UNO! *UN*ified *O*ffline Training Paradigm for Learning Path Recommendation

Linzhi Peng[1], Wentao Zhu[1], Ke Cheng[1], Heng Chang[2], Junchen Ye[1], Bowen Du[1, 3], Weifeng Lv[1]

[1]Beihang University, Beijing, China
[2]Tsinghua University, Beijing, China
[3]Zhongguancun Laboratory, Beijing, China
{lzpeng626, zhuwentao}@buaa.edu.cn, changh17@tsinghua.org.cn,
{ckpassenger, junchenye, dubowen, lwf}@buaa.edu.cn,

Abstract

With the wide adoption of online education platforms, adaptive learning systems have become increasingly important. Learning Path Recommendation (LPR) aims to dynamically adjust learning content to optimize learning efficiency based on individual student needs. However, current LPR methods suffer from sparse reward for precise assessment and only focus on anonymous sessions that overlook more personalized and effective paths. To address these challenges, we propose UNO, ***UN***ified ***O***ffline Training Paradigm for Learning Path Recommendation. This approach introduces an offline training paradigm in reinforcement learning based LPR to provide dense process rewards by a personalized advantage based on a reward model, which can estimate the students' internal knowledge levels on the learning targets. Additionally, we propose UniLPR model, a personalized recommendation system that unifies modeling the implicit relationships between students' long-term accumulation and evolving requirements for questions, and refines through Group Relative Policy Optimization(GRPO). Finally, we design learning tasks that encompass historical reviewing, recent learning, and long-term exploratory learning to simulate the comprehensive and diverse needs of students. Our UNO achieves state-of-the-art performance across all tasks, demonstrating its effectiveness.

Code — https://github.com/PengLinzhi/UNO-LPR

Introduction

With the wide adoption of online education platforms, a large amount of educational records have been accumulated for intelligent tutoring. Among adaptive learning systems, learning path recommendation (LPR) dynamically provides personalized learning paths based on students' learning history, targets, and the relationships between questions, to improve students' mastery of learning targets and enhance learning efficiency (Zhu et al. 2018; Yin et al. 2021).

Initially, rule-based LPR methods, leveraging knowledge structures and optimization algorithms, are effective in data-scarce settings, but fail to consider students' current knowledge levels, limiting their adaptability (Govindarajan, Kumar, and Kinshuk 2016; Zhu et al. 2018). Subsequent approaches, adopting sequential recommendation techniques

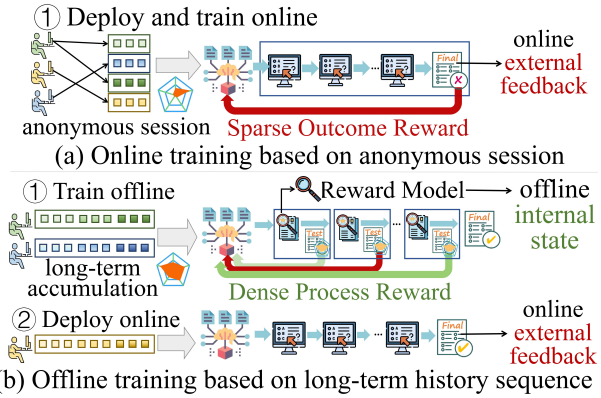

Figure 1: Online training based on anonymous sessions and sparse outcome reward is less efficient and adaptive than long-term offline training with dense process reward.

such as collaborative filtering and sequence-based models, aim to identify similar learning sequence patterns rather than directly improving learning efficiency (Wu et al. 2017; Zhou et al. 2018; Nasrin et al. 2025). Recently, reinforcement learning (RL) has emerged as a more effective approach to enhancing learning efficiency. The student learning process conforms to a Markov process, which establishes an LPR system as a sequential decision-maker (Chen et al. 2018; Kubotani, Fukuhara, and Morishima 2021; Yun et al. 2024). Advances in student Knowledge Tracing models(KT) have enabled realistic simulations of student states and behaviors (Piech et al. 2015; Wang and Sahebi 2023). In this RL framework, existing methods utilize KT models to simulate students as the environment, with the recommendation system selecting questions to guide students (Liu et al. 2019; Li et al. 2023; Zhang et al. 2024; Cheng et al. 2025).

Despite these successes, there are two limitations in current RL-based LPR methods as shown in Figure 1(a). They suffer from **sparse outcome rewards**, only evaluating the overall effectiveness of the whole sequence based on the final test scores, failing to assess the specific contribution of each question to the learning targets. If the whole sequence performs poorly, but contains questions that could help students, those effective questions may be mistakenly deemed ineffective. Besides, they are **lack of personalization** and

focus solely on the anonymous session. Specifically, they decompose and shuffle all records by student and session, and then model the anonymous sessions without long-term history. As a result, they fail to recognize students' personalized states that are not covered in the current session, recommending the same questions for similar session patterns.

In this paper, as depicted in 1(b), we argue that (1) accurately modeling the quantitative impact of each question on learning targets is essential for generating dense process rewards. However, the existing RL-based framework is restricted to the prevailing online training paradigm: the deployed model interacts with live students on the platform without isolated train/test splits, updating incrementally using external feedback like in the real scenario. This confines the reward function built on the final improvements after the recommendation. (2) With the long-term learning accumulation process, students' demand and reliance on the recommended questions dynamically evolve. Most methods retrieve candidates from the last acquired knowledge in the session on a static knowledge graph, and then apply decoupled probabilistic scoring only according to the student's mastery level within that session. If a student attempts to review, the system may redundantly provide prerequisite content as if the student had never encountered it before. (3) Student requirements for lifelong and diverse learning, such as reviewing previously learned content, previewing recent lessons, and exploring new items, are not fully considered by existing benchmarks. They mainly design the targets as the knowledge at the end of a session and recommend based on what was learned earlier within the same session.

