Computers and Artificial Intelligence; Vol.3, No.4; 2026
E-ISSN: 3006 -3728 P-ISSN: 3006-371X
DOI: https://doi.org/10.70267/cai.26v3n4.4659
Published by: Zeus Press

Large Language Models and Reinforcement Learning: A Taxonomy of Integration Paradigms, Challenges, and Future Directions

Xiqian Lu*

School of Artificial Intelligence and Computer Science (School of Software), Jiangnan University, Jiangsu, 214122, China

**Corresponding author: Xiqian Lu.*

Abstract

This paper examines the integration of large language models (LLMs) and reinforcement learning (RL) in recommender systems, focusing on their theoretical foundations and structural challenges. It highlights the transition from static prediction to sequential decision-making, emphasizing RL's strengths in long-term reward optimization and interaction modeling, and LLMs' advantages in semantic understanding and reasoning. Their complementary limitations—RL's weak semantic representation and LLMs' lack of long-term optimization—justify their integration. Existing research is classified into "LLM-enhanced RL" and "RL-shaped LLM," with roles including representation enhancement, reward modeling, policy generation, and environment simulation, under varying coupling levels. The paper proposes a unified three-dimensional framework based on information sources, optimization time scale, and coupling strength, showing that performance differences arise from structural positioning rather than model scale. Key challenges include balancing expressiveness and efficiency, long-term optimization and training stability, and generalization versus specialization. The paper also identifies limitations in evaluation protocols and experimental design, calling for standardized frameworks for long-term value assessment. Overall, integrating LLMs and RL is crucial for advancing recommender systems toward intelligent decision-making agents, with future work focusing on stable coupling and unified evaluation.

Keywords

large language models, reinforcement learning, recommend systems, hybrid intelligent systems, sequential decision making

1. Introduction

With the rapid development of the Internet and mobile applications, recommender systems have become indispensable infrastructure in e-commerce, social media, and online education. Their core objective is to predict user interests based on historical behavior and contextual information for personalized delivery. Early methods, such as collaborative filtering and content matching, relied on offline training to optimize short-term metrics. However, recommender systems should provide diverse choices rather than only optimizing short-term performance [1]. Traditional supervised learning methods often overlook user–system interactions and

fail to utilize real-time feedback [2], leading to limitations in long-term value optimization, multi-round interactions, and semantic understanding [3].

To address these issues, a recommendation has been formulated as a Sequential Decision Process using Reinforcement Learning (RL), which enables long-term reward optimization and dynamic interaction modeling [2]. However, RL has limitations in semantic understanding and content reasoning, making it difficult to capture complex user intentions. It also faces challenges in offline training and incurs high costs in online exploration [4]. Moreover, RL lacks interpretability, making it difficult to explain recommendation results [5].

Meanwhile, Large Language Models (LLMs) have demonstrated strong capabilities in semantic understanding, contextual modeling, and knowledge reasoning [6]. In recommender systems, LLMs can generate user profiles and support transparent recommendation [7], while their generative ability enables natural language explanations [8]. They are also applied in semantic enhancement, cold-start, and conversational recommendation. However, LLMs' optimization objectives differ from policy optimization in recommendation, and their offline training paradigm limits long-term interactive optimization [4]. In addition, their inference cost remains high.

Overall, RL and LLMs are highly complementary: RL excels in long-term decision-making, while LLMs provide semantic and generative intelligence. Most lack a structural understanding of capability boundaries and scenario differences, as well as sufficient theoretical analysis of evaluation, long-term value modeling, and system coupling. To address these gaps, this paper proposes a unified three-dimensional analytical framework. This framework explains performance differences across scenarios such as cold-start recommendation, long-term value optimization, real-time recommendation, and conversational recommendation.

2. Theoretical Background

2.1 Recommend Systems as Sequential Decision Processes

Traditional recommender systems typically treat recommendations as a static prediction problem, estimating immediate preference scores such as click-through rates or ratings based on historical behaviors [5]. This paradigm assumes stable user preferences and independent recommendation events. However, in real-world scenarios, user–system interactions are dynamic and continuous: recommendations influence user feedback and shape future interests, usage patterns, and platform engagement. As a result, single-step prediction is insufficient for modeling long-term value and evolving user behavior.

To address this limitation, a recommendation can be formulated as a sequential decision problem under the Markov Decision Process (MDP) framework [4]. In this setting, the objective shifts from optimizing immediate feedback to maximizing long-term cumulative rewards through policy learning, thereby improving user lifetime value and overall system effectiveness.

Moreover, sequential decision modeling captures the feedback-loop nature of recommendation, where user preferences evolve with past interactions and system policies influence future data distributions. This may lead to issues such as bias and interest convergence, while also requiring a balance between exploration and exploitation to support long-term value growth. These characteristics provide the theoretical foundation for applying reinforcement learning in dynamic recommendation scenarios [9,10].

2.2 Reinforcement Learning for Decision Optimization

Reinforcement learning (RL) provides a principled framework for optimizing long-term decision-making through interaction with the environment. In recommender systems, RL shifts the objective from short-term prediction to maximizing long-term cumulative rewards, thereby addressing the limitations of traditional approaches.

RL offers several advantages in recommendation scenarios. First, it enables explicit optimization of long-term user value, such as retention and lifetime value, which cannot be directly achieved by supervised learning [9]. Second, it naturally addresses the exploration–exploitation trade-off, allowing systems to balance between recommending known high-reward items and exploring potentially valuable new items [9]. In addition, RL

can optimize non-differentiable objectives such as novelty [11], adapt to dynamic user preference changes through interaction [5], and effectively model multi-round interactions, such as in conversational recommendation.

