Coursework Essay Probing into DeepSeek: Principle Analysis, Application Domains and Optimization Schemes

First Name	Yuepeng
Last Name	Peng
Student CUC ID	202229013088N
Student CU ID	14564165
Class	Class1
College	Hainan International college of Coventry University of China (Coventry College, Communication University of China)
Course Teacher	Qinglan Wei

Introduction

Reinforcement learning has long driven breakthroughs in AI. Yet its application in large language models faces persistent challenges: reward hacking, data dependency, and erratic outputs during cold starts. In this essay, we dissect how DeepSeek's Group Relative Policy Optimization not only overcomes these limitations but also enables LLMs to exhibit human-like reflection, self-correction, and contextual adaptability.

Beyond technical innovation, we envision GRPO-powered LLMs revolutionizing real-world domains like cultural tourism. By integrating local knowledge graphs, visitor behavior analyse, and immersive AR interfaces, DeepSeek could craft personalized travel experiences—transforming static scenic spots into interactive, AI-guided journeys.

Keywords: **Reinforcement Learning, Aha moments, Group Relative Policy Optimization , Cold Start, the local cultural and tourism industry.**

Related work on Github: KLucen/HW-for-DL: My Coursework for Deep Learning

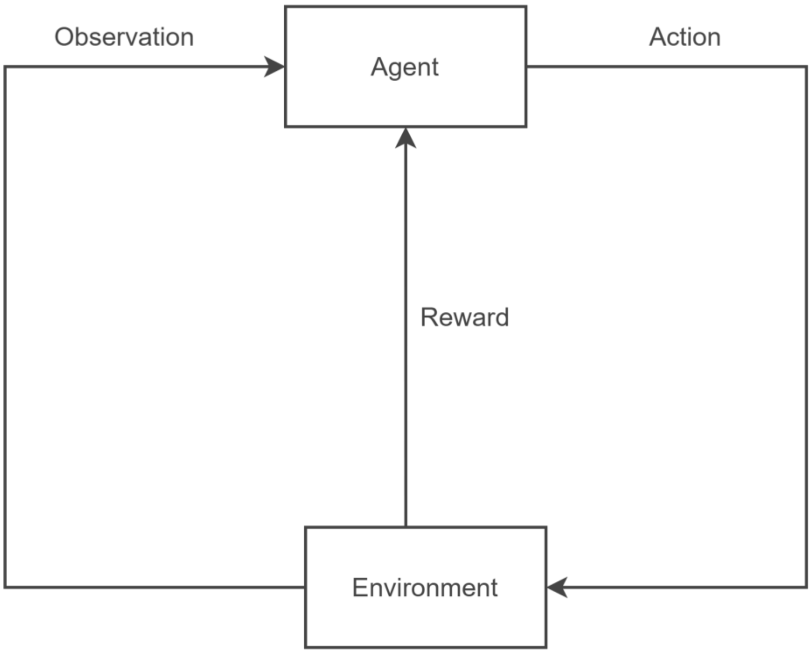
Link to my Cloud Drive: https://kod.cuc.edu.cn/#s/_o6_XHsg

Analysis of core principles

After a cursory review of the papers on the DeepSeek-R1 model and the V3 model, one can't help but marvel at the significant changes made by DeepSeek in the model innovation points. The training process of DeepSeek not only saves the resources required for training (such as: FP8 mixed precision training, Multi-head Latent Attention and DeepSeekMoE) but also has a major breakthrough in the performance of DeepSeek (such as: DeepSeekMoE, Fine-Tuning (SFT) and Reinforcement Learning). However, due to space and content limitations, in this article, I will only focus on **reinforcement learning in DeepSeek**.

When it comes to reinforcement learning, in fact, Google had already applied reinforcement learning to all Atari games as early as 2015, and the fact that AlphaGo defeated the Go champion in 2016 further demonstrated the unlimited potential of reinforcement learning in the field of artificial intelligence.

