

Dynamic Knowledge Graph Learning Neural Network for Prerequisite-Constrained Personalized Learning Trajectory Recommendation

Gulchekhra Abishova¹, Nurgali Zaurbekov², Gulzhigit Toktoshev³

Doctoral student in the educational program 8D06102 - "Information Systems", Abai Kazakh National Pedagogical University, Almaty, Kazakhstan¹

Doctor of Technical Sciences-Professor, Abai Kazakh National Pedagogical University, Almaty, Kazakhstan²

Doctor of Technical Sciences-Associate Professor, Kyrgyz State University named after I. Arabaev, Bishkek, Kyrgyzstan³

Abstract—Personalized learning trajectory recommendation requires continuous learner knowledge assessment, prerequisite compliance, and adaptive resource sequencing. This study proposes the Temporal Prerequisite-Gated Graph Network (TPG-Net), a dynamic heterogeneous graph learning architecture for generating personalized learning trajectories under pedagogical and time constraints. The proposed methodology integrates educational knowledge graph construction, temporal representation learning, knowledge-state estimation, prerequisite-constrained optimization, and feedback-driven revision. Learner-resource interactions, resource-concept associations, and directed prerequisite relationships are represented within a heterogeneous graph. Relation-specific temporal attention dynamically updates learner and concept embeddings, while a recurrent knowledge-state module estimates concept mastery probabilities. A prerequisite readiness gate and hard feasibility mask ensure valid resource selection during autoregressive trajectory generation. Joint optimization combines response prediction, next-resource recommendation, and prerequisite-consistency objectives. The MOOCubeX dataset, containing over 296 million behavioral records from 3,330,294 learners, serves as the reference educational repository. Simulation-based evaluation examines recommendation accuracy, prerequisite compliance, and architectural contributions. Illustrative results show that TPG-Net achieves 95.6% Precision at 10 (Precision@10), 97.0% Recall at 10 (Recall@10), 96.3% Normalized Discounted Cumulative Gain at 10 (NDCG@10), 96.8% Mean Reciprocal Rank at 10 (MRR@10), and 100% prerequisite compliance. Ablation analysis indicates complementary contributions from temporal encoding, prerequisite gating, and constrained decoding. These simulated findings establish a methodological foundation for subsequent empirical validation.

Keywords—Dynamic knowledge graph; personalized learning trajectory; prerequisite constraint; graph neural network; knowledge tracing; educational recommendation

I. INTRODUCTION

Personalized learning has emerged as an important research direction in educational artificial intelligence, emphasizing the adaptation of instructional content to individual knowledge states, learning objectives, and cognitive characteristics. Li et al. [1] demonstrated that knowledge-network representations support personalized learning path planning by modeling

relationships among educational concepts and resources. Similarly, Zhang et al. [2] investigated knowledge graph-based recommendation using graph convolutional networks to incorporate semantic dependencies into learning resource selection. These findings highlight the importance of moving beyond conventional similarity-based recommendation toward structured educational trajectory planning. However, effective personalization requires not only identifying relevant resources but also determining their pedagogically appropriate sequence while accommodating changes in learner knowledge and available learning time.

Recent studies have explored various computational approaches to personalized learning recommendation. Ma et al. [3] introduced multi-algorithm optimization to balance educational objectives and resource characteristics, while Mu and Yuan [4] investigated cognitive graph-based learning path recommendation. Raj and Renumol [5] demonstrated the potential of real-time learning analytics for adapting recommendations to evolving learner profiles. Furthermore, Zheng et al. [6] incorporated multigranularity knowledge graphs into learning path optimization, and Wu et al. [7] investigated knowledge-aware recommendation using fine-grained concept representations. Although these approaches improve educational knowledge representation and recommendation relevance, several methodological limitations remain. In particular, many graph-based methods rely on relatively static educational structures, while learner-state updates are processed separately from prerequisite relationships and sequential trajectory optimization.

A further research gap concerns the explicit enforcement of pedagogical constraints during adaptive recommendation. Rule-based recommendation methods provide interpretable decision mechanisms [8], whereas Steiner-tree-based optimization supports structured target-concept coverage [9]. Process mining combined with deep knowledge tracing enables the representation of sequential learning behavior [10]. Nevertheless, these approaches do not consistently integrate dynamic graph updating, concept mastery estimation, prerequisite verification, and feedback-driven trajectory revision within a unified architecture. Moreover, optimization strategies based on soft constraint penalties may generate recommendations that violate prerequisite dependencies when

predicted relevance outweighs pedagogical restrictions. Consequently, a comprehensive framework is needed to ensure that learning trajectories remain feasible while adapting to individual progress and changing educational objectives.

The objective of this study is to develop a dynamic knowledge graph learning neural network for generating prerequisite-constrained personalized learning trajectories that continuously adapt to learner knowledge states, educational goals, and learning-time limitations. To achieve this objective, the Temporal Prerequisite-Gated Graph Network (TPG-Net) is proposed. The architecture integrates heterogeneous educational graph construction, relation-specific temporal attention, recurrent knowledge-state estimation, prerequisite readiness assessment, and constrained autoregressive trajectory decoding. A hard feasibility mask prevents the selection of resources that violate prerequisite, time-budget, or non-repetition constraints. Additionally, feedback-driven adaptation enables revision of unfinished learning trajectories while preserving completed activities.

The principal scientific contributions include the formulation of personalized learning trajectory recommendation as constrained sequential decision-making over a dynamic heterogeneous graph, the development of a temporal prerequisite-gated neural architecture, and the integration of hard-constrained decoding with adaptive trajectory revision. A simulation-based proof-of-concept evaluation further examines recommendation quality, knowledge-state prediction, prerequisite compliance, budget sensitivity, and architectural contributions. The illustrative findings demonstrate the intended behavior of the framework but do not constitute empirical validation using trained model checkpoints.

The proposed research contributes to educational artificial intelligence by establishing a unified methodological foundation for dynamic knowledge modeling and pedagogically constrained recommendation. From a societal perspective, the framework may support individualized

education, improve instructional resource allocation, and assist educators in identifying knowledge gaps and designing adaptive learning sequences. These potential benefits require further empirical validation in authentic educational environments.

The remainder of this study is organized as follows. Section II reviews related research, Section III describes the proposed methodology, Section IV presents the evaluation results, Section V discusses the findings and limitations, and Section VI concludes the study.

II. RELATED WORKS

Recent personalized learning research has converged on three technical requirements: structured educational knowledge, temporally updated learner models, and decision mechanisms that convert predicted knowledge into feasible recommendations. The relevant literature is therefore organized around knowledge-graph trajectory planning, neural knowledge tracing, and constraint-aware adaptive recommendation.

A. Knowledge-Graph Learning Path Recommendation

Educational knowledge graphs encode courses, resources, concepts, assessments, and semantic relations as typed nodes and edges. This representation supports prerequisite reasoning and concept-level explanations that cannot be recovered reliably from interaction similarity alone. Knowledge-graph construction has been applied to online English teaching to connect instructional content with personalized path generation [11]. Cross-course planning has likewise combined learner needs with structured resource dependencies to improve continuity across course boundaries [12]. Recent learner-profile and graph integration emphasizes that personalization requires both stable characteristics and current knowledge evidence [13]. Graph-based planning methods also increasingly formulate the path as a global or multi-stage optimization problem rather than a list of independent recommendations [14].