However, RL also faces several challenges. Its state representation often relies on limited features or shallow embeddings, restricting its ability to capture complex semantic and multimodal information. Offline training struggles to estimate unseen states, while sparse rewards and large action spaces lead to low sample efficiency and training instability [13;11;14;15]. Moreover, RL lacks semantic understanding and reasoning capabilities, making it less effective in open-domain and explainable recommendation scenarios. Therefore, despite its strengths in long-term optimization, RL requires integration with other methods to better model complex user intentions.

2.3 Large Language Models for Semantic Understanding

When integrated into recommendation systems, they significantly enhance semantic understanding and text generation capabilities. LLMs are typically based on the Transformer architecture and are pretrained on large-scale corpora through self-supervised learning using the self-attention mechanism [16], enabling them to learn language distribution patterns and knowledge structures. After pretraining, they can be fine-tuned to adapt to specific tasks [17], such as text classification, question answering, or sequence labeling, demonstrating strong generalization abilities in semantic modeling, contextual understanding, and knowledge reasoning.

In recommendation systems, LLMs can encode user behaviors, reviews, and product descriptions into high-quality semantic vectors, thereby enhancing the semantic alignment between user and item representations [12]. Their application scenarios include cold-start recommendation, conversational recommendation, and user profile generation, enabling the capture of multi-level interest preferences [6,7,18]. LLMs can also extract recommendation rationales and generate natural language explanations, improving system transparency and user trust [16].

However, LLMs also have several limitations: they lack closed-loop feedback mechanisms, and their optimization objectives are not fully aligned with long-term user value [19]; their reasoning incurs high computational costs, which constrains real-time recommendation [12]; fine-tuning cycles are long, making it difficult to capture rapid changes in user interests [20]; and generated outputs may contain hallucinations, while prompt length limitations can lead to unstable responses [21].

2.4 Complementarity of LLM and RL

Reinforcement learning and large language models exhibit significant complementary capabilities in recommendation systems. RL excels at long-term policy optimization, multi-round interaction, and online learning, but has limited semantic understanding; LLMs are strong in semantic comprehension, text generation, and knowledge reasoning, but lack long-term optimization and online adaptation capabilities [22].

Based on this complementarity, current research has proposed several integration strategies. First, LLMs can be used at the state representation layer to provide rich semantic features for RL. For example, LLMs can serve as state models to generate semantically enriched user state representations for reinforcement learning, thereby enhancing the policy's perception of user intent [22]. Second, at the policy level, RL can guide LLMs to generate recommendation content or decision suggestions. For instance, feedback signals from recommendation systems can be directly used to optimize LLM-generated query rewriting [19]. Third, in terms of training paradigms, RL and LLMs can adopt either joint training or staged training approaches, enabling the system to maintain strong semantic understanding while optimizing long-term cumulative rewards [23]. In addition, multi-agent collaborative designs have also been explored, such as cooperation between selectors and recommenders to dynamically adjust reward functions and uncertainty estimation, thereby improving the system's adaptability in multi-round interaction scenarios [15].

In summary, integrating RL with LLMs not only compensates for the limitations of each approach but also forms a hybrid capability of "semantic intelligence + decision intelligence." This combination enables recommendation systems to achieve stronger adaptability in scenarios such as cold-start problems, multi-round interactions, long-term user value optimization, and conversational recommendation.

3. Literature Review

3.1 Reinforcement Learning–Based Recommendation Systems

3.1.1 RL Paradigms for Recommendation

Reinforcement learning (RL) formulates recommendation as a sequential decision-making problem, shifting the objective from short-term prediction to long-term cumulative reward optimization [5]. Unlike traditional methods that focus on immediate feedback such as click-through rates or ratings, RL models the dynamic interaction between users and systems, where current recommendations influence future user behavior and interest evolution.

Existing RL-based recommender systems can be broadly categorized into contextual bandit models, value-based methods, policy gradient methods, and offline reinforcement learning approaches [5]. Contextual bandit methods focus on optimizing immediate rewards and balancing exploration and exploitation, but lack the ability to model long-term user dynamics [5]. To address this limitation, recommendation is further modeled as a Markov Decision Process and optimized using deep reinforcement learning methods, including value-based approaches such as DQN [5,24], policy gradient methods [5,4], and Actor–Critic frameworks [5,11], which enable long-term policy optimization.

In addition, model-based and offline reinforcement learning methods have been proposed to improve sample efficiency and reduce the cost of online interaction [5]. Model-based approaches construct user simulators to support policy learning, while offline RL learns policies directly from logged data without real-time interaction [5,15]. However, these approaches face challenges such as model bias accumulation, limited coverage of state–action space, and out-of-distribution action issues [15]. Techniques such as uncertainty modeling and diversity constraints have been explored to mitigate these problems [14], but their stability and generalization remain limited.

Despite their advantages, RL-based recommender systems still exhibit several structural limitations. Large action spaces and sparse, delayed feedback increase training difficulty and reduce sample efficiency [5,13]. Moreover, RL methods typically rely on interaction data and lack strong semantic representation capabilities, leading to weak performance in cold-start scenarios and difficulty in modeling complex user intentions [3].