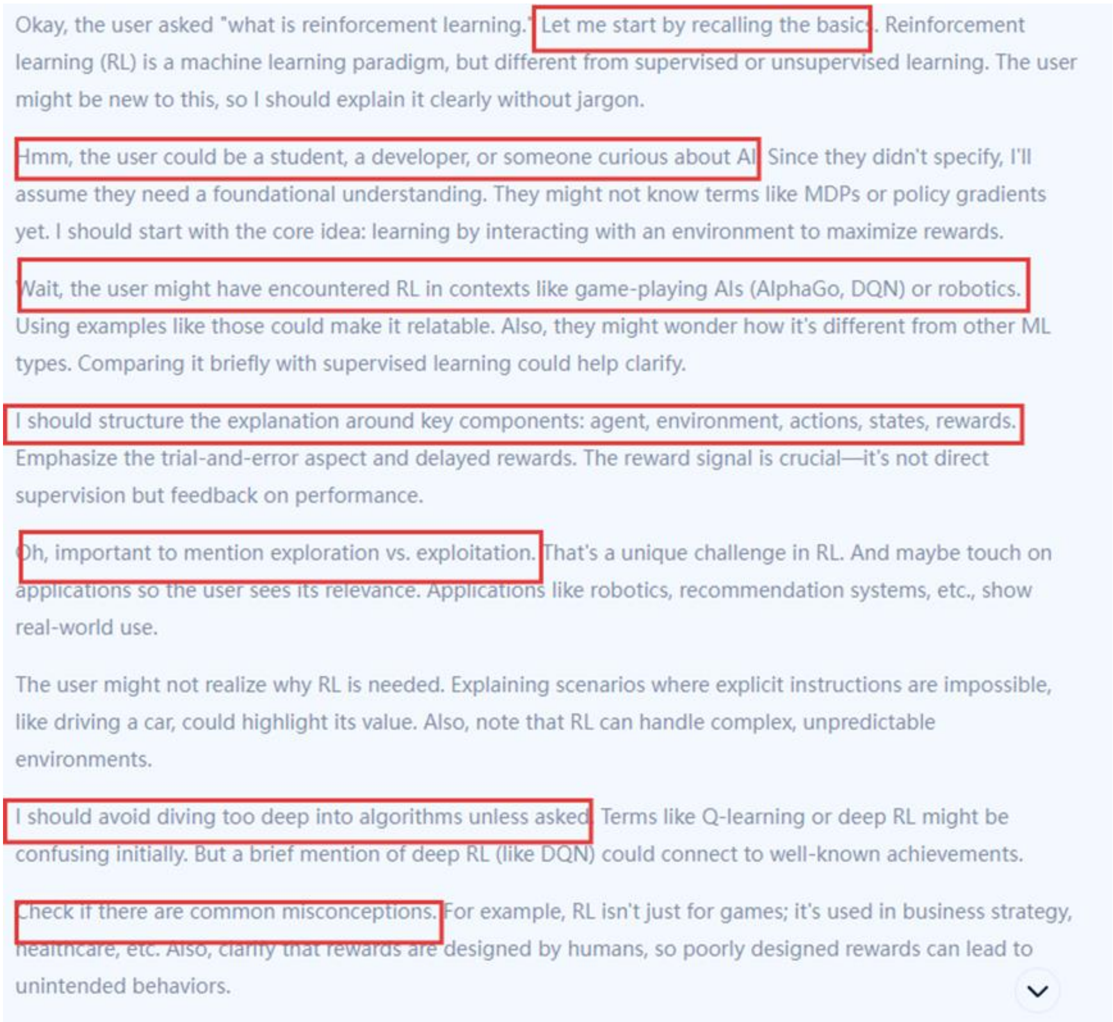
Now, let's take a look together at what the basic idea of reinforcement learning is exactly like. Here I will explain its working principle in the form of a diagram (Figure 1) : The Agent observes the environment (that is, determines the current state) and then takes specific actions. The actions taken will then affect the environment, thereby giving a reward to the Agent. This cycle repeats itself. The purpose of the Agent is to obtain **the maximum expected reward** after taking actions one after another. If we take Go as an example, the Agent makes a judgment on where to place its next move (action) based on the opponent's moves (observation). Only when the Agent defeats the opponent can it receive a positive reward; otherwise, the reward is negative.



(Figure 1: RL principle)

After understanding the principle of reinforcement learning, we will actually find a problem. If DeepSeek itself is the Agent in reinforcement learning theory, then is the Environment the user's question? I think the user's question belongs to the Environment, but the Environment is not merely the Environment. When we take a close look at the thinking process of DeepSeek in answering our questions and think carefully, we can find that DeepSeek itself is both an Agent and an Environment.

Here is the answer given by DeepSeek when I asked DeepSeek "what is reinforcement learning.":



Okay, the user asked "what is reinforcement learning." Let me start by recalling the basics. Reinforcement learning (RL) is a machine learning paradigm, but different from supervised or unsupervised learning. The user might be new to this, so I should explain it clearly without jargon.

Hmm, the user could be a student, a developer, or someone curious about AI. Since they didn't specify, I'll assume they need a foundational understanding. They might not know terms like MDPs or policy gradients yet. I should start with the core idea: learning by interacting with an environment to maximize rewards.

Wait, the user might have encountered RL in contexts like game-playing AIs (AlphaGo, DQN) or robotics. Using examples like those could make it relatable. Also, they might wonder how it's different from other ML types. Comparing it briefly with supervised learning could help clarify.

I should structure the explanation around key components: agent, environment, actions, states, rewards. Emphasize the trial-and-error aspect and delayed rewards. The reward signal is crucial—it's not direct supervision but feedback on performance.

Oh, important to mention exploration vs. exploitation. That's a unique challenge in RL. And maybe touch on applications so the user sees its relevance. Applications like robotics, recommendation systems, etc., show real-world use.

