Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.38268

Personalized Learning Path Optimization Based on Generative Artificial Intelligence

Zhikai Wang

School of Computer Science, University of Birmingham, Birmingham, UK
wellington589125@gmail.com

Abstract. Learning paths personalized by behavior have become common among personalized learning systems. However, the traditional methods are usually inefficient at building coherent paths as the learner's knowledge state changes dynamically, and the prerequisite structure exists. In light of this problem, we present a generative learning-path optimization framework based on EdNet-KT1 interaction dataset, Transformer knowledge tracing model, Qwen2.5-7B-Instruct candidate generator, and constraint-aware reranking. Mastery profiles of learners are extracted, and then five ten-resource candidate paths are generated and sorted according to learning gain, prerequisites, difficulty deviation, and repetition. We compare our method to collaborative filtering, GRU4Rec, and knowledge-tracing greedy recommendation on offline experiments. The generative optimizer gets NDCG@10 of 0.589, future-concept coverage of 0.721, and prerequisite violation rate of 0.031, and thus outperforms the best baseline on all three metrics. Ablation analysis shows that the prerequisite violation rate rises to 0.091 without constraints, and the learning gain drops to 0.104 without the learning gain term, respectively. Moreover, the full model still achieves better NDCG@10 performance on all learner history categories than other baselines, costing 0.84 s per learner with the memory peak of 13.6 GB. This research demonstrates the effectiveness of combining the generation of learning paths with pedagogical constraints.

Keywords: Generative Artificial Intelligence, Personalized Learning, Learning Path Optimization, Knowledge Tracing, Educational Recommendation

1. Introduction

Learning platforms have come to use behavioral traces more often for personalized content delivery, but large-scale systems are still facing difficulty in forming meaningful learning paths from scattered interactions. Recent surveys reveal that the application of artificial intelligence to facilitate adaptive sequencing can enable personalized paths, but in reality, such personalizations are still limited to isolated recommendations [1]. While rule-based and modular recommendation systems enhance interpretability, the adaptation capability relies highly on pedagogical logic and constraints specified in advance [2]. Hence, novel personalized learning studies via artificial intelligence have shifted toward the development of methods which incorporate estimation of learner state, recommendation, and generative reasoning in one adaptive process [3]. To this end, a computational pipeline is

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.38268

designed based on EdNet-KT1 interaction data, knowledge tracing via Transformer, Qwen2.5-7B-Instruct candidate generation, and constraint-aware reranking. Comparison and ablation experiments are conducted offline for assessing ranking relevance, concept coverage, violation of prerequisites, repetition, improvement of predicted mastery, and robustness against varying history lengths of learners under a reproducible experimental setting.

2. Literature review

2.1. Personalized learning path recommendation

In the case of personalized learning path recommendation, a change has been made from the static method of content matching to decision making that is sequentially done and includes the changing needs of learners. In one study, the research uses cascaded deep Q-networks to model behavior in such a way that the path construction will be helped by including short-term actions and long-term learning objectives [4]. Process mining along with deep knowledge tracing has been used in another research so that the branch selection will be based on the changing states of knowledge rather than having a fixed path [5]. These two techniques increase the adaptability; however, these still rely heavily on predefined candidate paths or policies.

2.2. Learner knowledge state modeling

Knowledge Tracing can be used for the representation of the state that is necessary for adaptive paths construction since it models the change in mastery between consecutive interactions. The study applies Hierarchical Transformer Attention to differentiate between session-to-session and intra-session learning dynamics, thereby providing more accurate representation of the learner's history [6]. Another study further develops this idea by implementing a dual graph model in order to consider connections between exercises and concepts in the process of predicting future performance [7]. However, although the Transformer and graph approaches better capture dependencies than score characteristics do, and higher prediction accuracy may be taken as a criterion for the next learning resource, the path optimizer still needs to translate the probability of mastery into sequences.

