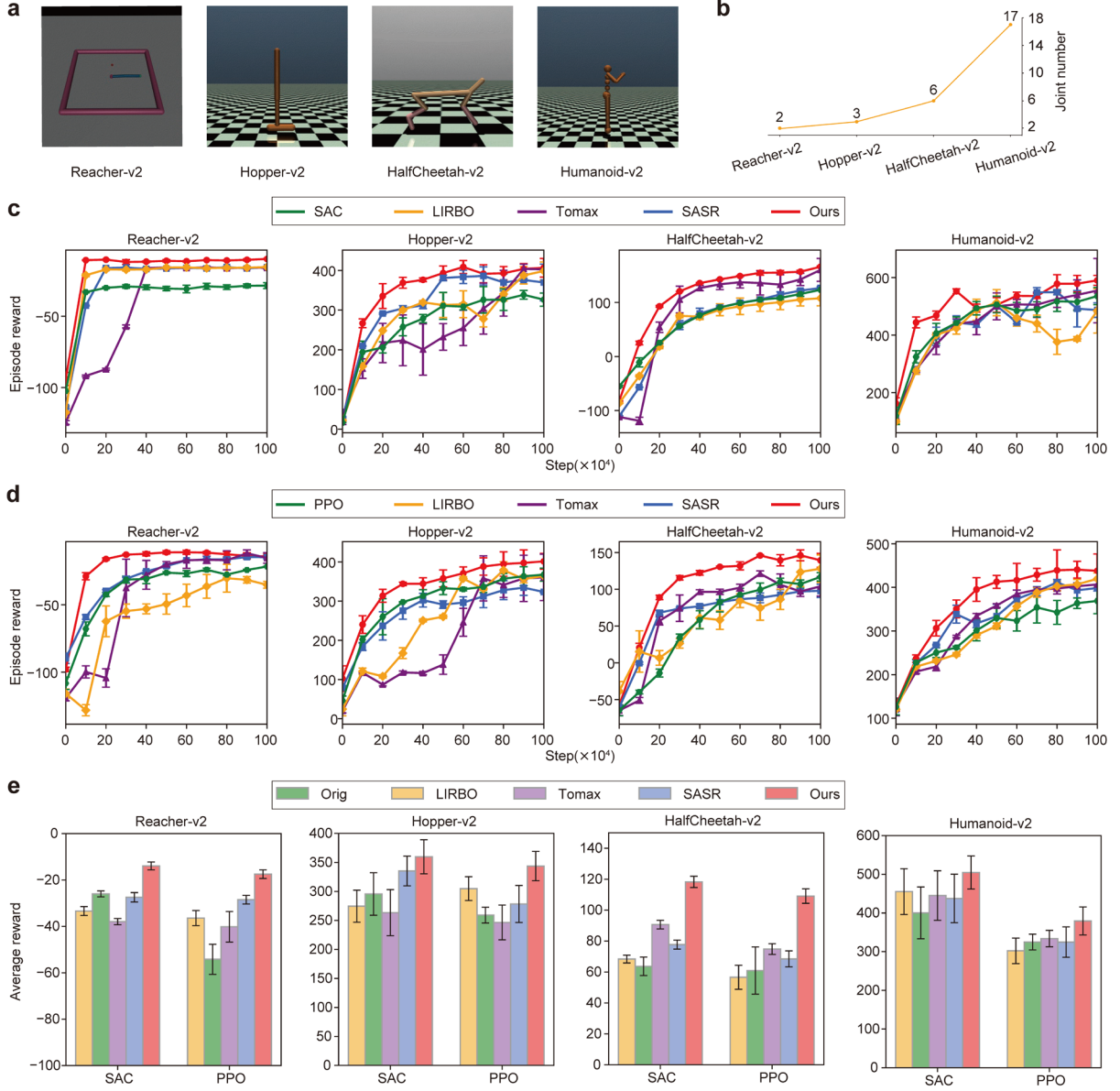


Fig. 5 Evaluation results for high-dimensional control tasks. a The high-dimensional control tasks, including *Reacher-v2*, *HalfCheetah-v2*, *Hopper-v2* and *Humanoid-v2*. b The number of controllable joints increases from *Reacher-v2* to *Humanoid-v2*, reflecting the increasing complexity of the tasks. c The reward curves during the learning processes of the SAC agent across the four tasks. The solid lines show the average episode rewards and the error bars indicate the standard deviations over five trials with different random seeds. d The reward curves during the learning processes of the PPO agent across the four tasks. e The average rewards with their standard deviations for different RL agents and different tasks. The results highlight the improved performance of the proposed approach across all tasks.

## Case Study of Real-World Tasks

### Case Study of Energy Management Task

Fig. 7(b) shows the rate of reduction in energy consumption when applying baseline RL and RL guided by the reward design approaches in data centers over a 1-year period. The numerical values are the percentage reduction rates. Fig. 7(c) illustrates the total energy consumption over 3, 6, 9, and 12 months for each RL agent. Although all RL methods reduce energy consumption, the reward function discovery approach further reduces consumption, particularly over longer times. Notably, as RL algorithms become more advanced, from the DQN to the more sophisticated SAC, the energy reduction rate increases (from 21% for DQN to 52% for SAC). The superiority of the proposed approach is more obvious when compared with the baseline models. With our method, all RL agents (including DQN, PPO, and SAC) achieve energy reductions that exceed 60%. These results highlight the generality and effectiveness of our approach for RL agents involved in energy management and other tasks. Therefore, the proposed optimal reward function discovery approach can be applied to RL agents operating in data centers and elsewhere.

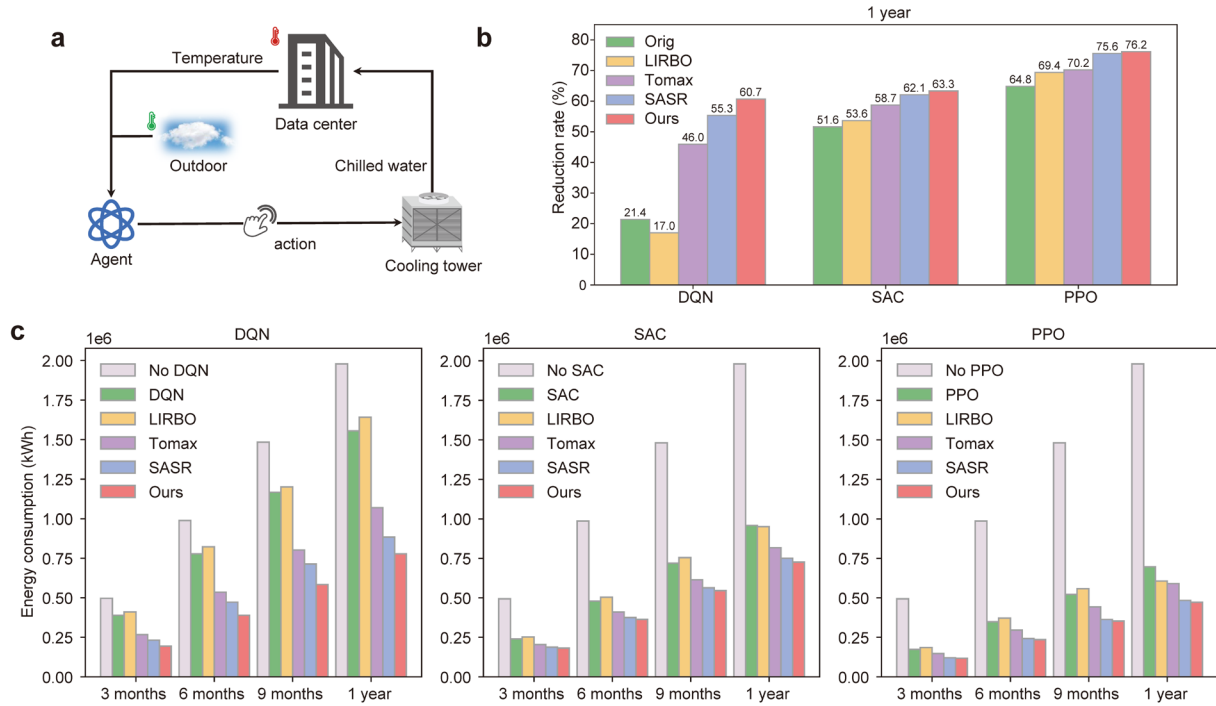


Fig. 7 Evaluation results for the energy management task. **a** Schematic of the data center cooling task, where an RL agent optimizes cooling system based on temperature and silicon data. **b** The 1-year reductions in energy consumption afforded by different reward design methods for different RL algorithms (DQN, SAC, and PPO). The agents that receive rewards from the discovered reward functions perform significantly better than others. **c** Energy consumption over 3, 6, 9, and 12 months. The RL approach using the discovered optimal reward function reduce energy consumption more effectively than do the comparative methods.

## Case Study of Unmanned Systems Control Task

Fig. 8(b) shows the reward curves when agent learning is guided by the reward function of the present work, reward function constructed on the basis of dynamic principles, and other reward design methods. Across the three different RL agents, the proposed approach significantly accelerates learning. For the PPO agent, our approach significantly improves both learning speed and the final performance, enabling the agent to successfully accomplish the task, whereas the hand-designed and other comparison methods fail to do so. For the SAC and TD3 agents, the improvement in learning speed is less pronounced, but our method performs comparable performance and ensures stable learning. These results demonstrate that the proposed reward function discovery approach is particularly useful when traditional rewards are insufficient, consistently ensuring robust performance across different RL algorithms.

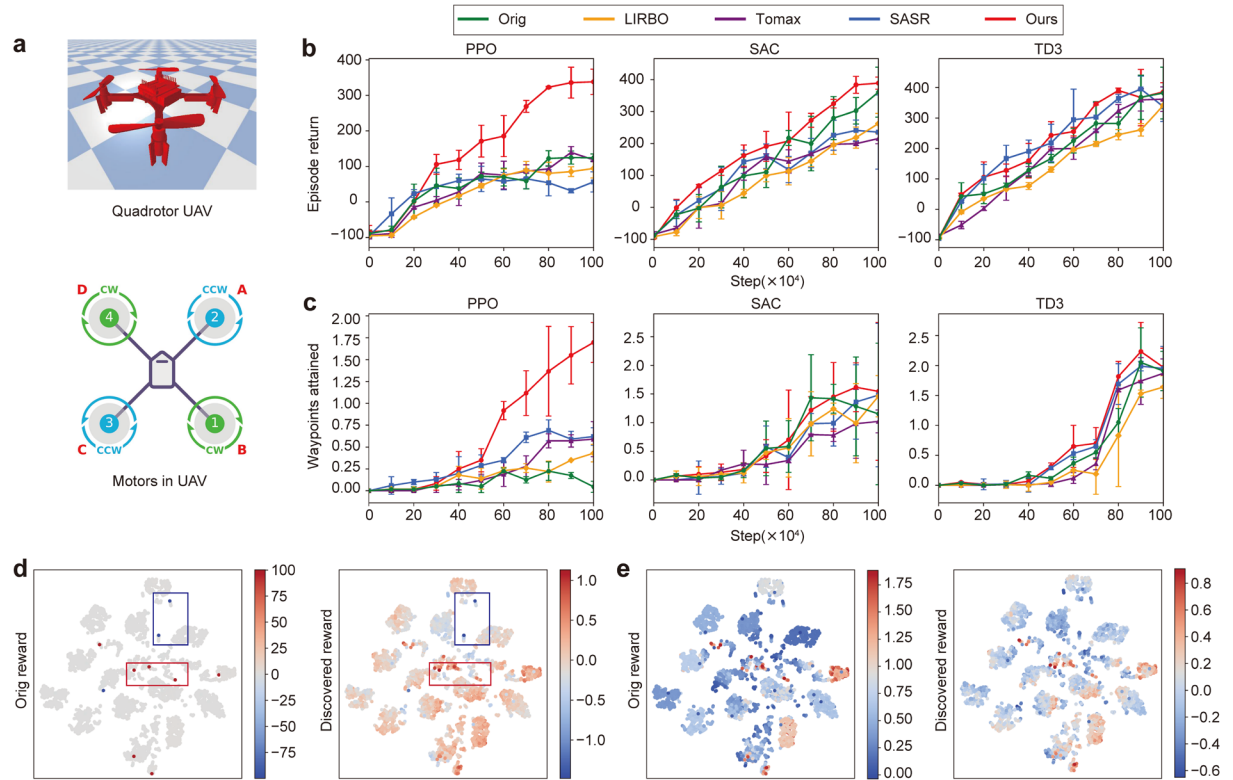


Fig. 8 Evaluation results for unmanned systems control task. a The Quadrotor UAV and the powered motors in a world model. b Learning curves showing reward progression during UAV trajectory tracking using the PPO, SAC, and TD3 agents. The proposed approach is associated with faster convergence and better performance. c The numbers of target trajectory points attained by the UAV during learning. d The state space visualizations of the UAV task, showing the reward distributions of the original reward function derived using dynamic principles and the discovered reward function. The discovered reward function successfully captures critical states. e The state space visualizations after removal of certain states associated with extremely high or low rewards, such as those

**involving collisions or attainment of waypoints. The reward distribution resembles those of reward functions derived using dynamic principles.**

Despite this issue, the empirical results are the strongest part of the paper. There is a solid set of environments, many of them well studied, exhibiting a variety of properties and levels of difficulty. As well the authors do well at varying the baseline RL algorithm itself, which shows generality of the method. And finally of course, the empirical results do generally support this method as being effective at improving learning performance.

**Response:** We sincerely thank the reviewer for the positive assessment of our empirical results. We greatly appreciate the recognition of our efforts to evaluate the proposed method across a diverse and well-studied set of environments, with different baseline RL algorithms. We are pleased that the reviewer considers that our results do generally support our method at improving learning performance. We agree that this comprehensive empirical evaluation represents a key strength of our work, and we are grateful for the affirmation and encouragement. We are willing to do in-depth research in this field. Thanks again.

## # Methods and Theory

Definition 1 (proximal) is not defined precisely enough as hidden inside of ‘nearly equal’ there is both (1) a choice of tolerance  $\epsilon > 0$ , and (2) a choice of distance metric (e.g.  $L_{\infty}$ ). This definition is essentially saying that when the term ‘proximal’ is used it is used to refer to the identified equivalence classes on policies or reward functions in terms of expected returns. Noticing this, it is worth pointing out that the  $\sim$  is a more common choice for notation on equivalence classes than  $\approx$ . Similarly, we might naturally consider  $V_R^\pi$  or  $v_R^\pi$  instead of  $U(\pi, \mathcal{M}[R])$ .

I’ll use this more standard notation for ease of typing comments out.

**Response:** Thanks for the insightful comments. We appreciate the opportunity to clarify these issues and to improve the rigor of our presentation. In response to your concern, in the revised manuscript, we have modified the definition of *proximal* to specify explicitly both the tolerance parameter  $\epsilon$  and the choice of distance metric, thereby eliminating any ambiguity of the term “*nearly equal*”.

In the revised manuscript, we have clearly stated that *proximal* refers to the identified equivalence classes of policies or reward functions in terms of expected returns under the specified metric and tolerance. Furthermore, based on your kind comment, we have replaced the symbol “ $\approx$ ” with “ $\sim$ ” when denoting equivalence classes.

In addition, for clarity, we now use  $G_R^\pi$  to represent the expected return of policy  $\pi$  under reward function  $R$ , in line with standard conventions. We now explicitly define the optimal policy  $\pi^*$  in the policy space  $\Pi$ , which is typically considered fixed.  $\pi_R^*$  is the learned optimal policy under reward

function  $R$ . The notation makes it clear that an agent receives reward signals  $r$  from a reward function  $R$ , interacts with a world model  $\mathcal{M}$ , and seeks to optimize its policy by maximizing the expected return  $G_R^\pi$ , with the ultimate goal of achieving the learned optimal policy  $\pi_R^*$ . The corresponding modification of method and theory are shown as below:

#### Preliminaries

**Notations** Throughout the article, the following symbols are used.  $\mathcal{S}$ : state space.  $\mathcal{A}$ : action space.  $\pi$ : policy.  $\theta$ : parameters of  $\pi$ .  $R$ : reward function.  $\omega$ : parameters of  $R$ .  $\Pi$ : policy space.  $\mathcal{R}$ : reward function space.  $\pi^*$ : optimal policy.  $R^*$ : optimal reward function.  $\pi_R^*$ : learned optimal policy under reward function  $R$ .  $\pi_{R^*}^*$ : learned optimal policy under optimal reward function  $R^*$ , and  $\pi_{R^*}^* = \pi^*$ .  $V_R^\pi$ : state value of policy  $\pi$  under reward function  $R$ .  $Q_R^\pi$ : state-action value of policy  $\pi$  under reward function  $R$ .  $A_R^\pi$ : state-action advantage of policy  $\pi$  under reward function  $R$ .  $G_R^\pi$ : discounted expected return of policy  $\pi$  under reward function  $R$ .  $\xi$ : trajectory, contains  $(s^{(0)}, a^{(0)}, r^{(0)}, \dots, s^{(t)}, a^{(t)}, r^{(t)}, \dots)$ , where  $r^{(t)} = R(s^{(t)}, a^{(t)})$ .  $\delta$ : a small positive constant.  $|\cdot|$ : absolute value of a real number.