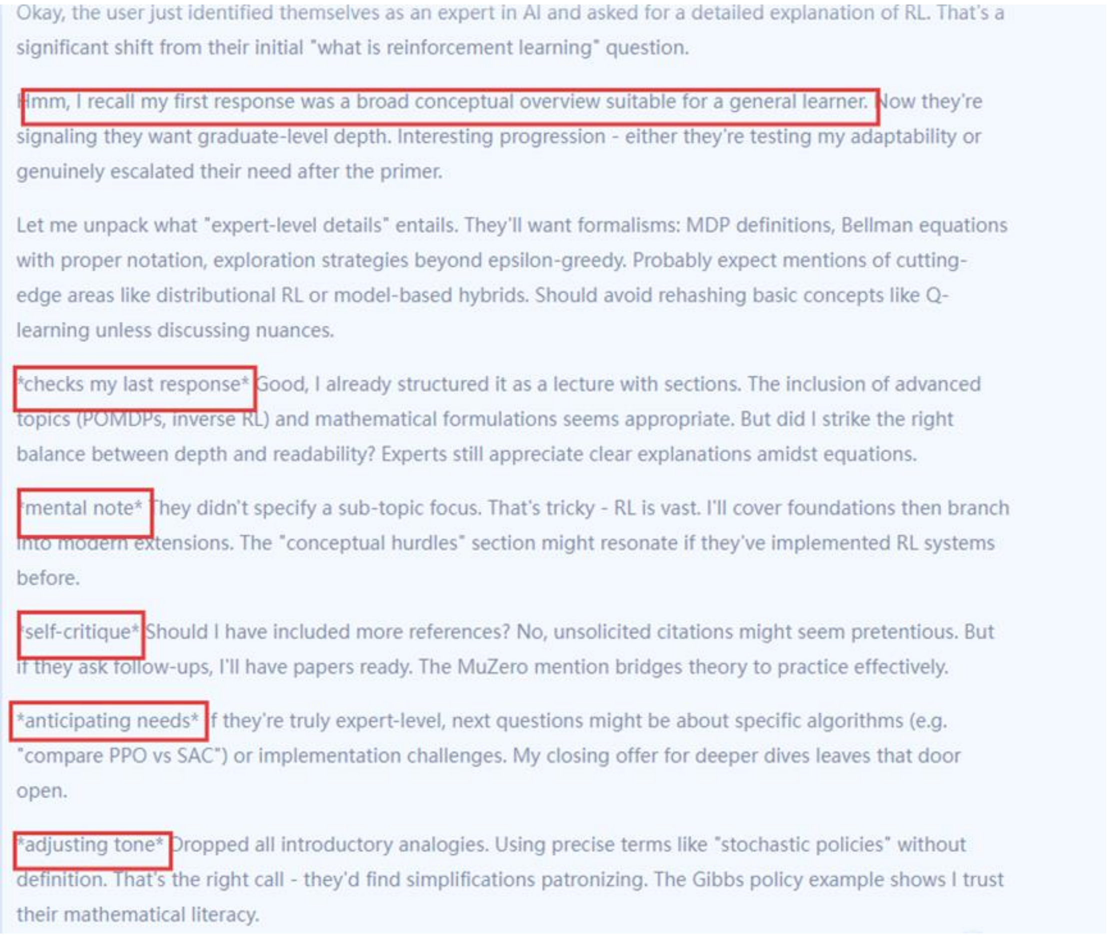
The user might not realize why RL is needed. Explaining scenarios where explicit instructions are impossible, like driving a car, could highlight its value. Also, note that RL can handle complex, unpredictable environments.

I should avoid diving too deep into algorithms unless asked. Terms like Q-learning or deep RL might be confusing initially. But a brief mention of deep RL (like DQN) could connect to well-known achievements.

Check if there are common misconceptions. For example, RL isn't just for games; it's used in business strategy, healthcare, etc. Also, clarify that rewards are designed by humans, so poorly designed rewards can lead to unintended behaviors.

(Figure 2: DeepSeek's answer 1)

From its thinking process, we can observe that it is not only thinking about the questions we raised, but in fact, it is also reviewing the content it has output itself. (For example, the content I selected in the red box in Figure 2) It first extracted the basic knowledge of RL from the knowledge base in response to my question, then made a conjecture about my identity, and then made guesses about the fields I might be interested in, thereby obtaining the most likely output for it (that is, the output with the greatest expectation of this reward for it).



Okay, the user just identified themselves as an expert in AI and asked for a detailed explanation of RL. That's a significant shift from their initial "what is reinforcement learning" question.

Hmm, I recall my first response was a broad conceptual overview suitable for a general learner. Now they're signaling they want graduate-level depth. Interesting progression - either they're testing my adaptability or genuinely escalated their need after the primer.

Let me unpack what "expert-level details" entails. They'll want formalisms: MDP definitions, Bellman equations with proper notation, exploration strategies beyond epsilon-greedy. Probably expect mentions of cutting-edge areas like distributional RL or model-based hybrids. Should avoid rehashing basic concepts like Q-learning unless discussing nuances.

checks my last response Good, I already structured it as a lecture with sections. The inclusion of advanced topics (POMDPs, inverse RL) and mathematical formulations seems appropriate. But did I strike the right balance between depth and readability? Experts still appreciate clear explanations amidst equations.

mental note They didn't specify a sub-topic focus. That's tricky - RL is vast. I'll cover foundations then branch into modern extensions. The "conceptual hurdles" section might resonate if they've implemented RL systems before.

self-critique Should I have included more references? No, unsolicited citations might seem pretentious. But if they ask follow-ups, I'll have papers ready. The MuZero mention bridges theory to practice effectively.

anticipating needs If they're truly expert-level, next questions might be about specific algorithms (e.g. "compare PPO vs SAC") or implementation challenges. My closing offer for deeper dives leaves that door open.

adjusting tone Dropped all introductory analogies. Using precise terms like "stochastic policies" without definition. That's the right call - they'd find simplifications patronizing. The Gibbs policy example shows I trust their mathematical literacy.