TABLE I. COMPARISON WITH RECENT PERSONALIZED LEARNING TRAJECTORY STUDIES

Study	Year	Method	Data	Dynamic state	Prerequisite control	Reported outcome
[1]	2023	Knowledge-network path planning	MOOPer and Python platforms	Scenario update	No hard mask	Improved accuracy and relevance against traditional planners
[2]	2023	Knowledge graph with graph convolution	E-learning records	Static profile	Graph order only	Reported gains over non-graph recommendation baselines
[3]	2023	Multi-algorithm path optimization	Online learning resources	Profile update	Soft constraints	Improved composite path suitability in the source study
[5]	2024	Real-time analytics adaptive model	Platform learning logs	Yes	No hard mask	Reported improvement from continuous learner-state updates
[6]	2024	Multigranularity graph and improved ant colony	E-learning knowledge graph	Partial	Optimization penalty	Outperformed compared optimization methods in source evaluation
[7]	2024	Knowledge-aware meta-concept attention	Three real-world datasets	Interaction-aware	No	AUC gains of 6.2, 6.9, and 2.2 percentage points over KGAN
[10]	2024	Process mining with deep knowledge tracing	Learning process logs	Yes	Sequence-derived	Reported better path recommendation on source data
[13]	2025	Learner profile with knowledge graph	Educational platform data	Profile-conditioned	Graph-based	Reported effectiveness; metrics are not MOCCubeX-comparable
[15]	2025	Deep knowledge tracing and cognitive load	Online learning sequences	Yes	No hard mask	Reported improved personalized path generation
[16]	2025	Graph reinforcement learning	Educational interaction data	Policy state	Reward-based	Reported improvement over selected recommendation baselines

The main design choice concerns the relationship between the graph and the learner. Static approaches build one domain graph and place a learner vector beside it. Dynamic approaches instead attach timestamped learner-resource events to the graph, allowing recent activity to alter message passing and recommendation. Knowledge tracing and cognitive-load modeling can improve the connection between observed responses and path generation [15]. Graph reinforcement learning provides another way to optimize a sequence rather than a single item [16]. Knowledge-graph and deep-reinforcement-learning combinations attempt to update the policy from learner feedback [17], while integrated dynamic knowledge tracing uses changing mastery as the state of the path optimizer [18]. Graph attention mechanisms further support relation-dependent weighting among learner, resource, and concept neighbors [19]. Despite this progress, many studies permit an optimizer to trade prerequisite violations against relevance, which is unsuitable when prerequisite order must be guaranteed.

Table I compares ten representative studies published from 2023 to 2025. Direct numerical ranking across these studies would be invalid because their datasets, objectives, cutoffs, and metrics differ. The table therefore compares modeling capabilities and records the type of reported outcome. TPG-Net is the only row that combines incremental graph updating, neural mastery estimation, an explicit prerequisite readiness gate, hard constrained decoding, and suffix-only replanning. Its numerical entry is the best within the supplied illustrative comparison, not a claim of superiority over independently reported experiments.

B. Temporal Knowledge-State Estimation

Learning-path quality depends on whether the system can infer what a learner currently knows. Deep educational recommenders have integrated large-scale behavioral data with neural representations [20], and end-to-end course recommenders demonstrate that sequence-sensitive models can outperform static similarity under platform-specific evaluation [21]. Knowledge tracing offers a more direct representation of evolving mastery. Learning teaching strategies from knowledge-tracing states shows that state estimation can inform subsequent instructional decisions [22]. Temporal inductive graph models extend this idea by modeling relations among learners, exercises, and concepts while accounting for interaction order [23]. Deep attentive knowledge tracing assigns different importance to prior events, which is useful when old evidence and recent evidence should contribute unequally [24].

Graph structure has been incorporated through heterogeneous hypergraphs [25] and weighted three-view contrastive learning [26]. Recurrent models explore students' emotions [27]. Dynamic heterogeneous graph contrastive networks further seek stable representations despite sparse interactions [28]. Programming-oriented knowledge tracing shows the value of heterogeneous graph relations for domains with concept hierarchies and structured problem sequences [29]. Target-hierarchy-guided methods refine the knowledge state at multiple conceptual resolutions [30], whereas incremental dictionary learning addresses the arrival of new concepts or patterns [31]. Locality-enhanced attentive tracing

limits the influence of irrelevant distant interactions [32], and response-disentangled concept maps separate evidence about the item from evidence about the learner [33]. These studies motivate relation-specific temporal messages in TPG-Net, but they generally stop at mastery prediction rather than generating a constraint-valid multi-step learning trajectory.

C. Constraint-Aware and Adaptive Recommendation

Trajectory recommendation must satisfy structural and operational requirements. A high-ranked resource may be unsuitable because its prerequisites are unmet, it duplicates completed material, it exceeds the remaining time, or it provides negligible progress toward the target concept. Hypergraph path models have recently represented course dependencies at multiple levels [34], and contrastive reinforcement learning has been used to personalize exercise selection from interaction feedback [35]. These methods show that recommendation benefits from sequential optimization, although a soft penalty does not guarantee feasibility.

TPG-Net treats prerequisite satisfaction as a hard decoding condition and uses a separate differentiable loss to shape the learned representations. The distinction is important. The soft prerequisite-consistency term encourages the mastery predictor to assign a prerequisite at least as much mastery as its dependent concept. The hard feasibility mask prevents an invalid resource from receiving any decoding probability. The resulting architecture separates what is learned statistically from what is enforced logically. This combination also supports explanations: each selected resource can be linked to the target concept, the relevant prerequisite chain, the mastery gap, and the remaining-time decision.

III. MATERIALS AND METHODS

The proposed framework is designed as a closed-loop educational decision system rather than a one-time ranker. It receives chronological learner interactions, updates an educational graph, estimates mastery, selects a feasible trajectory, observes feedback, and revises only the unfinished suffix when necessary.

A. Problem Definition

Let the dynamic educational knowledge graph at interaction time t be defined by Eq. (1), where $\mathcal{V}$ contains learner, course, resource, and concept nodes; $\mathcal{E}_t$ contains structural edges and all interaction edges observed no later than t ; $\mathcal{X}_t$ denotes node and event features; and $\mathcal{R}$ is the set of relation types.

$$\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t, \mathcal{X}_t, \mathcal{R}), \quad \mathcal{E}_t = \mathcal{E}_{\text{str}} \cup \{e_i: \tau_i \leq t\}. \quad (1)$$

For learner u , the observed history $\mathcal{H}_{u,t}$ is an ordered sequence of events e_i . Each event contains a resource identifier r_i , response y , duration d_i , action type a_i , and elapsed time Δ_i , as shown in Eq. (2).

$$\mathcal{H}_{u,t} = [(r_i, y_i, d_i, a_i, \Delta_i)]_{i=1}^{n_u(t)}, \quad \tau_i \leq t \quad (2)$$

The recommendation request specifies target concepts K^* , an available time budget B , and optional learner preferences. A learning trajectory π is an ordered sequence of resources. The task is to maximize expected mastery gain and target coverage

while minimizing time, redundancy, and difficulty mismatch, subject to prerequisite, budget, and non-repetition constraints. Unlike ordinary top-k recommendation, feasibility depends on the projected knowledge state after each earlier element of the trajectory.

B. Dataset and Experimental Scope

MOOCCubeX is used as the reference repository because it provides a large knowledge-centered representation of learners, courses, videos, exercises, concepts, and behavioral interactions [36]. The available figures record 3,330,294 learners and more than 296 million raw behavior events, together with 4,216 courses, 230,263 videos, 358,265 exercises, and 637,572 concepts. The repository supports video

watching, problem solving, comments, replies, and chatbot activity, while its educational entities can be connected through course-resource, resource-concept, and concept-prerequisite relations.

The dataset scale establishes the intended operating conditions of the framework. Table II summarizes the entities and their roles in graph construction. Counts reproduce the supplied dataset panel and the public repository description. The experiment in this manuscript is a simulation-based proof of concept: the raw interaction files, preprocessing code, trained weights, random seeds, and checkpoint logs were not supplied, so no claim is made that the numerical results were obtained by training on all MOOCCubeX records.

