HierLLM: Hierarchical Large Language Model for Question Recommendation

Yuxuan Liu[1], Haipeng Liu[1], and Ting Long✉[1]

[1] Jilin University, Changchun, China
{liuyuxuan23, liuhp22}@mails.jlu.edu.cn
longting@jlu.edu.cn

Abstract. Question recommendation is a task that sequentially recommends questions for students to enhance their learning efficiency. Previous methods typically model the question recommendation as a sequential decision-making problem, estimating students' learning state with the learning history, and feeding the learning state with the learning target to a neural network to select the recommended question from a question set. However, previous methods are faced with two challenges: (1) learning history is unavailable in the cold start scenario, which makes the recommender generate inappropriate recommendations; (2) the size of the question set is much large, which makes it difficult for the recommender to select the best question precisely. To address the challenges, we propose a method called hierarchical large language model for question recommendation (HierLLM), which is a LLM-based hierarchical model. HierLLM tackle the cold start issue with the strong reasoning abilities of LLM, and narrows the range of selectable questions via the hierarchical structure. Comprehensive and in-depth experiments demonstrate the outstanding performance of HierLLM. Our code will be released on `https://github.com/YuxuanLiu1112/HierLLM`.

Keywords: Question Recommendation · Large Language Model · Reinforcement Learning.

1 Introduction

Question recommendation personalizes learning by tailoring questions to a student's needs. As shown in Figure 1, if the target is mastering questions v_1, v_3, and v_9, and the student has answered $\{q_1, q_2, ..., q_8\}$, the system sequentially recommends questions to aid learning. Unlike traditional methods, question recommenders adapt to each student's learning path. Previous approaches model this as a sequential decision-making problem, optimized with reinforcement learning (RL) [4,14,15], using the student's history and target to select questions and iterating based on feedback.

However, previous methods face two main challenges from temporal and spatial perspectives: (1) temporal challenge: these methods rely on the student's learning history to estimate their learning state. In cold start scenarios, where

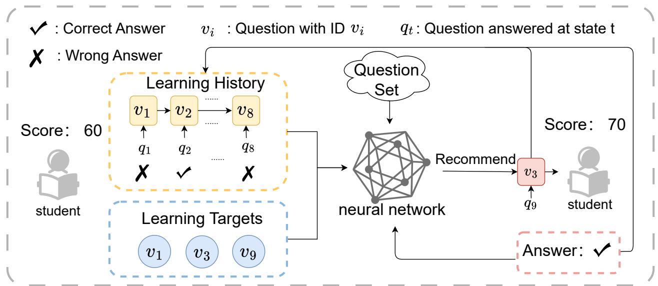

Fig. 1. Illustration of a recommender recommends questions.

no history is available, this results in inappropriate recommendations; (2) spatial challenge: previous methods are tasked with selecting a single question from a large question set, which, in real-world scenarios, may contain thousands or even hundreds of thousands of questions, making it difficult to precisely identify the most relevant question from such a vast decision space.

To address these limitations, we propose a method called Hierarchical Large Language Model for Question Recommendation (HierLLM), which utilizes a hierarchical structure built on a large language model (LLM). Leveraging the reasoning capabilities and extensive knowledge base of LLMs [19], HierLLM effectively mitigates the cold start problem. Additionally, by recognizing that the number of relevant concepts is much smaller than the total number of questions, HierLLM narrows the selection space by first identifying the relevant concept and then choosing the question associated with that concept. This approach significantly reduces the complexity of the recommendation process. Extensive experiments demonstrate that HierLLM achieves state-of-the-art performance.

2 Related Works

2.1 Personalized Question Recommendation

Many approaches have been proposed for question recommendation. Some use general recommendation algorithms [25,17] to suggest similar paths for students with shared learning targets. Others focus on personalized methods, such as CSEAL [15], which uses a navigation algorithm on the knowledge structure and an actor-critic algorithm for dynamic strategy updates, SRC [4] employs attention mechanisms to optimize based on student feedback, and GEHRL [14] uses hierarchical reinforcement learning to plan learning paths. However, these methods overlook the cold start and large question set problems.