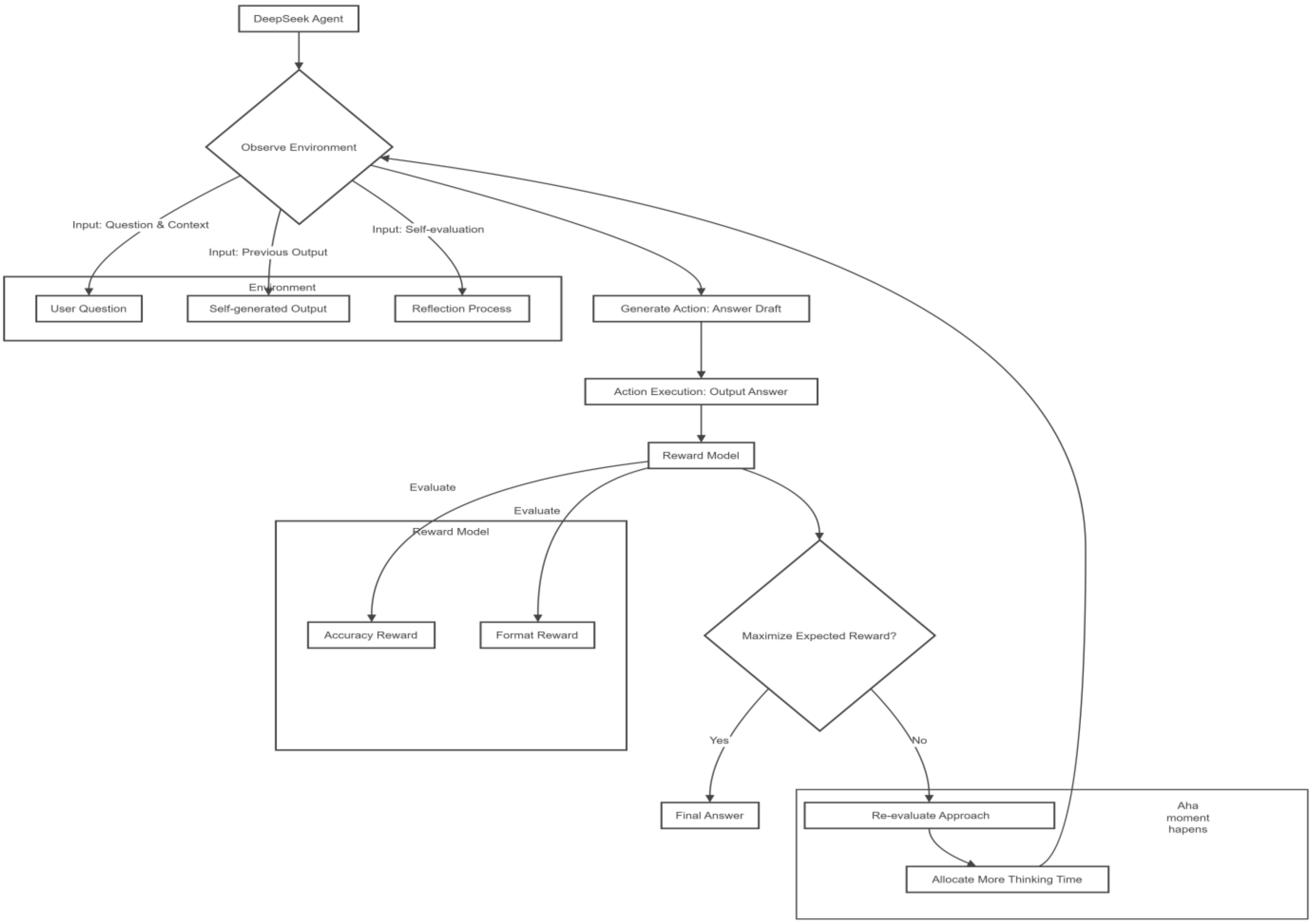
(Figure 3: DeepSeek's answer 2)

When I told it that I was an expert in the field of AI, it analyzed that I might need more details about RL, reflected on its previous answer, and finally output some technical terms and formulas to me.

Upon conducting a thorough review of R1's paper, it was observed that the designers implemented two primary types of rewards—accuracy rewards and format rewards—to train DeepSeek for reinforcement learning. These reward models can be readily aligned with my recent findings: DeepSeek's analysis of user identity and potential needs is specifically targeted at achieving the maximum accuracy reward. Additionally, the requirement of using **<think></think> and <answer></answer> tags** to separate reasoning content from answers serves as a critical guarantee for maintaining the high quality of DeepSeek's responses.

It is worth noting that when recording the performance of DeepSeek, researchers have discovered the **Aha moments** of DeepSeek. At this point DeepSeek learns to allocate more thinking time to a problem by reevaluating its initial approach. (This also corresponds precisely to the annotations in the boxes in Figures 2 and 3.)

