Reinforcement Learning for Recommender Systems: A Critical Review of Multi-Agent, Multi-Scenario, and Multi-Task Frameworks[†]

Aditya Shrey
Department of Computer Science
Vanderbilt University
Nashville, TN 37235
aditya.shrey@vanderbilt.edu

Abstract

Recommender systems are platforms designed to promote personalized services or products to users, such as targeted advertisements on social media. Reinforcement learning (RL) approaches have emerged as powerful tools to improve recommender systems, focusing on optimizing long-term engagement and balancing dynamic user preferences. In this paper, we examine the integration of RL across different facets of recommender systems, including Multi-Task Fusion [4], multi-scenario alignment [7], and hybrid approaches leveraging bi-clustering and MDPs for scalability [2]. Additionally, we explore state-of-the-art frameworks, such as UNEX-RL [6], which optimize multi-stage pipelines to enhance long-term rewards. By highlighting the advancements, challenges, and emerging directions of the current landscape, this paper provides a comprehensive perspective on designing robust and scalable RL-based recommender systems.

1 Introduction

Interacting with recommender systems [3] has become a common part of modern life, influencing everything from the products we shop to the media we consume. As these systems aim to deliver personalized experiences, there is a growing need for sophisticated techniques that address the complexity and scale of real-world applications. Reinforcement Learning (RL) [1] has emerged as a powerful paradigm for tackling challenges in recommender systems, allowing for dynamic decision-making that can optimize long-term user satisfaction and engagement. However, integrating RL into these systems is not straightforward, as it involves navigating diverse factors such as scalability, real-time constraints, and the interplay of multiple recommendation objectives.

The focus of this paper is to critically review and analyze RL approaches applied to recommender systems, particularly those that address challenges in handling data and decision-making processes. These methods vary widely in their treatment of the recommendation problem, as they consider factors such as multi-agent dynamics, sequential decision-making, and multi-stage architectures. This is an important and timely area of investigation, given the growing reliance on recommender systems across industries and the increasing complexity of the underlying data and user behavior. By understanding how these techniques address challenges like scalability, efficiency, and adaptability, we can better assess their potential for real-world deployment.

Our approach involves reviewing and critically comparing four recent papers that explore innovative RL methods in recommender systems. Each paper focuses on a distinct aspect of the problem: from

[†]submitted for CS 5891 Reinforcement Learning Fall 2024 Final Project

Preprint. Under review.

multi-task fusion [4] and multi-agent cooperation [7] to state representation [2] and multi-stage pipelines [6]. We will first provide an overview of these papers, summarizing their methodologies, key contributions, and applications. Following this, we will analyze each approach across common dimensions, such as scalability, efficiency, domain-specific applicability, and future promise, synthesizing insights to understand the state of the art and identify open challenges.

The rest of this paper is organized as follows: Section 2 presents background, providing context for the RL approaches discussed. Section 3 offers a detailed explanation of our approaches employed in the paper. Section 4 contains a detailed critical analysis of the selected papers, comparing their contributions and highlighting strengths and limitations. Section 4 further synthesizes these findings, considering broader implications and future directions for RL in recommender systems. Finally, Section 5 concludes the paper, noting key takeaways and potential opportunities for further research.

2 Background

With the rise of deep learning, increases in computing power, and the proliferation of large-scale data, personalized recommender systems have evolved into critical components of modern digital interactions [3]. These systems enable tailored recommendations to individual users, enhancing user satisfaction and driving profitability for businesses. Personalized recommender systems, as highlighted in recent surveys, are often categorized into three main types: collaborative filtering-based, context-aware-based, and hybrid methods [3]. Collaborative filtering relies on identifying patterns in user-item interactions, while context-aware systems integrate contextual information such as time or location. Hybrid systems combine these approaches to achieve more robust performance. Additionally, the emergence of knowledge-based recommender systems has expanded the field by incorporating external knowledge graphs, improving capabilities in addressing challenges such as the cold-start problem, which occurs when there is not initially enough information about a new item or user.