To address the aforementioned challenges, we propose UNO, *UN*ified *O*ffline Training Paradigm for Learning Path Recommendation. **Firstly, we introduce the offline training paradigm to utilize dense process rewards during training**. In the offline training stage, we employ a KT-based reward model that identifies the internal states on targets and assesses each recommended question by their cognitive improvement. And the reward model will be deactivated during testing to simulate a realistic deployment scenario. **Secondly, we propose UniLPR model, a personalized recommendation system that unifies modeling students' accumulating states and evolving demand for knowledge** by optimizing the unified representation of long-term individual knowledge levels and knowledge structures of questions through both supervised learning constrained by the history sequence and group relative policy optimization based on the dense rewards in the LPR task. **Finally, we adopt long-term learning history for personalized recommendation and develop three learning modes**: Historical Review, Recent Learning, and Exploratory Learning, to evaluate the model in diverse and lifelong learning demands.

Our contributions can be summarized as follows:

- We propose an offline training paradigm for learning path recommendation that introduces dense process rewards. This approach bridges the gap between simulation training and real-world deployment requirements, enhancing the precision and effectiveness of recommendations.
- We introduce UniLPR model, a personalized LPR system that unifies representations of long-term knowledge level and knowledge structure while optimizing through GRPO in recommendation tasks. It can capture implicit associations between students' evolving accumulation and dynamic requirements for questions.
- We extend three types of learning tasks that can cover various learning plans and target distributions. UNO achieves state-of-the-art performance across all these tasks, demonstrating its effectiveness in addressing long-term and diverse learning demands.

Related Work

Learning Path Recommendation

Learning Path Recommendation systems adjust learning content to optimize learning efficiency. Researchers first built rule-based LPR models in data-scarce settings, leveraging knowledge structures and optimization algorithms, including genetic algorithms and matrix factorization(Govindarajan, Kumar, and Kinshuk 2016; Nabizadeh et al. 2020). Next, deep learning is introduced in LPR (Wu et al. 2017; Nasrin et al. 2025). They adopt sequential recommendation techniques, such as collaborative filtering KNN (Cover and Hart 1967), GRU4Rec(Hidasi and Karatzoglou 2018) and SASRec(Kang and McAuley 2018), as they can better capture features in behaviors. HSTU innovatively unifies sequential user behaviors and items(Zhai et al. 2024; Zhang et al. 2025a). Dynamic graphs are also introduced to model behaviorals(Xu et al. 2024, 2025a,b).

Recently, LPR based on the reinforcement learning framework has become a leading approach (Kubotani, Fukuhara, and Morishima 2021; Yun et al. 2024). Because DKT can simulate student behavior with learning history (Piech et al. 2015; Liu et al. 2023), and has been validated for effectiveness in personalized lifelong learning (Wang and Sahebi 2023). Liu first employs DKT for student simulation (Liu et al. 2019) and uses a heuristic strategy to search on a static knowledge graph from the last acquired knowledge in the session. GEHRL proposes a hierarchical model to set subgoals (Li et al. 2023). SRC models the relationships among questions and generates a complete learning path in one time (Chen et al. 2023). DLPR maintains consistent difficulty and utilizes the A* algorithm to search the path (Zhang et al. 2024). GenMentor and DLELP utilize LLM agents to discover semantically relationships of questions (Wang et al. 2025a; Cheng et al. 2025).

Reinforcement Learning

Reinforcement Learning (RL), such as AC, PPO, formalizes sequential decision-making through Markov Decision Processes (Sutton et al. 1999; Schulman et al. 2017). Current advances introduce Group Relative Policy Optimization (GRPO), a PPO variant that enhances reasoning capabilities through group-wise advantage (Shao et al. 2024; Yang et al. 2025). RL now extensively trains foundation models through Deep Reinforcement Learning from Human Feedback(Christiano et al. 2017), which distills human values into reward signals, significantly reducing supervision costs while enabling complex behavior learning

(Bai et al. 2022; Gao, Schulman, and Hilton 2023). This RL paradigm also enables generative recommendation, OneRec and GFlowGR, to model lifecycle behaviors (Deng et al. 2025; Wang et al. 2025b). As a representative and focal approach, Process Reward Models are proposed to identify process errors by incorporating stepwise evaluations (Lightman et al. 2023), enabling finer-grained supervision on the reasoning process(Luo et al. 2024; Zhang et al. 2025b).

Preliminaries

Our work aims to recommend a learning path to maximize the promotion of a student on the learning targets with their answering history. Formally, given a student's learning sequence $\mathcal{H}_u = \{(h_1, a_1), \ldots, (h_n, a_n)\}$ and a target question set $\mathcal{T} \subseteq \mathcal{Q}$, where $\mathcal{Q}$ is the set of N questions, question $h_i \in \mathcal{Q}$, and $a_i \in \{0, 1\}$ is the correctness of the answer. The student can improve on the targets $\mathcal{T}$ after the optimal sequence $\mathcal{R}_u^* = (r_1, \ldots, r_k)$ within M steps from the path space $\mathcal{P} = \bigcup_{k=1}^{M} \mathcal{R}_k,\ \mathcal{R}_k = \{(r_1, \ldots, r_k) \mid r_i \in \mathcal{Q}\}$.

We quantify the task by Learning Path Effectiveness E_p, our goal is to identify the path that maximizes E_p:

$$\mathcal{R}_u^* = \arg\max_{\mathcal{R}_u \in \mathcal{P}}(E_p(\mathcal{R}_u)), \tag{1}$$

$$E_p(\mathcal{R}_u) = \frac{E_{\text{end}}(\mathcal{R}_u) - E_{\text{start}}}{E_{\text{sup}} - E_{\text{start}}}, \tag{2}$$

where E_{start} is the score at the start of the episode, E_{sup} denotes the full score, $E_{\text{end}}(\mathcal{R}_u)$ is the score after recommend $\mathcal{R}_u$, which can be predicted by Knowledge Tracing(KT). KT is to model students' behaviors with their learning history $\mathcal{H}_u$, to track knowledge level KL_t, and predict the correctness a of their next interaction by a scoring function:

$$E_{\text{end}}(\mathcal{R}_u) = \sum_{q_k \in \mathcal{T}} a_k = \sum_{q_k \in \mathcal{T}} \text{score}(KL_t, q_k), \tag{3}$$

$$KL_t = \text{KT}(\mathcal{H}_u^t). \tag{4}$$

After i_{th} step, update $\mathcal{H}_u^i = \mathcal{H}_u^{i-1} \cup \{(r_i, a_{n+i})\}$.