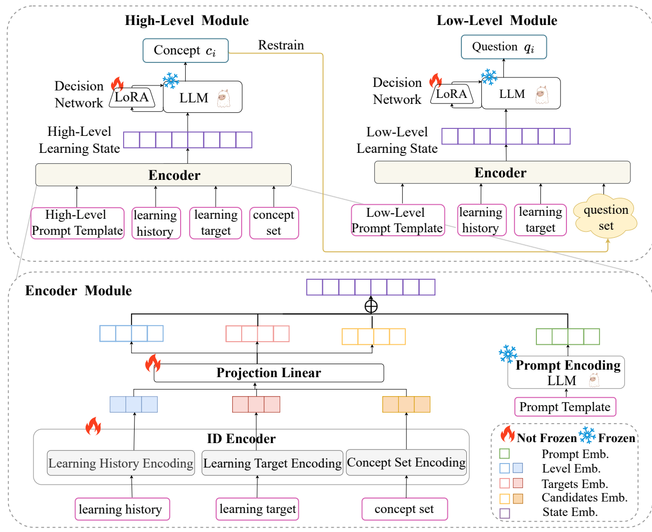

Fig. 2. The framework of HierLLM.

2.2 Application of LLM in Recommendations

Recent large language models (LLMs) have shown impressive reasoning and decision-making abilities across tasks [5,23], with applications in recommendation systems. These can be divided into two categories: 1) LLMs for feature enhancement, like KAR [22] and ONCE [16], which enhance traditional models by incorporating LLM-generated representations; and 2) LLMs for direct recommendations, such as TALLRec [1] using LoRA [11] for fine-tuning and A-LLMRec [12] integrating collaborative and textual knowledge.

3 Preliminaries

3.1 Terminology

Given a concept set $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ and a question set $\mathcal{Q} = \{v_1, v_2, \ldots, v_n\}$, where $m \ll n$ in most online learning systems, a student's learning history at step t is $\mathcal{H}_t = \{(q_1, y_1), (q_2, y_2), \ldots, (q_{t-1}, y_{t-1})\}$, with q_t as the question at step t and $y_t \in \{0, 1\}$ for correctness. The student's learning targets are represented by $\mathcal{G} = \{v_i\}^{|\mathcal{G}|}$, a set of question IDs they aim to master. The learning effect is defined as in [4,15,14]:

$$\Delta_u = \frac{E_a - E_b}{E_{max} - E_b}, \tag{1}$$

where E_a and E_b are the number of questions in learning target correctly answered by the student after and before learning the recommended questions, respectively, and E_{max} is the number of questions in learning target.

3.2 Problem formulation.

We propose a question recommendation model based on a Hierarchical Markov Decision Process (HMDP) to maximize students' long-term learning effect (Δ_u). The HMDP consists of high-level and low-level states ($\mathcal{S}_h, \mathcal{S}_l$), actions ($\mathcal{A}_h, \mathcal{A}_l$), transition probabilities ($\mathcal{P}_h, \mathcal{P}_l$), a reward function ($\mathcal{R}$) representing learning effect, and a discount factor (γ). The goal is to maximize the long-term learning effect by optimizing both policies:

$$\max_{\pi_h, \pi_l} J(\pi_h, \pi_l) = \max_{\pi_h, \pi_l} \Delta_u = \max_{\pi_h, \pi_l} \frac{1}{T} \sum_{t=0}^{T} r(q_t | \mathbf{s}_t^l, \mathcal{Q}_t(c_t | \mathbf{s}_t^h, \mathcal{C}))$$

4 Methodology

4.1 Overview

To address temporal and spatial challenges in personalized question recommendation, we propose HierLLM as shown in Figure 2. In the following, we will first present the high-level and low-level modules in the HierLLM framework, then discuss the details of encoder.

4.2 The High-level and Low-level Modules.

Given the learning history $\mathcal{H}_t$, learning target $\mathcal{G}$, and concept set $\mathcal{C}$, the high-level module recommends a concept, while the low-level module recommends a specific question within the selected concept.