Despite their successes, recommender systems face significant challenges. One major issue is the sparsity of interaction data, where users interact with only a small fraction of available items, creating gaps in representation and limiting the predictive power of models [3]. Similarly, biases in data—such as popularity bias or exposure bias—can skew recommendations, leading to unfair outcomes or suboptimal user experiences. Robustness and fairness also remain critical concerns, especially in applications requiring high reliability and equitable access [3]. These challenges demand innovative approaches that balance algorithmic efficiency with real-world complexities, driving the exploration of RL in this domain.

To consider recommender systems through the lens of RL, they can be modeled as sequential decision-making problems and formulated as Markov Decision Processes (MDPs) [1]. In this framework, the environment includes users and items, while the recommendation algorithm serves as the RL agent. The agent's goal is to maximize cumulative rewards, defined as:

$$\max_{\pi} \mathbb{E}\big[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)\big], \tag{1}$$

where $s_t \in \mathcal{S}$ represents the user behavior at time t , $a_t \in \mathcal{A}$ denotes the action of recommending an item, $r(s_t, a_t)$ is the reward function based on user feedback, γ is the discount factor accounting for the importance of future rewards, and T is the time horizon [1]. This formulation allows RL to capture the dynamic and sequential nature of user interactions, optimizing for long-term user satisfaction rather than immediate feedback.

In general, RL in recommender systems can be divided into four critical components: state representation, policy optimization, reward formulation, and environment building [1]. State representation defines the features or embeddings that capture user preferences, item attributes, and contextual information, forming the input space for the RL agent. Policy optimization focuses on determining the best recommendation strategy, leveraging algorithms such as Q-learning, policy gradients, and actor-critic methods. Reward formulation involves designing metrics that encapsulate user satisfaction, engagement, or business objectives, guiding the agent's learning process. Environment building pertains to creating suitable simulators or datasets that enable training and evaluation of RL models.

RL-based recommender systems still face several challenges. From an algorithmic perspective, RL methods often suffer from instability and sample inefficiency, particularly in model-free approaches. The "deadly triad" of RL—combining function approximation, bootstrapping, and off-policy training—can lead to divergence and hinder learning in high-dimensional spaces [1]. Data-related issues such as sparsity, bias, and the lack of sufficient interaction feedback further complicate the application of RL. Additionally, real-world constraints such as latency, scalability, and the necessity for fairness and interpretability remain significant barriers. Advancements in deep reinforcement learning (DRL) have made RL more practical for large-scale systems by enabling efficient handling of complex state and action spaces.

3 Approach

We take an investigative lens to analyze recent, cutting-edge approaches that apply RL in recommender systems. The focus is on four innovative methods, each addressing distinct challenges in the field: multi-task fusion using off-policy RL [4], multi-agent cooperation for multi-scenario optimization [7], state representation through bi-clustering and Markov Decision Processes (MDP) [2], and long-term reward optimization in multi-stage pipelines using multi-agent RL [6]. These approaches represent significant advancements that tackle complex issues in RL-based recommender systems. As these methods are relatively new, our analysis aims to critically evaluate their contributions while highlighting their implications for future research and practical applications.

To evaluate these approaches comprehensively, we adopt a structured process that first examines the unique methodologies of each paper before conducting a detailed analysis across common dimensions. We begin by first providing a concise overview of each paper, detailing its methodology, key innovations, and primary contributions. This is followed by a critical evaluation across shared dimensions, synthesizing insights about their algorithmic complexity, scalability, efficiency, and potential for broader applications. These dimensions are critical for understanding the real-world relevance of these methods. For example, scalability determines whether an approach can handle the demands of industrial-scale systems, while efficiency addresses the practical constraints of real-time recommendation settings. Furthermore, the societal implications of these methods—such as their ability to enhance user engagement, increase revenue, or improve resource allocation—make this analysis particularly meaningful in the broader context of technology and human interaction. The discussion considers future promise, highlighting the strengths and limitations of each method in addressing unresolved challenges in RL-based recommender systems. This structured approach enables us to understand not only the current state of the art but also the opportunities for advancing the field further.