The general RL problem without a designed reward function is the world model denoted as  $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \rho_0, T, \gamma)$ . The initial state distribution is  $\rho_0$ , which is the probability distribution over the initial states. The state transfer matrix is  $T: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ , and  $T(s, a, s')$  specifies the transition probability from state  $s$  to  $s'$  when the embodied RL agent chooses action  $a$ . The discount factor  $\gamma \in [0,1]$  determines the relative importance of immediate reward signals compared to future reward signals. We denote the world model  $\mathcal{M}$  with a designed reward function  $R$  as  $\mathcal{M}[R]$ , which is a Markov decision process (MDP). **In MDP, an agent observes state  $s$ , chooses action  $a$  based on policy  $\pi$ , receives rewards signal  $r$  from reward function  $R$ , and optimizes its policy  $\pi$  to  $\pi_R^*$  by maximizing  $G_R^\pi$ :**

$$\pi_R^* \in \arg \max_{\pi \in \Pi} G_R^\pi = \mathbb{E}_{\xi \sim \pi} \left[ \sum_{t=0}^{\infty} \gamma^t \cdot R(s^{(t)}, a^{(t)}) \right] \quad (1)$$

where  $\xi \sim \pi$  indicates that the trajectory  $\xi$  is derived by an agent obey policy  $\pi$  and interact with world model. Here, we assume that when the optimal reward function  $R^*$  is used to guide the agent's policy learning, we have  $\pi_{R^*}^* = \pi^*$ .

**Proposition 1.** Given a reward function  $R$  and the optimal reward function  $R^*$ , if  $R$  is proximal to  $R^*$ , then policy  $\pi$  under  $R$  will be proximal to the optimal policy  $\pi^*$ .

This proposition also has immediate issues with precision. Breaking it down:

\*  $R^*$  is "optimal" with respect to what notion of optimality? It seems instead the term optimal is simply being used to identify a reward as the 'target' for reward discovery.

\*  $R$  is proximal to  $R^*$ , but under what policy? Given there are two policies mentioned, I'll consider if the theory works for either one, but obviously this needs to be specified for this proposition to be complete.

\* “policy  $\pi$  under  $R$  will be proximal to the optimal policy  $\pi^*$ ”, this is not clearly phrased and can have many meanings. Are we saying the two policies are proximal wrt  $R$ ? Or are we saying that  $\pi$  is optimal wrt  $R$  and the two policies are proximal wrt  $R^*$ ?

**Response:** Thanks for the constructive comment. In this work, the term “*optimal reward function*” is used to denote the reward function that is the ultimate objective of the RL agent. Specifically,  $R^*$  is the target or ground-truth reward function that we seek to approximate via reward discovery. We have clarified this in the revised manuscript.

For the notion of *proximal* between reward functions, in the revised manuscript, we have modified Definition 1 to provide a clear and rigorous specification. In the updated definition, “ $R$  is *proximal* to  $R^*$ ” means that, for all policies  $\pi \in \Pi$ , the absolute difference between  $G_R^\pi$  and  $G_{R^*}^\pi$  is bounded by a small positive constant (within a specified tolerance). This ensures that the relationship holds across the policy space, and addresses the ambiguity in the original statement.

For the relationship between the policies, in the revised manuscript, we have rephrased the corresponding text for clarity. Specifically, let  $\pi_R^*$  denote the learned optimal policy under reward function  $R$ , and let  $\pi^*$  denote the learned optimal policy under optimal reward function  $R^*$  (i.e.,  $\pi^* = \pi_{R^*}^*$ ). Our intended meaning is that if  $R$  is *proximal* to  $R^*$ , then the policy  $\pi_R^*$  will be *proximal* to  $\pi^*$  with  $R^*$  and  $R$ , or equivalently, the expected returns will be approximately equal under  $R^*$  and  $R$ . We have clarified this relationship in the revised manuscript.

**Definition 1 (Proximal).** For two policies  $\pi_1$  and  $\pi_2$ , if the absolute difference between  $G_R^{\pi_1}$  and  $G_R^{\pi_2}$  is bounded by a small positive constant, i.e.,  $|G_R^{\pi_1} - G_R^{\pi_2}| \leq \delta$ , then  $\pi_1$  and  $\pi_2$  are proximal with  $R$ , denoted as  $\pi_1 \stackrel{R}{\sim} \pi_2$ . For two reward functions  $R_1$  and  $R_2$ , if the absolute difference between  $G_{R_1}^\pi$  and  $G_{R_2}^\pi$  is bounded by a small positive constant for all policies  $\pi \in \Pi$ , i.e.,  $|G_{R_1}^\pi - G_{R_2}^\pi| \leq \delta, \forall \pi \in \Pi$ , then  $R_1$  and  $R_2$  are proximal, denoted as  $R_1 \sim R_2$ .

**Proposition 1.** Given a reward function  $R$ , if  $R$  is proximal to the optimal reward  $R^*$ , then policy  $\pi_R^*$  will be proximal to the optimal policy  $\pi^*$  with  $R$  and  $R^*$ .

Proof. See Supplementary Theoretical Results.

Definition 1 shows that when the absolute difference between  $G_R^\pi$  and  $G_{R^*}^\pi$  is bounded by a small constant for any policy  $\pi$ , the reward function  $R$  qualifies as an approximately optimal reward function. However, in many practical applications, we are only interested in the performance of the learned optimal policy, i.e.,  $\pi_R^*$ , rather than the entire policy space. Therefore, we relax the optimal condition based on Proposition 1 and only require the expected returns of the policies  $\pi_R^*$  and  $\pi^*$  are approximately equal under  $R^*$ . That means, the optimal reward function  $R^*$  can be obtained through minimizing the absolute difference between the  $G_{R^*}^{\pi_R^*}$  and  $G_{R^*}^{\pi^*}$ , which is termed as the regret  $\mathfrak{L}$  in this work:

$$\mathfrak{L}(\pi_R^*, \pi^* | R^*) = |G_{R^*}^{\pi_R^*} - G_{R^*}^{\pi^*}|. \quad (2)$$

Just after the statement of Proposition 1 there is a fairly strong claim given without proof: “if the policy  $\pi$  optimized under a specific reward function  $R$  is proximal to the optimal policy  $\pi^*$ , then the reward function  $R$  can be considered the optimal reward function  $R^*$ .”

Which reward function are the two policies proximal wrt? I think the one that makes most sense here is  $R$ , so going forward with that the claim is essentially:

$$\pi \sim \pi^* \text{ wrt } R \Rightarrow R \sim R^*$$

First, “can be considered” needs to be specified formally. I think a natural choice would be for this to mean  $R \sim R^* \text{ wrt } \pi^*$ , as this would be fairly nice. However, none of this is true which can be seen with a simple counter example:  $R = R_0 = 0$  (the constant zero reward function).

So, we choose  $R = R_0$ , and for \*any\* policy  $\pi$  we have  $\pi \sim \pi^* \text{ wrt } R$  because all policies have value zero under this reward and as a result all policies are optimal wrt  $R$ , thus  $\pi$  being optimized under  $R$  is satisfied as is the proximal property. However, under any reasonable notion of equivalence  $R_0$  should not be considered the optimal reward function  $R^*$ .

**Response:** Thanks for the constructive comments. We greatly appreciate your careful analysis and the counterexample, which help us recognize the need for a more precise and rigorous formulation of our claim.

Regarding your concern of “Which reward function are the two policies proximal with respect to”, in the revised manuscript, we explicitly clarified this in Proposition 1. Specifically, we now state that the *proximal* between policies is that with respect to the optimal reward function  $R^*$  and the candidate reward function  $R$ , and that our primary focus is on *proximal* under  $R^*$ .

For the second concern of “can be considered”, we have revised our statement to provide a precise definition of *proximal* between  $R$  and  $R^*$  that does not depend on a particular policy. However, given the complexities of real-world applications, we also discuss a relaxed version in which we focus on the *proximal* of the optimal policies induced by the reward functions rather than strict equivalence of the entire policy space. In this context, we consider two reward functions to be *proximal* if their respective optimal policies yield similar expected returns under  $R^*$ .

The counterexample using the constant zero reward function  $R_0$  is well taken. As you pointed out, under  $R_0$ , all policies are optimal and have identical value; any policy  $\pi$  is *proximal* to  $\pi^*$  with respect to  $R_0$ . However, this does not imply that policy  $\pi$  is *proximal* to  $\pi^*$  with respect to  $R^*$ . In the revised manuscript, we have clarified this and modified the statement to avoid overgeneralization. For your reference, we included the revised statement below:

**Definition 1** shows that when the absolute difference between  $G_R^\pi$  and  $G_{R^*}^\pi$  is bounded by a small constant for any policy  $\pi$ , the reward function  $R$  qualifies as an approximately optimal reward function. However, in many practical applications, we are only interested in the performance of the learned optimal policy, i.e.,  $\pi_R^*$ , rather than the entire policy space. Therefore, we relax the optimal condition based on Proposition 1 and only require the expected returns of the policies  $\pi_R^*$  and  $\pi^*$  are approximately equal under  $R^*$ . That means, the optimal reward function  $R^*$  can be

obtained through minimizing the absolute difference between the  $G_{R^*}^{\pi_R^*}$  and  $G_{R^*}^{\pi^*}$ , which is termed as the regret  $\mathfrak{L}$  in this work:

$$\mathfrak{L}(\pi_R^*, \pi^* \mid R^*) = |G_{R^*}^{\pi_R^*} - G_{R^*}^{\pi^*}|. \quad (2)$$

Now turning to regret and Definition 2: “A reward function  $R$  is defined as the optimal reward function  $R^*$  if it minimizes the expected discounted regret  $\mathfrak{L}(\pi, R)$  over the space of possible policies  $\pi$ , where the policy  $\pi$  is learned by an embodied RL agent interacting with the world model  $\mathcal{M}[R]$ .” With  $L$  defined above with:

$$\mathfrak{L}(\pi, R) = v_R^{\pi^*} - v_R^\pi$$

Recall the issue raised previously,  $R_0$  perfectly achieves zero regret here. However, this isn’t the whole of the problem. Notice that  $\pi^*$  is some policy unrelated to a particular choice of  $R$ . So, there will be arbitrarily many reward functions  $R$ , with corresponding ‘optimized’ policies  $\pi$  such that  $\mathfrak{L}(\pi, R) < 0$ , whereas for the desired reward  $R^*$  we have  $\mathfrak{L}(\pi, R^*) = 0$  (for  $\pi$  optimized for  $R = R^*$  as the definition calls for). This objective, or definition for the optimal reward problem, appears to be misspecified at a foundational level.

A small notational note: it might make sense to use  $\mathfrak{L}_{\pi^*}$  to indicate the dependence on the given optimal policy.

**Response:** Thanks for the insightful comment. In the revised manuscript, we have corrected this definition to address the issues raised. Specifically, the optimal reward function is now defined in a manner that presupposes the existence of an optimal reward function  $R^*$ , which ensures that pathological cases in which  $R_0$  achieves zero regret cannot occur. Furthermore, we now explicitly state that for reward function  $R$ , the policy  $\pi_R^*$  is the policy that maximizes the expected return under  $R$ . When evaluating regret using the optimal reward function  $R^*$ , the expected return of any policy  $\pi$  will be less than or equal to that of the optimal policy  $\pi^*$  (i.e.,  $G_{R^*}^{\pi_R^*} \geq G_{R^*}^\pi, \forall \pi \in \Pi$ ), ensuring that regret is always non-negative.

Additionally, we now use  $\mathfrak{L}(\pi_R^*, \pi^* \mid R^*)$  to indicate the dependence of regret on the given optimal policy, making the notation more precise and transparent.

**Definition 2.** *A reward function  $R$  is defined as the optimal reward function if it minimizes the expected regret  $\mathfrak{L}(\pi_R^*, \pi^* \mid R^*)$  over the possible reward space  $\mathcal{R}$ , where the policy  $\pi_R^*$  is learned by an embodied RL agent interacting with the world model  $\mathcal{M}[R]$ . Formally, it is expressed as:*

$$R^* = \arg \min_{R \in \mathcal{R}} \mathbb{E}_{\pi_R^* \sim \mathcal{M}[R]} [\mathfrak{L}(\pi_R^*, \pi^* \mid R^*)], \quad (3)$$

*where  $\mathcal{R}$  denotes the space of all possible reward functions and  $\pi_R^* \sim \mathcal{M}[R]$  indicates that the policy  $\pi_R^*$  is derived under the dynamics influenced by the reward function  $R$ .*

The formal statement of Lemma 1 contains much less information, far less than needed for a precise statement, than the informal description just below it. Additionally, this lemma is entirely unnecessary.  $\bar{R}$  is a fixed reward function,  $\pi^*$  is a fixed policy, therefore the value of  $\pi^*$  under  $\bar{R}$  will be fixed, which is all this lemma seems to be saying.

**Response:** Thanks for the kind comment. As you correctly pointed out, Lemma 1 simply states that, as  $\bar{R}$  is a fixed reward function and  $\pi^*$  is a fixed policy, the value of  $\pi^*$  under  $\bar{R}$  is also fixed and deterministic. This renders the lemma redundant. Based on your suggestion, we have removed it from the revised manuscript and simplified the relevant discussion by directly stating that, given the fixed nature of both the reward function and the optimal policy, the objective function in equation (3) can be reformulated from a minimization of regret to a maximization of the objective value, omitting unnecessary notation. The corresponding modifications are shown as below:

In this work, the function is associated with the ultimate goal of the embodied RL agent and gives a reward when the task is completed. **We then measure the regret of different reward functions under the reward function  $\bar{R}$ :**

$$\mathfrak{L}(\pi_R^*, \pi^* \mid \bar{R}) = \left| G_{\bar{R}}^{\pi_R^*} - G_{\bar{R}}^{\pi^*} \right|. \quad (4)$$

Although many researchers have sought to approximate [65] and minimize [66] regret, and to derive regret bounds [67], direct regret computation and minimization using equation (2) remain challenging. **However, noting that the optimization objective of the optimal policy  $\pi^*$  under the reward function  $\bar{R}$  is deterministic as  $\bar{R}$  is fixed [68], we simplify the objective function in equation (3) and reformulate the process used to discover the optimal reward function from the minimization of regret to maximization of the objective value  $G_{\bar{R}}^{\pi_R^*}$  by omitting  $G_{\bar{R}}^{\pi^*}$ :**

$$\begin{aligned} R^* &= \operatorname{argmin}_{R \in \mathcal{R}} \mathbb{E}_{\pi_R^* \sim \mathcal{M}[R]} \left| G_{\bar{R}}^{\pi_R^*} - G_{\bar{R}}^{\pi^*} \right|, \\ &= \operatorname{argmax}_{R \in \mathcal{R}} \mathbb{E}_{\pi_R^* \sim \mathcal{M}[R]} \left[ G_{\bar{R}}^{\pi_R^*} \right]. \end{aligned} \quad (5)$$

Moving on to the proof of Proposition 1.

The first line of the proof indicates that the intended meaning is  $R \sim R^*$  wrt  $\pi$ . It then goes on to claim that this implies

$$|v_R^\pi - v_{R^*}^\pi| \leq \delta$$

This should be in Definition 1. Equations 22 and 23 both assume a unique optimal policy which is not in general true, so the equals should be replaced with “ $\in$ ” as the argmax may return a set of optimal policies.

The inequalities in 24 are very strange. First, it seems to assume  $R \sim R^*$  wrt  $\pi_R$  and  $R \sim R^*$  wrt  $\pi^*$ . Second, if a suitably formal definition was used then the corresponding thresholds  $\delta > 0$ , would

potentially be different for the two equivalence classes. Ignoring that for the moment, as of (24) what has been assumed without being clear about it is that  $R \sim R^*$  wrt both  $\pi_R$  and  $\pi^*$ .

**Response:** Thanks for the insightful comment. In terms of equations (22-23) in the original manuscript, we fully agree with the reviewer’s comment that the optimal policy is typically non-unique. Accordingly, we have revised the equation (1) in manuscript to use “ $\in$ ” instead of “ $=$ ” to indicate that the argmax may return a set of optimal policies rather than a single policy.