High-level Module. This module selects a relevant concept (c_t) for recommendation. It creates a high-level prompt ($\mathcal{P}_t^h$) integrating $\mathcal{H}_t$, $\mathcal{G}$, and $\mathcal{C}$, then encodes them to obtain the high-level learning state:

$$\mathbf{s}_t^h = \mathrm{Enc}(\mathcal{H}_t, \mathcal{G}, \mathcal{C}, \mathcal{P}_t^h), \tag{2}$$

where $\mathbf{s}_t^h \in \mathbb{R}^{d_m}$. The state is fed into a decision network (LLM) to select c_t:

$$c_t \sim \pi_c(\cdot | \mathbf{s}_t^h). \tag{3}$$

Low-level Module. Using c_t, the module filters the question set to form a relevant candidate set $\mathcal{Q}_t$. It then generates a low-level prompt $\mathcal{P}_t^l$ by integrating $\mathcal{H}_t$, $\mathcal{G}$, and $\mathcal{Q}_t$, and encodes it to obtain the low-level learning state:

$$\mathbf{s}_t^l = \mathrm{Enc}(\mathcal{H}_t, \mathcal{G}, \mathcal{Q}_t, \mathcal{P}_t^l), \tag{4}$$

where $\mathbf{s}_t^l \in \mathbb{R}^{d_m}$. The state is fed into a decision network (Llama2-7B) to recommend a specific question q_t:

$$q_t \sim \pi_l(\cdot | \mathbf{s}_t^l). \tag{5}$$

Both modules share the same encoder and LLM structure but have different parameters and roles.

Table 1. Prompt Template

Prompt template for the High-Level
Instruction: Based on students' learning history: [$\text{ID}_{\text{learning history}}$], learning targets: [$\text{ID}_{\text{targets}}$] and concepts: [$\text{ID}_{\text{concepts}}$], Predict the next concept. ### Response:
Prompt template for the Low-Level
Instruction: Based on students' learning history: [$\text{ID}_{\text{learning history}}$], learning targets: [$\text{ID}_{\text{targets}}$], and questions: [$\text{ID}_{\text{questions}}$], Predict the next question. ### Response:

4.3 The Encoder

Question candidate / Concept set encoding. We encode both concepts and questions in the same way. Let $\mathcal{X} = \{x_1, x_2, ..., x_k\}$ represent the set of concepts or question candidates. For each $x_i \in \mathcal{X}$, we first apply one-hot encoding, followed by an attention mechanism to obtain its attentive representation:

$$\begin{aligned} \mathbf{e_i^o} &= \text{one-hot}(x_i), \\ \mathbf{e_i^a} &= \text{Attn}(\mathbf{e}_i^o, \mathbf{E}_i^o), \end{aligned} \tag{6}$$

where $\mathbf{E}_i^o$ is the stacked one-hot encoding of $\mathcal{X}$.

The set representation $\mathbf{e}_t$ is obtained by average pooling:

$$\mathbf{e}_t = Mean([f_o(\mathbf{E}^o); \mathbf{E}^a]), \tag{7}$$

where $f_o(\cdot)$ is a linear transformation, and $\mathbf{E}^a \in \mathbb{R}^{|\mathcal{X}| \times d_a}$ is the attentive representation of $\mathcal{X}$. If $\mathcal{X}$ is the concept set, $\mathbf{e}_t \in \mathbb{R}^{2d_a}$ is denoted as $\mathbf{e}_t^c$; if $\mathcal{X}$ is the question candidate set, it is denoted as $\mathbf{e}_t^q$.

Learning target encoding. Like the question candidate or concept set encoding, we represent the learning target by averaging the attentive encoding in the set of the learning target. That is:

$$\mathbf{g}_t = Mean(\mathbf{E}_g^a), \tag{8}$$

where $\mathbf{E}_g^a \in \mathbb{R}^{|\mathcal{G}| \times 2d_a}$ is obtained by stacking the attentive representation of the questions in the learning target.