Environment Model (EM) is based on the pre-trained deep learning knowledge tracing (DKT) model (Piech et al. 2015) that first predicts KL_t and then **simulates a student's external score** E_s and $E_e(\mathcal{R}_u)$. To realistically simulate student behavior, EM generates answers a_{n+i} while restricting observable data to pre-/post-recommendation test scores and interaction responses, concealing internal state KL_t.

Reward Model (RM) is also a pre-trained DKT model that predicts and **provides the internal states** KL_t, which enables dense reward of each recommended question r_i by the student' evolving state of each step in offline training. At step i, we compute the process reward by following the task to quantify the improvement in the learning targets:

$$g_i = E_p(\mathcal{R}_u^i) - E_p(\mathcal{R}_u^{i-1}), \tag{5}$$

where $E_p(\mathcal{R}_u^i)$ denotes the current learning path effectiveness based on KL_i. RM provides step-wise feedback, which is deactivated during testing to simulate deployment online.

Methodology

Reinforcement Learning with Process Reward

In this work, we reformulate LPR as a Markov Decision Process with the following components:

State (S_t): The unified state S_t combines two fundamental dimensions to represent the complete learning recommendation context: (1) Knowledge level KL_t reflects evolving mastery levels of the student across different knowledge, continuously updated through learning accumulation. (2) Knowledge structure KS_t is the structural relationships between knowledge, the order of learning sequence $\mathcal{H}$ implicitly reflects the dynamic relationships between questions and the dependencies and requirements of the student (e.g., synergistic or transfer relationships).

Recommendation System (π_θ). In this work, we implement a recommendation system through a parameterized stochastic sampling policy to enable comprehensive exploration of the learning path space step by step. Formally, the policy recommending question r_t is formally defined as:

$$\pi_\theta(r_t|S_t) = p(r_t|KL_t, KS_t; \theta), \tag{6}$$

where θ denotes the trainable model parameters.

Environment. We emulate real-world student behaviors based on the environment model EM mentioned above, and the ultimate objective of our work is to maximize Learning Path Effectiveness within this student environment.

Reward Mechanism. We employ process rewards g_i generated by the reward model RM. We treat each individual's recommendation sequence as a personalized group and design a personalized advantage(PAdv) to mitigate inter-group discrimination while preserving intra-group personalization:

$$\tilde{g}_i = \frac{g_i - \mu_u}{\sigma_u + \epsilon_0}, \tag{7}$$

where g_i is the raw evaluation score for the r_i, $\mu_u = \frac{1}{k}\sum_{j=1}^{k} g_j$ and $\sigma_u = \sqrt{\frac{1}{k}\sum_{j=1}^{k}(\tilde{g}_j - \mu_u)^2}$ within a student's recommendation sequence, and $\epsilon_0 \to 0^+$.

UniLPR Recommendation System

We propose UniLPR model, as shown in Figure 2, we unify modeling students' learning accumulation and relationships with questions, and generate recommendations based on a unified representation of them.

Unified Sequence Encoding: We map and integrate students' performance and question sequences into a unified latent space. We first initialize question embedding matrix $\mathbf{X}_q \in R^{N \times d}$ and answer embedding matrix $\mathbf{X}_a \in R^{3 \times d}$. Specifically, $\mathbf{X}_a$ includes embeddings for the answering correctness [0, 1] and a hint action $\boldsymbol{a}_{[\text{REC}]}$ which signals the model to generate a recommendation question.

Given a student's history sequence $\mathcal{H}_u$ and target question $\mathcal{T}$, we reset the behavior sequence as $\mathcal{H}'_u = \{h_1, a_1, \ldots, h_n, a_n, r_1, a_{n+1}, \ldots r_i, a_{n+i}\}$, and encode it with embedding matrix forming as $\mathbf{X}_u = [\boldsymbol{x}_{h_1}, \boldsymbol{x}_{a_1}, \ldots, \boldsymbol{x}_{h_n}, \boldsymbol{x}_{a_n}, \boldsymbol{x}_{r_1}, \boldsymbol{x}_{a_{n+1}}, \ldots \boldsymbol{x}_{r_i}, \boldsymbol{x}_{r_{n+i}}]$, where $\boldsymbol{x}_h \in R^d$, $\boldsymbol{x}_a \in R^d$ and $\boldsymbol{x}_r \in R^d$. To integrate

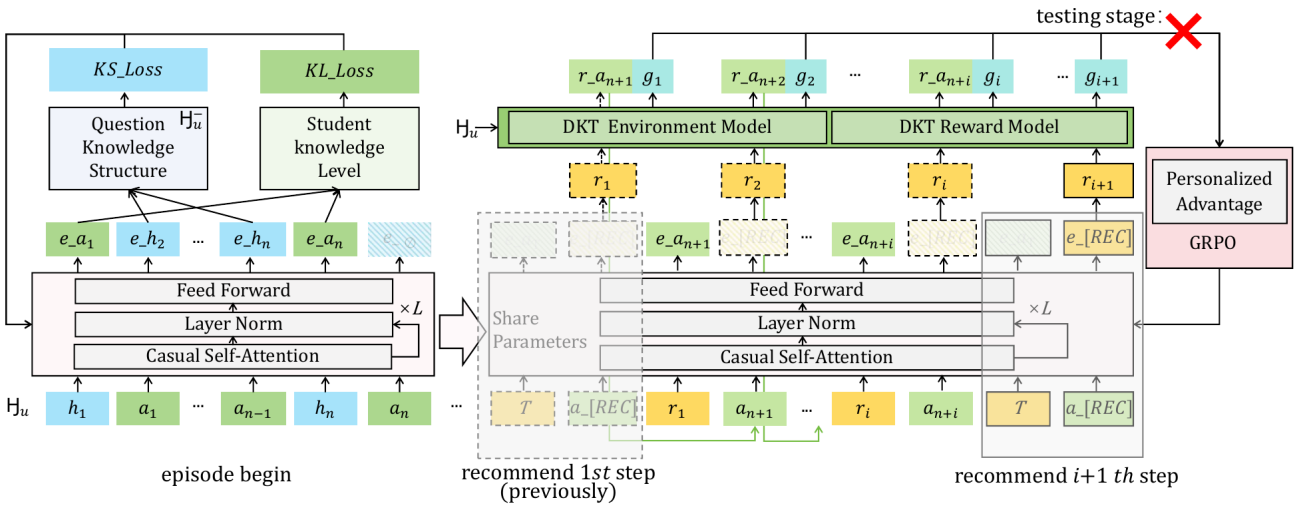

Figure 2: Framework overview. UniLPR employs a Transformer-based causal architecture. At the beginning of an episode, we unify modeling students' learning accumulation KL and relationships with questions KS through supervised learning. $1st$ step and $i+1$ th steps are as illustrated above, where the dashed box represents previously processed. UniLPR concatenates the embeddings of historical sequences, targets, and hint $a_{[\text{REC}]}$ to generate questions. DKT environment simulates student responses while the DKT reward model assesses questions through dense process rewards g_i, which is used to calculate the PAdv and to optimize the model through GRPO after the episode.