The general RL problem without a designed reward function is the world model denoted as  $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \rho_0, T, \gamma)$ . The initial state distribution is  $\rho_0$ , which is the probability distribution over the initial states. The state transfer matrix is  $T: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ , and  $T(s, a, s')$  specifies the transition probability from state  $s$  to  $s'$  when the embodied RL agent chooses action  $a$ . The discount factor  $\gamma \in [0,1)$  determines the relative importance of immediate reward signals compared to future reward signals. We denote the world model  $\mathcal{M}$  with a designed reward function  $R$  as  $\mathcal{M}[R]$ , which is a Markov decision process (MDP). **In MDP, an agent observes state  $s$ , chooses action  $a$  based on policy  $\pi$ , receives rewards signal  $r$  from reward function  $R$ , and optimizes its policy  $\pi$  to  $\pi_R^*$  by maximizing  $G_R^\pi$ :**

$$\pi_R^* \in \arg \max_{\pi \in \Pi} G_R^\pi = \mathbb{E}_{\xi \sim \pi} \left[ \sum_{t=0}^{\infty} \gamma^t \cdot R(s^{(t)}, a^{(t)}) \right] \quad (1)$$

where  $\xi \sim \pi$  indicates that the trajectory  $\xi$  is derived by an agent obey policy  $\pi$  and interact with world model. Here, we assume that when the optimal reward function  $R^*$  is used to guide the agent’s policy learning, we have  $\pi_{R^*}^* = \pi^*$ .

In terms of equation (24) in original manuscript, we now clarify the definition of “*proximal*” in the context of reward functions in Definition 1 of the Main File. Specifically, according to the revised Definition 1, if two reward functions  $R$  and  $R^*$  are *proximal*, then for any policy  $\pi$ ,  $|G_R^\pi - G_{R^*}^\pi| \leq \delta$ , where  $\delta$  is a small positive constant. This is now explicitly included in Definition 1, as you suggested. With this formalization, both policy  $\pi_R^*$  and optimal policy  $\pi^*$  satisfy the *proximal* definition, which justifies the inequalities in equation (24) in original manuscript.

In principle, the thresholds  $\delta$  could differ for different equivalence classes. However, for simplicity and clarity in our exposition, we use a single  $\delta$  to represent an arbitrarily small tolerance, and this is now clearly stated in the manuscript.

For your convenience, the revised equation (S2) in Supplementary Theoretical Results (*equation (24) in the original manuscript*) is copied here:

Proof of Proposition 1

*Proof of Proposition 1.* Before formally stating Proposition 1 in the main text, we again explain the notations for clarity.  $R^*$  is the optimal reward function.  $\pi^*$  represents the optimal policy in the policy space  $\Pi$ , which is often considered fixed.  $\pi_R^*$  indicates that an agent, guided by reward function  $R$ , optimizes its policy  $\pi$  by maximizing the expected return  $G_R^\pi$ , with the ultimate objective of optimizing its policy to  $\pi_R^*$ . Therefore, we have  $\pi_R^* \in \arg \max_{\pi \in \Pi} G_R^\pi$ . We assume  $\pi_{R^*}^* = \pi^*$ .

By Definition 1 in the main text, if the reward function  $R$  is *proximal* to the optimal reward function  $R^*$ , we have:

$$|G_R^\pi - G_{R^*}^\pi| \leq \delta, \forall \pi \in \Pi, \quad (\text{S1})$$

which gives:

$$\begin{aligned} |G_R^{\pi_R^*} - G_{R^*}^{\pi_R^*}| &\leq \delta, \\ |G_R^{\pi^*} - G_{R^*}^{\pi^*}| &\leq \delta, \end{aligned} \quad (\text{S2})$$

where  $\delta$  is a small positive constant.

The rest of the proof follows correctly, with one partial exception. Recall my concerns about precision of the definition of proximal. Here this comes up again because the result of the proof depends on this lack of precision. Taking  $\epsilon = 4 * \delta$  and saying this means that  $\pi \sim \pi^*$  wrt  $R$ , means that proximal has its meaning further eroded.

Since the definition of proximal you give, and actively rely on for your proof, allows for the distance between value functions to be arbitrarily large (as large as is needed for the narrative) it makes the result much less meaningful.

**Response:** Thanks for the kind comment. We appreciate your positive assessment of the rest of the proof.

Regarding your concern about the precision definition of “*proximal*”, we acknowledge that the use of  $\epsilon = 4\delta$  in the proof could potentially weaken the utility of the result if  $\delta$  is not sufficiently small. As we clarified in the last comment, in the revised Definition 1, the *proximal* parameter  $\delta$  serves as a uniform upper bound on the difference between the expected return for all policies. In our theoretical framework, we treat  $\delta$  as an infinitesimal quantity. Therefore, even when  $\epsilon = 4\delta$ , the bound retains the same infinitesimal order, and the overall *proximal* condition is preserved. We have revised the manuscript to clarify this point and now explicitly state the limitations of the *proximal* definition in terms of its impact thereof on the interpretability and strength of the theoretical guarantees.

Proof of Proposition 1

*Proof of Proposition 1.* Before formally stating Proposition 1 in the main text, we again explain the notations for clarity.  $R^*$  is the optimal reward function.  $\pi^*$  represents the optimal policy in the policy space  $\Pi$ , which is often considered fixed.  $\pi_R^*$  indicates that an agent, guided by reward function  $R$ , optimizes its policy  $\pi$  by maximizing the expected return  $G_R^\pi$ , with the ultimate objective of optimizing its policy to  $\pi_R^*$ . Therefore, we have  $\pi_R^* \in \operatorname{argmax}_{\pi \in \Pi} G_R^\pi$ . We assume  $\pi_{R^*}^* = \pi^*$ .

By Definition 1 in the main text, if the reward function  $R$  is *proximal* to the optimal reward function  $R^*$ , we have:

$$|G_R^\pi - G_{R^*}^\pi| \leq \delta, \forall \pi \in \Pi, \quad (\text{S1})$$

which gives:

$$\begin{aligned} |G_R^{\pi_R^*} - G_{R^*}^{\pi_R^*}| &\leq \delta, \\ |G_R^{\pi^*} - G_{R^*}^{\pi^*}| &\leq \delta, \end{aligned} \quad (\text{S2})$$

where  $\delta$  is a small positive constant.

Applying the triangle inequality to the bound  $|G_R^{\pi_R^*} - G_{R^*}^{\pi_R^*}|$ , we obtain the following equation:

$$|G_R^{\pi_R^*} - G_{R^*}^{\pi_R^*}| \leq |G_R^{\pi_R^*} - G_{R^*}^{\pi_R^*}| + |G_R^{\pi_R^*} - G_{R^*}^{\pi_R^*}| + |G_{R^*}^{\pi_R^*} - G_{R^*}^{\pi^*}|. \quad (\text{S3})$$

By substituting the bounds obtained in equation (S2), we obtain:

$$\begin{aligned} |G_R^{\pi_R^*} - G_{R^*}^{\pi_R^*}| &\leq \delta + |G_R^{\pi_R^*} - G_{R^*}^{\pi_R^*}| + \delta, \\ &\leq |G_R^{\pi_R^*} - G_{R^*}^{\pi_R^*}| + 2\delta, \\ &\stackrel{(a)}{\leq} G_{R^*}^{\pi_R^*} - G_{R^*}^{\pi_R^*} + 2\delta, \\ &\stackrel{(b)}{\leq} (G_R^{\pi_R^*} + \delta) - (G_{R^*}^{\pi_R^*} - \delta) + 2\delta, \\ &\leq G_R^{\pi_R^*} - G_{R^*}^{\pi_R^*} + 4\delta, \\ &\stackrel{(c)}{\leq} 4\delta, \end{aligned} \quad (\text{S4})$$

where (a) holds because  $\pi^*$  is the optimal policy under  $R^*$  (i.e.,  $G_{R^*}^{\pi^*} \geq G_{R^*}^\pi, \forall \pi \in \Pi$ ), which implies that:

$$|G_{R^*}^{\pi_R^*} - G_{R^*}^{\pi^*}| = G_{R^*}^{\pi_R^*} - G_{R^*}^{\pi^*} \geq 0, \quad (\text{S5})$$

and (b) holds on expansion of the absolute value inequalities of equation (S2):

$$\begin{aligned} G_R^{\pi_R^*} - G_{R^*}^{\pi_R^*} &\leq \delta, \\ G_{R^*}^{\pi_R^*} - G_{R^*}^{\pi^*} &\leq \delta, \end{aligned} \quad (\text{S6})$$

and (c) holds because  $\pi_R^*$  is the optimal policy under  $R$ , i.e.,  $G_R^{\pi_R^*} \geq G_R^\pi, \forall \pi \in \Pi$ .

As the absolute difference of  $G_R^{\pi_R^*}$  and  $G_{R^*}^{\pi_R^*}$  is bounded by  $4\delta$ , and  $\delta$  is a small positive constant, therefore, by Definition 1 in the main text,  $\pi_R^* \stackrel{R}{\sim} \pi^*$ .

Similar to equation (S4), applying the triangle inequality to the bound  $|G_{R^*}^{\pi_R^*} - G_{R^*}^{\pi^*}|$ , we have:

$$\begin{aligned}
|G_{R^*}^{\pi_R^*} - G_{R^*}^{\pi^*}| &\leq |G_{R^*}^{\pi_R^*} - G_{R^*}^{\pi_R}| + |G_{R^*}^{\pi_R} - G_{R^*}^{\pi^*}| + |G_{R^*}^{\pi^*} - G_{R^*}^{\pi^*}|, \\
&\leq \delta + |G_{R^*}^{\pi_R} - G_{R^*}^{\pi^*}| + \delta, \\
&\leq |G_{R^*}^{\pi_R} - G_{R^*}^{\pi^*}| + 2\delta, \\
&\leq G_{R^*}^{\pi^*} - G_{R^*}^{\pi_R} + 2\delta, \\
&\leq (G_{R^*}^{\pi^*} + \delta) - (G_{R^*}^{\pi_R} - \delta) + 2\delta, \\
&\leq G_{R^*}^{\pi^*} - G_{R^*}^{\pi_R} + 4\delta, \\
&\leq 4\delta.
\end{aligned} \tag{S7}$$

Therefore, if the reward function  $R$  is *proximal* to the optimal reward function  $R^*$ , we obtain a policy  $\pi_R^*$  that is *proximal* to the optimal policy  $\pi^*$  with  $R$  and  $R^*$ . Thus:

$$R \sim R^* \Rightarrow \pi_R^* \overset{R}{\sim} \pi^*, \pi_R^* \overset{R^*}{\sim} \pi^*. \tag{S8}$$

What do you *actually* care about saying here? For instance, why do we care if the discovered reward function is close (proximal) to the optimal one? Does this result support that goal?

**Response:** Thanks for the constructive comments. Our primary aim when investigating the *proximal* between the discovered reward function and the optimal reward function was to ensure that the learned behaviors of the RL agent were aligned with the intended objectives of the environment. In practice, reward discovery is often required when the true reward function is unknown or difficult to specify directly. By establishing theoretical guarantees that a discovered reward function is *proximal* to the optimal reward function, we aim to ensure that the resulting policy performs nearly as well as does the optimal policy, thereby achieving the desired outcomes in the environment. The results we present are intended to formalize and quantify this relationship, offering a principled way by which to evaluate the quality of discovered reward function.

To be clear, I think that the theory components can be fixed, but I am unsure if they will give interesting results. I strongly suspect that they can, but not without some work.

**Response:** We agree with all above comments that the current theoretical framework could be further strengthened and rendered more meaningful, and we have accordingly refined results to ensure that they yield practically relevant and interesting insights. In the revised manuscript, we have clarified the motivation for studying reward function *proximal* and have discussed the implications of our theoretical findings for real-world RL applications. We appreciate your encouragement and agree that, with further development, the theory can indeed provide valuable contributions to the field.

Reviewer #1 (Remarks on code availability):

I have looked at the code fairly briefly. It is built on top of CleanRL to handle the agent-environment interaction and RL optimization components. The reward function learning code is undocumented but it is well formatted and easy to read and understand even without further documentation or comments. Overall, I would say that the code is well done because it is very easy to contrast with the baseline CleanRL implementation to see differences (which are almost entirely modular).

Yes, there is a README that should be sufficient for testing out the code. There is an odd pattern of 'typo' like misspellings of words, but done consistently (e.g. reward\_funciton, optimaze\_reward), which is worth fixing.

**Response:** Thanks for the positive feedback on the clarity, modularity, and readability of our code, and for highlighting the ease of comparison with the baseline CleanRL implementation. In the revised manuscript, we have added comprehensive comments and detailed documentation of the code. We believe these that this will aid researchers who wish to use, understand, or extend our approach.

We also thank the reviewer for examining the README file and then highlighting the unfortunate misspellings in variable and function names. In response, we have carefully reviewed the code and corrected all errors. We believe these revisions will improve the usability and readability of our code, and we thank the reviewer for bringing this to our attention.

In summary, we sincerely thank Reviewer #1 for the detailed attention to, and review of, our work. All comments improved the quality of our manuscript, especially the perspectives taken and the theoretical statements.

## Response to Reviewer #2 (Remarks to the Author):

### > Summary

For embodied agents in Embodied AI (EAI), reinforcement learning (RL) is a promising method for realizing intelligent behavior to solve complex problems. However, RL suffers from the so-called reward-engineering problem, in which reward quality heavily depends on human expertise and is critical for improving agent performance. In this paper, the authors propose an optimal reward-function discovery method. Using a meta-gradient approach, they implement the reward discovery process as bilevel optimization: the upper level refines the reward function, and the lower level performs the traditional RL step. With the proposed method, agents discovered the optimal policy in fewer episodes, indicating high sample efficiency. The method worked well on both energy-management and UAV-control tasks. The authors also provide theoretical results for the proposed approach.

**Response:** Thanks for concisely summarizing the contributions of our proposed method for optimal reward-function discovery. We appreciate the thoughtful evaluation of our approach, including the meta-gradient bilevel optimization framework, the demonstrated sample efficiency, and the successful applications to both energy-management and UAV-control tasks. We also value the reviewer's affirmation of our theoretical analysis.

Below, we address the comments one-by-one, with the aim of clarifying further the benefits of our approach and describing the experimental analysis in more detail.

### > What are the noteworthy results?

The authors propose a meta-gradient-based bilevel optimization method to optimize the reward function. While traditional approaches rely on inverse RL or active learning, the proposed method operates without them. It generates a more informative dense reward signal than the inefficient sparse rewards of manually designed functions, thereby accelerating the agents' learning speed and improving their performance.

**Response:** Thanks for the overall positive assessment and for highlighting the advantages of our meta-gradient based bilevel optimization approach, compared to traditional inverse RL and active learning methods. We appreciate the reviewer's thoughtful view that our method generates more informative and dense reward signals than do other methods, accelerating agents learning speed and improving their performance. Thanks again.

> Will the work be of significance to the field and related fields? How does it compare to the established literature? If the work is not original, please provide relevant references.

Their proposed method is novel, but it closely resembles [14] or [39] (in the appendix). Although the implementation and design differ, the core idea remains very similar.

**Response:** Thanks for the kind comment. As the reviewer stated, the proposed method closely resembles those of [1] or [2] ([14] and [39] in the original Appendix), but both the implementation and design differ. Specifically, [1] ([14] in the original Appendix) formulated intrinsic reward learning as a bilevel optimization problem and leveraged hyperparameter optimization and conditional layer normalization when learning intrinsic rewards. In contrast, our core concept is direct optimization of the reward function *per se*, rather than designing a reward function and learning intrinsic rewards to assist it. Our approach integrates reward function optimization into the RL training loop; this enables autonomous discovery of optimal reward function. [2] ([39] in the original Appendix) introduced a meta-reward network that employed bilevel optimization of reward and policy learning to improve the efficiency of preference-based learning when labeled preference data were scarce. However, our method optimizes the reward function in an online RL interactive setting, thus reducing any reliance on preference datasets or prior knowledge. This fundamentally differs from the core objective of [2].

[1] Stadie, B., Zhang, L., Ba, J.: Learning intrinsic rewards as a bi-level optimization problem. In: Conference on Uncertainty in Artificial Intelligence, pp. 111 – 120 (2020).

[2] Liu, R., Bai, F., Du, Y., Yang, Y.: Meta-reward-net: implicitly differentiable reward learning for preference-based reinforcement learning. In: Proceedings of the 36th International Conference on Neural Information Processing Systems, pp. 22270–22284 (2022).