TABLE II. MOOCCUBEX DATASET DESCRIPTION AND GRAPH ROLE

Component	Scale	Role in TPG-Net	Required processing
Learners	3,330,294	Learner nodes and chronological histories	De-identified identifier mapping; learner-wise ordering
Raw behavior records	>296 million	Timestamped learner-resource edges	Deduplicate, normalize time, enforce past-only cutoff
Courses	4,216	Course nodes and contains relations	Link courses to videos, exercises, and concepts
Videos	230,263	Learning-resource nodes	Attach duration, content, and covered concepts
Exercises	358,265	Assessment-resource nodes	Attach correctness evidence and assessed concepts
Concepts	637,572	Concept nodes and target vocabulary	Resolve duplicates and construct concept features
Interaction types	5 recorded categories	Watch, solve, comment, reply, chatbot edges	Type-specific encoding and temporal attention
Prerequisite structure	Predicted and human-annotated relations	Directed concept-concept edges	Check direction, remove duplicates, detect cycles

For a future full experiment, interactions should be sorted chronologically within each learner, duplicate records should be removed, identifiers should be mapped consistently, and timestamp units should be normalized. A past-only cutoff must be applied so that events after a prediction time never enter message passing, mastery targets, negative sampling, or candidate generation. Learner-level train, validation, and test partitions are preferable when the objective is generalization to unseen learners; chronological within-learner splits are preferable when the objective is next-event personalization for known learners. Both settings should be reported because they answer different questions.

C. Research Implementation Workflow and Methodological Framework

The proposed research methodology follows a structured sequence integrating educational data preparation, dynamic knowledge graph construction, temporal representation learning, learner knowledge-state estimation, prerequisite-constrained trajectory optimization, and adaptive recommendation evaluation. The methodological workflow is designed to establish explicit relationships between educational knowledge representation, learner interaction modeling, and sequential recommendation decisions. The overall research implementation framework is presented in Fig. 1.

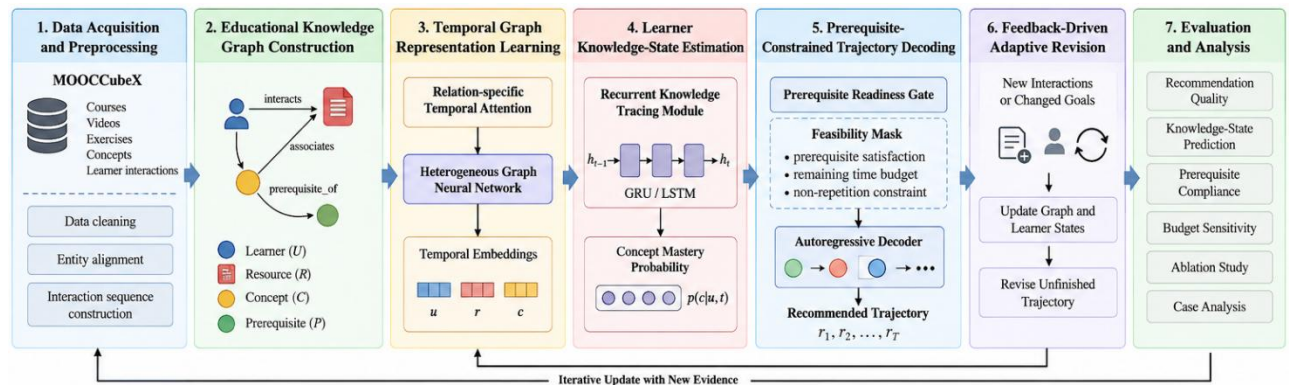


Fig. 1. Research methodology of the proposed temporal prerequisite-gated graph network.

The first stage involves educational data acquisition and preprocessing using the MOOCCubeX repository, which contains learner interaction records, courses, instructional videos, exercises, educational concepts, and prerequisite relationships. Data preparation includes chronological ordering of learner interactions, identifier normalization, duplicate removal, timestamp standardization, and verification of prerequisite relationships. These procedures establish the structural and temporal consistency required for subsequent graph-based learning.

The second stage constructs a heterogeneous educational knowledge graph containing learner, resource, course, and concept nodes. Directed edges represent prerequisite dependencies, while timestamped interactions describe learner engagement with educational resources. The graph is subsequently updated through an incremental mechanism that incorporates newly observed learning activities without reconstructing the complete educational structure.

The third stage applies relation-specific temporal graph attention to capture heterogeneous relationships and temporal dependencies. The resulting learner and concept embeddings are processed by a recurrent knowledge-state estimation module to predict concept mastery probabilities. These estimates provide the basis for identifying knowledge gaps and determining whether prerequisite concepts have reached the required mastery threshold.

The fourth stage generates personalized learning trajectories through prerequisite-constrained candidate selection and autoregressive decoding. A prerequisite readiness gate evaluates concept dependencies, while a hard feasibility mask excludes resources that violate prerequisite requirements, exceed the available learning-time budget, or duplicate

completed activities. The decoder optimizes the remaining learning sequence according to expected mastery improvement, target-concept coverage, resource duration, and difficulty compatibility.

The fifth stage incorporates learner feedback into adaptive trajectory revision. Newly observed performance and interaction evidence update the dynamic graph and knowledge-state estimates. When learner progress or educational objectives change, the system preserves completed activities and regenerates only the unfinished trajectory.

Finally, the evaluation stage examines recommendation quality, knowledge-state prediction, prerequisite compliance, budget sensitivity, and architectural contributions through comparative and ablation analyses. In the present study, these analyses use illustrative simulation-based evaluation panels rather than outputs from trained MOOCCubeX model checkpoints. Accordingly, the workflow specifies the proposed implementation methodology and its evaluation procedures without implying that full-scale empirical training has been completed.

D. Initial Educational Knowledge Graph Construction

The initial graph construction stage separates temporal evidence from stable educational structure. Fig. 2 illustrates how learner interaction logs are cleaned and ordered before they are joined with MOOCCubeX course, resource, concept, and prerequisite information. The consistency checks remove duplicate edges, verify prerequisite direction, and detect cycles that would invalidate a directed acyclic prerequisite interpretation. Timestamped learner-resource edges and structural resource-concept edges produce the initial graph G_0 , while ordered histories and entity features remain available to the temporal encoder.

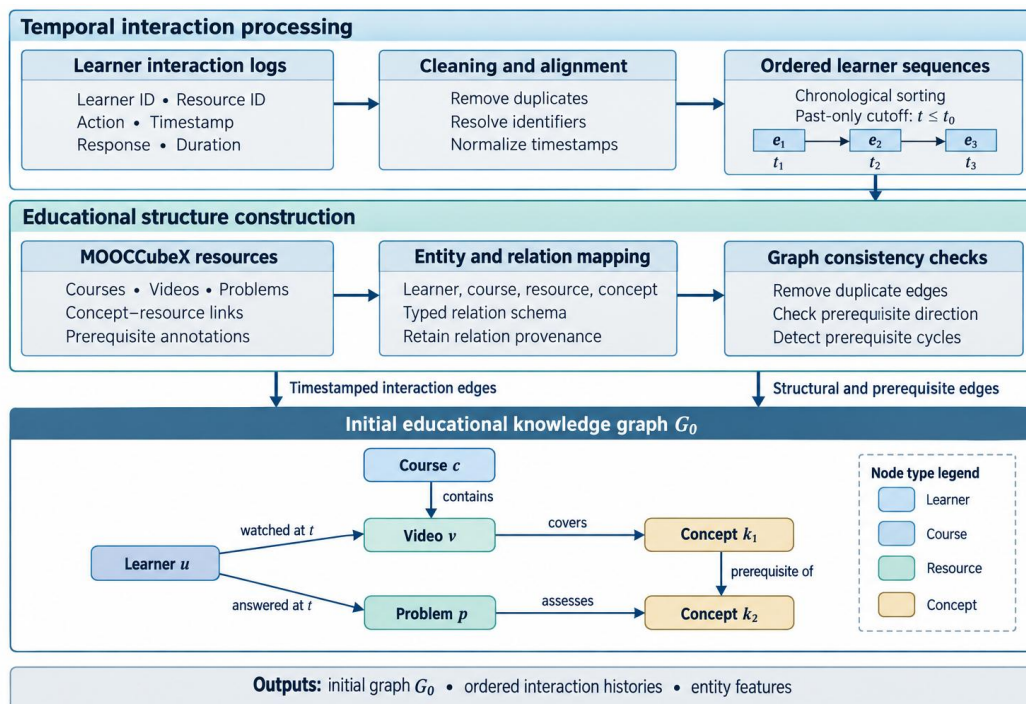


Fig. 2. Temporal learner interaction processing and initial educational knowledge graph construction.