2.3. Generative path planning

The generative models bring the ability of sequence generation that the traditional recommendation systems lack and allow creating possible learning pathways from the structured information about the learner context. Studies highlight that the large language models are promising tools for personalized educational interfaces, providing recommendations and tutoring with a particular focus on issues of reliability, transparency, and controllability [8]. Another work shows how generative AI can personalize instructional materials by adapting them to the knowledge level of a specific learner in real time [9]. It becomes clear that there is a tension between these two approaches, with one side offering the ability to expand the recommendation space and the other producing improper sequences or invalid content.

DOI: 10.54254/2755-2721/2026.38268

3. Experimental methods

3.1. Learning state modeling

The pipeline for the experiments was programmed using the Python 3.11 programming language and involves using the EdNet-KT1 public dataset as the source of behavior. Question IDs, tags on knowledge, correctness, and timestamps are maintained while learners who had less than 20 interactions are dropped. Each of the learners' histories is sorted by time and then limited to sequences of 100 interactions or fewer. The learners are split into training, validation, and testing sets by a ratio of 70:15:15 respectively without record splitting across the three sets. The difficulty of the resources is derived from the errors in the training set, and the prerequisite directed edges are learned from frequent concept transitions before successful responses. The transformer has two layers with 128 hidden sizes and 4 heads of attention and encodes the interaction sequence before producing the concept mastery probabilities. It is reasonable to use the Transformer model for knowledge tracing due to its capability to model structured dependencies among questions and skills [10]. The binary cross-entropy is optimized using the AdamW optimizer at a learning rate of 0.001 and batch size 64 until there are five consecutive epochs with no improvement in the validation loss.

3.2. Candidate path generation

The structured learner state is transformed into a static prompt that includes weak concepts, strong concepts, difficulty of candidate-resource, and prerequisites connections. This prompt is passed to the locally hosted Qwen2.5-7B-Instruct LLM. This LLM produces five candidate paths consisting of ten resources in JSON format using temperature of 0.3 [11]. For each candidate path, predicted learning gain is calculated from the change in mastery probabilities, as shown in Equation (1).

$$G(P_i)=\frac{1}{L}\sum_{j=1}^{L}\left(\hat{m}_{u,c_j}^{post}-m_{u,c_j}^{pre}\right) \tag{1}$$

Where L denotes path length, $\hat{m}_{u,c_j}^{post}$ is the current mastery probability, and $\hat{m}_{u,c_j}^{post}$ is the predicted mastery after completing the candidate path. The four path criteria are normalized to [0,1]. A weighted reranking function integrates learning gain, prerequisite compliance, difficulty matching, and repetition control, as shown in Equation (2).

$$S(P_i)=0.40\tilde{G}_i+0.25\left(1-\tilde{V}_i\right)+0.20\left(1-\tilde{D}_i\right)+0.15\left(1-\tilde{R}_i\right) \tag{2}$$

Where $\tilde{G}_i$, $\tilde{G}_i$, $\tilde{D}_i$, and $\tilde{R}_i$ represent normalized learning gain, prerequisite violations, difficulty deviation, and repetition rate. The highest-scoring path is selected, while prompts, generated sequences, component scores, and final rankings are retained for reproducibility.

3.3. Path evaluation design

Path evaluation uses the held-out learner set and compares collaborative filtering, GRU4Rec, a knowledge-tracing greedy recommender, and the complete generative optimizer. Each method

DOI: 10.54254/2755-2721/2026.38268

outputs ten resources per learner. Ranking relevance is measured against subsequent observed interactions using NDCG@10, as shown in Equation (3).

$$NDCG@10=\frac{DCG@10}{IDCG@10}$$

$$DCG@10=\sum_{k=1}^{10}\frac{2^{rel_k}-1}{\log_2(k+1)} \tag{3}$$

Where rel_k indicates whether the resource ranked at position k appears in the learner's future interaction sequence, and IDCG@10 denotes the ideal discounted cumulative gain. Structural validity is measured using prerequisite violation rate, as shown in Equation (4).

$$PVR=\frac{\sum_{i=1}^{N}v_i}{\sum_{i=1}^{N}e_i} \tag{4}$$

Where v_i is the number of violated prerequisite transitions in path i, e_i is the number of prerequisite-sensitive transitions, and N is the number of evaluated paths. Future-concept coverage, repetition frequency, and estimated improvement in mastery using a frozen knowledge tracing model are also noted. The experiments are conducted in PyTorch 2.3 on an RTX 4090 GPU with five different random seeds. Two ablation cases exclude either the prerequisite or learning gain term. Average values, standard deviation, 95% confidence interval, and statistical significance tests are performed under the same test condition.