As the reviewer also mentioned in the previous comment: “*While traditional approaches rely on inverse RL or active learning, the proposed method operates without them*”, indeed, our approach aims to optimize the reward function directly during the RL interaction; we do not employ inverse RL or active learning. To clarify further the differences between our work and existing studies, we have added text to the Introduction in the Main File and Supplementary Related Works accordingly. The corresponding contents are copied here for your convenience:

#### Introduction

The discovery of the optimal reward function within the embodied RL agent is conceptualized as a meta learning problem, enabling the integration of optimal reward function discovery with the pursuit of the optimal policy. **Although meta learning problems have been explored in prior works [49, 56–58], we here propose a meta gradient algorithm specifically tailored to this setting, creating a bilevel optimization framework in which RL policy optimization is the lower-level process and the discovery of the optimal reward function is the upper-level process.**

#### Supplementary Related Works

##### Meta Learning in Reinforcement Learning

Meta learning methods have been widely applied to RL tasks [23–28] including hyperparameter optimization [25, 29], model learning [24, 30], and algorithm generalization [28]. In this paper, we focus

on a meta learning framework specific to reward function learning [31, 32]. One approach used to reward meta learning involves optimizing the reward functions across diverse task distributions for adaptation to new tasks [33, 34]. A promising alternative is to establish a bilevel optimization framework based on meta learning. Following the theory proposed in [21], researchers in [14] posited that reward learning could be framed as a bilevel optimization problem, where policy optimization in an inner loop shapes the rewards and the reward function parameters are optimized in the outer loop. Similarly, [2] viewed reward learning as a bilevel optimization framework but focused on how the optimized reward function is integrated within the framework. Our approach aligns with this idea. The framework has been expanded in various ways. For example, [35–37] incorporated it into multi-agent RL, allowing each agent to learn a personal reward function. Currently, the key challenges posed by the use of bilevel framework, to optimize reward functions, include effective definition of the upper-level optimization objectives. Usually, the definition is the same as that of the optimal reward function, as explored in [38], which sought to optimize imperfect rewards. Meanwhile, [39] and [5] focused on alignment of reward functions with human preferences. **Our approach differs fundamentally from previous methods in terms of how the upper-level optimization objective is defined within the bilevel framework. Prior works often rely on indirect or heuristic formulations, but our method introduces a direct and principled objective. Specifically, we define the optimal reward function by explicitly minimizing policy regret. The policy performance of the agent is the metric used for reward optimization, which effectively resolves the long-standing challenge of specifying the upper-level optimization goal. Thus, our approach establishes a clear and theoretically grounded connection between reward function learning and policy optimization, setting it apart from existing methods.**

> Does the work support the conclusions and claims, or is additional evidence needed?

Considering the previous comment, comparisons with the methods in [14] or [39] would clarify that the authors' proposed approach is more effective than theirs; the work currently only compares against naive RL baselines. It would also be preferable to include visualizations of the learned reward to assess what reward was actually learned. While this may be difficult for high-dimensional control tasks, it could at least be done for classical control tasks.

**Response:** Thanks for the constructive comments.

For the comment of “*comparisons with the methods in [14] or [39] would clarify that the authors' proposed approach is more effective than theirs; the work currently only compares against naive RL baselines*”, in the original manuscript, two existing baseline rewards, including sparse reward signals and manually well-designed reward functions, were used to verify the effectiveness of the proposed approach in terms of the episode reward and convergence speed. The selected methods are commonly used for comparisons in the existing studies. In the revised manuscript, based on the reviewer's comment, we included the direct experimental comparison of [1] ([14] in the original appendix). The authors of [1] developed a bilevel optimization framework to learn intrinsic rewards that augmented external reward signals, whereas our approach directly optimizes the reward function *per se*. In [2] ([39] in the original appendix), reward learning required

large-scale human preference datasets which are not readily accessible for our current study. However, if the human preference datasets of the case studies are available, a performance comparison with [2] would be possible.

In addition, we extensively reviewed the literature and selected several recent state-of-the-art reward design methods [3,4]. These served as additional baselines to further demonstrate the effectiveness of our approach. Specifically, [3] presented an efficient adaptive reward shaping mechanism that used the success rate for shaping, and the kernel density estimation (KDE) and random Fourier features (RFF) to derive a beta distribution from which the success rate was sampled. In [4], the key role of reward maximization in RL was also emphasized. However, unlike the reward function design method, [4] directly maximized the maximum value of the reward signal obtained instead of maximizing the cumulative return of RL.

[1] Stadie, B., Zhang, L., Ba, J.: Learning intrinsic rewards as a bi-level optimization problem. In: Conference on Uncertainty in Artificial Intelligence, pp. 111 – 120 (2020).

[2] Liu, R., Bai, F., Du, Y., Yang, Y.: Meta-reward-net: implicitly differentiable reward learning for preference-based reinforcement learning. In: Proceedings of the 36th International Conference on Neural Information Processing Systems, pp. 22270–22284 (2022).

[3] Ma, H., Luo, Z., Vo, T.V., Sima, K., Leong, T.-Y.: Highly efficient self-adaptive reward shaping for reinforcement learning. In: Proceedings of the 13th International Conference on Learning Representations (2025).

[4] Veviurko, G., Böhmer, W., De Weerd, M.: To the max: reinventing reward in reinforcement learning. In: Proceedings of the 41st International Conference on Machine Learning, pp. 49455–49470 (2024).

The results of these additional comparisons are presented in Fig. 3(c-e), Fig. 5(c-e), Fig. 7(b-c), Fig. 8(b-c), demonstrating that our method achieves superior performance in terms of learning stability and convergence speed. The corresponding comparison results are copied here for your convenience:

## Results

Furthermore, **the reward functions discovered by the proposed approach are compared to the baseline rewards, including sparse reward signals, manually designed reward functions, and rewards from three state-of-the-start reward learning methods, including LIRBO [49], Tomax [30], SASR [53].** The comparison serves to illustrate further the relationship between the reward functions and the tasks, as well as how these functions help embodied RL agents to achieve better performance. **The experimental setups for both the numerical experiments and the real-world tasks, and the details on how the comparative methods are implemented, are described in the Supplementary Experimental Details.**

### Case Study of Sparse-Reward Tasks

**Fig. 3(c-d) shows the average episodic rewards of embodied RL agents when using our proposed approach compared to those of agents that receive sparse reward signals and reward signals from baseline methods. Agents under the proposed reward function outperform the baseline agents in terms of both convergence speed and learning stability.** When receiving sparse reward signals, the agent finds it challenging to learn the optimal policy and achieve fast convergence because of the limited number of samples that can be effectively utilized, resulting in low sample efficiency. This low efficiency requires the agent to engage in extensive trial-and-error, thereby slowing the convergence rate. **Reward function design methods have relieved this challenge. LIRBO replaces sparse rewards with intrinsic reward signals, and SASR uses a shaping auxiliary function to the same end. Tomax directly learns historical maximum rewards but sometimes does not perform well, as the initial random exploration in sparse-reward tasks does not deliver an adequate reward to the agent.** In contrast, the proposed approach delivers dense reward signals directly, thereby improving the sample efficiency.

Fig. 3(e) shows the average episodic returns and standard errors throughout the entire learning process. **The PPO and DQN agents that employ our proposed approach comprehensively outperform the agents under baseline methods. Additionally, as shown in Fig. 3(c-e), our approach enhances learning stability and the convergence speed.**

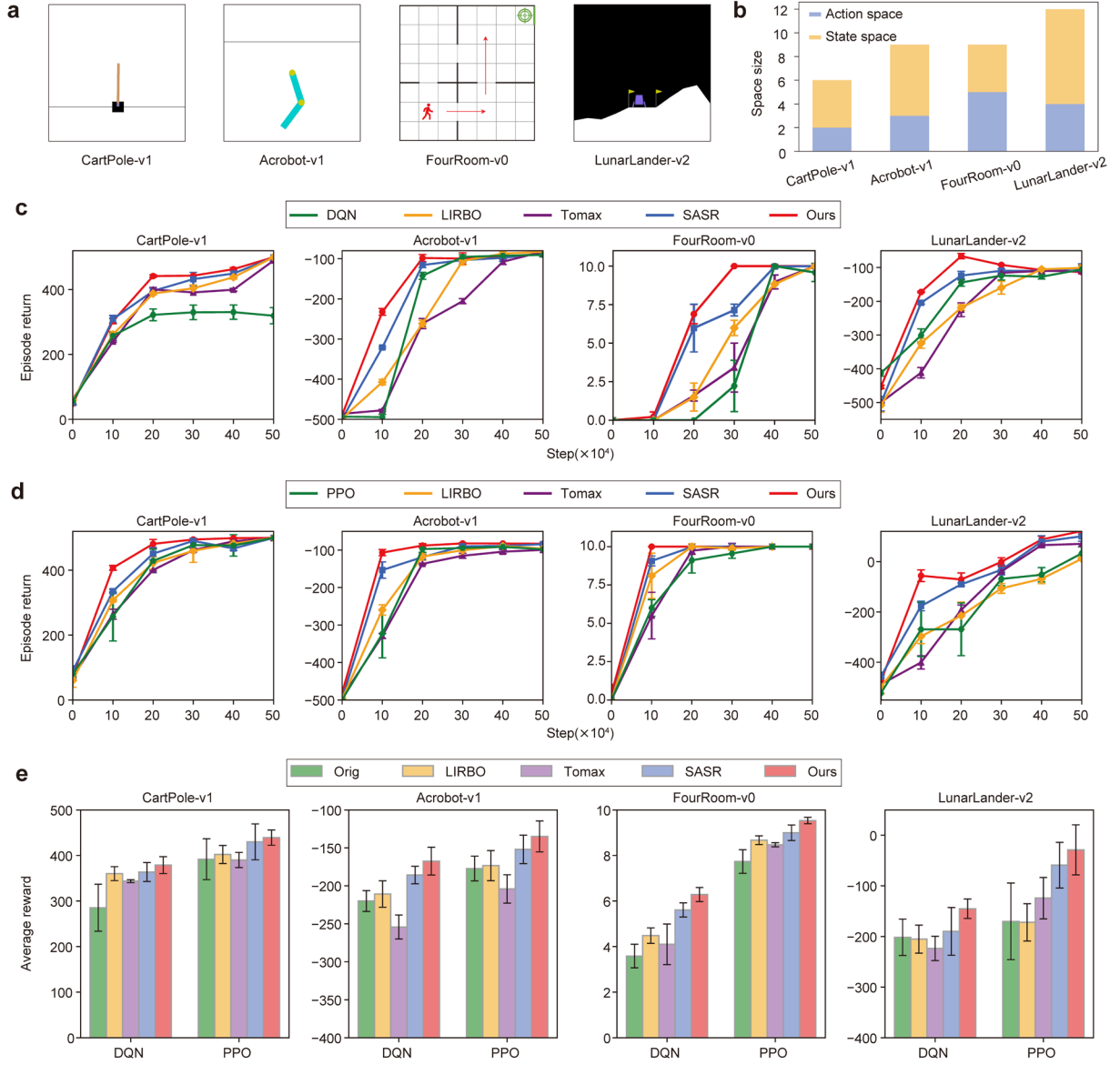


Fig. 3 Evaluation results for sparse-reward tasks. **a** The environments used in the sparse-reward task, including *CartPole-v1*, *Acrobot-v1*, *FourRoom-v0* and *LunarLander-v2*. **b** The sizes of the observation and action spaces of the four tasks, illustrating the varying complexity levels. **c** The reward curves created during the learning processes of the DQN agent with different reward functions on different tasks. The solid lines show the average rewards and the error bars indicate the standard deviations across five trials using different random seeds. **d** The reward curves created during the learning processes of PPO agent with different reward functions on different tasks. **e** The average rewards obtained by the embodied RL agent during all episodes, with the standard deviations of the rewards received when different algorithms performed each task.

### Case Study of High-Dimensional Control Tasks

Fig. 5(c-d) shows the cumulative reward curves of embodied RL agents after each episode across the four tasks, with the error bars indicating the variabilities across the five parallel experiments. The proposed approach exhibits superior performance, even outperforming the reward functions of OpenAI designed by humans. **Compared to other baseline methods, our approach is more effective when PPO and SAC agents are employed. The convergence speed is faster and the episodic returns are higher. SASR and LIRBO perform similarly, but SASR achieves better when a PPO agent is applied because SASR samples the success rate of historical data.** Fig. 5(c-d) also reveals that the proposed approach accelerates agent convergence and improves the cumulative expected returns, particularly when the number of agent joints increases. Fig. 5(e) presents the means and variance of the cumulative returns given during learning and compares the rewards of the proposed approach with those afforded by the baseline rewards. It is clear that, for both the SAC and PPO agents, the proposed approach not only outperforms the baseline but also reduces risk induced by randomness.

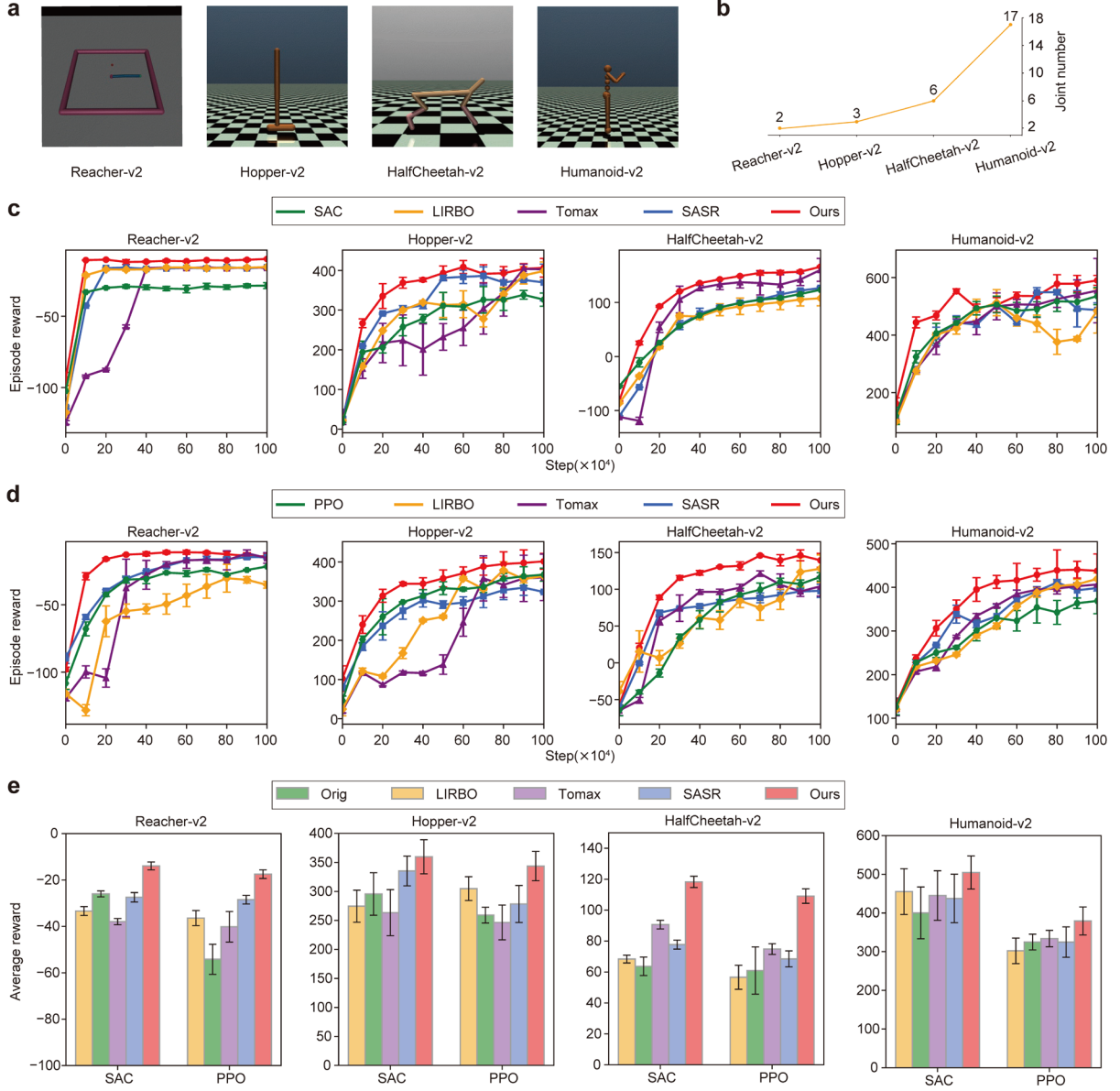


Fig. 5 Evaluation results for high-dimensional control tasks. a The high-dimensional control tasks, including *Reacher-v2*, *HalfCheetah-v2*, *Hopper-v2* and *Humanoid-v2*. b The number of controllable joints increases from *Reacher-v2* to *Humanoid-v2*, reflecting the increasing complexity of the tasks. c The reward curves during the learning processes of the SAC agent across the four tasks. The solid lines show the average episode rewards and the error bars indicate the standard deviations over five trials with different random seeds. d The reward curves during the learning processes of the PPO agent across the four tasks. e The average rewards with their standard deviations for different RL agents and different tasks. The results highlight the improved performance of the proposed approach across all tasks.

## Case Study of Real-World Tasks

### Case Study of Energy Management Task

Fig. 7(b) shows the rate of reduction in energy consumption when applying baseline RL and RL guided by the reward design approaches in data centers over a 1-year period. The numerical values are the percentage reduction rates. Fig. 7(c) illustrates the total energy consumption over 3, 6, 9, and 12 months for each RL agent. Although all RL methods reduce energy consumption, the reward function discovery approach further reduces consumption, particularly over longer times. Notably, as RL algorithms become more advanced, from the DQN to the more sophisticated SAC, the energy reduction rate increases (from 21% for DQN to 52% for SAC). The superiority of the proposed approach is more obvious when compared with the baseline models. With our method, all RL agents (including DQN, PPO, and SAC) achieve energy reductions that exceed 60%. These results highlight the generality and effectiveness of our approach for RL agents involved in energy management and other tasks. Therefore, the proposed optimal reward function discovery approach can be applied to RL agents operating in data centers and elsewhere.