3.1.2 Core Capability Boundaries of RL4Rec

From the evolution of the above methods, it can be seen that reinforcement learning provides a systematic theoretical framework for optimizing long-term user value in recommender systems, enabling recommendation tasks to shift from short-term prediction to dynamic decision-making and control. However, in practical applications, RL4Rec (Reinforcement Learning for Recommendation) still faces several structural capability limitations. Limited representation capability. Traditional reinforcement learning methods usually rely on handcrafted features or low-dimensional embedding representations, which makes it difficult to capture the high-dimensional and complex relationships between user interests and item semantics. This limitation is particularly evident in cold-start scenarios, where learning effective representations for new users or new items is especially challenging [3].

Reward sparsity and complex reward design. In real-world systems, user feedback is often sparse and delayed. Designing a reward function that can both reflect immediate user satisfaction and guide the optimization of long-term value remains a core challenge for applying reinforcement learning in recommendation scenarios [5].

Training instability and low sample efficiency. Whether using value-based methods, policy gradient methods, or Actor–Critic frameworks, reinforcement learning algorithms may suffer from slow convergence and training oscillations. Moreover, they typically require large-scale interaction data, leading to relatively low sample efficiency [5].

Offline evaluation bias. Due to the high cost of online experiments, most studies rely on offline datasets for validation. However, the distribution mismatch between static logged data and the real dynamic environment often causes offline evaluation results to fail to accurately reflect the model's performance in real deployment settings [25].

By enhancing representation capabilities, assisting with reward design, and improving policy generalization, LLMs may help overcome the performance bottlenecks of existing reinforcement learning–based recommendation frameworks.

3.1.3 Structural Comparison of RL-based Recommenders

Table 1: Structural Comparison of RL-based Recommender Systems (RL4Rec)

Paradigm	Long-term Reward Modeling	Data Requirement	Sample Efficiency	Cold-start Capability	Training Stability	Structural Limitation
Contextual Bandits (MAB)	×	Low	High	Weak	High	No state transition modeling
Value-based Methods (e.g., DQN)	√	High	Medium	Weak	Medium	Large discrete action space challenge
Policy Gradient Methods	√	High	Low	Weak	Low	High variance in gradient estimation
Actor–Critic Methods	√	High	Medium	Weak	Medium	Value estimation bias; training oscillation
Offline Reinforcement Learning	√	High (logged data)	Medium	Weak	Medium	Distributional shift and extrapolation error

As shown in Table 1, RL-based recommender systems demonstrate theoretical advantages in modeling long-term cumulative rewards, as they can explicitly optimize long-term user value through a sequential decision-making feedback loop. However, their strong reliance on large-scale interaction data creates an inherent disadvantage in cold-start scenarios. In addition, limited sample efficiency and training instability remain key challenges in high-dimensional recommendation environments. These structural bottlenecks reflect the limitations of traditional RL4Rec frameworks in semantic representation capability, and also provide a theoretical basis for introducing large language models (LLMs) to enhance representation and generalization abilities.

3.2 Large Language Models in Recommender Systems

3.2.1 Roles of LLMs in Recommender Systems

Large language models (LLMs) have been widely applied in recommender systems due to their strong semantic understanding and generative capabilities. Existing studies generally categorize their roles into three types: semantic encoders, generators, and reasoning planners.

As semantic encoders, LLMs leverage textual information such as item descriptions, user reviews, and interaction histories to construct richer user–item representations. Compared with traditional ID-based embeddings, this approach enables better semantic alignment and improves performance in cold-start scenarios [6,16]. Empirical studies such as SEALR and TALLRec further demonstrate that LLM-based representations can capture fine-grained user preferences and exhibit strong few-shot and cross-domain generalization capabilities [12,6].

As generators, LLMs enable recommender systems to produce natural language outputs, supporting conversational recommendation and explainable recommendation scenarios. By understanding user intent from dialogue context, LLMs can generate personalized recommendations and provide explanations, thereby improving user interaction and system transparency [21,18]. However, purely generative approaches may suffer from hallucination issues, where generated items are inconsistent with real data. To address this, retrieval–generation hybrid architectures are often adopted to constrain the generation process and improve reliability [12].

As reasoners and planners, LLMs can infer users' latent preferences from complex behavioral and contextual information, supporting more sophisticated recommendation decisions. Through structured prompting and instruction-based inputs, LLMs are able to model implicit user intent and generate recommendations that better align with user needs [12,18].

Despite these advantages, LLM-based recommender systems lack explicit mechanisms for long-term optimization and interactive feedback, limiting their effectiveness in sequential decision-making scenarios.

3.2.2 Structural Limitations of LLM-Only Recommenders

LLMs demonstrate strong capabilities in semantic understanding, content generation, and reasoning. However, building recommender systems solely based on LLMs still faces several structural limitations that constrain their effectiveness in real-world scenarios.

First, LLMs lack long-term optimization capability. Traditional reinforcement learning methods can optimize policies based on long-term user feedback and maximize cumulative rewards [22], whereas LLMs operate as static models that rely on current inputs and pretrained knowledge, without explicit mechanisms for sequential decision optimization [20]. As a result, they are not well suited for tasks requiring long-term planning.

Second, LLMs face challenges in policy updating. Their knowledge is encoded in model parameters, and updating this knowledge typically requires retraining or fine-tuning, which is costly in dynamic environments. Hybrid strategies combining periodic fine-tuning with retrieval-augmented generation (RAG) have been proposed to improve adaptability [20], but these approaches still indicate limited ability for real-time adaptation.

Third, hallucination remains a major issue in generative recommendation. LLMs may produce non-existent items or inaccurate content when reliable evidence is lacking [21]. Although retrieval–generation architectures can mitigate this problem by constraining the candidate space [21], the final output quality still depends heavily on the retriever.