Learning history encoding. Given the student's learning history $\mathcal{H}_t = \{(q_1, y_1), (q_2, y_2), ..., (q_t, y_t)\}$, we encode each record (q_i, y_i) by

$$\mathbf{z}_i = f_z\left([\mathbf{e}_i;\ y_i]\right), \tag{9}$$

where $f_z(\cdot)$ is a linear function, $\mathbf{e}_i \in \mathbb{R}^{|\mathcal{X}|}$ is the one-hot encoding of question q_i, y_i is the answer correctness, and $\mathbf{z}_i \in \mathbb{R}^{d_z}$ is the record representation.

We initialize the learning history as $\mathbf{h}_0 = \mathbf{0}$, and process the records through an RNN (implemented with LSTM [8]):

$$\mathbf{h}_i = \text{RNN}(\mathbf{z}_i, \mathbf{h}_{i-1}), \text{for } 1 \le i \le t, \tag{10}$$

After processing all records, the final learning history representation is $\mathbf{h}_t \in \mathbb{R}^{d_h}$, where d_h is the dimension of the learning history.

Prompt encoding. As we discussed previously, the prompts are the text. Both high-level and low-level prompts are encoded using a pretrained LLM:

$$\mathbf{m}_t = PLLM(\mathcal{P}_t), \tag{11}$$

where $\mathcal{P}_t$ represents either the high-level prompt $\mathcal{P}_t^h$ or low-level prompt $\mathcal{P}_t^l$, with $\mathbf{m}_t \in \mathbb{R}^{d_m}$. If it's a high-level prompt, it's denoted as $\mathbf{m}_t^h$; if low-level, $\mathbf{m}_t^l$.

Finally, we obtain the high-level learning state of the student via:

$$\mathbf{s}_t^h = f_{p_1}(\mathbf{h}_t) + f_{p_2}(\mathbf{g}_t) + f_{p_3}(\mathbf{e}_t^c) + \mathbf{m}_t^h. \tag{12}$$

where $f_{p_1}(\cdot)$, $f_{p_2}(\cdot)$, $f_{p_3}(\cdot)$, and $f_{p_4}(\cdot)$ are linear projection layers.

Similarly, the low-level learning state of the student is obtained via:

$$\mathbf{s}_t^l = f_{p_1}(\mathbf{h}_t) + f_{p_2}(\mathbf{g}_t) + f_{p_4}(\mathbf{e}_t^q) + \mathbf{m}_t^l, \tag{13}$$

4.4 Optimization

To enable personalized question recommendation, we freeze the pre-trained LLM parameters and fine-tuned using low-rank adaptation (LoRA) [11]. We defined three learning targets to improve model performance.

We optimize the high-level and low-level module with policy gradient:

$$\mathcal{L}_h = -\sum_{t=1}^{T} \hat{r}_t \log \pi_h(c_t|\mathbf{s}_t^h), \tag{14}$$

$$\mathcal{L}_l = -\sum_{t=1}^{T} \hat{r}_t \log \pi_l(q_t|\mathbf{s}_t^l), \tag{15}$$

where $\hat{r}_t$ the accumulated reward of selecting q_t in long term, and

$$\hat{r}_t = r_t + \gamma \hat{r}_{t+1}, \tag{16}$$

with r_t being the learning effect (Eq. 1) and γ the discount factor.

Following [4], we use $\mathbf{h}_t$ in a binary classifier to predict response correctness and optimize the representation based on the prediction and ground truth.

$$\begin{aligned} \hat{y}_t &= \delta(f_p(\mathbf{h}_t)), \\ \mathcal{L}_p &= -\sum_{t=1}^{T} \left(y_t \ln \hat{y}_t + (1-y_t)\ln(1-\hat{y}_t)\right), \end{aligned} \tag{17}$$

Here, δ denotes the Sigmoid function, f_p represents the multi-layer perceptron (MLP) [21]. y_t is the ground truth, indicating the student's response correctness.

Therefore, HierLLM is optimized by

$$\mathcal{L} = \mathcal{L}_h + \mathcal{L}_l + \alpha \mathcal{L}_p, \tag{18}$$

where $\mathcal{L}_p$ is computed in low-level module. α is the hyperparameter.