4 Critical Comparison

4.1 Overview

A brief overview of the four papers, all of which represent recent innovations in RL for recommender systems, is outlined before a dive into the analysis of the algorithmic complexity, scalability, efficiency, and applicability to other areas is performed. In this section, we aim to provide an introduction to each paper and the key ideas that are presented with respect to RL in recommender systems. Particularly, we emphasize what is new compared to prior innovations in this field. We will explore the general approach, experimental results, and any challenges that may still persist. Examining the papers at this level situates a deep analysis into the four dimensions presented afterwards.

IntegratedRL-MTF: Off-Policy Reinforcement Learning for Multi-Task Fusion

In large-scale recommender systems, multiple user behaviors—such as clicks, watching time, and likes—are often predicted separately and then combined into a single score to maximize user satisfaction. This process, referred to as Multi-Task Fusion (MTF), is a critical stage that determines the ultimate recommendation results [4]. Optimizing MTF poses significant challenges, as it requires effectively integrating multiple user behaviors into a cohesive policy. Liu et al. propose IntegratedRL-MTF, an off-policy RL algorithm specifically designed to address this problem. Their approach centers on answering the fundamental question: how can a recommendation system maximize cumulative rewards over a session while simultaneously optimizing multiple user behaviors?

In this framework, the recommendation system serves as the agent, while the user is modeled as the environment, aligning with standard RL formulations. Liu et al. frame the problem as an MDPs, where the state space captures user profiles and behavioral history (e.g., interests, click patterns), the action space consists of fusion weights for different behaviors, and the reward function reflects user feedback such as valid consumption and engagement time. A novel aspect of IntegratedRL-MTF is its progressive training mode, which alternates between offline model training and online exploration. This iterative process enables the policy to refine itself continuously, addressing one of the key limitations of traditional off-policy RL methods, which often impose strict constraints to avoid out-of-distribution (OOD) errors, thereby limiting their performance.

The key innovation of IntegratedRL-MTF lies in its integration of the RL model with an efficient exploration policy that prioritizes high-value state-action pairs [4]. By eliminating low-value exploration spaces, the system improves model efficiency and enhances user experience, as fewer suboptimal actions are taken during training. Furthermore, IntegratedRL-MTF introduces a penalty mechanism to mitigate extrapolation errors, improving its robustness and reliability in high-dimensional environments. These design choices make the algorithm particularly effective for large-scale applications with complex behavioral patterns.

Liu et al. validated their approach through comprehensive offline and online experiments on Tencent News, a platform serving hundreds of millions of users. Offline evaluation employed a weighted GAUC metric to measure ranking quality based on user engagement, with IntegratedRL-MTF achieving a score of 0.7953, outperforming advanced RL baselines such as DDPG [5]. Online experiments further confirmed the model's superiority, demonstrating a 4.64% increase in valid user consumption and a 1.74% improvement in user duration time. These results underscore the algorithm's ability to balance exploration and exploitation effectively, focusing on long-term user satisfaction. Challenges remain, particularly in scaling the approach to other domains. The reliance on a well-calibrated exploration policy necessitates careful tuning to avoid bias or overfitting, and domain-specific adaptations may be required for broader real-world deployment.

MA-RDPG: Multi-Agent Reinforcement Learning for Multi-Scenario Combination

Recommender systems on large platforms often involve multiple interconnected scenarios, such as search, recommendations, and advertisements, which are typically optimized in isolation. This independent optimization can lead to inefficiencies and conflicts, as users frequently navigate across scenarios. Zhao et al. address this issue by proposing the Multi-Agent Recurrent Deterministic Policy Gradient (MA-RDPG) algorithm, a novel multi-agent reinforcement learning (MARL) framework that aligns these scenarios under a shared objective [7]. By treating the problem as a partially observable multi-agent decision problem (POMDP), where each scenario acts as an agent, the framework facilitates cooperation and optimizes platform-wide performance.