4. Results

4.1. Recommendation performance

According to Table 1, the generative optimizer yielded the highest recommendation quality performance in five runs. The NDCG@10 metric got a score of 0.589±0.004, versus 0.536±0.004 obtained by the knowledge-tracing greedy baseline, which is a 9.9% relative improvement. The future concept coverage rose from 0.667±0.007 to 0.721±0.006, while the prerequisite violation rate dropped to 0.031±0.002, being lower than that of GRU4Rec and collaborative filtering. The paired t-tests with the best baseline showed significance at p=0.002 for NDCG@10, p=0.004 for coverage, and p<0.001 for violation rate. Such findings suggest that better ranking relevance and greater structural coherence were gained at the same time. Qwen generated more sequences of feasible candidates, whereas the weighted reranking process removed transitions not supported by the graph. Sequential baselines also performed better than collaborative filtering, supporting the importance of sequential interactions, yet higher violation rates demonstrated that sequence modeling alone could not guarantee pedagogical validity of sequencing. Lower standard deviations also indicated the stability of the whole framework in multiple runs.

DOI: 10.54254/2755-2721/2026.38268

Table 1. Recommendation performance on the held-out learner set

Method	NDCG@10	Future-Concept Coverage	Prerequisite Violation Rate
Collaborative Filtering	0.421 ± 0.006	0.584 ± 0.009	0.142 ± 0.004
GRU4Rec	0.497 ± 0.005	0.631 ± 0.008	0.119 ± 0.003
KT-Greedy	0.536 ± 0.004	0.667 ± 0.007	0.073 ± 0.003
Generative Path Optimizer	0.589 ± 0.004	0.721 ± 0.006	0.031 ± 0.002

4.2. Module contribution results

Ablation results show the roles of the two reranking elements. The omission of the prerequisite term resulted in a PVR increase from 0.031 to 0.091 and a drop in NDCG@10 from 0.589 to 0.557, demonstrating that structural constraints prevent unrealistic state transitions. Omitting the learning gain term led to a reduction in mastery gain from 0.126±0.004 to 0.104±0.005 and an increase in repetitions from 0.084±0.003 to 0.097±0.004. As illustrated in Figure 1, the complete system obtained NDCG@10 scores of 0.541, 0.574, 0.602, and 0.619 in four groups of learners based on their history, which outperformed both ablation systems. The longer interaction history led to better performance, and the complete setup kept its superiority regardless of sparsity level. It took the complete system 0.84 s for each learner to generate and rerank five paths, and its maximum GPU memory consumption was 13.6 GB.