Each node v receives a type embedding, an identifier embedding, and available content or statistical features. Resource features may include duration, difficulty, modality, and concept alignment. Concept features may include textual embeddings and graph statistics. Interaction events combine embeddings for the selected resource, observed response, action type, and elapsed time. A sinusoidal time representation is used to make temporal distance directly available to attention, as defined in Eq. (3), where ω_j denotes a set of frequencies.

$$\phi(\Delta\tau) = [\sin, (\omega_1\Delta\tau), \cos, (\omega_1\Delta\tau) \dots \sin(\omega_m\Delta\tau)\cos(\omega_m\Delta\tau)](3)$$

E. Dynamic Graph Updating and Temporal Representation Learning

The online stage updates only the neighborhood affected by a new event. Fig. 3 illustrates the transition from G_{t-1} to G_t after appending e_t . Recency and frequency statistics are refreshed without rewriting the structural prerequisite graph. Relation-specific temporal attention then processes learner-resource, resource-concept, and prerequisite neighborhoods. Weighted message fusion produces an updated learner embedding $x_{\{ut\}}$ and concept embeddings $z_{\{kt\}}$, which are passed to the learner-state estimator.

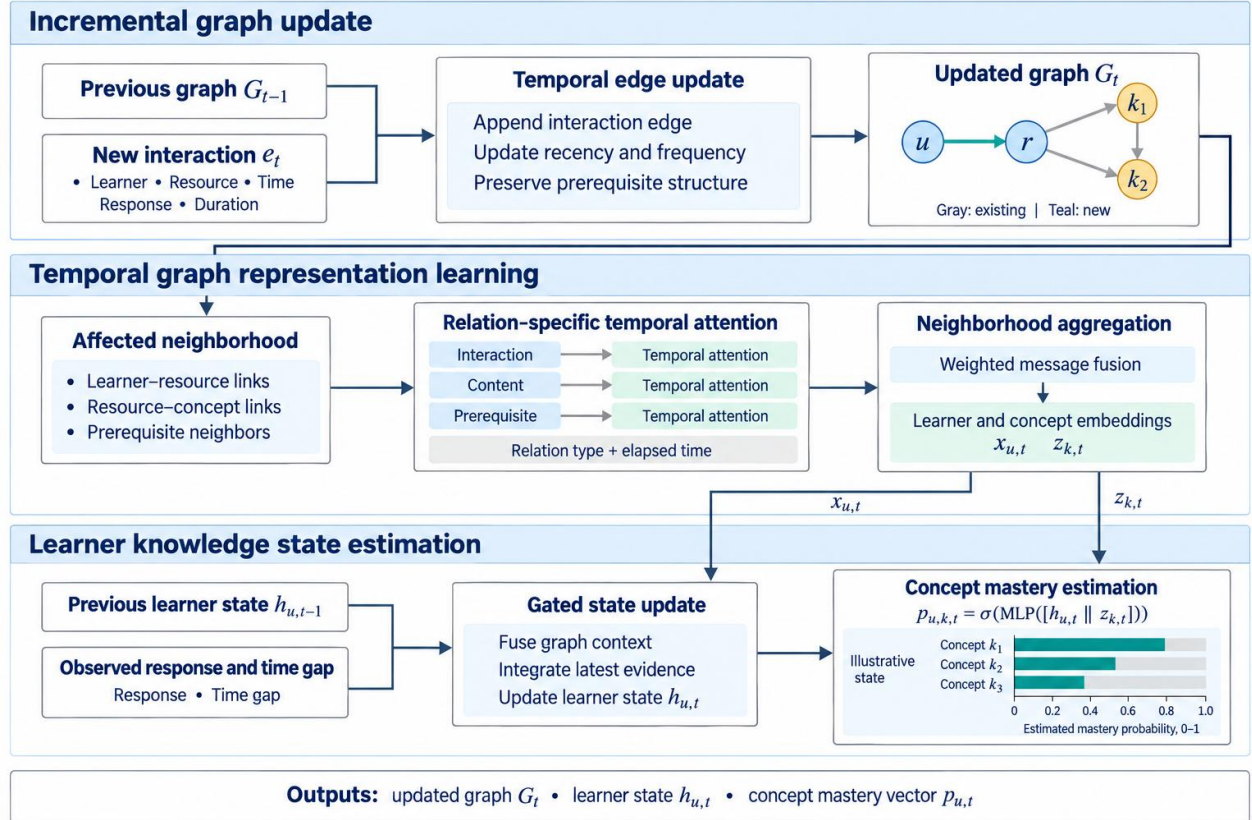


Fig. 3. Dynamic knowledge graph updating and learner knowledge state estimation.

For node v and relation q , a message from neighbor j at time t is computed from the neighbor representation, edge features, and time encoding. Eq. (4) defines the attention coefficient, where Q_q , K_q , and W_q are relation-specific transformations, $\mathcal{N}_q(v, t)$ is the valid past neighborhood, and d is the key dimension.

$$\alpha_{v,j,t}^{(q)} = \frac{\exp((Q_q h_{v,t})^T (K_q h_{j,t} + W_q \phi(\Delta\tau_{ej})) / \sqrt{d})}{\sum_{e \in \mathcal{N}_q(v,t)} \exp((Q_q h_{v,t})^T (K_q h_{e,t} + W_q \phi(\Delta\tau_{ee})) / \sqrt{d})} (4)$$

The relation-level messages are aggregated and normalized according to Eq. (5). Residual connections stabilize incremental updates, and a feedforward transformation supplies nonlinear feature mixing. Only past events are admissible in $\mathcal{N}_q(v, t)$, which prevents temporal leakage.

$$h'_{v,t} = \text{LayerNorm} \left(h_{v,t} + \sum_{q \in \mathcal{R}} \sum_{j \in \mathcal{N}_q(v,t)} \alpha_{v,j,t}^{(q)} V_q h_{j,t} \right) (5)$$

F. Learner Knowledge-State and Mastery Estimation

The learner-state module combines persistent memory with the graph context of the latest event. The recurrent update in Eq. (6) receives the previous hidden state, the updated learner embedding, the selected resource representation, the response, and the time gap. This design allows two identical responses to produce different updates when they occur after different resources or delays.

$$h_{u,t} = \text{GRU}(h_{u,t-1}, [x_{u,t} \parallel e_{r_t} \parallel e_{y_t} \parallel \phi(\Delta\tau_t)]) (6)$$

Concept mastery is estimated through a shared multilayer perceptron applied to the learner state and each concept

embedding. Eq. (7) returns a probability $p_{u,k,t}$ between zero and one. Sharing the head across concepts supports large concept vocabularies while allowing concept-specific behavior through $z_{k,t}$.

$$p_{u,k,t} = \sigma(\text{MLP}([h_{u,t} \parallel z_{k,t}])) \quad (7)$$

Mastery probabilities serve two roles. They predict whether the learner will answer an assessment item correctly, and they determine whether a concept's prerequisites are ready for use by the trajectory decoder. Calibration is therefore important: a score of 0.8 should correspond approximately to an 80% observed success rate within a sufficiently large bin.