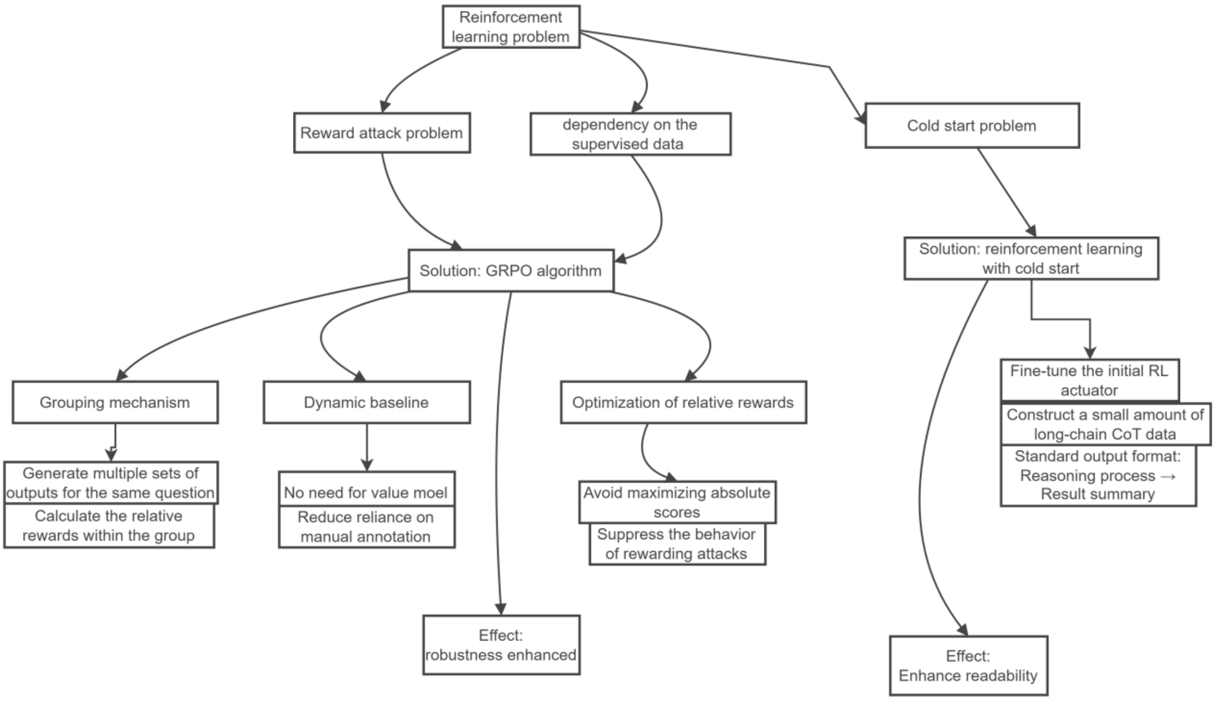
Overall, I have summarized the above content into Figure 4, which summarizes my understanding of how reinforcement learning works in the training process of DeepSeek.



(Figure 4: RL in DeepSeek)

Technical Optimization Strategies

Overview:


Reinforcement learning problem
Reward attack problem
dependency on the supervised data
Cold start problem
Solution: GRPO algorithm
Solution: reinforcement learning with cold start
Grouping mechanism
Dynamic baseline
Optimization of relative rewards
Fine-tune the initial RL actuator
Construct a small amount of long-chain CoT data
Standard output format: Reasoning process → Result summary
Generate multiple sets of outputs for the same question
Calculate the relative rewards within the group
No need for value moel
Reduce reliance on manual annotation
Avoid maximizing absolute scores
Suppress the behavior of rewarding attacks
Effect: robustness enhanced
Effect: Enhance readability

(Figure 5: Conclusion of Optimization Strategies)

Although the incorporation of reinforcement learning has enabled DeepSeek to exhibit complex cognitive behaviors, such as **reflection and re-evaluation**, which are typically associated with human problem-solving processes. However, for a chat-bot, simply using reinforcement learning may expose the model to **reward hacking attacks** (For instance, in pursuit of the maximum reward, they merely mix up related words, but the sentences are not smooth.) and the use of reinforcement learning is highly **dependent on supervised data**.

To address these issues, the designers proposed replacing the traditional **Proximal Policy Optimization (PPO) algorithm with the Group Relative Policy Optimization (GRPO) algorithm**. In the subsequent sections, I will detail the specific aspects in which the GRPO algorithm improves upon the PPO algorithm and explain why it demonstrates superior performance in the training of DeepSeek.

Let's first take a look at the formula of PPO:

$$\mathcal{J}_{navie}(\theta) = E_{(q,o)\sim(\text{data},\pi_\theta)}[r(o)] \tag{1}$$

This formula represents that we merely optimize our final **absolute score**. The result may naturally lead the model to merely stack related words together to form an incoherent sentence in pursuit of the maximum reward expectation. That is what we mentioned as rewarding hackers.