The MA-RDPG algorithm combines two key elements to handle the POMDP setting: Deep Recurrent Q-Networks (DRQN) and an actor-critic approach inspired by Deterministic Policy Gradient (DPG). DRQN leverages recurrent neural networks to encode historical observations, enabling agents to infer unobserved states and maintain temporal context. The actor-critic setup uses a centralized critic to evaluate overall platform performance and scenario-specific actors to generate actions, while an LSTM-based communication module facilitates information sharing between agents. This design ensures that agents cooperate effectively, balancing individual scenario goals with global optimization.

Experiments on an e-commerce dataset demonstrated that MA-RDPG significantly improves gross merchandise volume (GMV) compared to baseline models, particularly in scenarios like in-store search and main search. The framework achieved stable convergence and enhanced overall platform performance. Challenges persist, including scaling the approach to a larger number of agents and scenarios, as the experiments only involved two interconnected components.

Bi-clustering and MDP State Representations for RL in Recommender Systems

Iftikhar et al. propose a novel approach to modeling recommender systems using bi-clustering techniques within an MDP framework. Their method begins by addressing the static limitations of conventional collaborative filtering, which struggles with evolving user preferences and dynamic environments. To overcome these limitations, the user-item voting matrix is transformed into a binary matrix, which is subsequently processed through bi-clustering algorithms (BiMax and Bibit) [2]. These algorithms identify localized patterns of user-item interactions, grouping similar users and

items into subsets or "bi-clusters." The bi-clusters are then mapped onto a square grid, representing states within the MDP. This mapping, guided by the scaled mean square residual (SMSR) fitness function, ensures that high-quality clusters are positioned strategically for efficient state transitions. Reinforcement learning techniques, such as Q-learning and SARSA, are then applied to derive the optimal policy for generating recommendations [2].

The methodology involves multiple stages of refinement to ensure scalability and computational efficiency. The process begins with the initial bi-clustering, followed by merging or decomposition to match the grid size, ensuring a manageable state space for the MDP. Each bi-cluster is placed using a space-filling curve, such as Cantor-diagonal traversal, to maintain proximity for clusters with similar SMSR values. The state transitions are governed by an ϵ-greedy policy, balancing exploration and exploitation, while the reward function evaluates the overlap between current and subsequent states using Jaccard similarity. This structured approach enables the RL agent to effectively navigate the grid, maximizing cumulative rewards and user satisfaction while maintaining computational feasibility. The start state is determined using the Improved Triangle Similarity measure, which accounts for global and local voting patterns to ensure robust initial recommendations [2].

Experimental evaluations were conducted on the ML-100K and FilmTrust datasets to assess the performance of the proposed method. Key metrics such as precision, recall, F-measure, and latency were used for comparison against baseline methods. The results demonstrated that the approach outperformed traditional CF and clustering-based methods in terms of precision and recommendation accuracy. For example, in the ML-100K dataset, the algorithm achieved significant improvements in learning time, with average policy learning times of 0.05 seconds per user. The number of steps required to reach the goal state decreased steadily across episodes, indicating that the agent effectively learned optimal policies over time. Similarly, higher rewards were observed as the agent improved its navigation within the state space, showcasing its adaptability and efficiency in generating tailored recommendations [2].

While bi-clustering effectively reduces the state space, the quality and granularity of clusters are heavily dependent on the chosen parameters and algorithms, such as the minimum number of rows and columns (mnr and mnc). Poor parameter tuning can result in excessive merging or decomposition, potentially degrading recommendation quality. Additionally, the scalability of the method to larger datasets or more diverse item categories remains a concern, as the computational complexity of bi-clustering and grid traversal grows with the size of the dataset.