Table 2. The dataset.

Dataset	Students	Concepts	Questions	Records
ASSIST09	2968	121	15003	185110
Junyi	10000	39	2163	882198

5 Experiment

In this section, we first introduce the experimental setup, followed by a detailed discussion of experiment results.

5.1 Datasets and Simulators

Datasets. We use two public datasets ASSIST09[1] [6] and Junyi[2] [3] to evaluate HierLLM. Table 2 shows the statistics.

Simulators. We develop simulators based on [13,4,15] to emulate student behavior, using a Rule-based Simulator with IRT and a KT-based Simulator (DKT, IEKT), producing KSS, D-A, D-J, I-A, and I-J simulators.

5.2 Baselines

We evaluate our model against two groups: non-RL methods (Random, GRU4REC [10], FMLP [24], GPT-3.5 [2]) and RL methods (DQN [18], AC [20], SAC [9], TD3 [7], CSEAL [15], SRC [4], GEHRL [14]).

5.3 Experiment Setting

We fine-tune Llama-7B with LoRA. We set $d_a = d_z = 64$, $d_h = 128$, $d_m = 4096$, and 1 attention head. $\alpha = 1$ and $\gamma = 1$ for all cases. Experiments are run on NVIDIA A40 GPUs (40GB) with the Adam optimizer and learning rates $\{1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}\}$, with optimized baseline parameters.

5.4 Evaluation Metrics

We evaluate performance by calculating E_a and E_b for each simulated student and averaging the learning effect across all students.

5.5 Experiment Result

We test simulated students with 10, 30 steps for KSS and 10, 30, 200 steps for KT-based simulators. Table 3 shows HierLLM outperforms all baselines, with better performance as steps increase. RL-based methods outperform non-RL approaches, supporting MDP for question recommendation. GPT-3.5's suboptimal performance highlights the need for fine-tuning and domain expertise.

[1] https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data

[2] https://www.kaggle.com/datasets/Junyiacademy/ learning-activity-public-dataset-by-Junyi-academy

Table 3. Learning effects of different methods on 5 simulators, tested five times. **Bold** indicates the best performance, underline the second-best, * denotes p-value < 0.05.

Simulator	t	Random	GRU4REC	FMLP	GPT-3.5	DQN	AC	SAC	TD3	CSEAL	SRC	GEHRL	HierLLM
KSS	10	0.0668	0.0033	0.0023	0.0150	0.0793	0.0263	0.0482	0.0353	0.1558	0.1157	0.1528	**0.2262***
	30	0.1901	0.0006	0.0024	0.0468	0.1469	0.0388	0.0788	0.0848	0.4835	0.2416	0.3173	**0.4937**
I-J	10	0.0070	-0.0121	-0.0525	-0.0155	0.1593	-0.0085	-0.0048	-0.0162	-0.0106	0.0317	0.0611	**0.2905**
	30	-0.0009	0.0042	-0.0733	-0.0621	0.2487	-0.0032	0.0112	-0.0413	0.0245	0.0672	0.1357	**0.4438**
	200	0.0307	0.1306	-0.0871	-0.0301	0.4795	0.1548	0.1087	-0.0115	0.2687	0.2089	0.3130	**0.6470**
D-J	10	0.0008	0.0010	-0.0024	-0.0108	0.0077	-0.0088	0.0069	-0.0006	-0.0005	0.0009	0.0000	**0.0182***
	30	0.0014	0.0000	-0.0030	-0.0128	0.0109	-0.0079	0.0063	-0.0003	-0.0004	0.0006	0.0018	**0.0194**
	200	0.0015	0.0040	-0.0030	-0.0019	0.0096	-0.0097	0.0049	0.0040	0.0014	0.0028	0.0027	**0.0176***
I-A	10	0.0570	0.0508	-0.0078	-0.0062	0.1574	0.1697	0.2205	-0.0077	-0.0217	0.0618	0.0337	**0.3533**
	30	0.1309	0.1003	-0.0504	0.0792	0.2904	0.2648	0.3358	-0.0329	-0.0606	0.1411	0.1269	**0.5812**
	200	0.2734	0.0953	-0.1036	-0.0651	0.5176	0.4127	0.5683	0.1108	0.1419	0.3041	0.2984	**0.7848***
D-A	10	0.0001	-0.0052	-0.0544	-0.0541	0.0150	0.0055	0.0073	0.0066	-0.0360	0.0004	-0.0034	**0.0498***
	30	0.0012	-0.0053	-0.0730	-0.0811	0.0175	0.0066	0.0073	0.0076	-0.0436	0.0015	0.0058	**0.0509***
	200	0.0000	-0.0020	-0.0743	-0.0649	0.0082	0.0046	0.0140	0.0124	-0.0096	0.0021	0.0021	**0.0510***