Discrimination without calibration could cause a hard threshold to admit resources too early or defer them unnecessarily.

G. Prerequisite-Constrained Candidate Selection

Candidate selection begins with the difference between target mastery and current mastery. Fig. 4 illustrates the three-stage process of goal-conditioned retrieval, prerequisite-constrained eligibility, and trajectory optimization. Relevant resources are retrieved for unmet target concepts and their prerequisite support concepts. Each candidate is checked against the mastery threshold, remaining time, and completed-resource set. Invalid candidates are deferred rather than scored, and the feasible candidates enter sequential search.

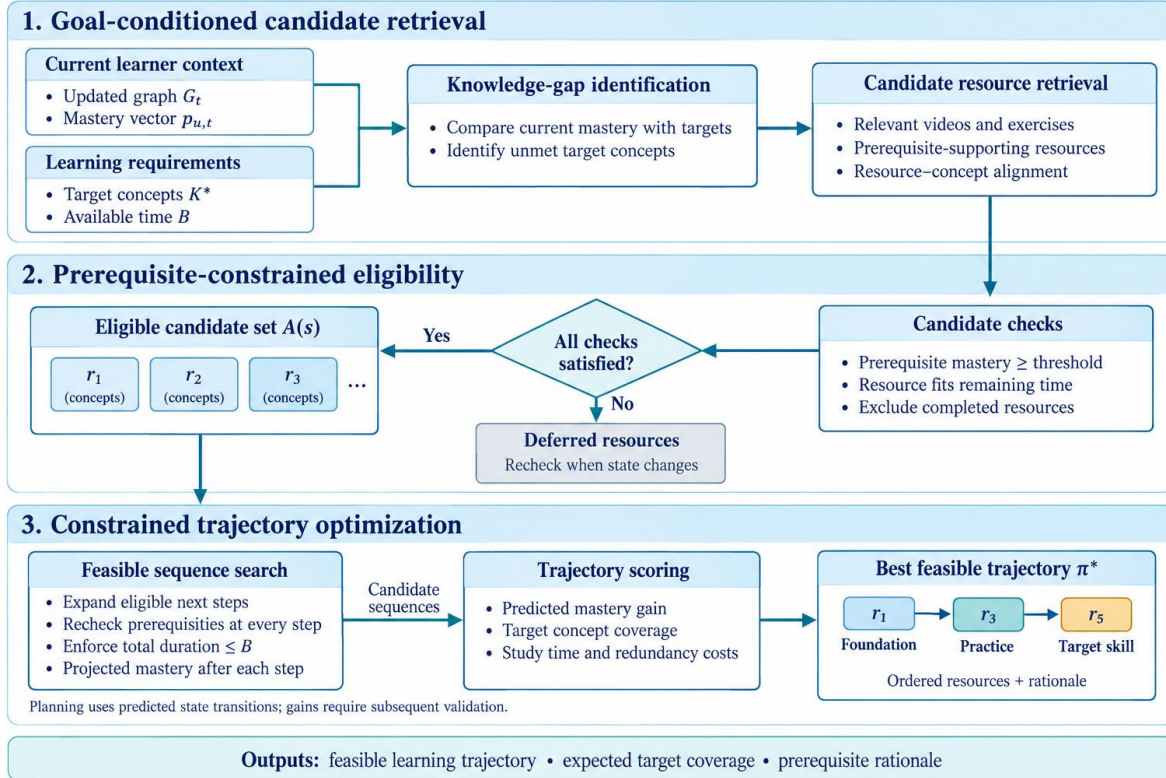


Fig. 4. Prerequisite-constrained candidate selection and personalized learning trajectory optimization.

For concept k with prerequisite set $\text{Pre}(k)$, the readiness gate in Eq. (8) is the minimum mastery of its prerequisites. A concept without prerequisites receives a gate value of one. The minimum operator implements the logical requirement that all prerequisites should be sufficiently mastered.

$$g_{u,k,t} = \begin{cases} \min_{j \in \text{Pre}(k)} p_{u,j,t}, & \text{Pre}(k) \neq \emptyset, \\ 1, & \text{Pre}(k) = \emptyset. \end{cases} \quad (8)$$

Let $C(r)$ be the concepts covered by resource r , $\text{dur}(r)$ its expected duration, S_t the completed resource set, B_t the remaining budget, and θ the prerequisite-mastery threshold. The hard eligibility indicator is given by Eq. (9). The product equals one only when every concept requirement is ready, the resource fits the budget, and the resource has not already been completed.

$$m_t(r) = 1[r \notin S_t] \prod_{k \in C(r)} 1[g_{u,k,t} \geq \theta] \quad (9)$$

The decoder applies the mask before softmax, assigning negative infinity to invalid logits. Consequently, an ineligible resource receives zero selection probability regardless of its learned relevance. The STOP token remains feasible so that the model can terminate when no resource provides sufficient expected utility or when the remaining budget is too small.

H. Feedback Integration and Adaptive Revision

Deployment requires the plan to change when observed progress diverges from predicted progress. Fig. 5 illustrates the closed-loop revision process. Performance signals include correctness, attempts, and time, while experience signals include perceived difficulty, relevance, and preference. The new event refreshes the graph and mastery state. The system either delivers the next planned resource or preserves the completed prefix and replans the unfinished suffix with the updated goal, state, and remaining budget.

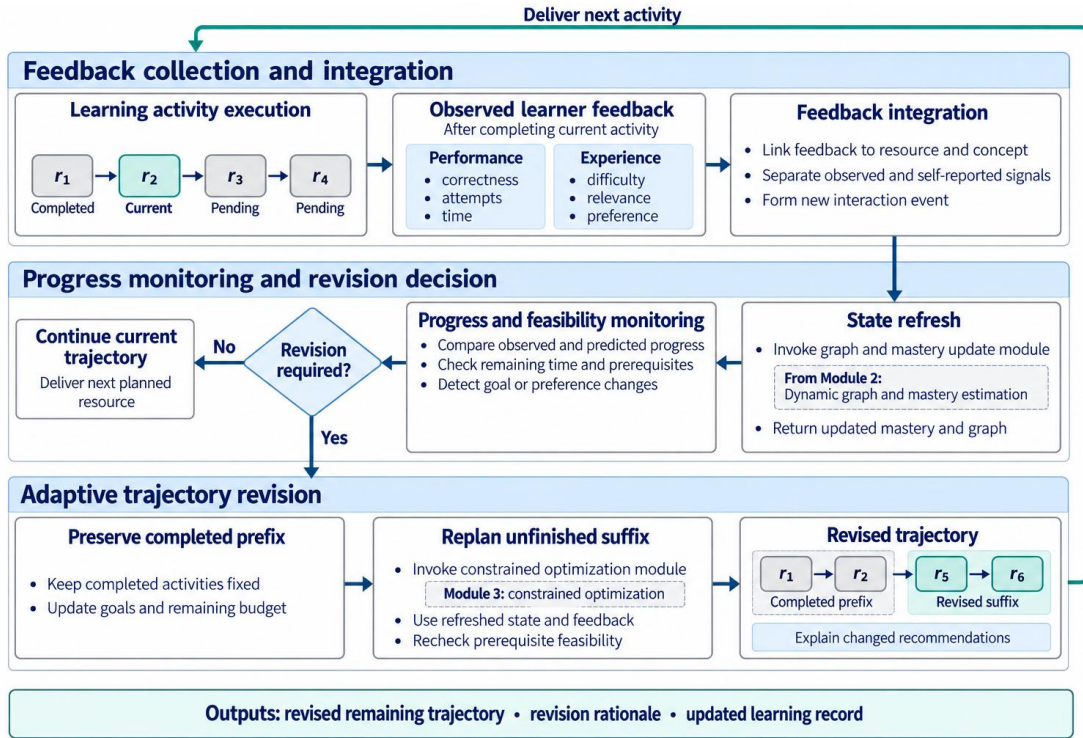


Fig. 5. Learner feedback integration and adaptive learning trajectory revision.