UNEX-RL for Multi-Stage RL in Recommender Systems

The UNEX-RL framework proposed by Zhang et al. addresses the challenge of applying RL to industrial-scale multi-stage recommender systems, which are common in large platforms that serve millions of users daily [6]. Traditional single-agent RL frameworks struggle in this context because multi-stage systems involve distinct stages—such as matching, pre-ranking, ranking, and re-ranking—each with unique observation spaces and objectives. The UNEX-RL framework integrates MARL with a unidirectional execution design to optimize long-term rewards. This approach capitalizes on the cascading structure of multi-stage systems, where each stage progressively reduces the candidate pool, making downstream decisions increasingly precise. By treating each stage as an independent agent and facilitating cooperation among them, UNEX-RL effectively aligns multi-stage processes to achieve cohesive optimization of user engagement metrics.

The core of UNEX-RL is its Cascading Information Chain (CIC), a novel training mechanism designed to address two major challenges: observation dependency (OD) and cascading effect (CE). OD arises when changes in upstream stages cause unpredictable downstream observations, complicating critic learning in MARL. CE reflects the ripple effects of upstream actions on downstream agent decisions, which can destabilize actor learning. CIC mitigates these challenges by sequentially propagating information across stages, ensuring consistent training and improving cooperation among agents. In addition, variance reduction techniques like stopping gradient and category quantile rescaling further enhance training stability and reduce noise in reward signals. These innovations enable UNEX-RL to overcome traditional MARL limitations in multi-stage environments while maintaining scalability and robustness.

Zhang et al. validate UNEX-RL through extensive offline and online experiments, including deployment on Kuaishou, a short-video platform with over 100 million users. Offline tests on public datasets such as KuaiRand demonstrate significant gains in metrics like watch time and session length

compared to baseline methods, including DDPG and TD3. Online A/B experiments reinforce these findings, with UNEX-RL achieving a 0.953% improvement in daily watch time and a 0.558% gain over TD3, underscoring its effectiveness in optimizing long-term rewards. Despite these successes, scalability remains a concern when extending the framework to systems with more stages or agents, as the computational complexity of cascading dependencies can increase.

4.2 Algorithmic Complexity

Algorithmic complexity varies significantly across the approaches based on their design and implementation choices. IntegratedRL-MTF employs a progressive training mode combined with an efficient exploration policy, which, while improving state-space exploration, introduces offline training overhead and requires careful tuning of the exploration policy to ensure optimal performance. MA-RDPG leverages a MARL framework that enables coordination among agents in a partially observable environment using RNNs. While the flexibility afforded by POMDP assumptions simplifies some aspects, the presence of multiple agents inherently increases the complexity, as each agent must be properly tuned for its specific task. Similarly, UNEX-RL also utilizes a MARL framework but incorporates a CIC for inter-agent coordination, along with variance reduction techniques to enhance training stability. These additions, while theoretically robust, further elevate its complexity. By contrast, the bi-clustering with MDP approach is the least algorithmically complex, as it focuses on reducing the state space through pre-processing with bi-clustering algorithms, which introduces relatively low overhead. This simplicity can be advantageous for systems requiring lower computational demands.

4.3 Scalability

Scalability examines how well each approach can adapt to large-scale industrial contexts, such as social media platforms or e-commerce systems, while also considering its algorithmic complexity. IntegratedRL-MTF, designed specifically for single-stage systems, demonstrates strong scalability for large-scale systems within its domain. Its reliance on domain-specific adaptations for generalized tasks limits its versatility in broader contexts. MA-RDPG, on the other hand, employs a MARL framework, which inherently becomes less scalable as the number of scenarios and agents increases. While this design is crucial for certain applications, the added complexity of coordinating multiple agents poses challenges for scaling to even larger systems. UNEX-RL faces similar issues but at an even higher level of complexity due to its reliance on CIC mechanisms. These dependencies, while effective for ensuring consistency across stages, demand significant computational overhead to manage cascading interactions efficiently, making scalability dependent on optimization. While the bi-clustering with MDP approach is less algorithmically complex, the appraoch is also constrained in scalability. While bi-clustering reduces the state space, its computational complexity increases with the size of the dataset and the diversity of item categories, limiting its effectiveness for industrial-scale applications.