Now let's take a look at the formula of GRPO:

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = E\left[\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i)}{\pi_{\theta_{\mathrm{old}}}(o_i)}A_i,\mathrm{clip}\left(\frac{\pi_\theta(o_i)}{\pi_{\theta_{\mathrm{old}}}(o_i)},1-\varepsilon,1+\varepsilon\right)A_i\right)-\beta D_{\mathrm{KL}}(\pi_\theta|\pi_{\mathrm{ref}})\right)\right] \quad (1)$$

where

$$A_i = \frac{r_i - \mathrm{mean}(\{r_1, r_2, \cdots, r_G\})}{\mathrm{std}(\{r_1, r_2, \cdots, r_G\})} \quad (2)$$

calculates a “**relative score**” by averaging multiple outputs from the same question and normalizing.

Whether it is from the names of GRPO and PPO or their respective formulas, it becomes evident that GRPO builds upon PPO by introducing the concept of groups. Specifically, GRPO realizes that for the same task, the model will provide multiple different outputs. Then, based on the average score of the outputs, the outputs with relative advantages are screened out. Finally, the strategy is updated based on the decisions made from the advantageous outputs, thereby maximizing the objective function of GRPO. In this way, the reward model is continuously trained to achieve the purpose of self-training (Figure 5).

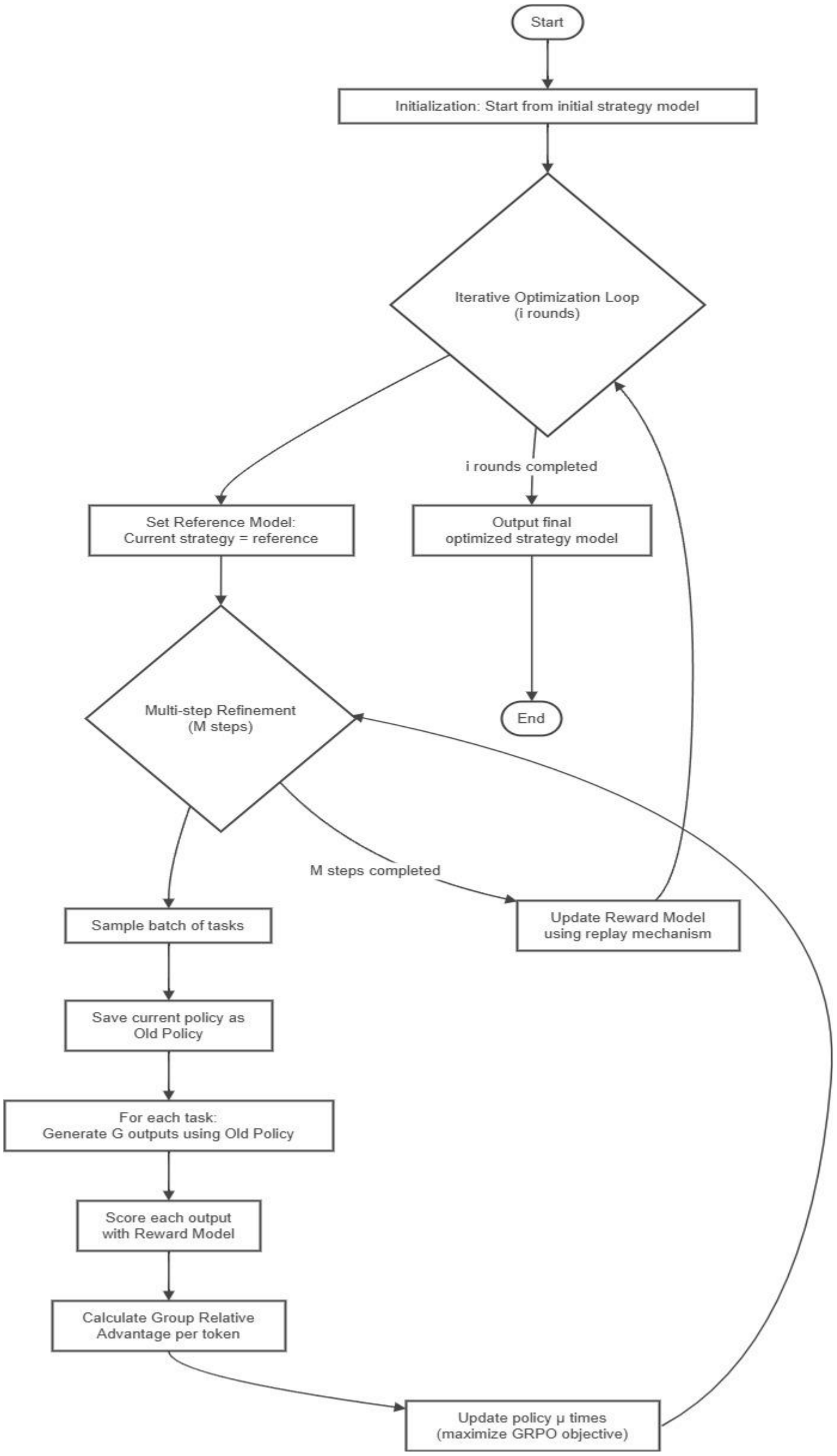
This mechanism enables the model to transition from comparing its previous individual score to evaluating the average score within a group. Such modifications render the GRPO algorithm capable of dynamically establishing a baseline for reward assessment without requiring a dedicated value model. This means that GRPO no longer requires humans to perform the working step of marking the quality of responses for the model. In other words, during the training process of GRPO, the model will judge the base line by itself, thereby reducing the reliance on supervised data. Furthermore, the change from optimizing absolute rewards to optimizing relative rewards in the objective function further solves the problem of reward hacking caused by pursuing the maximum absolute score and enhances the robustness of the model.