the learning targets and suggest the model to give a recommendation based on the targets, we compute targets representation by $\boldsymbol{x}_{\mathcal{T}} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\boldsymbol{x}_t$ and append it to the question sequence with the hint action $\boldsymbol{x}_{[\text{REC}]}$ to generate the sequence embedding $\mathbf{X}'_u \in R^{2(n+i+1)\times d}$:

$$\mathbf{X}'_u = \mathbf{X}_u \,||\, [\boldsymbol{x}_{\mathcal{T}}, \boldsymbol{x}_{[\text{REC}]}]. \tag{8}$$

We employ position embeddings to specify the position of each step in the question sequence, while preserving the order information of step-wise process rewards later. Without loss of generality, we annotate positions for answers,

$$\mathbf{X}_p = \left[\mathbf{P}_q^{(1)}, \mathbf{P}_a^{(1)}, \ldots, \mathbf{P}_q^{(n+i+1)}, \mathbf{P}_a^{(n+i+1)}\right], \tag{9}$$

where $\mathbf{P}_q^{(k)} \in R^d$, $\mathbf{P}_a^{(k)} \in R^d$ respectively encode the position of the k-th question and the position of the k-th answer.

The final representation combines the sequence embedding and the position embeddings:

$$\mathbf{X} = \mathbf{X}'_u + \mathbf{X}_p. \tag{10}$$

Joint Recommendation: The unified representation $\mathbf{X}$ consists of the student's learning accumulation KL and the dependencies on questions KS, which are fully used to make a joint recommendation. We process $\mathbf{X}$ through a Transformer decoder with causal masking. The transformer consists of L layers, each containing a self-attention mechanism and a feed-forward network:

$$\text{Attention}^{(l)}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \mathbf{M}\right)\mathbf{V}, \tag{11}$$

$$\mathbf{Z}^{(l)} = \text{LayerNorm}\left(\mathbf{X}^{(l-1)} + \text{Attention}^{(l-1)}\right), \tag{12}$$

$$\mathbf{X}^{(l)} = \text{GeLU}(\mathbf{Z}^{(l)}\mathbf{W}_1)\mathbf{W}_2, \tag{13}$$

where M is a causal mask matrix preserving the sequential dependency, $\mathbf{W}_1, \mathbf{W}_2 \in R^{d\times d}$ are trainable weights.

$$\mathbf{M}_{ij} = \begin{cases} 0 & i \geq j \\ -\infty & i < j \end{cases}.$$

The output of the transformer decoder is the final representation:

$$\mathbf{E} = \mathbf{X}^{(L)} = [\boldsymbol{e}_{a_1}, \boldsymbol{e}_{h_2}, \ldots \boldsymbol{e}_{r_i}, \boldsymbol{e}_{a_{n+i}}, \boldsymbol{e}_{\emptyset}, \boldsymbol{e}_{\mathcal{T}}, \boldsymbol{e}_{[\text{REC}]}],$$

In particular, $\mathbf{e}_{\emptyset}$ and $\mathbf{e}_{\mathcal{T}}$ represent the last answer a_{n+i} and the learning targets $\mathcal{T}$. Corresponding to the hint action $a_{[\text{REC}]}$, $\mathbf{e}_{[\text{REC}]}$ is the final representation used for question recommendation. We compute similarity scores across the entire question bank:

$$s(q) = \frac{\exp(\boldsymbol{e}_{[\text{REC}]}^\top \boldsymbol{x}_q/\tau)}{\sum_{q'\in\mathcal{Q}}\exp(\boldsymbol{e}_{[\text{REC}]}^\top \boldsymbol{x}_{q'}/\tau)} \quad , \forall q \in \mathcal{Q}\,, \tag{14}$$

where τ is a temperature parameter controlling distribution sharpness. The next recommended question r_{i+1} is then sampled from this probability distribution:

$$r_{i+1} \sim p(q) = \text{softmax}(s(q)). \tag{15}$$

Unified Optimization

At the beginning of the episode, input the raw historical sequence of a student, and then obtain the answer presentation and the question presentation:

$$\mathbf{E}_a = [\boldsymbol{e}_{a_1}, \boldsymbol{e}_{a_2}, \ldots, \boldsymbol{e}_{a_n}], \mathbf{E}_q = [\boldsymbol{e}_{h_2}, \ldots, \boldsymbol{e}_{h_n}]\,.$$

To enhance the model's capacity for modeling KL and KS, we introduce dual supervised tasks integrated with the UniLPR, (1) a binary cross-entropy (BCE) loss for student response prediction, and (2) a noise-contrastive estimation (NCE) loss on question representations to improve modeling sequential dependencies in the recommendation space:

$$\begin{aligned}\mathcal{L}_{\text{KL}} = -\frac{1}{n}\sum_{i=1}^{n}\Big[a_i \log(\sigma(\mathbf{W}_a \boldsymbol{e}_{a_i} + b_a)) \\ + (1-a_i)\log(1-\sigma(\mathbf{W}_a \boldsymbol{e}_{a_i} + b_a))\Big],\end{aligned} \tag{16}$$

where σ denotes the sigmoid function, $\mathbf{W}_a \in R^{1\times d}$ and $b_a \in R$ are learnable parameters,

$$\mathcal{L}_{\text{KS}} = -\frac{1}{n-1}\sum_{i=2}^{n}\log\frac{\exp(\boldsymbol{e}_{q_i}^{\top}\mathbf{x}_{h_i}/\tau)}{\sum_{k\in\mathcal{N}_i}\exp(\boldsymbol{e}_{h_i}^{\top}\mathbf{x}_{h_k}/\tau)}, \tag{17}$$

where τ is a parameter controlling distribution sharpness, $\mathcal{N}_i = \{h_i\} \cup \mathcal{H}_i^-$ contains one positive item and randomly sampled negatives $\mathcal{H}_i^- \subset \mathcal{Q} - \{h_i\}, |\mathcal{H}_i^-| = N_{neg}$.