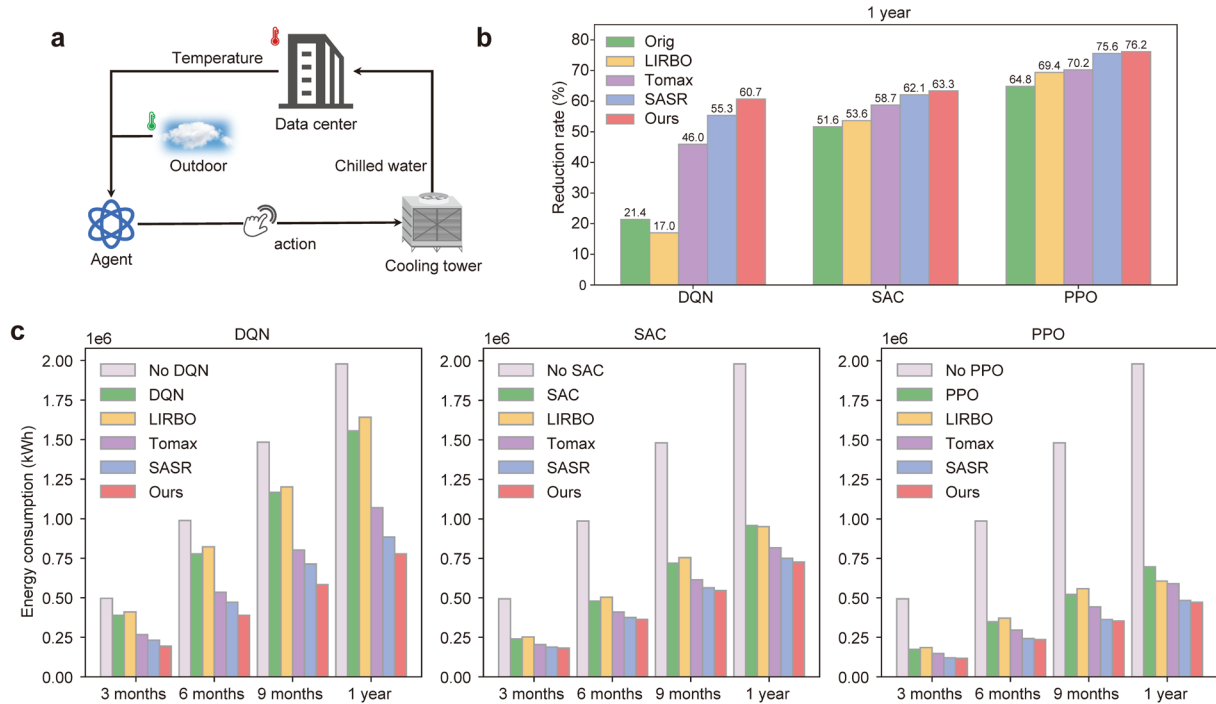


Fig. 7 Evaluation results for the energy management task. **a** Schematic of the data center cooling task, where an RL agent optimizes cooling system based on temperature and silicon data. **b** The 1-year reductions in energy consumption afforded by different reward design methods for different RL algorithms (DQN, SAC, and PPO). The agents that receive rewards from the discovered reward functions perform significantly better than others. **c** Energy consumption over 3, 6, 9, and 12 months. The RL approach using the discovered optimal reward function reduce energy consumption more effectively than do the comparative methods.

## Case Study of Unmanned Systems Control Task

Fig. 8(b) shows the reward curves when agent learning is guided by the reward function of the present work, reward function constructed on the basis of dynamic principles, and other reward design methods. Across the three different RL agents, the proposed approach significantly accelerates learning. For the PPO agent, our approach significantly improves both learning speed and the final performance, enabling the agent to successfully accomplish the task, whereas the hand-designed and other comparison methods fail to do so. For the SAC and TD3 agents, the improvement in learning speed is less pronounced, but our method performs comparable performance and ensures stable learning. These results demonstrate that the proposed reward function discovery approach is particularly useful when traditional rewards are insufficient, consistently ensuring robust performance across different RL algorithms.

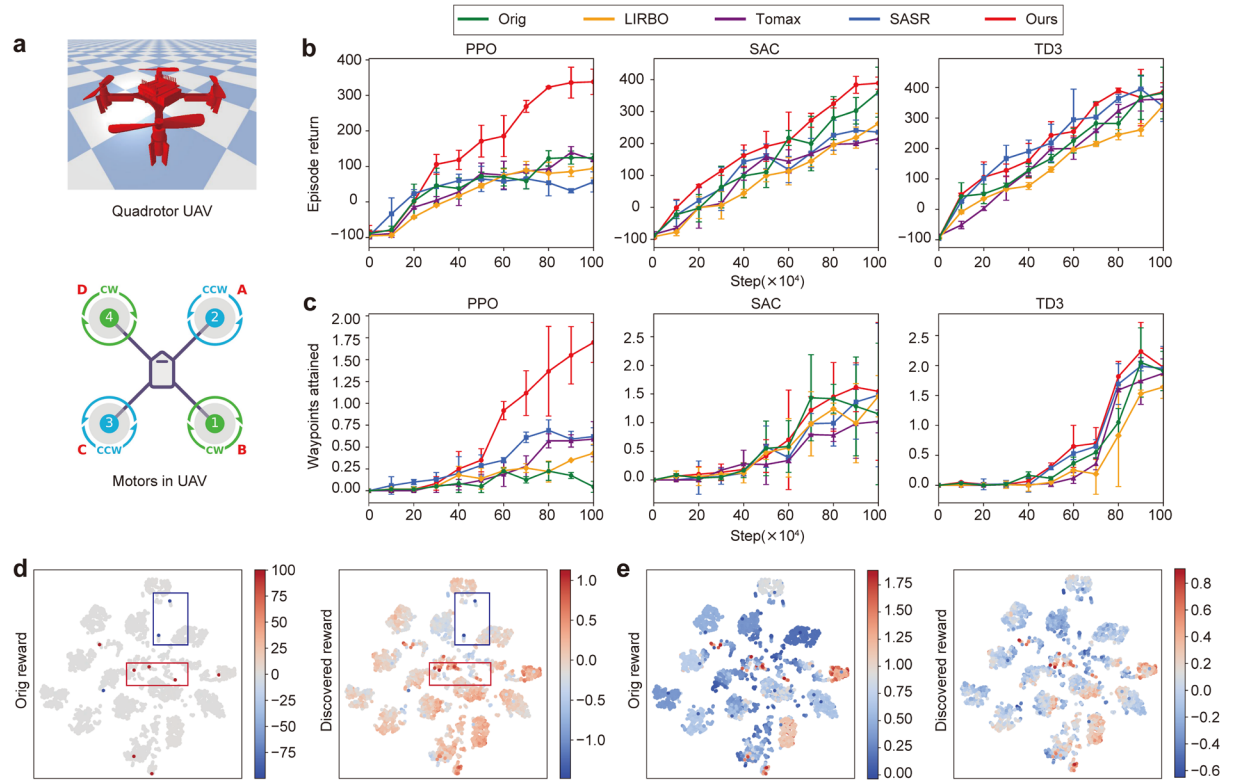


Fig. 8 Evaluation results for unmanned systems control task. a The Quadrotor UAV and the powered motors in a world model. b Learning curves showing reward progression during UAV trajectory tracking using the PPO, SAC, and TD3 agents. The proposed approach is associated with faster convergence and better performance. c The numbers of target trajectory points attained by the UAV during learning. d The state space visualizations of the UAV task, showing the reward distributions of the original reward function derived using dynamic principles and the discovered reward function. The discovered reward function successfully captures critical states. e The state space visualizations after removal of certain states associated with extremely high or low rewards, such as those

**involving collisions or attainment of waypoints. The reward distribution resembles those of reward functions derived using dynamic principles.**

For the comment of “*It would also be preferable to include visualizations of the learned reward to assess what reward was actually learned. While this may be difficult for high-dimensional control tasks, it could at least be done for classical control tasks*”, in the original manuscript, we have employed t-SNE to visualize the learned reward representations, including the high-dimensional control tasks and the unmanned systems control task. However, as the reviewer stated, the complexity of high-dimensional spaces makes such visualizations challenging and sometimes less interpretable, but the reward function visualization is more feasible and interpretable for classical control tasks. To address this, we have selected a classic control task (*Acrobot-v1*) with a discrete action space, which allows for a clear visualization of the reward function. The agent in the *Acrobot-v1* system consists of two links connected linearly to form a chain, with one end of the chain fixed. The joint between the two links is actuated. The goal is to apply torques on the actuated joint to swing the free end of the linear chain above a given height while starting from the initial state of hanging downwards. The state space observed by the agents includes the sines and cosines of the two rotational joint angles as well as their angular velocities:  $(\cos(\theta_1), \sin(\theta_1), \cos(\theta_2), \sin(\theta_2), \dot{\theta}_1, \dot{\theta}_2)$ . The action space is discrete, deterministic, and represents the torque applied on the actuated joint between the two links.

In the revised manuscript, Fig. 4(e-h) illustrates the reward surfaces of the sparse reward function and the discovered reward function for each discrete action in the *Acrobot-v1* task, visualized as a function of the two joint angles  $\theta_1$  and  $\theta_2$ , while fixing the angular velocities  $\dot{\theta}_1$  and  $\dot{\theta}_2$  at 0 (for the distribution of the reward functions at different angular velocities, we present them in the Supplementary Information). Each surface illustrates the reward values of our reward function for a specific action: application of -1 torque (action 0), 0 torque (action 1), and 1 torque (action 2) to the actuated joint. The results reveal that our reward function assigns higher values to actions 0 and 2 (-1 and 1 torque, respectively) when the joint angles are away from the downward resting position, encouraging the agent to apply torque when momentum generation is favorable. In contrast, action 1 (0 torque) receives relatively higher rewards near the neutral region, reflecting a learned preference for stabilizing when the system is aligned. This structured reward landscape aligns well with the task objective of swinging the outer link above a target height and demonstrates that the reward function of this work captures meaningful agent control behavior consistent with the dynamics of the *Acrobot-v1* system. The corresponding analyses are copied here for your convenience:

**Fig. 4(e-h) illustrates the reward surfaces of the sparse reward function and the discovered reward function for each discrete action in *Acrobot-v1* task. The reward surfaces are visualized as functions of the two rotational joint angles,  $\theta_1$  and  $\theta_2$ , when their angular velocities  $\dot{\theta}_1$  and  $\dot{\theta}_2$  are 0. Fig. 4(e) shows that the sparse reward function assigns a reward of -1 for all state and action. Fig. 4(f-h) shows the rewards assigned by our reward function for specific actions: application of -1 torque (action 0), 0 torque (action 1), or 1 torque (action 2) to the actuated joint. The results show that the**

reward function of this work assigns higher values to actions involving -1 and 1 torque when the joint angles deviate from the downward resting position. This encourages the agent to apply torque in situations when momentum generation can be effectively created. Conversely, the action corresponding to 0 torque receives higher rewards when the agent is near the neutral region, indicating a learned preference for stabilization or energy conservation when the system is aligned. This structured reward landscape is consistent with the task objective of swinging the outer link above a target height, demonstrating that the reward function of this work successfully captures meaningful control behaviors aligned with the dynamics of the *Acrobot-v1* task. Visualizations of the reward functions at different angular velocities can be in found the Supplementary Figs. 2-4.

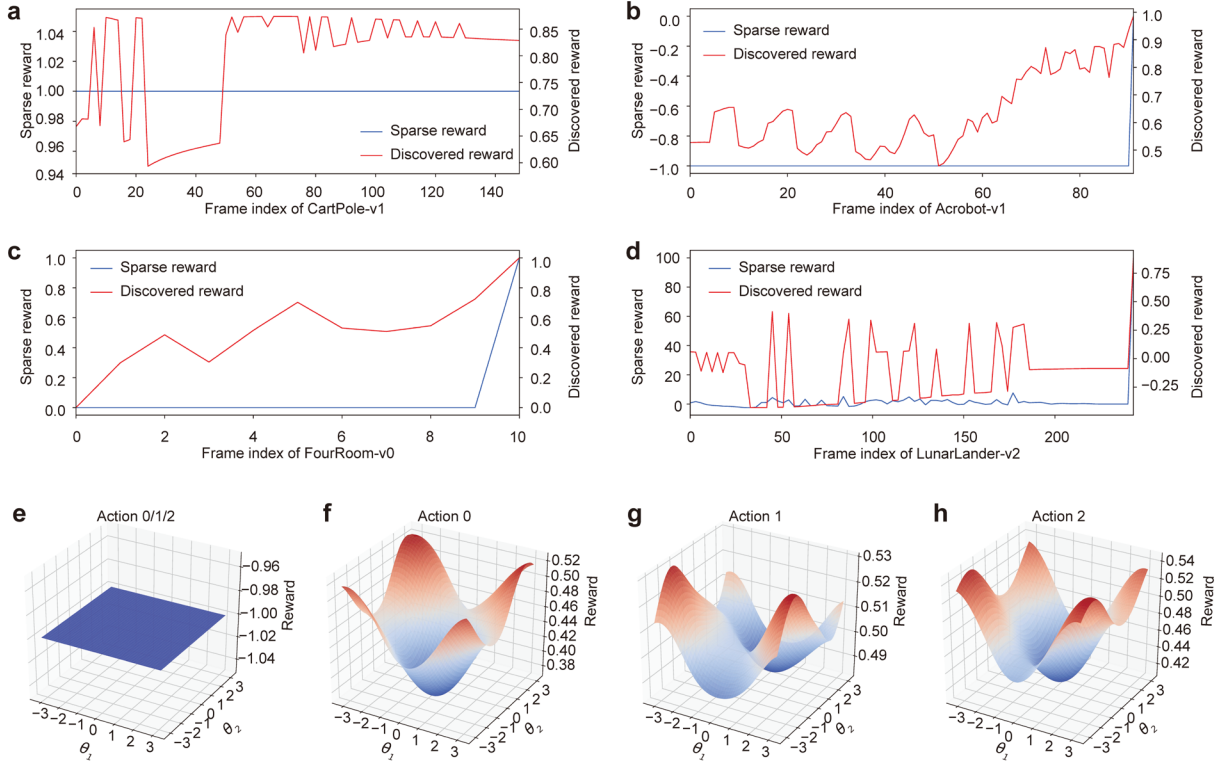


Fig. 4 Reward visualization. a Reward signals during a single interaction episode of the *CartPole-v1* task. The sparse reward signals (blue lines) and the reward signals discovered in this work (red lines) are shown. b Reward signals during a single interaction episode of the *Acrobot-v1* task, showing the difference between the sparse and discovered reward signals. c Reward signals during a single interaction episode of the *FourRoom-v0* task, illustrating the increased signal density achieved in this work. d Reward signals during a single interaction episode of the *LunarLander-v2* task. e The surface of the sparse reward function. For any state and action, the sparse reward function always delivers a reward signal of -1. f The reward surface for action 0 (-1 torque) of *Acrobot-v1* as a function of the joint angles  $\theta_1$  and  $\theta_2$  with angular velocities fixed at 0. g The reward surface for action 1 (0 torque). h The reward surface for action 2 (1 torque).

> Are there any flaws in the data analysis, interpretation and conclusions? Do these prohibit publication or require revision?

In the energy-management task, the consumption reduction achieved by SAC and PPO appears smaller than that of DQN (Fig. 7(c)). In the UAV control task, Fig. 8(b) does not clearly demonstrate an acceleration in learning: while PPO shows noticeable improvement, the gains for TD3 and SAC are comparatively modest. Moreover, the advantage illustrated in Fig. 8(e) over Fig. 8(d) is not immediately apparent. It would be helpful to clarify the form of the reward function that was learned, as this would shed light on how the proposed method drives performance improvements. Finally, the t-SNE visualizations are somewhat challenging to interpret; additional explanation or annotation could improve their clarity.

**Response:** Thanks for the constructive comments.

For the first comment of “*In the energy-management task, the consumption reduction achieved by SAC and PPO appears smaller than that of DQN (Fig. 7(c))*”, the height of each bar in Fig. 7(c) is the total energy consumption under the control of each RL agent. Since PPO and SAC are generally considered to be more advanced than DQN, the DQN consumes more energy, as revealed by the baseline results. Using our proposed reward discovery method, the performance of all baseline models (including DQN, PPO, and SAC) improved. The energy demand fell. The absolute reduction is smaller for PPO and SAC than DQN because PPO and SAC are already lower-energy, whereas DQN could be improved in this respect.