Fourth, computational cost poses a significant challenge for deployment. While techniques such as ID-based representations can improve efficiency [12], LLM-based systems remain substantially more expensive than traditional recommendation models, limiting their scalability.

Finally, LLMs struggle to satisfy strict numerical or combinatorial constraints. In scenarios such as dietary recommendation, generated results may violate explicit constraints unless combined with external optimization methods [21].

This motivates the integration of LLMs with reinforcement learning, where LLMs provide semantic understanding and generation, while RL enables long-term optimization and adaptive decision-making.

3.2.3 Structural Comparison of LLM-based Recommenders

Overall, the above analysis shows that LLMs mainly play three roles in recommender systems: semantic encoders, generators, and reasoning planners. However, different methods exhibit significant differences in terms of data dependency structures, computational cost, and long-term optimization capability. Table 2 summarizes the capability boundaries of representative LLM-based recommendation methods.

Table 2: Structural Comparison of LLM-only Recommendation Methods

Method	Primary Role	Data Dependency	Computational Cost	Long-term Optimization	Cold-start Capability	Structural Limitation
SEALR	Emotion-aware semantic encoder	User reviews + textual metadata	High	×	Strong	No sequential decision loop
TALLRec	Few-shot alignment framework	Limited recommendation samples	Medium–High	×	Strong	Single-step relevance optimization
Retrieval–Generation Cascade	Generative re-ranking	Candidate pool + textual context	High	×	Moderate	Dependent on retrieval quality
Constrained LLM-based Frameworks	Generator + external optimizer	Structured constraints + textual input	High	×	Moderate	No adaptive policy learning
PALR	Reasoning-enhanced personalization	User behavior + textual signals	High	×	Moderate	High update cost; no dynamic feedback loop

3.3 Integration of LLM and RL: Emerging Paradigms

3.3.1 Motivation and Perspective Shift

The integration of LLMs and RL is driven by their complementary strengths: RL excels at long-term sequential decision-making, while LLMs provide strong semantic understanding and reasoning capabilities. Recent research has explored two main directions: LLM-enhanced RL, where LLMs improve representation, reward modeling, and policy learning [8], and RL shaping LLMs, where RL aligns generation behavior with task objectives, as exemplified by RLHF [17]. However, existing work largely focuses on one-way enhancement and lacks a unified view of their interaction mechanisms. To address this gap, this paper proposes an asymmetric bidirectional framework that jointly analyzes these two paradigms from the perspectives of role, coupling, and optimization structure.

3.3.2 LLM-Enhanced Reinforcement Learning

In this paradigm, LLMs are integrated into different components of the RL pipeline to enhance perception, decision-making, and learning capabilities. Their roles can be systematically categorized into representation enhancers, reward models, policy generators, and environment simulators.

As representation enhancers, LLMs transform raw observations such as natural language or multimodal inputs into semantically enriched state representations, improving the agent's ability to understand complex environments. In this setting, LLMs are typically used as frozen modules and do not participate in policy optimization [8]; As reward models, LLMs generate or shape reward signals by translating human intentions or task descriptions into optimizable objectives. This is particularly useful in sparse-reward or multi-objective settings, and is closely related to the reward modeling paradigm in RLHF [17]. LLM-based reward models also enable optimization of non-differentiable objectives such as novelty or diversity [26]; As policy generators, LLMs directly produce actions or action sequences and are optimized through RL objectives. In this tightly coupled setting, the decision process is reformulated as sequence generation, where the LLM is updated via policy gradients to maximize cumulative rewards [19,27]; As environment simulators, LLMs model environment dynamics by generating future states and rewards, enabling model-based RL with improved sample efficiency. However, the reliability of such approaches depends on the accuracy of the generated trajectories, and simulation bias may affect policy learning [8].

Based on the interaction between LLMs and RL, three levels of system coupling can be identified. In weak coupling, LLMs act as fixed components and do not participate in RL optimization. In modular coupling, LLMs provide intermediate signals such as rewards or state predictions without receiving gradient updates. In tight coupling, RL objectives are directly backpropagated to LLM parameters, enabling joint optimization of generation and decision-making [8].

Correspondingly, three main training paradigms can be identified. The supervised pretraining followed by RL fine-tuning paradigm first equips the model with general capabilities and then optimizes task-specific objectives. The LLM-based reward offline RL paradigm uses LLM-generated rewards to guide learning from static datasets, avoiding costly online interaction [26]. The LLM-simulator-based model RL paradigm leverages synthetic trajectories generated by LLMs to improve sample efficiency [8].

Despite its advantages, LLM-enhanced RL faces several structural limitations. LLM-based reward signals may be noisy and sensitive to prompts, leading to unstable optimization [17]. LLM-based simulators may introduce bias when generated trajectories deviate from real dynamics [8]. In tightly coupled settings, high computational cost and scalability constraints become significant challenges., particularly when the complexity of the environment increases and the need for real-time processing intensifies. In addition, RL optimization may lead to catastrophic forgetting of pretrained knowledge, requiring careful balancing between adaptation and knowledge retention [17].

3.3.3 RL Shaping Large Language Models

In contrast to LLM-enhanced RL, this paradigm treats reinforcement learning as a tool for optimizing the behavior of large language models, where the generation process itself is viewed as a policy. Rather than embedding LLMs into the RL pipeline, this direction focuses on post-training optimization, aiming to align model outputs with human preferences or task-specific objectives.