We then optimize UniLPR, which serves as π_θ, using Group Relative Policy Optimization (GRPO) to leverage dense process rewards for precise question effectiveness evaluation. In each optimization epoch, we utilize the personalized advantage $\tilde{g}$ to maximize E_p:

$$\mathcal{L}_\pi = -E_{\mathcal{H}_k\sim p^\pi}\left[\min\left(\rho_k(\theta)\tilde{g}_k, \text{clip}(\rho_k(\theta), 1-\epsilon, 1+\epsilon)\tilde{g}_k\right)\right], \tag{18}$$

$$\rho_k(\theta) = \frac{\pi_\theta(r_{k+1}|\mathcal{H}_k)}{\pi_{\text{old}}(r_{k+1}|\mathcal{H}_k)}, \tag{19}$$

where π_{old} denotes the model used to compute action probabilities in (15), and π_θ is the current model being optimized.

We employ the Adam optimizer with learning rate scheduling to optimize. Before generating recommendations, we train the model using $\lambda_1\mathcal{L}_{KL} + \lambda_2\mathcal{L}_{KS}$ to establish robust unified state representations. After the episode, we optimize $\mathcal{L}_\pi$ to refine UniLPR.

Experiments

Datasets

Our experiments utilize public datasets Assist09 and Junyi. To pretrain EM and RM, we filter out questions with fewer than 5 answers and students with fewer than 5 responses. We also exclude cases where EM simulates a full score before recommendation. Table 1 shows the statistics of all datasets.

Dataset	student	question	interaction	avg length
Assist09	4.2k	123	525.5k	124.6
Junyi	198.5k	719	39,360.1k	198.3

Table 1: Statistics of all datasets.

Evaluation

Baseline Models We compare our method with the following baselines, which include vanilla reinforcement learning algorithms, current learning path recommendation models, and sequential recommendation models. We use grid search with cross-validation to match hyperparameters.

AC: Use a DKT to model students' learning states and vanilla actor-critic as recommender (Sutton et al. 1999).

PPO: Use a DKT to model students' learning states and vanilla PPO as recommender (Schulman et al. 2017).

CSEAL: Use a pre-trained DKT and a cognitive navigation algorithm to narrow the search space, and use vanilla AC to learn and update (Liu et al. 2019).

GEHRL-ST: Use a hierarchical reinforcement learning architecture that employs sub-tree pruning algorithms. The high-level policy is optimized using vanilla AC, and the low-level policy is optimized using PPO (Li et al. 2023).

GEHRL-EB: Similarly, use the node vector algorithm for pre-trained graph embeddings and narrow the search space based on distance (Li et al. 2023).

SRC: Employ a concept-aware encoder to optimize knowledge level along with knowledge structure, and an LSTM-based decoder with deduplication and greedy strategies, optimized by policy gradient (Chen et al. 2023).

GRU4Rec: A GRU-based sequential recommendation model (Hidasi and Karatzoglou 2018).

SASRec: Use a transformer-based sequential recommendation model with unidirectional causal self-attention (Kang and McAuley 2018).

Experimental Setting We first split the dataset by students into training and test sets with a ratio of 0.85 and 0.15. Then train for 5000 episodes and test on 1000 episodes, with 3 random seeds in each run. In the test phase, we stop updating model parameters and extend three patterns of learning targets. We pretrain the KT models following Piech (Piech et al. 2015), a more detailed task settings and implementation of EM and RM are in the code.

Recent Learning(rct): Following Liu(Liu et al. 2019), utilize the first 60% of the sequence to recommend for the last 20% and set steps range from $[5, 10, 20]$.

Historical Review(his): Represent the review of questions previously learned. Randomly select 10 questions in history sequence and set steps range from $[10, 20, 30]$.

Exploratory Learning(exp): Randomly select 10 questions from questions that the student has never met before and set steps range from $[10, 20, 30]$.

Evaluation Metrics We monitor convergence speed by scoring the average E_p of 3 runs during training and validate convergence by the last 10% of training episodes. We assess real-world effectiveness by the average E_p in the testing stage in Table 2.

Experiment Result

Table 2 demonstrates that UNO consistently outperforms all baseline methods across diverse tasks and experimental configurations. This superiority stems from its ability to effectively capture process rewards in the recommended