**Fig. 7(b) shows the rate of reduction in energy consumption when applying baseline RL and RL guided by the reward design approaches in data centers over a 1-year period.** The numerical values are the percentage reduction rates. Fig. 7(c) illustrates the total energy consumption over 3, 6, 9, and 12 months for each RL agent. **Although all RL methods reduce energy consumption, the reward function discovery approach further reduces consumption, particularly over longer times. Notably, as RL algorithms become more advanced, from the DQN to the more sophisticated SAC, the energy reduction rate increases (from 21% for DQN to 52% for SAC). The superiority of the proposed approach is more obvious when compared with the baseline models. With our method, all RL agents (including DQN, PPO, and SAC) achieve energy reductions that exceed 60%. These results highlight the generality and effectiveness of our approach for RL agents involved in energy management and other tasks.** Therefore, the proposed optimal reward function discovery approach can be applied to RL agents operating in data centers and elsewhere.

For the second comment of “*Fig. 8(b) does not clearly demonstrate an acceleration in learning: while PPO shows noticeable improvement, the gains for TD3 and SAC are comparatively modest*”, in the unmanned systems control task, the baseline reward function was constructed using expert knowledge based on the UAV dynamics. This hand-designed reward function enables RL agents to complete the task, but may not afford stable and consistent support across various RL algorithms. For example, as shown in Fig. 8(b), SAC and TD3 agents perform well when the hand-designed reward function is in play, but the PPO agent struggles; performance is not well. In contrast, the rewards discovered by our approach allow various RL agents (including PPO, SAC,

and TD3) to exhibit better and more stable performance. The proposed method significantly improved the action of the PPO agent, but not those of the TD3 and SAC agents. Therefore, Fig. 8(b) indeed shows that our method enhances the performance of various RL agents compared to that of a hand-designed reward function.

**Fig. 8(b) shows the reward curves when agent learning is guided by the reward function of the present work, reward function constructed on the basis of dynamic principles, and other reward design methods. Across the three different RL agents, the proposed approach significantly accelerates learning. For the PPO agent, our approach significantly improves both learning speed and the final performance, enabling the agent to successfully accomplish the task, whereas the hand-designed and other comparison methods fail to do so. For the SAC and TD3 agents, the improvement in learning speed is less pronounced, but our method performs comparable performance and ensures stable learning. These results demonstrate that the proposed reward function discovery approach is particularly useful when traditional rewards are insufficient, consistently ensuring robust performance across different RL algorithms.**

For the third comment of *“the advantage illustrated in Fig. 8(e) over Fig. 8(d) is not immediately apparent. It would be helpful to clarify the form of the reward function that was learned, as this would shed light on how the proposed method drives performance improvements. Finally, the t-SNE visualizations are somewhat challenging to interpret; additional explanation or annotation could improve their clarity”*, in our reward discovery approach, the reward function was parameterized and the weights were continuously updated via meta gradient-based optimization. As the reviewer notes, it is challenging to derive a closed-form analytical expression for the learned reward function. To address this, we correlated each high-dimensional state in the state space with the corresponding reward value, and then employed the t-SNE method to project the high-dimensional state representations into 2D space for visualization. This illustrates how the reward function varies across different regions of the state space and yields insights into the structure and distribution of the learned reward functions. Specifically, as shown in Fig. 6 and Fig. 8, similar states map closely in the 2D space. The color of each point indicates the reward signal (red: higher rewards; blue: lower rewards). We agree that it is not easy to interpret the t-SNE visualizations, and we have added additional explanatory text and annotations in Supplementary Experiment Details in the updated manuscript.

Regarding Fig. 8(d) and Fig. 8(e) in our original manuscript, each figure contains two visualizations. The left subplots are the reward distributions across the entire state space and the right subplots are the distributions after removing critical states (i.e., when the UAV attains a target trajectory point or crashed, typically associated with very high or low rewards respectively). Fig. 8(d) shows the original hand-designed reward function based on dynamic principles. The reward distribution is very sparse (most states receive negligible rewards; only a few states are linked to significant positive or negative values). Such sparsity becomes even more pronounced after special states are filtered out (right subplot). In contrast, Fig. 8(e) illustrates the reward function discovered by our approach. The reward function not only captures the critical states (collisions and goal attainments) but also distributes more informative and denser reward signals across the

other state space, reducing sparsity and better reflecting the underlying state-reward relationships. This is particularly evident in the filtered state space, where the discovered reward function maintains meaningful distinctions among states, unlike the original reward function which remains nearly uniform. In response to your comments, we have modified Fig. 8 to aid interpretation. Specifically, the reward distribution in the original state space is shown in Fig. 8(d), and the reward distribution in the filtered state space is shown in Fig. 8(e). Furthermore, Fig. 8(d) highlights the key states in which an agent attains a target point or crashes. The corresponding modifications are copied here for your convenience:

#### Case Study of Unmanned Systems Control Task

The differences between the discovered reward function and the carefully designed reward function based on dynamic principles are illustrated in Fig. 8(d-e). For the manually designed reward, the UAV receives a reward of 100 upon reaching the target position and -100 upon crashing, while rewards for all other states remain relatively small. **Such a design leads to a very sparse reward distribution, as shown in left of Fig. 8(d). When further analyzing the reward distributions, we exclude those of special states; the resulting filtered state space is that of Fig. 8(e). These visualizations clearly demonstrate the advantages of our approach. First, our method fully captures the critical states when a UAV attains a target trajectory point or crashes. Specifically, in Fig. 8(d), the lower red box refers to states wherein the UAV successfully attains a target point, associated with a strong positive reward, and the upper blue box highlights crash states associated with negative rewards. These critical states are appropriately highlighted in the reward space. More importantly, our approach significantly reduces reward sparsity by delivering denser and more informative reward signals throughout the state space, rather than concentrating the rewards on only a few special states. Furthermore, our discovered reward function effectively encodes the underlying theoretical relationship between states and rewards. This is evidenced by the fact that states with similar dynamics are assigned reward values that closely match those calculated when using the physical principles of the system. Our method not only preserves the essential characteristics of the task but also facilitates more efficient learning by providing richer feedback to the agent. Visualizations of the reward distributions for other reward function design methods are provided in Supplementary Fig. 6 for further comparison.**

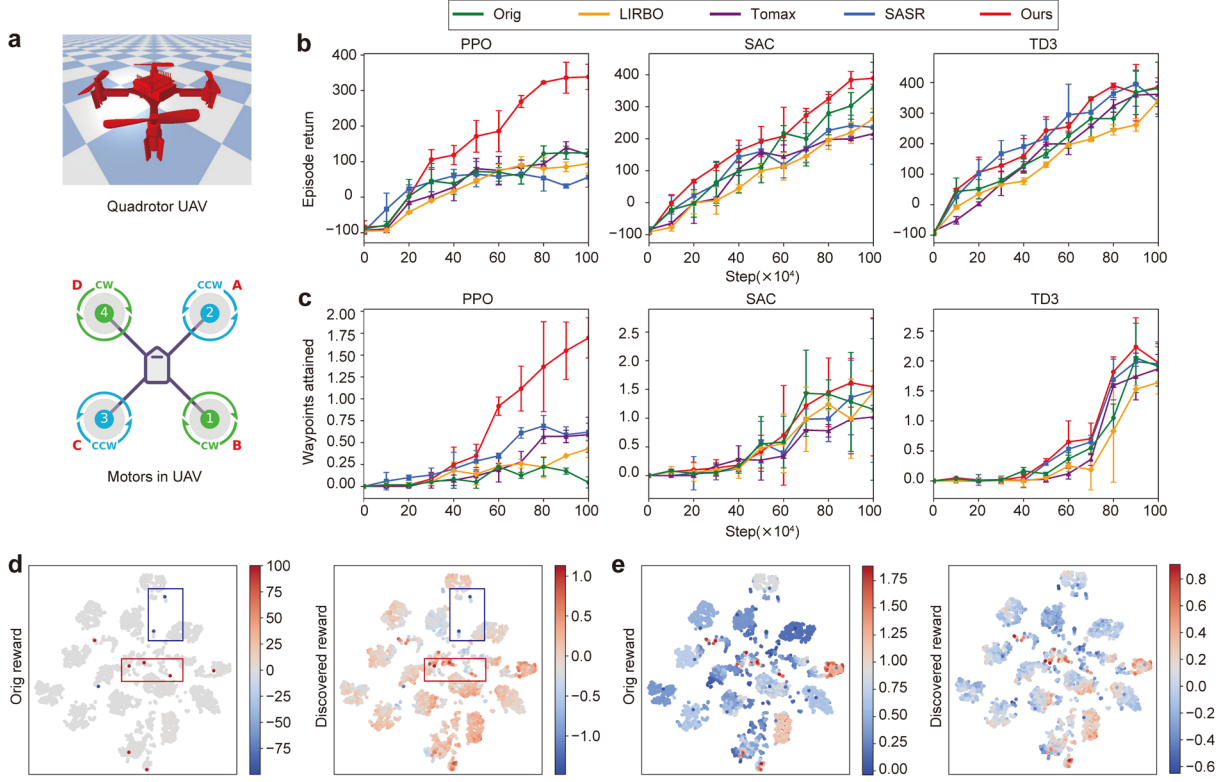


Fig. 8 Evaluation results for unmanned systems control task. a The Quadrotor UAV and the powered motors in a world model. b Learning curves showing reward progression during UAV trajectory tracking using the PPO, SAC, and TD3 agents. The proposed approach is associated with faster convergence and better performance. c The numbers of target trajectory points attained by the UAV during learning. d The state space visualizations of the UAV task, showing the reward distributions of the original reward function derived using dynamic principles and the discovered reward function. The discovered reward function successfully captures critical states. e The state space visualizations after removal of certain states associated with extremely high or low rewards, such as those involving collisions or attainment of waypoints. The reward distribution resembles those of reward functions derived using dynamic principles.

In addition, we used a classic control task (*Acrobot-v1* in Fig. 3(a)) with a discrete action space that allows clear visualization of the reward function. The agent in the *Acrobot-v1* system consists of two links connected linearly to form a chain, with one end of the chain fixed. The joint between the two links is actuated. The goal is to apply torques on the actuated joint to swing the free end of the linear chain above a given height while starting from the initial state of hanging downwards. The state space observed by the agents includes the sines and cosines about the two rotational joint angles as well as their angular velocities:  $(\cos(\theta_1), \sin(\theta_1), \cos(\theta_2), \sin(\theta_2), \dot{\theta}_1, \dot{\theta}_2)$ . The action space is discrete, deterministic, and represents the torque applied on the actuated joint between the two links. In the revised manuscript, Fig. 4(e-h) illustrates the reward surfaces of the sparse reward function and the discovered reward function for each discrete action in the *Acrobot-v1* task, visualized as a function of the two joint angles  $\theta_1$  and  $\theta_2$ , when the angular velocities  $\dot{\theta}_1$

and  $\dot{\theta}_2$  are 0 (for the distribution of the reward functions at different angular velocities, we present them in the Supplementary Figs. 2-4). Each surface illustrates the reward values of our reward function for a specific action: application of -1 torque (action 0), 0 torque (action 1), and 1 torque (action 2) to the actuated joint. The results reveal that our reward function assigns higher values to actions 0 and 2 (-1 and 1 torque, respectively) when the joint angles are away from the downward resting position, encouraging the agent to apply torque when momentum generation is favorable. In contrast, action 1 (0 torque) receives relatively higher rewards near the neutral region, reflecting a learned preference for stabilizing when the system is aligned. This structured reward landscape aligns well with the task objective of swinging the outer link above a target height and demonstrates that the reward function of this work captures meaningful agent control behavior consistent with the dynamics of the *Acrobot-v1* system.

For your convenience, we copied the revised figure and its description here:

**Fig. 4(e-h) illustrates the reward surfaces of the sparse reward function and the discovered reward function for each discrete action in *Acrobot-v1* task. The reward surfaces are visualized as functions of the two rotational joint angles,  $\theta_1$  and  $\theta_2$ , when their angular velocities  $\dot{\theta}_1$  and  $\dot{\theta}_2$  are 0. Fig. 4(e) shows that the sparse reward function assigns a reward of -1 for all state and action. Fig. 4(f-h) shows the rewards assigned by our reward function for specific actions: application of -1 torque (action 0), 0 torque (action 1), or 1 torque (action 2) to the actuated joint. The results show that the reward function of this work assigns higher values to actions involving -1 and 1 torque when the joint angles deviate from the downward resting position. This encourages the agent to apply torque in situations when momentum generation can be effectively created. Conversely, the action corresponding to 0 torque receives higher rewards when the agent is near the neutral region, indicating a learned preference for stabilization or energy conservation when the system is aligned. This structured reward landscape is consistent with the task objective of swinging the outer link above a target height, demonstrating that the reward function of this work successfully captures meaningful control behaviors aligned with the dynamics of the *Acrobot-v1* task. Visualizations of the reward functions at different angular velocities can be found in the Supplementary Figs. 2-4.**

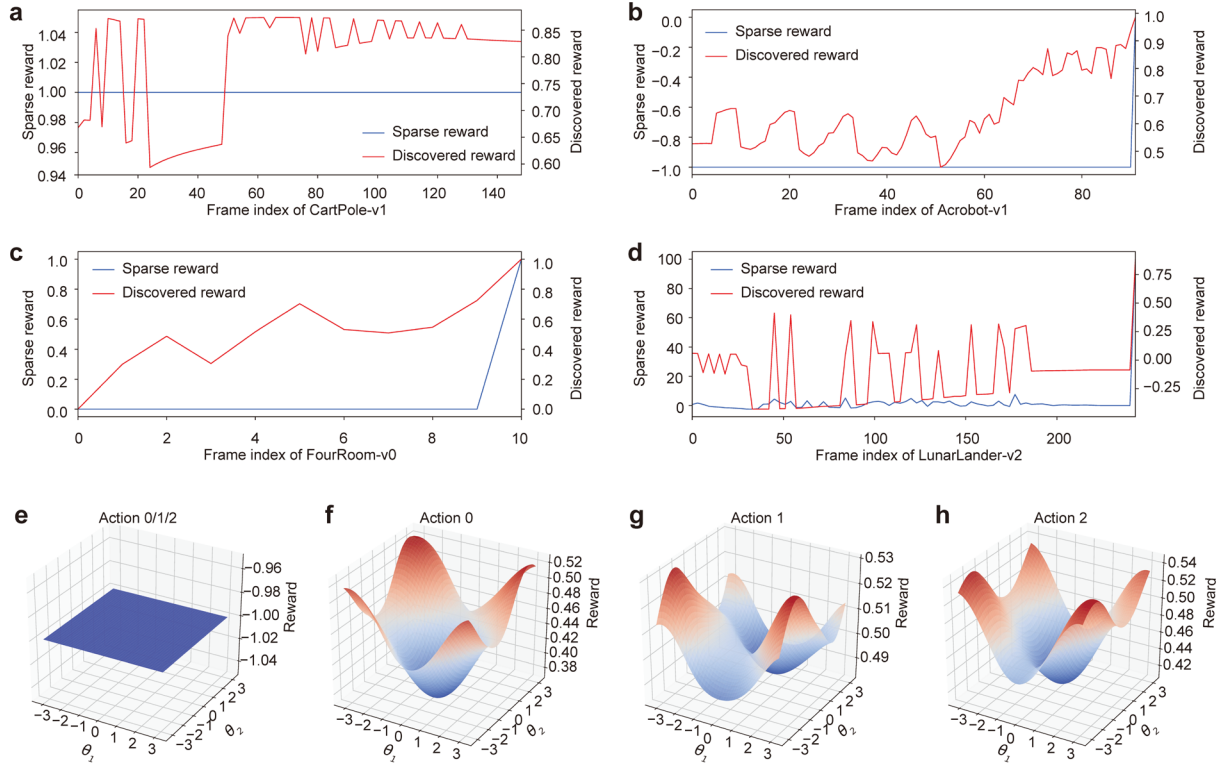


Fig. 4 Reward visualization. a Reward signals during a single interaction episode of the *CartPole-v1* task. The sparse reward signals (blue lines) and the reward signals discovered in this work (red lines) are shown. b Reward signals during a single interaction episode of the *Acrobot-v1* task, showing the difference between the sparse and discovered reward signals. c Reward signals during a single interaction episode of the *FourRoom-v0* task, illustrating the increased signal density achieved in this work. d Reward signals during a single interaction episode of the *LunarLander-v2* task. e The surface of the sparse reward function. For any state and action, the sparse reward function always delivers a reward signal of -1. f The reward surface for action 0 (-1 torque) of *Acrobot-v1* as a function of the joint angles  $\theta_1$  and  $\theta_2$  with angular velocities fixed at 0. g The reward surface for action 1 (0 torque). h The reward surface for action 2 (1 torque).

> Is the methodology sound? Does the work meet the expected standards in your field?

The methodology is fundamentally sound; however, a more detailed comparison with state-of-the-art approaches would better highlight the advantages of the proposed method. I therefore recommend conducting additional, comprehensive experiments to this end.

**Response:** Thanks for the reviewer's valuable comment. In the original manuscript, to verify the effectiveness of the proposed approach, sparse reward signals and manually well-designed reward functions are selected to compare the performance with respect to the episode reward and convergence speed. These methods are commonly used for comparison in the existing studies.