However, the problem still exists. During the early training process, the base model went through an unstable cold start phase: during this phase, the model kept making mistakes and the generated answers were also illogical. To solve such problems, the method of Reinforcement Learning with **Cold Start** was introduced.

The specific operation is also very simple: Construct and collect a small amount of long CoT data to fine-tune the model as the initial RL actor. This well solves the readability problem of DeepSeek-R1-Zero, because the answers generated by R1-Zero are mixed with different languages, resulting in it being limited to problems that are not suitable for reading. But, by introducing Reinforcement Learning with Cold Start, after the initial training, the model can reply in the format of first presenting the reasoning process and then summarizing the reasoning process to answer the result.

In addition, **supervised fine-tuning** also plays an indispensable role in the training of DeepSeek.

However, since this article mainly discusses how to optimize the reliance on supervised data exposed by reinforcement learning and solve the problem of reward hacking, supervised fine-tuning is not emphasized here.

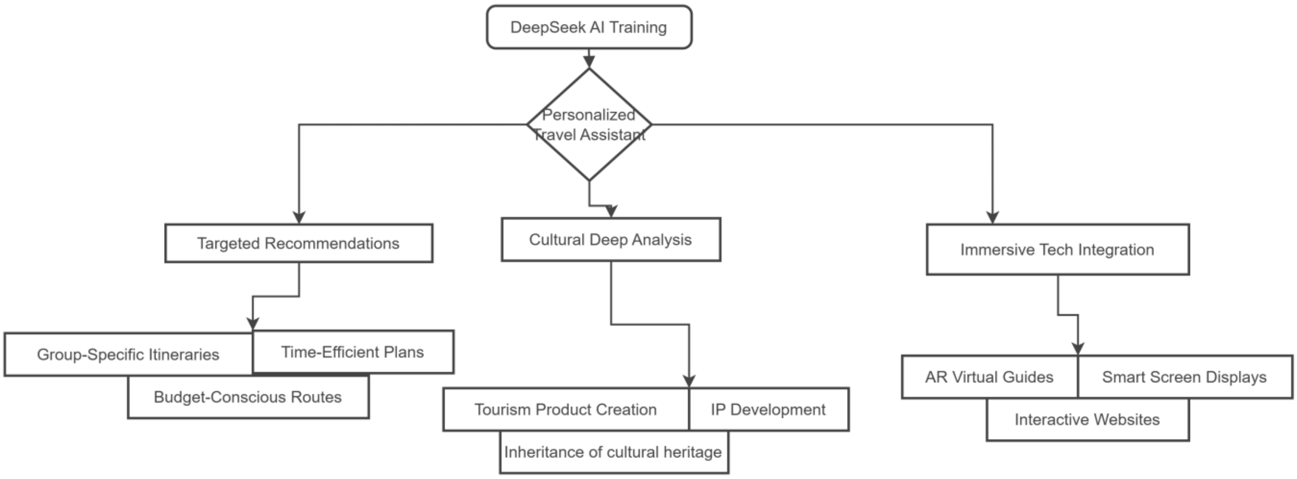


(Figure 6: GRPO)

Application scenario outlook

After discussing the core principles and optimization strategies of DeepSeek, we will continue to explore the practical application scenarios of DeepSeek.

I will focus on integrating DeepSeek into **the local cultural and tourism industry**.