Pattern	Dataset	Steps	AC	PPO	CSEAL	GEHRL-ST	GEHRL-EB	SRC	GRU4Rec	SASRec	UNO
rct	Junyi	5	28.79 ± 0.98	29.08 ± 0.47	25.58 ± 0.16	24.48 ± 0.86	25.88 ± 0.98	16.04 ± 3.31	16.88 ± 2.77	26.98 ± 5.15	**34.78 ± 1.64**
		10	35.84 ± 0.85	37.64 ± 1.01	30.19 ± 0.61	33.03 ± 1.29	34.89 ± 1.00	17.00 ± 2.64	11.67 ± 2.95	25.98 ± 3.91	**46.01 ± 4.08**
		20	45.71 ± 0.81	44.88 ± 1.42	34.21 ± 1.03	39.94 ± 2.38	44.16 ± 0.43	15.67 ± 0.80	20.18 ± 6.77	38.20 ± 3.65	**54.05 ± 1.44**
	Assist09	5	29.65 ± 1.18	30.50 ± 2.25	25.49 ± 1.74	25.52 ± 1.53	26.95 ± 0.79	38.90 ± 4.12	15.61 ± 11.49	23.63 ± 9.69	**41.50 ± 0.21**
		10	32.24 ± 1.40	29.81 ± 1.25	28.04 ± 2.40	30.36 ± 2.39	33.99 ± 0.39	39.66 ± 3.04	16.77 ± 17.60	38.74 ± 2.14	**42.01 ± 0.20**
		20	30.51 ± 2.48	30.94 ± 1.71	29.50 ± 1.90	33.91 ± 2.02	32.18 ± 1.64	41.82 ± 0.00	20.48 ± 13.28	30.71 ± 5.85	**42.04 ± 0.41**
his	Junyi	10	11.03 ± 1.10	10.46 ± 3.16	4.17 ± 1.09	6.98 ± 0.59	7.29 ± 0.93	-2.58 ± 3.20	1.21 ± 2.75	10.65 ± 2.88	**29.19 ± 5.67**
		20	15.55 ± 0.91	17.30 ± 3.95	10.20 ± 0.80	12.37 ± 1.44	16.18 ± 0.66	-7.57 ± 13.35	11.77 ± 2.56	8.18 ± 4.90	**34.32 ± 2.48**
		30	19.71 ± 0.94	19.37 ± 1.98	10.93 ± 2.65	16.37 ± 2.52	18.84 ± 1.29	-5.20 ± 7.73	-4.61 ± 8.29	14.63 ± 3.33	**34.74 ± 3.88**
	Assist09	10	8.51 ± 6.42	8.77 ± 1.14	-1.48 ± 6.96	8.27 ± 2.17	11.82 ± 2.96	30.37 ± 1.94	5.10 ± 23.80	8.24 ± 8.92	**35.32 ± 0.59**
		20	10.47 ± 1.52	17.96 ± 8.12	1.08 ± 2.26	11.19 ± 1.35	12.12 ± 2.11	34.85 ± 0.00	8.23 ± 34.68	8.56 ± 14.15	**35.08 ± 0.02**
		30	9.41 ± 3.24	17.22 ± 1.18	-2.25 ± 1.99	10.27 ± 2.75	11.95 ± 1.84	23.22 ± 16.45	9.71 ± 25.62	1.05 ± 22.69	**35.15 ± 0.22**
exp	Junyi	10	0.73 ± 0.62	8.96 ± 0.63	-1.59 ± 0.71	0.59 ± 0.71	-0.08 ± 0.07	5.67 ± 1.34	3.12 ± 3.68	1.49 ± 1.21	**14.50 ± 2.27**
		20	0.36 ± 0.44	7.36 ± 2.68	-0.80 ± 0.85	0.75 ± 0.07	0.88 ± 0.23	7.85 ± 3.98	1.41 ± 1.31	0.83 ± 1.62	**16.17 ± 1.05**
		30	0.20 ± 0.50	8.03 ± 2.33	-1.92 ± 0.41	0.60 ± 0.86	0.67 ± 0.36	-6.82 ± 5.34	4.28 ± 3.57	0.22 ± 0.82	**16.40 ± 1.63**
	Assist09	10	-2.10 ± 0.70	-2.65 ± 1.54	-3.69 ± 0.63	-4.91 ± 1.75	-4.28 ± 1.09	7.31 ± 1.08	-5.31 ± 13.37	-2.88 ± 1.69	**7.95 ± 0.38**
		20	-1.27 ± 1.19	-1.89 ± 1.17	-3.31 ± 0.86	-4.25 ± 0.49	-2.66 ± 1.65	7.78 ± 0.62	-0.73 ± 4.17	-4.58 ± 5.92	**8.66 ± 0.00**
		30	-1.98 ± 0.92	-1.46 ± 1.01	-1.48 ± 4.84	-4.28 ± 0.81	-2.82 ± 1.08	8.22 ± 0.00	1.90 ± 5.33	-2.68 ± 0.36	**8.64 ± 0.02**

Table 2: Test performance comparison across different learning patterns, datasets, and steps (values are mean±std %).

learning paths, thereby guiding the recommendation process more precisely. By integrating students' evolving accumulation and dynamic associations with questions, UNO fully explores in the unified space of KS and KL, and achieves optimal performance in backtracking paths, local predictions, and large-scale global space searches. In contrast, existing LPR models exhibit limitations in personalized long-term and diverse learning scenarios, primarily due to the sparse outcome reward and reliance on static anonymous session-based path search strategies.

(1) AC and PPO models fail to learn any knowledge structure. They directly map all questions to high-dimensional spaces and choose paths through nearly identical and random weighted selection. While the sequential models underperform owing to the absence of LPR constraints, SASRec demonstrates relatively better performance than GRU4Rec, validating the adaptability of the Transformer.

(2) LPR models still face the inability to capture dense process rewards. By establishing hierarchical recommendation, GEHRL incorporates additional sub-session rewards, thereby demonstrating superior performance compared to CSEAL. But they all treat the session or sub-session as a unit. Some questions within the session may be highly effective, their impact is obscured by the aggregated sequence, making it difficult to identify truly beneficial questions.

(3) The poor performance of CSEAL and GEHRL-ST is also related to their reliance on heuristic path search on the static knowledge graph, which does not consider students' evolving knowledge accumulation and requirements. GEHRL-EB, utilizing node2vec graph neural encoding, provides more path opportunities. SRC jointly encodes KL and KS, and optimizes using outcome rewards through policy gradient, exploring a joint choice space. But when the targets and candidates increase substantially in the Junyi dataset, SRC degrades severely due to it generates a complete path in one time without dense reward.

Ablation Study

Effectiveness of Process Reward. To verify the effectiveness of the dense process rewards, we conduct 2 sets of experiments under 10 steps for Recent Learning (rct-10), 20 steps for Historical Review (his-20), and 20 steps for Exploratory Learning (exp-20), as is reported in Table 3.
(1) We apply UniLPR to AC and PPO with only sparse reward, revealing that the LPR task is highly dependent on the reward mechanism. (2) By replacing the backbone with SASRec and GRU4Rec, we find that incorporating process rewards significantly enhances their performance. This confirms that the reward model combined with GRPO forms an efficient RL framework. In addition, all tested models underperformed compared to UNO, demonstrating UniLPR's superior compatibility with GRPO.

Dataset	Exp	UniLPR+AC	UniLPR+PPO	GRU+GRPO	SAS+GRPO
Junyi	rct-10	36.12 ± 1.01	35.59 ± 0.41	18.64 ± 5.33	30.26 ± 3.61
	his-20	17.32 ± 0.70	17.35 ± 0.50	31.44 ± 0.12	26.83 ± 8.39
	exp-20	0.58 ± 0.33	-0.35 ± 0.53	2.13 ± 5.28	13.16 ± 4.63
assist09	rct-10	34.72 ± 1.14	33.79 ± 1.09	32.31 ± 8.10	40.07 ± 2.47
	his-20	13.81 ± 2.59	12.41 ± 2.74	19.98 ± 17.07	33.63 ± 0.76
	exp-20	-3.20 ± 0.12	-3.81 ± 0.63	2.16 ± 2.41	8.24 ± 1.95

Table 3: Study the effectiveness of process reward.