4.4 Data Efficiency

Data efficiency refers to the ability of an algorithm to derive meaningful insights and policies from limited data, a critical consideration in large-scale recommender systems where data collection can be expensive or sparse. IntegratedRL-MTF achieves significant data efficiency by prioritizing high-value state-action pairs, effectively reducing unnecessary exploration and focusing on areas that yield the most informative results. Similarly, UNEX-RL improves data efficiency by leveraging sequential information across different stages, which minimizes redundancy and noise during implementation. In contrast, MA-RDPG, while incorporating temporal context through its recurrent neural networks, still requires substantial amounts of data to ensure robust performance across multiple agents and scenarios. The bi-clustering with MDP approach depends heavily on the quality of the bi-clustering process to maintain data efficiency, as poorly constructed clusters could lead to less informative state representations. While each approach has its nuances, understanding these facets highlights their relative strengths and limitations in efficiently utilizing available data.

4.5 Potential for Broader Applications

The potential for broader applications reflects an algorithm's ability to adapt to various contexts within recommender systems rather than being confined to a specific use case. IntegratedRL-MTF, with its focus on a multi-task framework, is well-suited for large-scale platforms requiring multi-behavior optimization, making it particularly applicable in diverse contexts such as social media, where users engage in various activities simultaneously. Similarly, UNEX-RL demonstrates strong adaptability to multi-stage pipelines. Although its design and implementation are initially complex, its ability to handle cascading dependencies makes it highly versatile for large, industrial-scale systems. MA-RDPG, while effective for interconnected scenarios like search and advertising, may face challenges when applied to more complex environments with independent or loosely connected scenarios, limiting its broader applicability. The bi-clustering approach is advantageous in systems with sparse data, which are common in many recommender settings. Its effectiveness diminishes in environments with highly diverse domains, where the variability of user and item categories could complicate its clustering process.

Table 1: Comparison of four RL-based approaches in recommender systems: IntegratedRL-MTF (off-policy RL for multi-task fusion), MA-RDPG (multi-agent RL for multi-scenario coordination), bi-clustering with MDP (state-space reduction for efficient recommendations), and UNEX-RL (multi-stage MARL with cascading information chains). The approaches are evaluated across four dimensions: algorithmic complexity, scalability, data efficiency, and potential for broader applications.

Approach	**Algorithmic Complexity**	**Scalability**	**Data Efficiency**	**Broader Applications**
IntegratedRL-MTF	Requires progressive training and efficient exploration policies, introducing offline training overhead.	Designed for single-stage systems; domain-specific adaptations limit broader generalization.	Prioritizes high-value state-action pairs, reducing unnecessary exploration.	Suited for multi-behavior optimization on large-scale platforms like social media.
MA-RDPG	Uses MARL framework with recurrent networks for POMDP settings, increasing complexity.	Scaling to more scenarios and agents adds significant coordination overhead.	Temporal context helps but requires substantial training data for robust performance.	Ideal for interconnected scenarios but less effective for loosely connected or independent tasks.
Bi-clustering with MDP	Focuses on reducing state space via bi-clustering, with relatively low computational overhead.	Bi-clustering reduces state space but complexity increases with dataset size and diversity.	Data efficiency depends on the quality of bi-clustering, with poor clustering affecting state representation.	Effective for sparse data systems but less adaptable to diverse or dynamic domains.
UNEX-RL	Relies on MARL with CIC and variance reduction techniques, adding significant theoretical complexity.	Scalability depends on efficient handling of cascading dependencies and computational overhead.	Sequential information propagation reduces redundancy and noise.	Adaptable to complex multi-stage pipelines across various industrial contexts like short-video platforms or large e-commerce systems.