One representative pathway is RL as an alignment mechanism, typically instantiated by reinforcement learning from human feedback (RLHF). In this setting, a reward model is first trained using human preference data, and is then used to guide policy optimization, enabling LLMs to produce outputs that better align with human expectations [17]. Unlike classical reinforcement learning, the "environment" in RLHF is not a dynamic system but a static feedback mechanism based on preference data. As a result, this paradigm is closer to preference-driven distribution adjustment than to interactive decision-making.

Another important direction is RL as an outcome-level optimizer, where task-specific evaluation metrics are used as reward signals to directly optimize generation quality. In such approaches, the reward is typically defined based on final outputs, such as answer correctness or downstream system performance, and the model is trained to maximize these objectives [19,27]. Compared with alignment-based methods, this paradigm emphasizes improving task performance rather than matching human preferences.

Despite these advances, the RL → LLM paradigm differs fundamentally from classical reinforcement learning. First, it generally lacks real-time interaction with an environment, relying instead on offline feedback or evaluators. Second, rewards are often defined at the sequence level and are delayed, leading to the token-level credit assignment problem in long-text generation. Third, RL fine-tuning may cause catastrophic forgetting, where pretrained knowledge is degraded during optimization, requiring mechanisms to balance adaptation and knowledge retention [17].

Overall, RL shaping LLMs can be understood as policy optimization over language generation distributions, rather than traditional sequential decision-making, highlighting its complementary role to LLM-enhanced RL in integrated recommender systems.

3.3.4 Comparative Analysis and Research Gaps

The above analysis shows that the LLM→RL direction has developed a relatively complete role taxonomy: representation enhancers, reward models, policy generators, and environment simulators collectively cover the key components of the RL pipeline, forming a continuous spectrum from weak coupling to modular coupling and further to tight coupling.

In contrast, the RL→LLM direction is currently still dominated by alignment and outcome-level optimization, with a relatively limited set of roles. In this direction, RL mainly functions as an external optimizer, using reward signals to adjust generation strategies, and has not yet developed a level of structural diversification comparable to that of LLM→RL direction.

A closer examination reveals that the differences between the two classes of integration paradigms are often fundamentally reflected in three key questions: Does decision-making primarily rely on semantic priors or interactive feedback? Is a long-horizon sequential optimization loop established? And is there tight gradient-level coupling between LLMs and RL?

Overall, the integration of LLMs and RL is evolving from "unidirectional enhancement" toward "bidirectional synergy." However, to truly understand the performance differences of various integration paradigms in scenarios such as cold-start conditions, long-horizon optimization, and real-time deployment, it remains necessary to establish a higher-level unified analytical perspective. Based on this motivation, the next chapter will introduce a three-dimensional structural framework to systematically synthesize and interpret existing research on LLM–RL integration.

4. Discussion & Synthesis

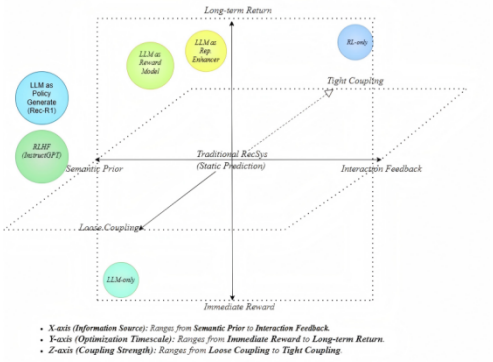
4.1 Unified Analytical Framework: From Method Taxonomy to Structural Explanation

Existing studies on the integration of LLMs and RL often classify approaches based on model architecture or technical pathways, for example distinguishing representation-enhanced, reward-enhanced, or policy-generation-based methods [8,22]. However, while such taxonomies are useful for tracing the evolution of techniques, they are less effective at systematically explaining the performance differences of various paradigms across typical scenarios such as cold-start settings, long-term value optimization, real-time recommendation, and conversational recommendation. To address this limitation, this paper proposes a three-dimensional unified analytical framework, illustrated in Figure 1, which structurally synthesizes existing

integration paradigms along three abstract dimensions: information source, optimization time horizon, and system coupling strength.

This framework emphasizes that the performance differences among recommendation paradigms fundamentally stem from their positions within the three-dimensional structural space, rather than from simple variations in model scale or training details.

Figure 1: Three-Dimensional Framework for LLM-RL Integration



4.2 Information Source Dimension: Semantic Priors and Interaction Feedback

The information source dimension focuses on what type of information the model primarily relies on when making decisions. Existing approaches can be broadly categorized into three types.

Semantic-prior-driven (LLM-dominated): decision-making primarily relies on the semantic knowledge and representation capacity embedded in large-scale pretrained language models [6,16];

Interaction-feedback-driven (RL-dominated): decision-making relies on user historical behaviors and online reward signals to optimize policies [2,5].

Prior–feedback co-driven (hybrid paradigm): a bridging mechanism is established between semantic understanding and feedback-based learning [22].

This dimension directly influences the model's performance in cold-start scenarios and cross-domain transfer.

The essence of the cold-start problem lies in the scarcity or even absence of interaction data. The RL4Rec survey points out that reinforcement learning models rely on state–action–reward sequences for value estimation, making it difficult to construct stable policies when historical interactions are insufficient [2,5].

In contrast, LLMs can leverage textual descriptions, review information, and external knowledge to perform semantic reasoning, enabling a certain level of generalization even in few-shot or zero-shot scenarios [6,12,16]. For example, both TALLRec and SEALR demonstrate that large language models can still effectively model user preferences under low-data conditions [6,12].

Therefore, a structural conclusion can be drawn: the effectiveness of cold-start capability is primarily determined by the model's information dependency structure, rather than by the model size itself.