Stability of Varying Lengths of Historical Learning Sequence. To investigate the impact of unified modeling

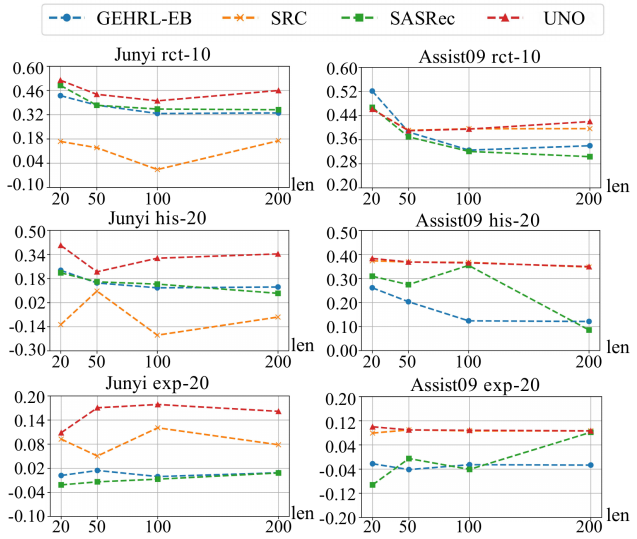

Figure 3: Stability of varying lengths of historical sequence.

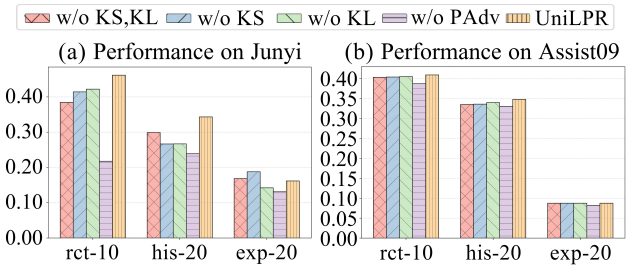

Figure 4: Ablation study of each optimization module.

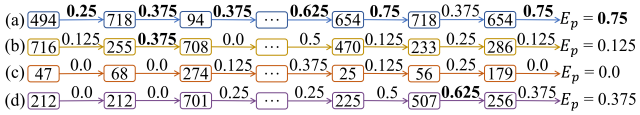

Figure 5: The process learning path effectiveness of student 29144 in the Junyi during rec-20 testing stage: comparison between (a) UNO, (b) AC, (c) CSEAL, and (d) GEHRL-EB.

on the recommendation in the lifelong learning scenario, we compare UNO with the best step-by-step LPR method GEHRL-EB, the multi-step model SRC, and the best sequential model SASRec. Experiments are conducted under configurations rec-10, his-20, exp-20, with history sequence length $len \in [20, 50, 100, 200]$, as illustrated in Figure 3. The results show that UNO outperforms all baselines in the vast majority of settings. While other models degrade or fluctuate as KL and KS complexity grow with increasing historical length, UNO maintains superior stability, demonstrating exceptional capabilities for modeling students' evolving accumulation and demands, and better adaptability for complex path exploration.

Optimization Module Ablation Study. In Figure 4, we conduct an ablation study to validate the effectiveness of joint optimization designs in UNO under standard configurations (rec-10, his-20, exp-20). The study involves examining the use of KS Loss, KL Loss, and Personalized Advantage(PAdv). We remove each module separately and refer to the remaining parts as w/o KS, KL, w/o KS, w/o KL, w/o PAdv. The ablation study results show that the complete UNO performs most stably when integrating and optimizing both the KS and KL. Treating them separately limits performance due to their weak individual associations with recommendation tasks, which proves that complete student state representation must dynamically combine both dimensions for capturing the evolving relation between KL accumulation and KS. The Junyi dataset further reveals that this implicit relationship between KL and KS becomes increasingly meaningful with larger data scales. Additionally, incorporating Personalized Advantage (PAdv) in the model significantly stabilizes model expression, enhances generalization ability, and makes the model more robust in rating extreme cases, leading to overall better performance.

Case Study. Figure 5 shows a case study, which illustrates how a student's learning efficiency is improved step by step. We compare with AC, CSEAL, and GEHRL-EB during the recommendation process. It can be observed that in this long-term learning process, due to the lack of process rewards, even if the models can accurately select some effective questions, they are still distracted by invalid questions during the recommendation process. This leads to the student's scores not only remaining low but also showing a downward trend. In contrast, UNO could offer precise guidance, recommending that students consolidate their existing learning level. It then supports students in achieving steady performance gains by continuously exploring and optimizing pathways throughout the learning process.

Conclusion

In this paper, we address the critical limitations of current learning path recommendation systems by proposing UNO, a novel unified offline training paradigm. We innovate dense process rewards provided by a KT-based reward model during offline training. We propose UniLPR to unify modeling long-term accumulation and evolving requirements for knowledge, which enables personalized path exploration. We also introduce GRPO to jointly optimize the recommendation model. Finally, the extended three learning patterns comprehensively validate the UNO's capability to address diverse lifelong learning demands.

Experimental results demonstrate that UNO outperforms existing methods across all evaluations, particularly in capturing long-term patterns and optimizing cognitive progression. This highlights the importance of process reward and unified representation in adaptive learning systems.

Several promising directions warrant further investigation, including dynamic temperature adjustment for better exploration and session aggregation or KV-cache for efficiency. These advancements could further bridge the gap between theoretical research and practical educational applications in the learning path recommendation.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. U2469205, the Fundamental Research Funds for the Central Universities of China under Grant No. JKF-20240769.

References

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Chen, X.; Shen, J.; Xia, W.; Jin, J.; Song, Y.; Zhang, W.; Liu, W.; Zhu, M.; Tang, R.; Dong, K.; et al. 2023. Set-to-sequence ranking-based concept-aware learning path recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 5027–5035.

Chen, Y.; Li, X.; Liu, J.; and Ying, Z. 2018. Recommendation system for adaptive learning. *Applied psychological measurement*, 42(1): 24–41.