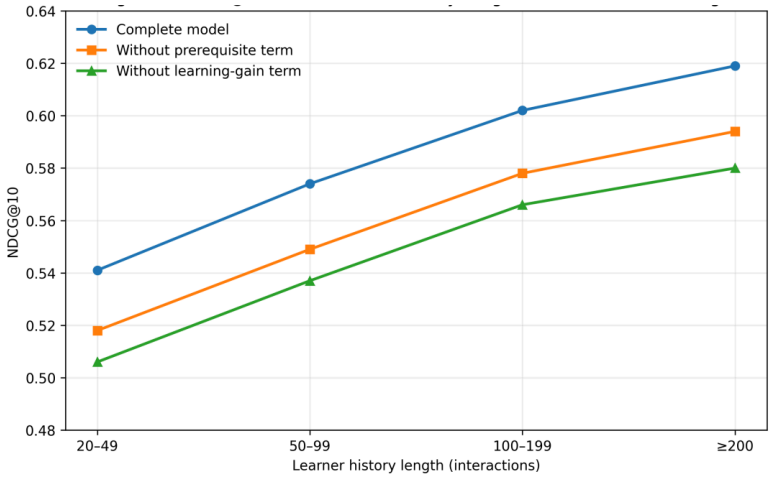

Figure 1. NDCG@10 across learner-history lengths under ablation settings

5. Discussion

It can be seen that the primary benefit comes from decoupling candidate generation from pedagogical filtering. Qwen2.5-7B-Instruct widens the sequence space, yet the low violation rate occurs only after prerequisite-aware reranking, suggesting that diverse generations are effective under the constraints of the deterministic filter. The findings of the ablation study also indicate that scoring by learning gain has influence not just on the estimation of the mastery level but also on the reduction of repetition, implying that targeting of weak concepts impacts resource diversity indirectly. The fact that the consistent NDCG difference is kept over both groups with short and long history is crucial, since knowledge tracing usually provides unreliable information for sparse

DOI: 10.54254/2755-2721/2026.38268

learners; thus, the ability to keep the lead in the group with 20–49 interactions is explained by the use of structurally designed prompts in the presence of insufficient behavioral data. It is necessary to note that the predicted mastery improvement is based on the predictions of the frozen model and not on the post-learning performance, meaning that the value is to be treated as a proxy for optimization.

6. Conclusion

The proposed personal learning path pipeline was created by integrating EdNet-KT1 data preprocessing, Transformer knowledge tracing, Qwen2.5-7B-Instruct generation, and reranking under constraints. Offline experiments confirmed that this architecture combination is able to increase ranking quality, future concept coverage, prerequisite adherence, and predicted mastery gain without being sensitive to learner history length. Ablation study revealed the relationship between the use of prerequisites and structural validity as well as learning gain and repetition reduction. Currently, there are certain limitations of this evaluation based on offline behavior data, predicted prerequisites, and estimated mastery values obtained via the model. In future research, the validation of the paths needs to be conducted via experiments with learners, real prerequisite graphs provided by instructors, and small-scale generators.

References

[1] Bayly-Castaneda, K., Ramirez-Montoya, M. S., & Morita-Alexander, A. (2024). Crafting personalized learning paths with AI for lifelong learning: A systematic literature review. *Frontiers in Education, 9*. Frontiers.

[2] Pelánek, R., Effenberger, T., & Jarušek, P. (2024). Personalized recommendations for learning activities in online environments: A modular rule-based approach. *User Modeling and User-Adapted Interaction, 34*(4), 1399–1430.

[3] Farhood, H., et al. (2025). Artificial intelligence-based personalised learning in education: A systematic literature review. *Discover Artificial Intelligence, 5*(1), 331.

[4] Ma, D., et al. (2024). Learning path recommendation with multi-behavior user modeling and cascading deep Q networks. *Knowledge-Based Systems, 294*, 111743.

[5] Zhang, F., Feng, X., & Wang, Y. (2024). Personalized process–type learning path recommendation based on process mining and deep knowledge tracing. *Knowledge-Based Systems, 303*, 112431.

[6] Ke, F., et al. (2024). HiTSKT: A hierarchical transformer model for session-aware knowledge tracing. *Knowledge-Based Systems, 284*, 111300.

[7] Cui, C., et al. (2024). DGEKT: A dual graph ensemble learning method for knowledge tracing. *ACM Transactions on Information Systems, 42*(3), 1–24.

[8] Kasneci, E., et al. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences, 103*, 102274.

[9] Jauhiainen, J. S., & Guerra, A. G. (2024). Generative AI and education: Dynamic personalization of pupils' school learning material with ChatGPT. *Frontiers in Education, 9*. Frontiers Media SA.

[10] Liang, Z., et al. (2024). GELT: A graph embeddings based lite-transformer for knowledge tracing. *Plos One, 19*(5), e0301714.

[11] Deshmukh, P., Samrat, H., & Pervin, N. (2026). Personalized learning path generation using large language models and sequential recommendations. *Data & Knowledge Engineering*, 102595.