In hybrid paradigms, when LLMs serve as state representation enhancers to provide semantic representations for RL [22], the cold-start problem can be alleviated to some extent while retaining the capability for long-term optimization. Therefore, along the information source dimension, hybrid approaches typically lie in the intermediate region between semantic-driven and feedback-driven methods.

4.3 Optimization Time-Scale Dimension: Immediate Decisions Vs. Long-Term Cumulative Rewards

The second key dimension is the optimization time scale, which refers to whether the model explicitly models long-term cumulative rewards. Reinforcement learning methods are typically based on the Markov Decision Process (MDP) framework, where policy optimization is achieved by maximizing long-term

discounted rewards [2,5]. These approaches have demonstrated potential in optimizing long-term objectives, such as user retention [9]. In contrast, LLM-only recommender systems usually generate single-step decisions based on the current context, and their training objectives are primarily language modeling or instruction alignment, rather than the maximization of long-term value [19,20].

Although LLMs possess strong reasoning capabilities, their generation mechanism does not inherently establish a long-term feedback loop through interaction with the real environment [20]. Studies such as Rec-R1 address this limitation by introducing reinforcement learning signals, using downstream recommendation metrics as reward functions to optimize the LLM, thereby compensating for this deficiency [19].

Therefore, the key to long-term value optimization does not lie in whether an LLM is introduced, but in whether an optimizable sequential decision-making loop is established.

From this perspective, RL-only models and tightly coupled LLM–RL models possess theoretical advantages in long-term optimization. In contrast, LLM-only models, even if they perform well on short-term relevance metrics, still struggle to guarantee improvements in long-term user lifetime value.

Therefore, from a structural mechanism standpoint, the capability for long-term value optimization depends on whether a learnable sequential decision-making loop is established, rather than on whether a large-scale semantic model is introduced.

4.4 System Coupling Strength Dimension: Weak Coupling, Modular Coupling, and Tight Coupling

The third dimension concerns system coupling strength, namely whether the LLM is incorporated into the gradient optimization loop of RL. Existing studies on LLM-enhanced RL have systematically categorized this integration at the role level [8], which can be further abstracted as follows. Weak coupling: the LLM functions as a representation enhancer or information processor, with its parameters kept frozen [8,22]. Modular coupling: the LLM is used to generate reward signals or simulate environment dynamics [8,26]. Tight coupling: the LLM functions as a policy generator whose parameters are directly updated through RL [19,27].

The coupling strength directly influences both the expressive capability of the system and the complexity of training. As the degree of coupling increases, the model's ability in policy generation and task adaptation improves significantly. However, this also raises the risks of training instability and amplified reward noise [8].Moreover, in tightly coupled training settings such as RLHF, reinforcement learning optimization may degrade the model's original capabilities, leading to the so-called catastrophic forgetting phenomenon [17].

Therefore, at the structural level, the coupling strength of the system determines the trade-off boundary between expressive capability and training stability. This structural trade-off explains why, in industrial practice, representation-enhanced approaches or retrieval–generation cascade architectures are more commonly adopted, while large-scale tightly coupled end-to-end reinforcement learning is used less frequently.

4.5 Scenario-Based Comparative Analysis under the Three-Dimensional Framework

Based on the above three-dimensional unified analytical framework, the adaptability of different paradigms in typical application scenarios can be compared in a structured manner. Cold-start scenarios: Cold-start capability is primarily determined by the information source dimension. Semantic-driven LLM-only models and representation-enhanced hybrid models tend to perform better in this scenario [6,12], whereas RL-only models exhibit a structural disadvantage due to their reliance on interaction data [2]. Although tightly coupled hybrid models theoretically combine semantic understanding with long-term optimization capability, their training cost is relatively high.

Long-term user value optimization: The capability for long-term optimization is mainly determined by the optimization time-scale dimension. Reinforcement learning methods based on MDPs can explicitly optimize long-term cumulative rewards [2,5], and have been validated in tasks such as user retention optimization [9]. In contrast, LLM-only models struggle to guarantee the maximization of long-term returns [20]. Tightly coupled paradigms that introduce an RL loop to fine-tune LLMs provide a potential pathway for achieving "semantic understanding + long-term policy optimization" [19].

Conversational recommendation scenarios: Conversational recommendation inherently combines natural language understanding with multi-turn decision optimization. LLMs demonstrate significant advantages in semantic understanding and generation [18], while RL can model policy adaptation across multi-turn interactions [2,5]. Therefore, conversational recommendation represents one of the application scenarios with the strongest theoretical justification for deep integration between LLMs and RL. Existing studies have begun to explore incorporating RL signals into generative models to improve the quality of conversational recommendations [19].

In summary, the three-dimensional framework is not merely a classification tool, but also constitutes a structural mechanism for explaining the performance differences across various application scenarios.

4.6 Structural Trade-offs Between Intelligence and Efficiency

Based on the above analysis, three core tension relationships can be further abstracted. The tension between representational capacity and computational cost; The tension between long-term optimization capability and training stability [8]; The tension between pretraining generalization capability and task-specific specialization [17].

In tightly coupled paradigms, RL fine-tuning may weaken the model's pretrained generalization capability [17]. In contrast, weakly coupled paradigms tend to offer higher system stability but are limited in their ability to perform long-term optimization. Therefore, the core challenge in designing hybrid paradigms does not lie in whether to integrate the two approaches, but rather in how to establish a structural balance between semantic intelligence and decision intelligence.