Cheng, X.; Zhang, Z.; Wang, J.; Fang, L.; He, C.; Guan, Q.; Pan, S.; and Luo, W. 2025. Education-Oriented Graph Retrieval-Augmented Generation for Learning Path Recommendation. *arXiv preprint arXiv:2506.22303*.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Cover, T.; and Hart, P. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1): 21–27.

Deng, J.; Wang, S.; Cai, K.; Ren, L.; Hu, Q.; Ding, W.; Luo, Q.; and Zhou, G. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*.

Gao, L.; Schulman, J.; and Hilton, J. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, 10835–10866. PMLR.

Govindarajan, K.; Kumar, V. S.; and Kinshuk. 2016. Dynamic Learning Path Prediction — A Learning Analytics Solution. In *2016 IEEE Eighth International Conference on Technology for Education (T4E)*, 188–193.

Hidasi, B.; and Karatzoglou, A. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*, 843–852.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.

Kubotani, Y.; Fukuhara, Y.; and Morishima, S. 2021. Rltutor: Reinforcement learning based adaptive tutoring system by modeling virtual student with fewer interactions. *arXiv preprint arXiv:2108.00268*.

Li, Q.; Xia, W.; Yin, L.; Shen, J.; Rui, R.; Zhang, W.; Chen, X.; Tang, R.; and Yu, Y. 2023. Graph enhanced hierarchical reinforcement learning for goal-oriented learning path recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 1318–1327.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Liu, Q.; Tong, S.; Liu, C.; Zhao, H.; Chen, E.; Ma, H.; and Wang, S. 2019. Exploiting cognitive structure for adaptive learning. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 627–635.

Liu, Z.; Liu, Q.; Chen, J.; Huang, S.; Gao, B.; Luo, W.; and Weng, J. 2023. Enhancing deep knowledge tracing with auxiliary tasks. In *Proceedings of the ACM Web Conference 2023*, 4178–4187.

Luo, L.; Liu, Y.; Liu, R.; Phatale, S.; Guo, M.; Lara, H.; Li, Y.; Shu, L.; Zhu, Y.; Meng, L.; et al. 2024. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*.

Nabizadeh, A. H.; Gonçalves, D.; Gama, S.; Jorge, J.; and Rafsanjani, H. N. 2020. Adaptive learning path recommender approach using auxiliary learning objects. *Computers & Education*, 147: 103777.

Nasrin, A.; Qian, L.; Obiomon, P.; and Dong, X. 2025. Enhancing Learning Path Recommendation via Multi-task Learning. *arXiv preprint arXiv:2507.05295*.

Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L. J.; and Sohl-Dickstein, J. 2015. Deep knowledge tracing. *Advances in neural information processing systems*, 28.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Wang, C.; and Sahebi, S. 2023. Continuous personalized knowledge tracing: Modeling long-term learning in online environments. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, 2616–2625.

Wang, T.; Zhan, Y.; Lian, J.; Hu, Z.; Yuan, N. J.; Zhang, Q.; Xie, X.; and Xiong, H. 2025a. Llm-powered multi-agent framework for goal-oriented learning in intelligent tutoring system. In *Companion Proceedings of the ACM on Web Conference 2025*, 510–519.

Wang, Y.; Zhou, S.; Lu, J.; Liu, Q.; Li, X.; Zhang, W.; Li, F.; Wang, P.; Xu, J.; Zheng, B.; et al. 2025b. GFlowGR: Fine-tuning Generative Recommendation Frameworks with Generative Flow Networks. *arXiv preprint arXiv:2506.16114*.

Wu, C.-Y.; Ahmed, A.; Beutel, A.; Smola, A. J.; and Jing, H. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, 495–503.

Xu, Y.; Zhang, W.; Lin, X.; and Zhang, Y. 2025a. UniDyG: A Unified and Effective Representation Learning Approach for Large Dynamic Graphs. *IEEE Transactions on Knowledge and Data Engineering*.

Xu, Y.; Zhang, W.; Zhang, Y.; Orlowska, M.; and Lin, X. 2024. TimeSGN: Scalable and effective temporal graph neural network. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 3297–3310. IEEE.

Xu, Y.; Zhang, W.; Zhang, Y.; Xu, X.; and Lin, X. 2025b. Fast and accurate temporal hypergraph representation for hyperedge prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 1727–1738.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yin, H.; Sun, Z.; Sun, Y.; and Huang, G. 2021. Automatic learning path recommendation for open source projects using deep learning on knowledge graphs. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, 824–833. IEEE.

Yun, Y.; Dai, H.; An, R.; Zhang, Y.; and Shang, X. 2024. Doubly constrained offline reinforcement learning for learning path recommendation. *Knowledge-Based Systems*, 284: 111242.

Zhai, J.; Liao, L.; Liu, X.; Wang, Y.; Li, R.; Cao, X.; Gao, L.; Gong, Z.; Gu, F.; He, J.; Lu, Y.; and Shi, Y. 2024. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 58484–58509. PMLR.

Zhang, H.; Shen, S.; Xu, B.; Huang, Z.; Wu, J.; Sha, J.; and Wang, S. 2024. Item-difficulty-aware learning path recommendation: From a real walking perspective. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4167–4178.

Zhang, L.; Song, K.; Lee, Y. Q.; Guo, W.; Wang, H.; Li, Y.; Guo, H.; Liu, Y.; Lian, D.; and Chen, E. 2025a. Killing two birds with one stone: Unifying retrieval and ranking with a single generative recommendation model. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2224–2234.

Zhang, Z.; Zheng, C.; Wu, Y.; Zhang, B.; Lin, R.; Yu, B.; Liu, D.; Zhou, J.; and Lin, J. 2025b. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*.

Zhou, Y.; Huang, C.; Hu, Q.; Zhu, J.; and Tang, Y. 2018. Personalized learning full-path recommendation model based on LSTM neural networks. *Information sciences*, 444: 135–152.

Zhu, H.; Tian, F.; Wu, K.; Shah, N.; Chen, Y.; Ni, Y.; Zhang, X.; Chao, K.-M.; and Zheng, Q. 2018. A multi-constraint learning path recommendation algorithm based on knowledge map. *Knowledge-Based Systems*, 143: 102–114.