Table 4. Learning effects Δ_u under cold start problem.

Simulator	t	DQN	CSEAL	SRC	GEHRL	HierLLM
KSS	10	0.0718	0.1601	0.1242	0.1602	**0.2256**
	30	0.1273	0.4295	0.1937	0.3290	**0.4846**
I-J	10	0.1733	-0.0063	0.0626	0.0315	**0.2098**
	30	0.1620	0.0401	0.1256	0.1976	**0.3779**
	200	0.2521	0.1393	0.2346	0.2710	**0.5838**
D-J	10	0.0028	-0.0015	0.0002	0.0072	**0.0090**
	30	0.0017	-0.0001	0.0018	0.0078	**0.0096**
	200	0.0085	-0.0092	0.0002	0.0032	**0.0103**
I-A	10	0.1375	-0.0166	0.0450	0.0473	**0.3078**
	30	0.3151	-0.0846	0.1433	0.0936	**0.4956**
	200	0.4770	0.2225	0.2732	0.2667	**0.7817**
D-A	10	0.0175	-0.0310	-0.0003	0.0026	**0.0274**
	30	0.0181	-0.0514	0.0004	0.0168	**0.0281**
	200	0.0256	-0.0211	-0.0001	0.0002	**0.0281**

5.6 Cold Start Study

The cold start problem arises when there is no prior learning history. In our experiment, we simulate students with zero history, relying on LLMs' reasoning. Table 4 shows HierLLM outperforms other models at all time steps ($t = 10$, $t = 30$, $t = 200$), demonstrating its ability to generate quality recommendations without prior history. Performance improves with more recommendation steps, highlighting its capacity for incremental optimization.

5.7 Ablation Study

We conduct an ablation study with three variants: (1) **-High**, removing the high-level module and treating all questions as candidates in the low-level module; (2) **-LLM**, replacing LLM in both modules with a regular MLP; (3) **LLMFrozen**, training HierLLM without fine-tuning. Figure 3 shows that HierLLM and -High perform similarly in KSS due to similar concepts and questions, while removing any component reduces performance, emphasizing their importance. HierLLM outperforms -LLM, demonstrating the LLM's ability to model sequences and reason. LLMFrozen exceeds -LLM in D-J, I-A, and D-A, but without fine-tuning, performance drops, highlighting the need for fine-tuning.

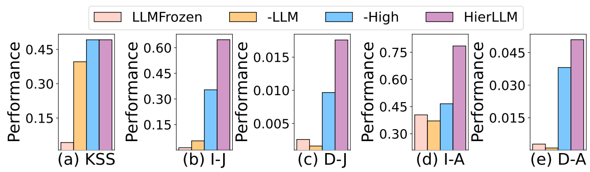

Fig. 3. Ablation Study

6 Conclusion

We propose HierLLM, a hierarchical model for question recommendation that addresses cold start and large question set challenges, achieving state-of-the-art performance. Future work we will explore LLMs beyond Llama2-7B.

Acknowledgement

This work is partially supported by the National Natural Science Foundation of China (62307020, U2341229) and the Ministry of Science and Technology of China (2023YFF0905400).