I. TPG-Net Architecture and Trajectory Objective

The complete neural architecture links representation learning to constrained decoding. Fig. 6 illustrates four principal blocks: input representations, a temporal heterogeneous graph encoder, a prerequisite-gated knowledge encoder, and a constrained trajectory decoder. Typed graph

neighborhoods are processed by relation-specific graph attention and time encoding. A recurrent learner memory produces mastery probabilities, the readiness gate modulates concept features, and goal-conditioned cross-attention scores resource candidates. Planning rollouts refresh projected mastery and remaining budget after each selected resource.

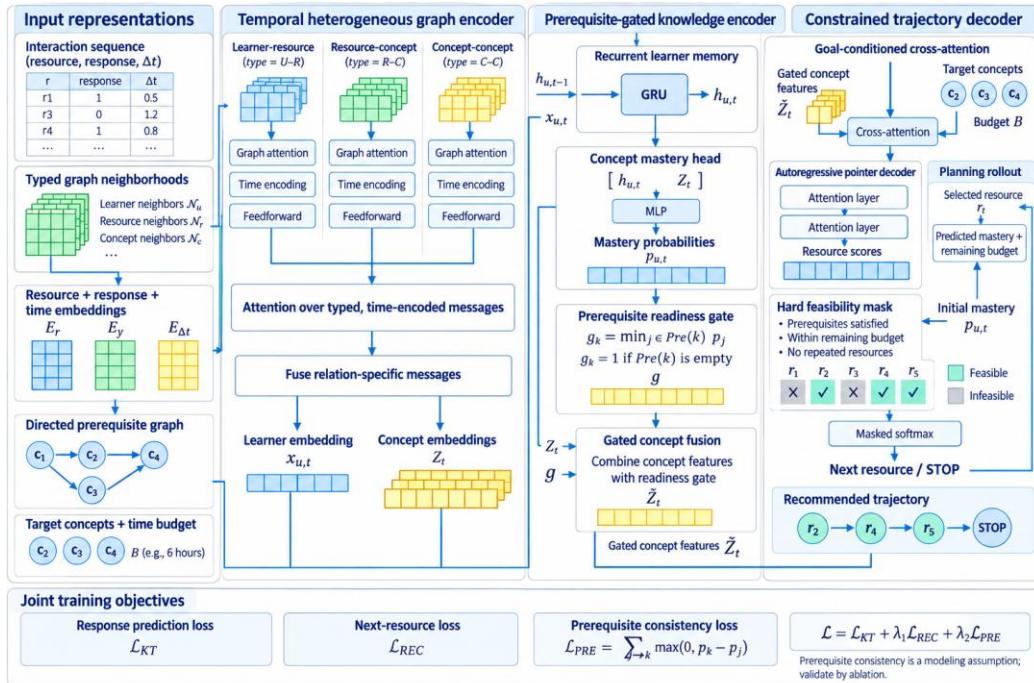


Fig. 6. Temporal prerequisite-gated graph network architecture.

For a candidate trajectory $\pi = (r_1, \dots, r_L)$, the decoder maximizes the utility in Eq. (10). The terms represent expected mastery gain, target-concept coverage, total duration, repeated or redundant content, and difficulty mismatch. Lambda

$$J(\pi) = \lambda_g \Delta M(\pi) + \lambda_c \text{Cov}(\pi, K^*) - \lambda_t \sum_{t=1}^L \text{dur}(r_t) - \lambda_r \text{Red}(\pi) - \lambda_d \text{Diff}(\pi, u) \quad (10)$$

Projected mastery after selecting a resource is updated by a bounded transition model. Eq. (11) illustrates a residual update in which $q_{r,k}$ is the resource-concept alignment and $\Delta_{u,r,t}$ is the predicted learning gain. The clipping operator prevents probabilities outside the unit interval.

$$\hat{p}_{u,k,t+1} = \text{clip}(p_{u,k,t} + q_{r,k} \Delta_{u,r,t} (1 - p_{u,k,t})) \quad (11)$$

The selected resource becomes part of the rollout state, the budget is reduced by its expected duration, and the feasibility mask is recomputed. Beam search can retain several high-utility partial trajectories, while greedy decoding provides lower latency. Because prerequisite feasibility is checked after each projected transition, a resource that is initially unavailable may become eligible later in the same plan.

J. Joint Training Objectives

The first training task predicts the observed response. Binary cross-entropy is computed over learner-item events according to Eq. (12), with optional class weights for imbalanced correctness labels.

$$\mathcal{L}_{\text{KT}} = - \sum_{(u,r,t)} [y_{u,r,t} \log \hat{y}_{u,r,t} + (1 - y_{u,r,t}) \log(1 - \hat{y}_{u,r,t})] \quad (12)$$

The second task predicts the next observed resource from a sampled candidate set. Eq. (13) uses cross-entropy over the masked decoder distribution. Training negatives must be sampled from resources available at the prediction time to prevent future-information leakage.

$$\mathcal{L}_{\text{REC}} = - \sum_{(u,t)} \log P(r_{t+1} | l, \mathcal{G}_t, h_{u,t} K^* B_t) \quad (13)$$

The third task encourages prerequisite-consistent mastery. For every directed prerequisite edge j to k , Eq. (14) penalizes a dependent concept whose predicted mastery exceeds that of its prerequisite by more than margin epsilon. The term is differentiable and complements the hard decoding mask.

$$\mathcal{L}_{\text{PRE}} = \sum_{u,t} \sum_{j \rightarrow k} \max(0, \hat{p}_{u,k,t} - p_{u,j,t} + \epsilon) \quad (14)$$

The total loss is the weighted sum in Eq. (15). The coefficients β_1 and β_2 should be selected using validation performance and constraint behavior. Reporting their sensitivity is important because excessive prerequisite regularization could over-smooth mastery estimates, whereas a very small value may leave concept states structurally inconsistent.

$$\mathcal{L} = \mathcal{L}_{\text{KT}} + \beta_1 \mathcal{L}_{\text{REC}} + \beta_2 \mathcal{L}_{\text{PRE}} \quad (15)$$

A revision trigger can be defined after the learner state and training objectives are specified. Eq. (16) uses η_p to control acceptable progress deviation, η_b as the minimum budget feasibility margin, and I_{goal} to indicate that the target set

coefficients are nonnegative and can be tuned on validation requests. Hard constraints are not absorbed into this utility because doing so would permit a sufficiently high gain to compensate for an invalid prerequisite.

changed. A trigger does not discard completed work; it invokes constrained decoding only for the remaining suffix.

$$\rho_t = 1[|M_t^{\text{obs}} - M_t^{\text{pred}}| > \eta_p] \vee 1[B_t < B_t^{\text{required}} + \eta_b] \vee 1[I_{\text{goal}} = 1] \quad (16)$$

K. Evaluation Design and Metrics

The evaluation design separates mastery prediction from trajectory quality. Response discrimination is measured by the area under the receiver operating characteristic curve and average precision. Accuracy is reported at a fixed 0.5 threshold, while the Brier score and expected calibration error evaluate probability reliability. Recommendation is evaluated with Precision@10, Recall@10, normalized discounted cumulative gain at 10, and mean reciprocal rank at 10. Trajectory-level measures include prerequisite compliance, target-concept coverage, total duration, repeated-resource rate, and replanning stability.