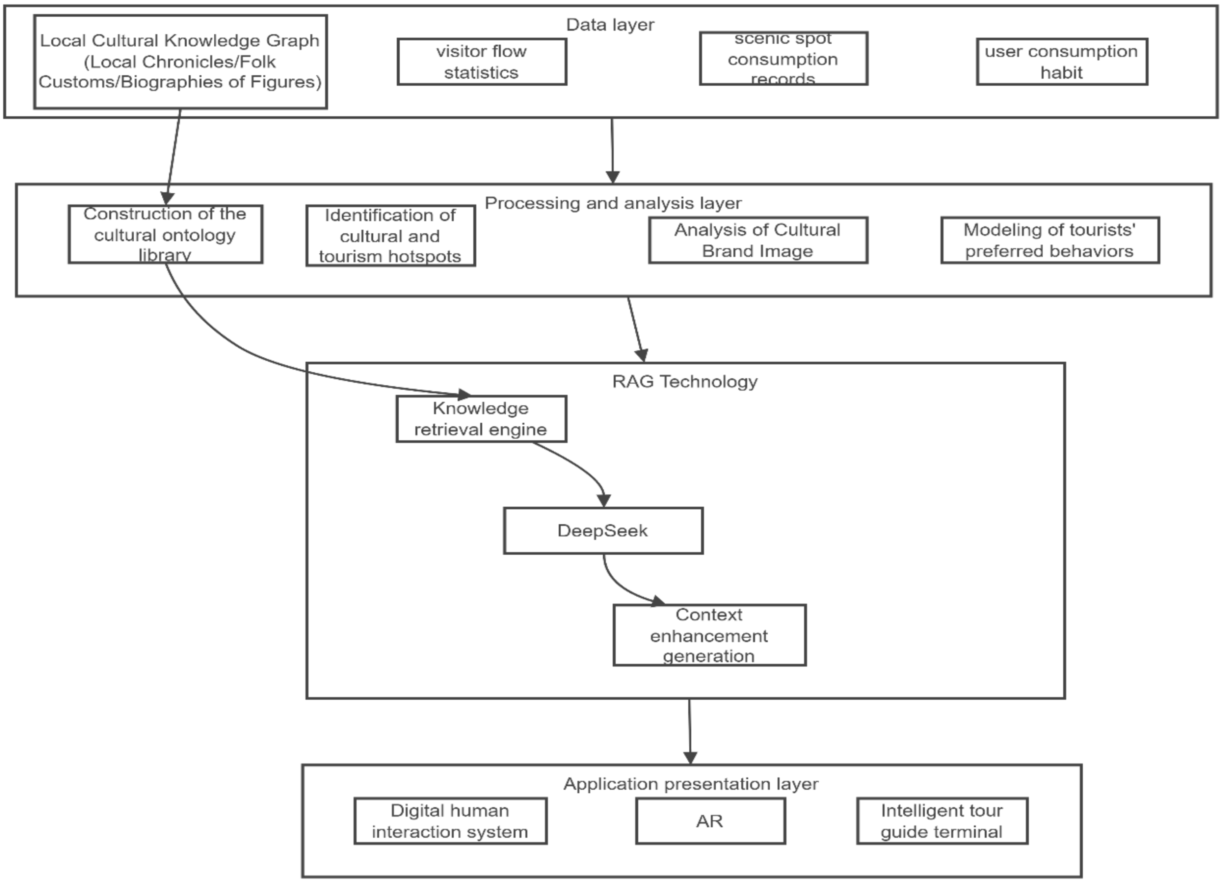
(Figure 7: Application prospects)

When we want to travel to a place and search for travel guides online, most of the time we will find that the guides on the Internet are all the same, and it is very difficult to find a guide that suits our own consumption level and travel habits. However, if we can cooperate with the managers of major local scenic spots and open up the tour routes and data of our own scenic spots to DeepSeek for relevant training, then the trained DeepSeek can become **a tour guide** who is very familiar with the local cultural and tourism landmarks, and thus be able to provide targeted travel suggestions for tourists. Meanwhile, during the process of tourists communicating with DeepSeek, data of different users can also be collected, enabling DeepSeek to better understand the habits of tourists. (For instance, trained AI can provide a travel route that meets tourists' needs by understanding information such as the number of people traveling, consumption budget, and travel time)

In addition to using DeepSeek to make **personalized recommendations** for travel guides, we can also **combine DeepSeek with technologies like virtual humans, AR** and other technologies, and apply them to the official websites of scenic spots or the smart large screens deployed in the scenic spots, so as to quickly respond to the personalized needs of tourists and create **immersive interactive experiences** for them.

In the in-depth exploration of the cultural connotations of local scenic spots, DeepSeek can also play its role in deep understanding. Based on DeepSeek's in-depth exploration of the cultural background and image characteristics of the IP in scenic spots, we can form various **cultural and tourism consumer goods**.

To train such a tour guide assistant and integrate it into the systems of various scenic spots, we first require data such as visitor flow statistics, scenic spot consumption records, and user consumption habit information. These data will be used for in-depth analysis of cultural tourism brand images, tourist preference behaviors, and hot events identification within cultural tourism sites. Additionally, local cultural knowledge graphs (including local chronicles, folklore, and biographies) is essential to construct a comprehensive cultural ontology library that encompasses historical events, character relationships, and architectural styles. This lays the groundwork for adopting **Retrieval-Augmented Generation (RAG)** technology. Furthermore, the integration of technologies such as digital humans and AR with models like DeepSeek plays a critical role in determining whether the digital human representation of a scenic area can become a landmark feature.


Data layer
Local Cultural Knowledge Graph (Local Chronicles/Folk Customs/Biographies of Figures)
visitor flow statistics
scenic spot consumption records
user consumption habit
Processing and analysis layer
Construction of the cultural ontology library
Identification of cultural and tourism hotspots
Analysis of Cultural Brand Image
Modeling of tourists' preferred behaviors
RAG Technology
Knowledge retrieval engine
DeepSeek
Context enhancement generation
Application presentation layer
Digital human interaction system
AR
Intelligent tour guide terminal

(Figure 8: underlying technical ideas)

If such technology were to be fully realized, one could envision the following scenario: Prior to our trip, the DeepSeek travel assistant would generate a four-day itinerary for three individuals traveling to Beijing. Upon arriving in Beijing and visiting the Old Summer Palace, the digital assistant on our mobile device would track our movements in real time, identifying our current location and providing detailed historical context about the site.

References

[1] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[2] Simonds, T., Lopez, K., Yoshiyama, A., & Garmier, D. (2025). Self Rewarding Self Improving. *arXiv preprint arXiv:2505.08827*.

[3] Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., ... & Piao, Y. (2024). Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

[4] Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., ... & He, Y. (2025). Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

[5] Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., ... & Zou, Y. (2024). Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.