In the revised manuscript, based on the reviewer's comment, we conducted additional experiments that directly compare our approach with the method proposed in [1], as well as another two recent state-of-the-art reward design methods [2][3]. Specifically, [2] proposed an efficient adaptive reward shaping mechanism, which used the success rate as the shaping reward, and used kernel density estimation (KDE) and random Fourier features (RFF) to derive a beta distribution from which the success rate is sampled. [3] emphasized the important role of reward maximization in RL. However, compared with reward function design method, [3] directly maximized the maximum value of the reward signal obtained in the scene instead of maximizing the cumulative return of RL.

[1] Stadie, B., Zhang, L., Ba, J.: Learning intrinsic rewards as a bi-level optimization problem. In: Conference on Uncertainty in Artificial Intelligence, pp. 111 – 120 (2020).

[2] Ma, H., Luo, Z., Vo, T.V., Sima, K., Leong, T.-Y.: Highly efficient self-adaptive reward shaping for reinforcement learning. In: Proceedings of the 13th International Conference on Learning Representations (2025).

[3] Vevurko, G., Böhmer, W., De Weerd, M.: To the max: reinventing reward in reinforcement learning. In: Proceedings of the 41st International Conference on Machine Learning, pp. 4945–49470 (2024).

The results of these additional comparisons are presented in Fig. 3(c-e), Fig. 5(c-e), Fig. 7(b-c), Fig. 8(b-c), demonstrating that our method achieves superior performance in terms of learning stability and convergence speed. This is similar to the previous comment. The corresponding comparison results are copied here again for your convenience:

## Results

Furthermore, **the reward functions discovered by the proposed approach are compared to the baseline rewards, including sparse reward signals, manually designed reward functions, and rewards from three state-of-the-start reward learning methods, including LIRBO [49], Tomax [30], SASR [53].** The comparison serves to illustrate further the relationship between the reward functions and the tasks, as well as how these functions help embodied RL agents to achieve better performance. **The experimental setups for both the numerical experiments and the real-world tasks, and the details on how the comparative methods are implemented, are described in the Supplementary Experimental Details.**

## Case Study of Sparse-Reward Tasks

**Fig. 3(c-d) shows the average episodic rewards of embodied RL agents when using our proposed approach compared to those of agents that receive sparse reward signals and reward signals from baseline methods. Agents under the proposed reward function outperform the baseline agents in terms of both convergence speed and learning stability.** When receiving sparse reward signals, the agent finds it challenging to learn the optimal policy and achieve fast convergence because of the limited number of samples that can be effectively utilized, resulting in low sample efficiency. This low efficiency requires the agent to engage in extensive trial-and-error, thereby slowing the convergence rate. **Reward**

function design methods have relieved this challenge. LIRBO replaces sparse rewards with intrinsic reward signals, and SASR uses a shaping auxiliary function to the same end. Tomax directly learns historical maximum rewards but sometimes does not perform well, as the initial random exploration in sparse-reward tasks does not deliver an adequate reward to the agent. In contrast, the proposed approach delivers dense reward signals directly, thereby improving the sample efficiency.

Fig. 3(e) shows the average episodic returns and standard errors throughout the entire learning process. The PPO and DQN agents that employ our proposed approach comprehensively outperform the agents under baseline methods. Additionally, as shown in Fig. 3(c-e), our approach enhances learning stability and the convergence speed.

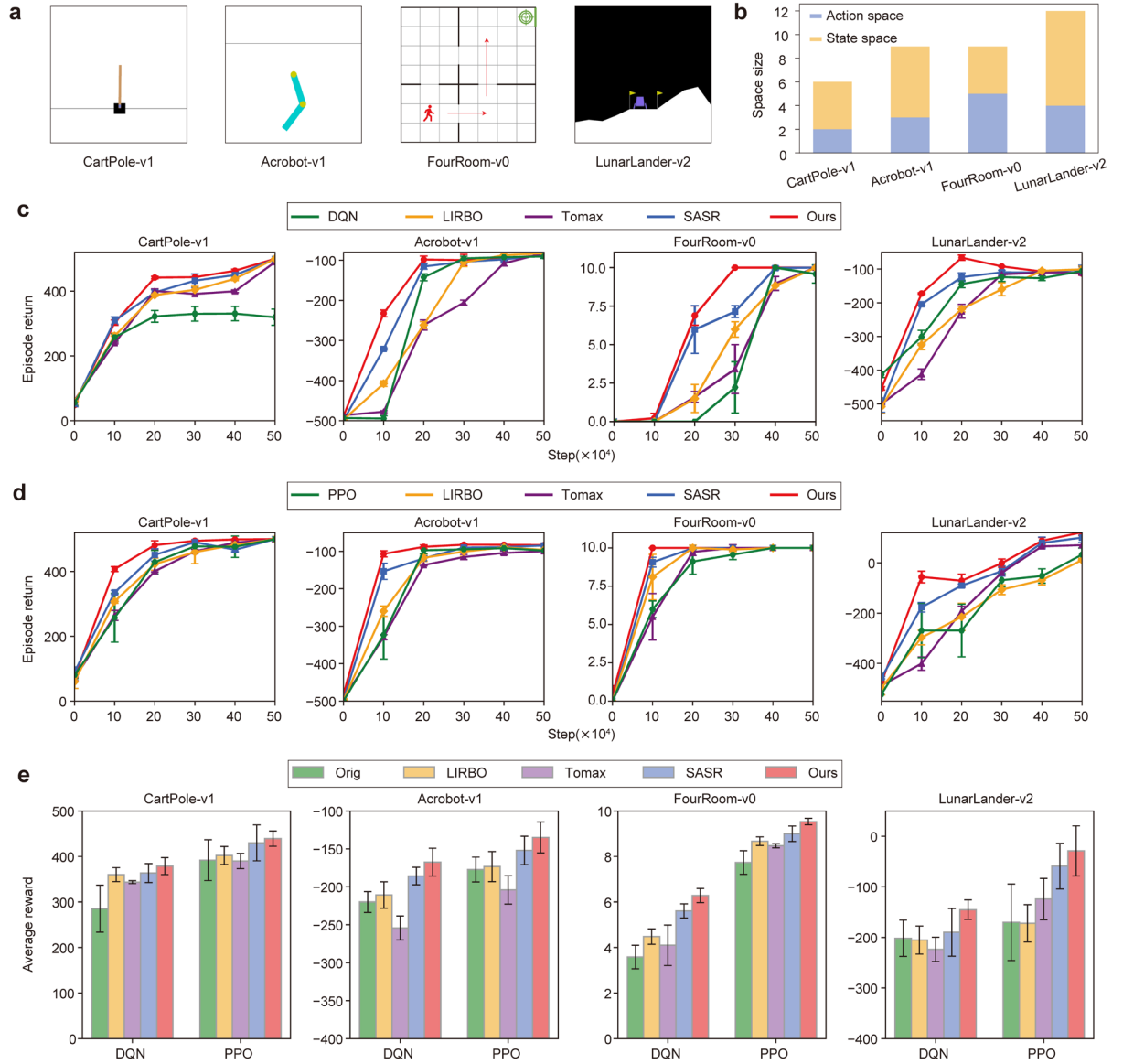


Fig. 3 Evaluation results for sparse-reward tasks. a The environments used in the sparse-reward task, including *CartPole-v1*, *Acrobot-v1*, *FourRoom-v0* and *LunarLander-v2*. b The sizes of the observation

and action spaces of the four tasks, illustrating the varying complexity levels. **c The reward curves created during the learning processes of the DQN agent with different reward functions on different tasks. The solid lines show the average rewards and the error bars indicate the standard deviations across five trials using different random seeds. d The reward curves created during the learning processes of PPO agent with different reward functions on different tasks. e The average rewards obtained by the embodied RL agent during all episodes, with the standard deviations of the rewards received when different algorithms performed each task.**

#### Case Study of High-Dimensional Control Tasks

Fig. 5(c-d) shows the cumulative reward curves of embodied RL agents after each episode across the four tasks, with the error bars indicating the variabilities across the five parallel experiments. The proposed approach exhibits superior performance, even outperforming the reward functions of OpenAI designed by humans. **Compared to other baseline methods, our approach is more effective when PPO and SAC agents are employed. The convergence speed is faster and the episodic returns are higher. SASR and LIRBO perform similarly, but SASR achieves better when a PPO agent is applied because SASR samples the success rate of historical data.** Fig. 5(c-d) also reveals that the proposed approach accelerates agent convergence and improves the cumulative expected returns, particularly when the number of agent joints increases. Fig. 5(e) presents the means and variance of the cumulative returns given during learning and compares the rewards of the proposed approach with those afforded by the baseline rewards. It is clear that, for both the SAC and PPO agents, the proposed approach not only outperforms the baseline but also reduces risk induced by randomness.

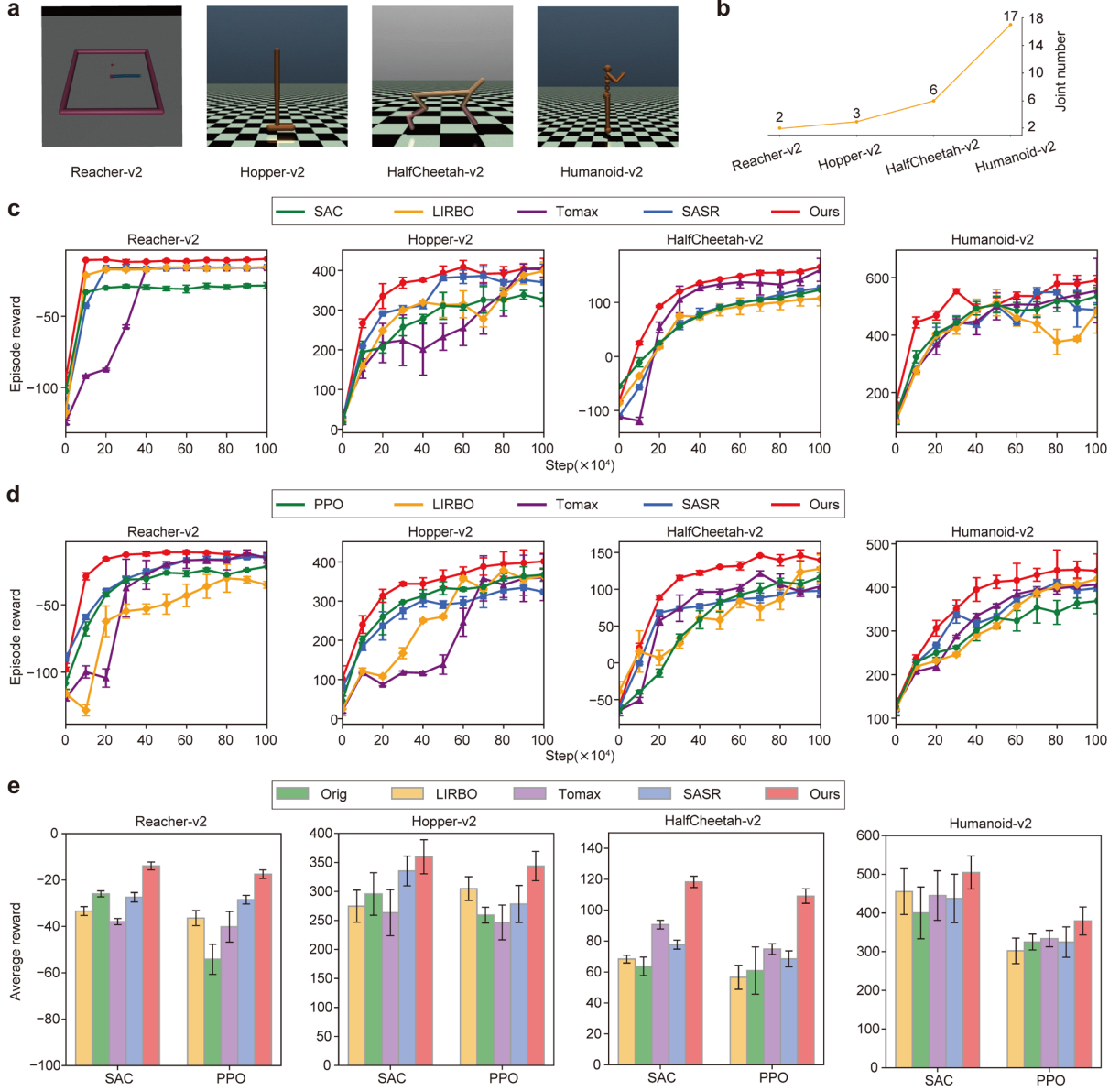


Fig. 5 Evaluation results for high-dimensional control tasks. a The high-dimensional control tasks, including *Reacher-v2*, *HalfCheetah-v2*, *Hopper-v2* and *Humanoid-v2*. b The number of controllable joints increases from *Reacher-v2* to *Humanoid-v2*, reflecting the increasing complexity of the tasks. c The reward curves during the learning processes of the SAC agent across the four tasks. The solid lines show the average episode rewards and the error bars indicate the standard deviations over five trials with different random seeds. d The reward curves during the learning processes of the PPO agent across the four tasks. e The average rewards with their standard deviations for different RL agents and different tasks. The results highlight the improved performance of the proposed approach across all tasks.

## Case Study of Real-World Tasks

### Case Study of Energy Management Task

Fig. 7(b) shows the rate of reduction in energy consumption when applying baseline RL and RL guided by the reward design approaches in data centers over a 1-year period. The numerical values are the percentage reduction rates. Fig. 7(c) illustrates the total energy consumption over 3, 6, 9, and 12 months for each RL agent. Although all RL methods reduce energy consumption, the reward function discovery approach further reduces consumption, particularly over longer times. Notably, as RL algorithms become more advanced, from the DQN to the more sophisticated SAC, the energy reduction rate increases (from 21% for DQN to 52% for SAC). The superiority of the proposed approach is more obvious when compared with the baseline models. With our method, all RL agents (including DQN, PPO, and SAC) achieve energy reductions that exceed 60%. These results highlight the generality and effectiveness of our approach for RL agents involved in energy management and other tasks. Therefore, the proposed optimal reward function discovery approach can be applied to RL agents operating in data centers and elsewhere.

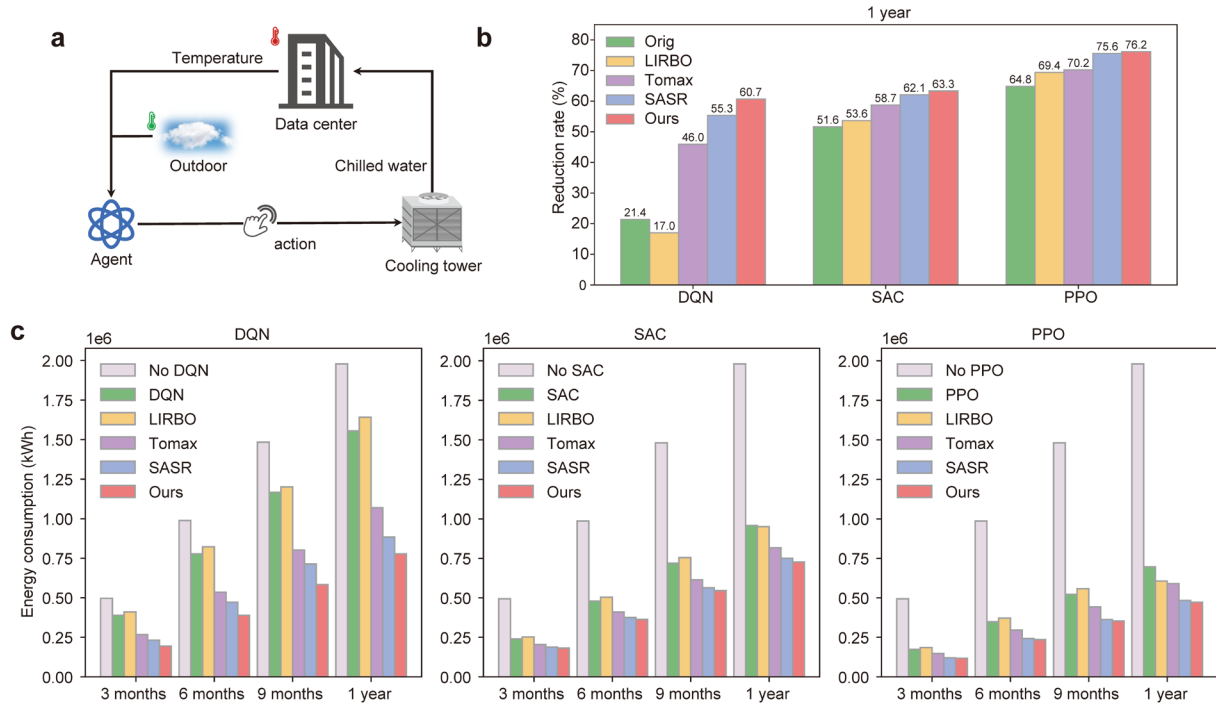


Fig. 7 Evaluation results for the energy management task. **a** Schematic of the data center cooling task, where an RL agent optimizes cooling system based on temperature and silicon data. **b** The 1-year reductions in energy consumption afforded by different reward design methods for different RL algorithms (DQN, SAC, and PPO). The agents that receive rewards from the discovered reward functions perform significantly better than others. **c** Energy consumption over 3, 6, 9, and 12 months. The RL approach using the discovered optimal reward function reduce energy consumption more effectively than do the comparative methods.