The data organization connects platform scale with the graph schema used by TPG-Net. Fig. 7 depicts four complementary views: a concept-centered entity diagram, educational entity counts, learner-interaction coverage, and a schematic prerequisite subgraph. Courses contain videos and problems, resources link to concepts, and learners generate timestamped watch or answer edges. The prerequisite arrows are interpreted as directed relations from a prerequisite concept to a dependent concept. The topology shown in the panel is illustrative and is not treated as an empirical degree distribution.

Precision@10 is the proportion of the ten recommended resources that are relevant, Recall@10 is the proportion of relevant resources recovered in the top ten, and NDCG@10 discounts correct resources that appear later in the list. MRR@10 depends on the rank of the first relevant recommendation. Prerequisite compliance is the fraction of selected steps whose prerequisite concepts exceed theta at the moment of selection. Target coverage is the proportion of requested target concepts addressed by at least one feasible selected resource. These metrics should be computed per learner and summarized with confidence intervals in a future empirical study.

The current numerical analysis uses the fixed values embedded in the supplied figures. These values form a deterministic illustrative score matrix, not output recovered from raw logs. No statistical significance tests are reported because repeated runs and per-learner scores were unavailable. This protocol prevents synthetic panels from being misrepresented as trained-model evidence while still allowing the architectural hypotheses and expected metric relationships to be examined.

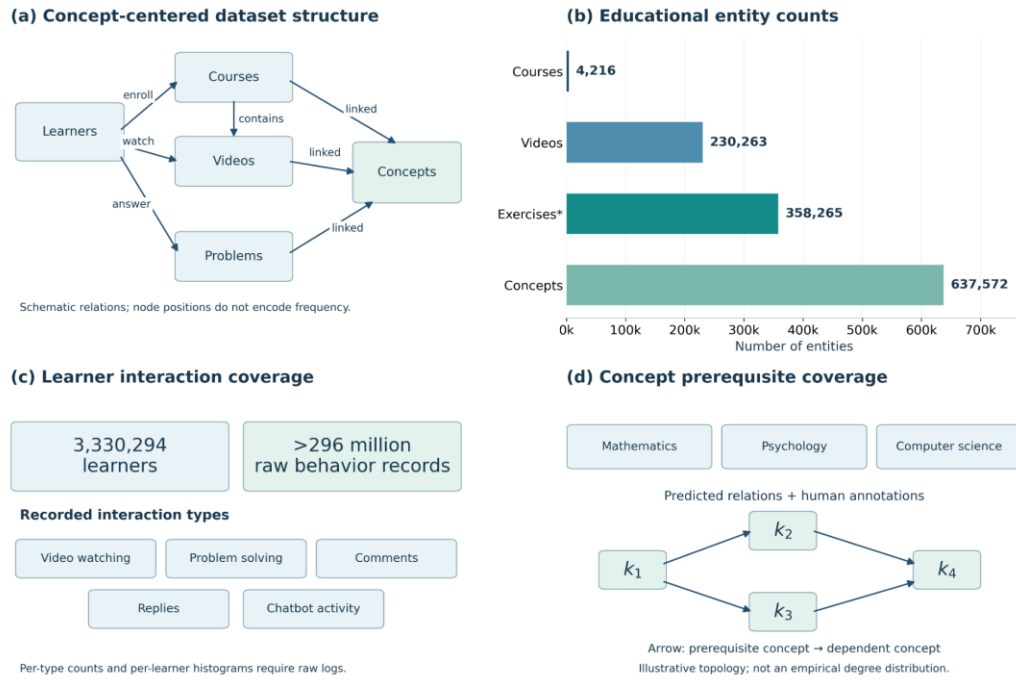


Fig. 7. MOOCCubeX dataset structure and distributions of learner interactions, educational resources, and concept prerequisites.

IV. RESULTS

The illustrative results evaluate whether the proposed design behaves consistently with its intended objectives. They test ranking quality, mastery discrimination and calibration, constraint behavior under different budgets, component contributions, and adaptive trajectory revision.

A. Comparative Trajectory Recommendation

The internal comparison includes popularity ranking, Item-KNN, GRU4Rec, SASRec, an unconstrained graph model, and

TPG-Net. Fig. 8 demonstrates that TPG-Net has the highest supplied value for every ranking metric: 95.6% Precision@10, 97.0% Recall@10, 96.3% NDCG@10, and 96.8% MRR@10. The unconstrained graph model is the strongest comparator with 92.1%, 93.8%, 92.6%, and 93.2%, respectively. The absolute TPG-Net improvements over that comparator are 3.5, 3.2, 3.7, and 3.6 percentage points. The consistent gain across precision, recall, ranking discount, and first-hit position suggests that the illustrative advantage is not confined to one metric definition.

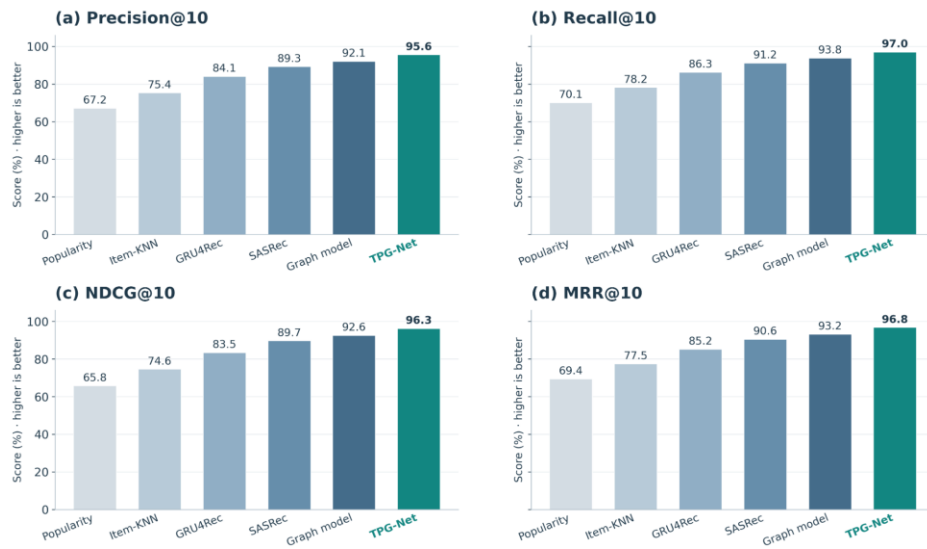


Fig. 8. Comparative learning trajectory recommendation performance across evaluation metrics.

The progression among baselines is also internally coherent. Popularity ranges from 65.8% to 70.1%, Item-KNN

from 74.6% to 78.2%, GRU4Rec from 83.5% to 86.3%, and SASRec from 89.3% to 91.2%. The graph model improves on

SASRec, reflecting the assumed value of concept-resource structure, while TPG-Net adds temporal encoding and prerequisite-constrained decoding. Because these are fixed illustrative values, the differences should be treated as target behavior for later replication rather than evidence of statistically significant generalization.

B. Knowledge-State Prediction and Calibration

Mastery prediction must distinguish correct from incorrect responses and produce probabilities suitable for threshold decisions. Fig. 9 illustrates an ROC AUC of 0.9947, average precision of 0.9948, and accuracy of 96.01% on a synthetic sample of 12,000 labels. The confusion matrix contains 5,787 correct predictions for the incorrect class and 5,734 correct predictions for the correct class, with 234 and 245 errors. The

calibration panel reports a Brier score of 0.0286 and an expected calibration error of 0.0032, while the reliability curve remains close to the diagonal. These values describe a deliberately favorable simulated predictor; the figure itself states that no model was trained or evaluated.