5 Discussion

This paper presents a comprehensive exploration of RL techniques applied to recommender systems, offering insights into their algorithmic underpinnings, scalability, efficiency, and applicability. We

began by discussing the foundational principles of recommender systems and their formulation within RL frameworks. Subsequently, we critically reviewed four novel approaches: IntegratedRL-MTF for multi-task fusion, MA-RDPG for multi-scenario cooperation, bi-clustering with MDP for scalable state representation, and UNEX-RL for optimizing multi-stage pipelines. Through an analysis of these methods across key dimensions—algorithmic complexity, scalability, data efficiency, and broader applications—we sought to evaluate their contributions and implications for large-scale, real-world recommender systems.

The findings underscore both the potential and limitations of these methods. IntegratedRL-MTF demonstrates impressive performance in integrating multi-task objectives but is constrained by the overhead associated with offline training and the need for precise policy tuning. MA-RDPG highlights the promise of multi-agent cooperation for interconnected scenarios but reveals challenges in scaling to larger or more complex environments. Bi-clustering with MDP offers a computationally efficient solution for sparse datasets but struggles with scalability in diverse or dynamic domains. UNEX-RL showcases exceptional adaptability to multi-stage architectures but requires substantial computational resources to manage cascading dependencies. Across all methods, we observe a growing trend towards leveraging hybrid frameworks, integrating techniques such as multi-agent RL, bi-clustering, and advanced exploration strategies to tackle the inherent complexities of modern recommender systems.

Looking ahead, several challenges and opportunities remain in advancing RL for recommender systems. One critical direction involves simplifying the design of multi-agent frameworks to ensure scalability and ease of deployment in large-scale systems. This includes addressing the computational overhead of training and coordination while maintaining robustness in highly dynamic environments. Another avenue for exploration lies in enhancing the interpretability and fairness of RL-based recommendations, particularly as these systems become increasingly integral to user-facing platforms. As multi-task and multi-scenario applications grow in complexity, developing methods that seamlessly integrate diverse objectives without compromising performance will be crucial. Future research should also explore the integration of pre-trained models and transfer learning to reduce reliance on extensive training data, thus improving data efficiency and adaptability.

References

[1] M. Mehdi Afsar, Trafford Crump, and Behrouz Far. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, 2022.

[2] Arta Iftikhar, Mustansar Ali Ghazanfar, Mubbashir Ayuba, Saad Ali Alahmari, Nadeem Qazi, and Julie Wall. A reinforcement learning recommender system using bi-clustering and markov decision process. *Expert Systems With Applications*, 2024.

[3] Yang Li, Kangbo Liu, Ranjan Satapathy, Suhang Wang, and Erik Cambria. Recent developments in recommender systems: A survey. *IEEE Computational Intelligence Magazine*, 2024.

[4] Peng Liu, Cong Xu, Ming Zhao, Jiawei Zhu, Bin Wang, and Yi Ren. An off-policy reinforcement learning algorithm customized for multi-task fusion in large-scale recommender systems. In *Proceedings of ACM Conference (Conf'24)*. ACM, 2024.

[5] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[6] Gengrui Zhang, Yao Wang, Xiaoshuang Chen, Hongyi Qian, Kaiqiao Zhan, and Ben Wang. Unex-rl: Reinforcing long-term rewards in multi-stage recommender systems with unidirectional execution. *Associations for the Advancements of Artificial Intelligence*, 2024.

[7] Yang Zhao, Chang Zhou, Jin Cao, Yi Zhao, Shaobo Liu, Chiyu Cheng, and Xingchen Li. Multi-scenario combination based on multi-agent reinforcement learning to optimize the advertising recommendation system. In *2024 5th International Conference on Artificial Intelligence and Electromechanical Automation (AIEA)*. IEEE, 2024.

Acknowledgements

A deep sense of gratitude is extended to Dr. Marcos Quinones-Guirero and Dr. Gautam Biswas for their dedication to teaching reinforcement learning and their invaluable support throughout the semester.