4.7 Methodological Limitations and Evaluation Bias

Despite the significant progress made in integration research, common methodological issues still remain. Inconsistent evaluation standards. Different studies adopt evaluation metrics such as NDCG, click-through rate, long-term rewards, or human preference scores [19,26], resulting in the lack of a unified protocol for evaluating long-term value. Bias in offline evaluation. Most RL4Rec studies rely on static logged data for validation, which may lead to issues such as distribution shift and bias in value estimation [25]. Inadequate cold-start experimental settings. Some studies simulate cold-start scenarios merely by reducing the amount of training data, rather than evaluating under true zero-shot conditions or with genuinely new items [6]. Subjectivity in reward design. Reward signals generated by LLMs are sensitive to prompt design, and there is often a lack of rigorous convergence analysis for such reward formulations [8,26]. Insufficient reporting of computational costs. Although some studies have begun to address inference efficiency issues [12,20], systematic reporting of industrial-scale latency and throughput remains limited overall.

Therefore, it is necessary to establish reproducible and comparable evaluation protocols within a unified analytical framework to facilitate fair comparisons among different integration paradigms. From a broader perspective, the integration of LLMs and RL is not merely a simple combination of model scales, but rather a reconfiguration within a three-dimensional structural space, where the capability boundaries of each approach are determined by its structural positioning.

5. Conclusion

This paper investigates the integration of large language models (LLMs) and reinforcement learning (RL) in recommendation systems, and systematically reviews the development and core paradigms of related research. First, it outlines the theoretical background, explaining the shift of recommendation systems from static prediction to sequential decision-making, and analyzes the strengths of RL in long-term value optimization and the advances of LLMs in semantic understanding and generation. It then categorizes existing integration approaches from two perspectives—LLM-enhanced RL and RL-shaped LLM—summarizing key fusion patterns including representation enhancement, reward modeling, policy generation, and environment simulation, and discussing the structural characteristics of different training paradigms.

Building on this, the Discussion section proposes a three-dimensional unified analytical framework that structurally synthesizes different integration paradigms from three abstract dimensions: information source, optimization time scale, and system coupling strength. This framework reveals that the performance

differences of various models in scenarios such as cold-start recommendation, long-term user value optimization, real-time recommendation, and conversational recommendation essentially stem from their structural positioning along these three dimensions. Models driven by semantic priors possess natural advantages in cold-start scenarios; models based on sequential decision-making feedback loops demonstrate stronger theoretical justification for optimizing long-term value; while the degree of system coupling determines the trade-off between intelligence and efficiency in recommendation systems.

Through this unified analytical framework, this paper further argues that the integration of LLMs and reinforcement learning is not a simple aggregation of techniques, but rather a process of seeking structural balance between "semantic intelligence" and "decision intelligence." As the degree of coupling increases, models achieve improved capabilities in policy representation and task adaptability. However, this also introduces challenges such as increased computational costs, greater training instability, and difficulties in knowledge retention. Therefore, no single integration paradigm is inherently superior; instead, different approaches exhibit distinct advantages across different application scenarios.

In addition, this paper also reflects on several methodological limitations in current research, including inconsistent evaluation standards, biases in offline experiments, loosely defined cold-start settings, and insufficient reporting of efficiency metrics. These factors, to some extent, affect the fair comparison among different integration methods and also limit the unification and empirical validation of theoretical frameworks.

Overall, the integration of LLMs and reinforcement learning provides an important pathway for recommender systems to evolve from prediction models toward intelligent decision-making agents. Future research can further explore several directions. First, establishing unified protocols for long-term value evaluation and multi-dimensional performance benchmarks. Second, developing more stable and efficient tightly coupled training mechanisms to mitigate issues such as reward noise and catastrophic forgetting. Third, considering real-world deployment requirements by designing scalable architectures that balance expressive capability and computational cost. Fourth, deepening the collaborative mechanisms between semantic understanding and policy optimization in complex scenarios such as conversational recommendation and multimodal recommendation.

References

[1] H. Ko, S. Lee, Y. Park, and A. Choi, "A survey of recommendation systems: Recommendation models, techniques, and application fields," Electronics,2022, doi: 10.3390/electronics11010141.

[2] Lin, Y., Liu, Y., Lin, F., Zou, L., Wu, P., Zeng, W., Chen, H., Miao, C.: A survey on reinforcement learning for recommender systems. IEEE Trans. Neural Netw. Learn. Syst. 34(5), 13164-13184(2023)

[3] Rezaei, M., Tabrizi, N.: Recommender system using reinforcement learning: a survey. In: Proc. IEEE Int. Conf. Electrical Engineering, Big Data and Algorithms (EEBDA), pp. 148-159 (2022)

[4] X. Xin, T. Pimentel, A. Karatzoglou, P. Ren, K. Christakopoulou, and Z. Ren, "Rethinking reinforcement learning for recommendation: A prompt perspective," in Proc. 45th Int. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '22), Madrid, Spain, 2022, pp. 1347-1357.

[5] M. M. Afsar, T. Crump, and B. Far, "Reinforcement learning based recommender systems: A survey," ACM Comput. Surv., vol. 55, no. 7, pp. 1-38, Dec. 2022, Art. no. 145.

[6] K. Bao, J. Zhang, Y. Zhang, W. Wang, F. Feng, and X. He, "TALLRec: An effective and efficient tuning framework to align large language model with recommendation," in Proc. 17th ACM Conf. Recommender Systems (RecSys '23), Singapore, 2023, pp. 1007-1014.

[7] B. Sguerra, E. V. Epure, H. Lee, and M. Moussallam, "Biases in LLM-generated musical taste profiles for recommendation," in Proc. 19th ACM Conf. Recommender Systems (RecSys '25), Prague, Czech Republic, 2025, pp. 527-532.