## Case Study of Unmanned Systems Control Task

Fig. 8(b) shows the reward curves when agent learning is guided by the reward function of the present work, reward function constructed on the basis of dynamic principles, and other reward design methods. Across the three different RL agents, the proposed approach significantly accelerates learning. For the PPO agent, our approach significantly improves both learning speed and the final performance, enabling the agent to successfully accomplish the task, whereas the hand-designed and other comparison methods fail to do so. For the SAC and TD3 agents, the improvement in learning speed is less pronounced, but our method performs comparable performance and ensures stable learning. These results demonstrate that the proposed reward function discovery approach is particularly useful when traditional rewards are insufficient, consistently ensuring robust performance across different RL algorithms.

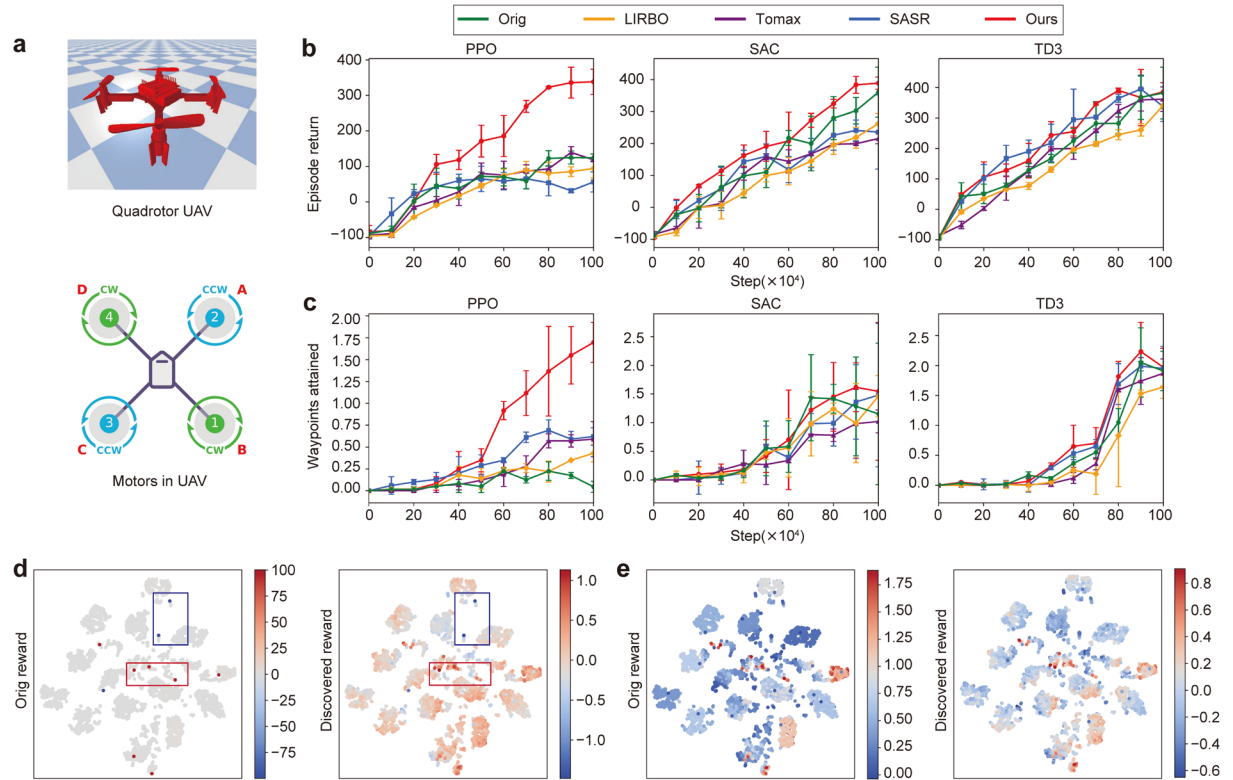


Fig. 8 Evaluation results for unmanned systems control task. a The Quadrotor UAV and the powered motors in a world model. b Learning curves showing reward progression during UAV trajectory tracking using the PPO, SAC, and TD3 agents. The proposed approach is associated with faster convergence and better performance. c The numbers of target trajectory points attained by the UAV during learning. d The state space visualizations of the UAV task, showing the reward distributions of the original reward function derived using dynamic principles and the discovered reward function. The discovered reward function successfully captures critical states. e The state space visualizations after removal of certain states associated with extremely high or low rewards, such as those

involving collisions or attainment of waypoints. The reward distribution resembles those of reward functions derived using dynamic principles.

> Is there enough detail provided in the methods for the work to be reproduced?

The methods as described are sufficient for reproduction; however, we will further strengthen the manuscript by including additional implementation details to support a more thorough evaluation.

**Response:** Thanks for the constructive comment. According to the reviewer’s suggestion, we have expanded the Supplementary Experimental Details to facilitate thorough evaluation of reproducibility. The corresponding contents are copied here for your convenience:

#### Reward Function Discovery approach

All experiments employed a computing cluster with a NVIDIA GeForce RTX 4090. All experiments were run at least 5 times in parallel. All hyperparameters are listed in Supplementary Table 1. Specifically, we integrated reward function discovery into the standard RL procedure, using the “reward optimization steps” parameter to control how often the reward function was updated. Although the details varied slightly across the different RL agents (Algorithms 4-7), we made effort to ensure that our proposed reward function discovery method was broadly applicable and generalizable to a wide range of RL agents.

#### Comparison Methods

In the experimental setup, several state-of-the-art reward function design methods were used for comparison. Specifically:

SASR [43] introduced a self-adaptive success rate-based reward shaping method to address the sparse-reward problem in RL, wherein the shaped reward functions were constructed by leveraging the success rates obtained from past interactions. These success rates were sampled from a dynamically updated beta distribution, which was efficiently estimated using kernel density estimation with random Fourier features. This approach enhanced the efficiency and adaptability of reward shaping in environments with sparse feedback.

Tomax [66] recognized the importance of reward maximization in RL, which focused on utilizing reward functions more efficiently. Specifically, a max-reward RL method was used to maximize the maximum reward encountered within an episode, rather than the cumulative return. This reformulation simplified the design and optimization of reward functions.

LIRBO [14] formulated the intrinsic RL reward learning problem as a bilevel optimization problem and employed the self-tuning networks approach for hyperparameter optimization. This method aimed to train the intrinsic reward function more efficiently. A best-response function was learned, and a neural network module was introduced. This engaged in conditional layer normalization that embed the parameters of the reward function into the policy network. This

integration ensured a stable and efficient training process. By avoiding the high computational cost of backpropagation throughout the entire training process, LIRBO improved both the scalability and practicality of intrinsic reward learning.

#### Reward Function Visualization Methods

In our approach, the reward function is parameterized (as shown in Supplementary Fig. 7). The weights are continuously updated via a meta-gradient based optimization. Therefore, it is challenging to write a closed-form analytical expression for the function. In sparse-reward tasks, the agent observations are simple and the action space is discrete, facilitating visualization of reward function surfaces in the state-action space. In high-dimensional control tasks and real-world tasks, both the observation space and the action space become significantly more complex and are typically continuous. To address this, we employed the t-SNE method [72] to project the high-dimensional state space onto two dimensions, enabling us to visualize the distribution of reward signals within the state space.

**Reward Curves.** Fig. 4(a-d) in the main text present the reward signal curves observed during a single interaction between the agent and the task. Specifically, we used a well-trained agent to interact with the task for one complete episode, and recorded the reward signals generated by different reward functions. This allowed us to compare the reward signals given to the agent at the same states when different reward function designs were used.

**Reward Surfaces.** We considered the *Acrobot-v1* task, a classic benchmark in RL that involves underactuated control. The system consists of a two-link pendulum with one end fixed and a single actuated joint between the two links. The goal is to apply one of three discrete  $-1, 0$ , or  $+1 N \cdot m$  torques to the actuated joint to swing the free end of the outer link above a target height, starting from the initial state where both links hang downwards. The observation space includes the sines and cosines of both joint angles  $\theta_1, \theta_2$ , and their angular velocities  $\dot{\theta}_1, \dot{\theta}_2$ . This yields a 6-dimensional continuous state representation. The agent must learn to coordinate oscillatory motions when building sufficient momentum to attain the goal. We systematically varied the two joint angles from  $-\pi$  to  $\pi$  while keeping the joint angular velocities fixed, thereby generating 6-dimensional states. In total, 100 states were generated. For each state, the reward function delivers reward signals associated with different actions, and the rewards are then used to construct the reward surfaces as functions of the two joint angles.

**t-SNE.** To visualize the distribution of reward signals in high-dimensional state spaces, we associate each high-dimensional state with the corresponding reward value and project these state representations into a 2-dimensional space using the t-distributed Stochastic Neighbor Embedding (t-SNE) method [72]. t-SNE is a nonlinear dimensionality reduction technique that preserves local data structure by mapping similar states close together in the 2D space. In our visualization, each point represents a state, and the color thereof the associated reward signal (red: higher rewards;

blue: lower rewards). This enables an intuitive interpretation of how reward signals are distributed across complex, high-dimensional state spaces.

The hyperparameters of the various agents and task parameter settings of the main text are listed in Supplementary Table 1. As stated in the main text, the experimental hyperparameters were not specifically fine-tuned for individual tasks. Multiple experiments were conducted to ensure comprehensive evaluation of the algorithms. Only the random seed changed across the experiments.

**Supplementary Table 1** The experimental parameter settings.

| Parameter                    | Sparse-reward task   |                    | High-dimensional task |                    | Energy management task |                    |                    | Unmanned systems task |                    |                    |
|------------------------------|----------------------|--------------------|-----------------------|--------------------|------------------------|--------------------|--------------------|-----------------------|--------------------|--------------------|
|                              | DQN                  | PPO                | PPO                   | SAC                | DQN                    | PPO                | SAC                | PPO                   | SAC                | TD3                |
| hidden dim                   | [120, 84]            | [64, 64]           | [64, 64]              | [256, 256]         | [64, 32]               | [64, 64]           | [64, 64]           | [64, 64]              | [256, 256]         | [256, 256]         |
| max time steps               | $5 \times 10^5$      | $5 \times 10^5$    | $1 \times 10^6$       | $1 \times 10^6$    | -                      | -                  | -                  | $1 \times 10^6$       | $1 \times 10^6$    | $1 \times 10^6$    |
| buffer size                  | $1 \times 10^4$      | -                  | -                     | $1 \times 10^6$    | $1 \times 10^4$        | $1 \times 10^4$    | $1 \times 10^4$    | -                     | $1 \times 10^6$    | $1 \times 10^6$    |
| batch size                   | 128                  | 64                 | 64                    | 256                | 64                     | 64                 | 64                 | 2,048                 | 256                | 256                |
| learning rate                | $2.5 \times 10^{-4}$ | $1 \times 10^{-4}$ | $3 \times 10^{-4}$    | $3 \times 10^{-4}$ | $3 \times 10^{-4}$     | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ | $3 \times 10^{-4}$    | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ |
| policy optimization steps    | 10                   | 1024               | 2,048                 | 2                  | -                      | -                  | -                  | 5,000                 | 2                  | 2                  |
| offline optimization rounds  | -                    | -                  | -                     | -                  | 200                    | 200                | 200                | -                     | -                  | -                  |
| running time                 | -                    | -                  | -                     | -                  | 1 month                | 1 month            | 1 month            | -                     | -                  | -                  |
| learning start steps         | 5,000                | -                  | -                     | 5,000              | -                      | -                  | 5,000              | -                     | 5,000              | 25,000             |
| target optimization steps    | 500                  | -                  | -                     | -                  | -                      | -                  | -                  | -                     | 1                  | -                  |
| target smoothing coefficient | -                    | -                  | -                     | 0.05               | -                      | -                  | -                  | -                     | 0.05               | 0.05               |
| target network update rate   | 1                    | -                  | -                     | 1                  | -                      | -                  | -                  | -                     | 0.05               | 0.05               |
| exploration noise rate       | 0.5                  | -                  | -                     | -                  | -                      | -                  | -                  | -                     | 0.2                | 0.1                |
| policy training epochs       | -                    | 10                 | 10                    | -                  | -                      | 10                 | -                  | 10                    | -                  | -                  |
| clip rate                    | -                    | 0.2                | 0.2                   | -                  | -                      | 0.2                | -                  | 0.2                   | -                  | -                  |
| gae lambda                   | -                    | 0.95               | 0.95                  | -                  | -                      | -                  | -                  | 0.95                  | -                  | -                  |
| entropy regularization rate  | -                    | -                  | -                     | 0.2                | -                      | -                  | 0.2                | -                     | 0.2                | -                  |
| policy noise rate            | -                    | -                  | -                     | -                  | -                      | -                  | -                  | -                     | -                  | 0.2                |
| noise clip rate              | -                    | -                  | -                     | -                  | -                      | -                  | -                  | -                     | -                  | 0.5                |
| trajectory buffer size       | 100                  | 100                | 100                   | 100                | 100                    | 100                | 100                | 100                   | 100                | 100                |
| learning rate                | $1 \times 10^{-4}$   | $3 \times 10^{-4}$ | $1 \times 10^{-4}$    | $1 \times 10^{-4}$ | $1 \times 10^{-4}$     | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ | $3 \times 10^{-4}$    | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ |
| reward optimization steps    | 1,000                | 1,024              | 2,048                 | 2,048              | -                      | -                  | -                  | 5,000                 | 5,000              | 5,000              |
| sample action number         | -                    | -                  | 1,000                 | 1,000              | -                      | -                  | -                  | 1,000                 | 1,000              | 1,000              |

We again thank Reviewer #2 for the detailed attention given to our work and the careful review. All comments helped us to improve the quality of the manuscript. We hope that the revision meets the reviewer's expectations of a high-quality publication in *Nature Communications*.

## Response to Reviewer #1 (Remarks to the Author):

The author response and changes made address my two primary concerns with the work (limited empirical comparison with other methods and precision of, especially theoretical, claims). The updated language in the article, improved theory statements and inclusion of additional experimental work satisfies these and significantly strengthen the work.

**Response:** We sincerely thank the reviewer for the careful reading of our manuscript and for the thoughtful feedback. We are very pleased that the revised version addresses your two primary concerns. Thank you again for your constructive feedback and support.

My only remaining, perhaps pedantic, suggestion has to do with the framing of this work and its dependence on knowledge of the task relative to existing methods. The authors repeatedly distinguish their approach from others in terms of dependency on "external knowledge", e.g. "In contrast, our approach aims to discover reward functions in a fully autonomous manner, guided solely by environmental interactions and generalization signals, without relying on predefined expert trajectories, heuristics, or task-specific priors."

This seems to claim that the proposed method does not depend on any prior information about the "task", which obviously is not correct. There is indeed information about the task, just as in other methods, but this method and others certainly differ in the type and form of that "external knowledge". The type of dependency in this article is most closely related to those such as LIRBO and SASR, making them great choices for comparison in the experiments.

I think it is important to be more clear that this method does indeed require some such knowledge, in what form, and then to contrast with other in these terms. The argument cannot be that this approach requires no knowledge of the task, instead of must be that the type of dependency and type of knowledge here is preferable (e.g. sparse reward or fitness vs. expert demonstrations). This detail is decipherable from the methods section, but in the main the claims do not match this detail, and clarity would be much improved on this point by simply stating the dependency and comparison upfront.

**Response:** We sincerely thank the Reviewer #1 for this insightful and constructive suggestion. We agree with your comment that in our method, like other approaches, inherently depends on certain forms of task-related knowledge. The key distinction, as you correctly pointed out, lies not in the absence of such knowledge, but rather in the type and form of dependency.

Following your suggestion, we have revised the manuscript to make this argument more explicit and precise, as shown below:

## **Introduction**

**The core insight is that the optimal reward function can be automatically discovered on the basis of the past performance of embodied RL agents with minimal reliance on task-specific prior knowledge.**

We sincerely thank you for the time, effort, and thoughtful evaluation that greatly contributed to improving the quality of our work.

## **Response to Reviewer #2 (Remarks to the Author):**

I have confirmed the authors' reply and have no further questions.

The authors have added comparative experiments with existing methods, and it is evident that the performance of the proposed method is comparable to or better than the existing ones.

The visualizations have also been improved, making it easier for readers to understand

**Response:** We sincerely thank Reviewer 2 for the positive and encouraging evaluation. We are pleased that the revisions have met your expectations.

Thank you again for your time, effort, and supportive comments, which helped us further improve the quality of our work.