References

1. Bao, K., Zhang, J., Zhang, Y., Wang, W., Feng, F., He, X.: Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems. pp. 1007–1014 (2023)
2. Brown, T.B.: Language models are few-shot learners. arXiv preprint ArXiv:2005.14165 (2020)
3. Chang, H.S., Hsu, H.J., Chen, K.T., et al.: Modeling exercise relationships in e-learning: A unified approach. In: EDM. pp. 532–535 (2015)
4. Chen, X., Shen, J., Xia, W., Jin, J., Song, Y., Zhang, W., Liu, W., Zhu, M., Tang, R., Dong, K., et al.: Set-to-sequence ranking-based concept-aware learning path recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 5027–5035 (2023)
5. Eigner, E., Händler, T.: Determinants of llm-assisted decision-making. arXiv preprint arXiv:2402.17385 (2024)
6. Feng, M., Heffernan, N., Koedinger, K.: Addressing the assessment challenge with an online system that tutors as it assesses. User modeling and user-adapted interaction **19**, 243–266 (2009)
7. Fujimoto, S., Hoof, H., Meger, D.: Addressing function approximation error in actor-critic methods. In: International conference on machine learning. pp. 1587–1596. PMLR (2018)
8. Graves, A., Graves, A.: Long short-term memory. Supervised sequence labelling with recurrent neural networks pp. 37–45 (2012)

9. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement. In: Proceedings of the 35th International Conference on Machine Learning. July 10th-15th, Stockholm, Sweden. vol. 1870 (1861)
10. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)
11. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
12. Kim, S., Kang, H., Choi, S., Kim, D., Yang, M., Park, C.: Large language models meet collaborative filtering: An efficient all-round llm-based recommender system. arXiv preprint arXiv:2404.11343 (2024)
13. Kubotani, Y., Fukuhara, Y., Morishima, S.: Rltutor: Reinforcement learning based adaptive tutoring system by modeling virtual student with fewer interactions. arXiv preprint arXiv:2108.00268 (2021)
14. Li, Q., Xia, W., Yin, L., Shen, J., Rui, R., Zhang, W., Chen, X., Tang, R., Yu, Y.: Graph enhanced hierarchical reinforcement learning for goal-oriented learning path recommendation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 1318–1327 (2023)
15. Liu, Q., Tong, S., Liu, C., Zhao, H., Chen, E., Ma, H., Wang, S.: Exploiting cognitive structure for adaptive learning. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 627–635 (2019)
16. Liu, Q., Chen, N., Sakai, T., Wu, X.M.: A first look at llm-powered generative news recommendation. arXiv preprint arXiv:2305.06566 (2023)
17. Liu, Z., Dong, L., Wu, C.: Research on personalized recommendations for students' learning paths based on big data. International Journal of Emerging Technologies in Learning (iJET) **15**(8), 40–56 (2020)
18. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
19. Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., Wu, X.: Unifying large language models and knowledge graphs: A roadmap. IEEE Transactions on Knowledge and Data Engineering (2024)
20. Sutton, R.S., McAllester, D., Singh, S., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. Advances in neural information processing systems **12** (1999)
21. Taud, H., Mas, J.F.: Multilayer perceptron (mlp). Geomatic approaches for modeling land change scenarios pp. 451–455 (2018)
22. Xi, Y., Liu, W., Lin, J., Cai, X., Zhu, H., Zhu, J., Chen, B., Tang, R., Zhang, W., Zhang, R., et al.: Towards open-world recommendation with knowledge augmentation from large language models. arXiv preprint arXiv:2306.10933 (2023)
23. Zhang, Y., Mao, S., Ge, T., Wang, X., de Wynter, A., Xia, Y., Wu, W., Song, T., Lan, M., Wei, F.: Llm as a mastermind: A survey of strategic reasoning with large language models. arXiv preprint arXiv:2404.01230 (2024)
24. Zhou, K., Yu, H., Zhao, W.X., Wen, J.R.: Filter-enhanced mlp is all you need for sequential recommendation. In: Proceedings of the ACM web conference 2022. pp. 2388–2399 (2022)
25. Zhou, Y., Huang, C., Hu, Q., Zhu, J., Tang, Y.: Personalized learning full-path recommendation model based on lstm neural networks. Information sciences **444**, 135–152 (2018)