[8] Y. Cao et al., "Survey on Large Language Model-Enhanced Reinforcement Learning: Concept, Taxonomy, and Methods," in IEEE Transactions on Neural Networks and Learning Systems, vol. 36, no. 6, pp. 9737-9757, June 2025, doi: 10.1109/TNNLS.2024.3497992.

[9] Z. Xue, Q. Cai, B. Yang, L. Hu, P. Jiang, K. Gai, and B. An, “AURO: Reinforcement learning for adaptive user retention optimization in recommender systems,” in Proc. ACM Web Conf. (WWW '25), Sydney, NSW, Australia, 2025, pp. 391-401.

[10] D. C. Rajapakse and D. Jannach, “Reassessing the effectiveness of reinforcement learning based recommender systems for sequential recommendation,” in Proc. 48th Int. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '25), Padua, Italy, 2025, pp. 3306-3312.

[11] Z. Ren, N. Huang, Y. Wang, P. Ren, J. Ma, J. Lei, X. Shi, H. Luo, J. Jose, and X. Xin, “Contrastive state augmentations for reinforcement learning-based recommender systems,” in Proc. 46th Int. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '23), Taipei, Taiwan, 2023, pp. 922-931.

[12] N. Lee and J. Kim, “SEALR: Sequential emotion-aware LLM-based personalized recommendation system,” in Proc. 48th Int. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '25), Padua, Italy, 2025, pp. 2906-2910.

[13] S. Wang, X. Chen, and L. Yao, “Policy-guided causal state representation for offline reinforcement learning recommendation,” in Proc. ACM Web Conf. (WWW '25), Sydney, NSW, Australia, 2025, pp. 402-412.

[14] W. Shu, Y. Zeng, Y. Tang, T. Sha, N. Luo, Y. Cheng, X. Liu, F. Zhou, and P. Jiang, “Reward balancing revisited: Enhancing offline reinforcement learning for recommender systems,” in Companion Proc. ACM Web Conf. (WWW Companion '25), Sydney, NSW, Australia, 2025, pp. 1308-1311.

[15] Y. Zhang, R. Qiu, X. Xu, J. Liu, and S. Wang, “DARLR: Dual-agent offline reinforcement learning for recommender systems with dynamic reward,” in Proc. 48th Int. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '25), Padua, Italy, 2025, pp. 2192-2202.

[16] Y. Deldjoo, Z. He, J. McAuley, A. Korikov, S. Sanner, A. Ramisa, R. Vidal, M. Sathiamoorthy, A. Kasirzadeh, and S. Milano. A review of modern recommender systems using generative models (gen-recsys). In Proc. 30th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD '24), Barcelona, Spain, 2024, pp. 6448-6458.

[17] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In Advances in Neural Information Processing Systems (NeurIPS), 2022, pp. 1-15.

[18] S. Yun and Y.-k. Lim. User experience with llm-powered conversational recommendation systems: A case of music recommendation. In Extended Abstracts of the CHI Conf. Human Factors in Computing Systems (CHI EA '25), Yokohama, Japan, 2025.

[19] J. Lin, T. Wang, and K. Qian. Rec-r1: Bridging generative large language models and user-centric recommendation systems via reinforcement learning. arXiv preprint arXiv:2503.24289v4, 2025.

[20] C. Meng, H. Ling, J. Wang, Y. Liu, S. Zhang, D. Hong, M. Gao, O. Dalal, E. Chi, L. Hong, H. Lu, and N. Han. Balancing fine-tuning and rag: A hybrid strategy for dynamic llm recommendation updates. In Proc. 19th ACM Conf. Recommender Systems (RecSys '25), Prague, Czech Republic, 2025, pp. 919-922.

[21] N. Ishii and K. Higuchi. A framework for personalized recommendation based on llms with constrained combinatorial optimization. In Extended Abstracts of the CHI Conf. Human Factors in Computing Systems (CHI EA '25), Yokohama, Japan, 2025.

[22] J. Wang, A. Karatzoglou, I. Arapakis, and J. M. Jose. Reinforcement learning-based recommender systems with large language models for state reward and action modeling. In Proc. 47th Int. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '24), Washington, DC, USA, 2024, pp. 375-385.

[23] A. Bodaghi, B. C. M. Fung, and K. A. Schmitt. Augmentoxic: Leveraging reinforcement learning to optimize llm instruction fine-tuning for data augmentation to enhance toxicity detection. ACM Transactions on the Web, 19(4), Article 38, 2025.

[24] F. Liu, H. Guo, X. Li, R. Tang, Y. Ye, and X. He. End-to-End Deep Reinforcement Learning based Recommendation with Supervised Embedding. In Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM ’20), Houston, TX, USA, 2020, pp. 384–392.

[25] Romain Deffayet, Thibaut Thonet, Jean-Michel Renders, and Maarten de Rijke. 2023. Offline Evaluation for Reinforcement Learning-Based Recommendation: A Critical Issue and Some Alternatives. SIGIR Forum 56, 2 (2023). doi:10.1145/3582900.3582905

[26] A. Sharma, H. Li, X. Li, and J. Jiao, “Optimizing novelty of top-k recommendations using large language models and reinforcement learning,” in Proc. 30th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD '24), Barcelona, Spain, 2024, pp. 5669–5679.

[27] X. Zhao, Y. Chen, Z. Wang, H. Zhou, and J. Liu, “Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning,” arXiv preprint arXiv:2503.09516v5, 2025.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Copyrights