The calibration view is operationally important because the prerequisite gate converts continuous probabilities into eligibility decisions. A poorly calibrated model could obtain strong rank discrimination but still fail at a fixed threshold. The supplied reliability curve suggests that the simulated probabilities are suitable for thresholding, but a trained TPG-Net would require calibration assessment on held-out learners, subgroup analysis, and sensitivity testing across several values of theta.

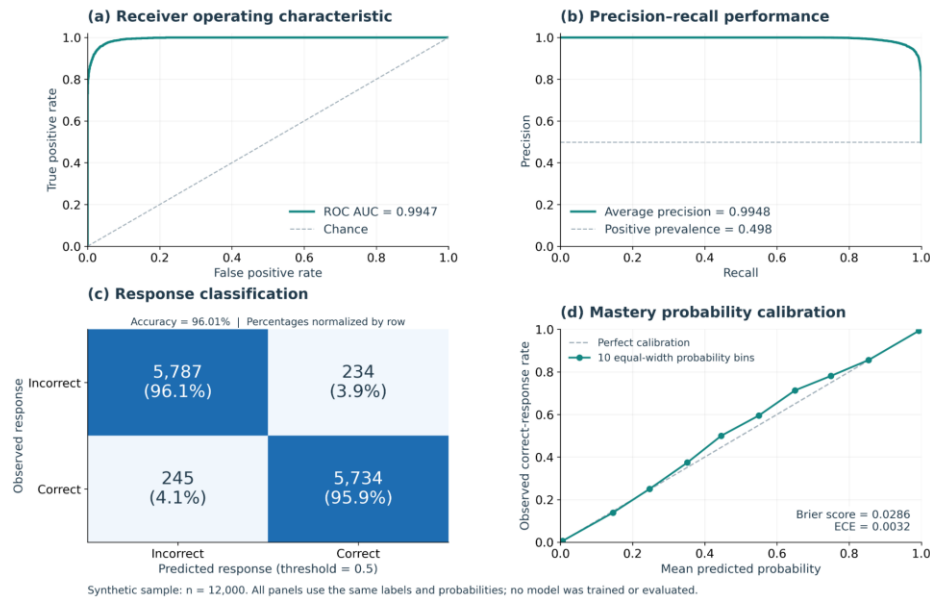


Fig. 9. Knowledge state prediction performance and mastery probability calibration.

C. Budget Sensitivity and Constraint Compliance

Learning-time constraints change which targets can be reached, but they should not weaken prerequisite enforcement. Fig. 10 demonstrates that TPG-Net maintains 100% prerequisite compliance at budgets of 30, 60, 90, 120, 150, and 180 minutes. In contrast, the supplied baselines remain below

96% and generally decline slightly as the budget grows. Target-concept coverage for TPG-Net rises from 37.9% at 30 minutes to 58.6%, 76.4%, 88.7%, 94.8%, and 96.5% at the successive budgets. This monotonic saturation pattern is plausible because additional time expands the feasible path but provides diminishing coverage gains near the target set limit.

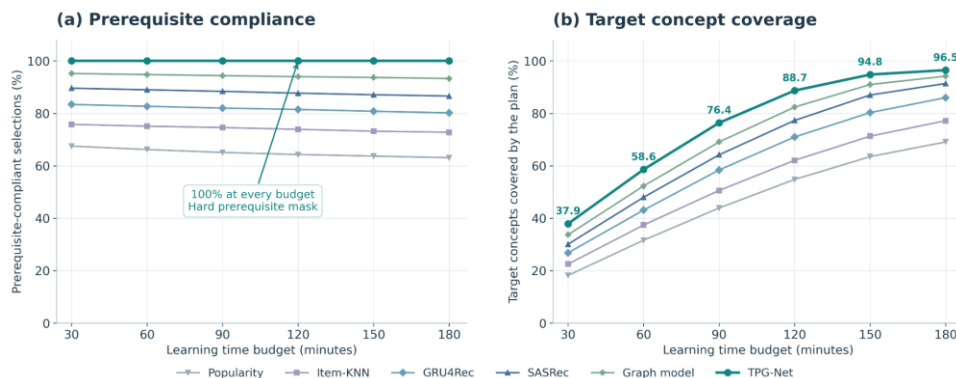


Fig. 10. Prerequisite compliance and target concept coverage under different learning time budgets.

The hard-mask design explains the flat compliance curve. Invalid resources are removed before normalization, so a larger budget cannot make an unmet prerequisite acceptable. Budget growth instead increases the number and duration of valid resources that may be selected. The comparison also separates two desirable properties that a single relevance metric could hide: structural validity remains perfect in the simulation while educational coverage improves with time.

D. Ablation Analysis

The ablation study examines temporal graph encoding, prerequisite gating, and constrained decoding. Fig. 11 illustrates that removing temporal encoding reduces Precision@10 from 95.6% to 92.4%, Recall@10 from 97.0% to 93.7%, NDCG@10 from 96.3% to 92.5%, MRR@10 from 96.8% to 93.1%, and target coverage from 96.5% to 91.8%.

Removing prerequisite gating produces intermediate reductions, with Precision@10 of 93.2%, Recall@10 of 94.4%, NDCG@10 of 93.1%, MRR@10 of 93.9%, and target coverage of 92.7%. Both variants retain 100% compliance because constrained decoding remains active.

Removing constrained decoding creates the most informative tradeoff. Ranking metrics fall to 94.1% Precision@10, 95.2% Recall@10, 94.4% NDCG@10, and 94.8% MRR@10, while prerequisite compliance decreases sharply to 89.6%. Target coverage increases to 97.4%, slightly above the full model, because the unconstrained decoder may reach additional targets through pedagogically invalid shortcuts. This pattern supports separate reporting of coverage and compliance: higher target coverage is not necessarily better when the path violates prerequisite structure.

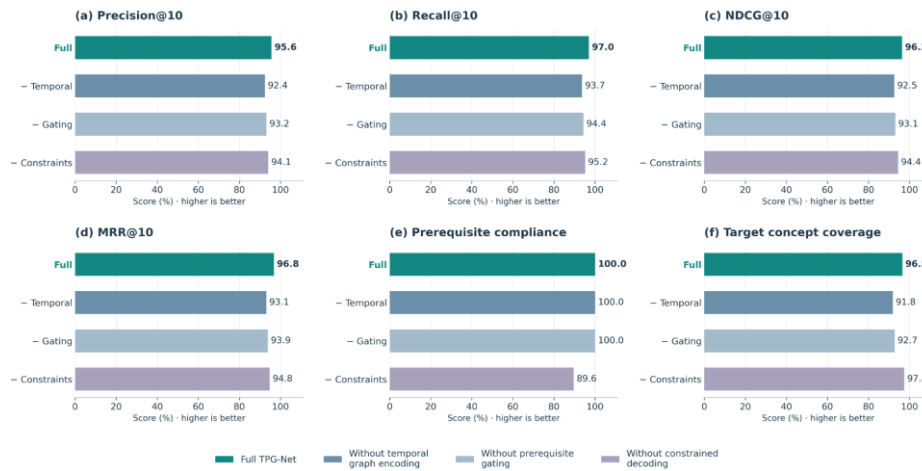


Fig. 11. Ablation analysis of temporal graph encoding, prerequisite gating, and constrained decoding.

E. Adaptive Trajectory Revision

The revision scenario tests whether the framework preserves completed work while responding to new evidence. Fig. 12 illustrates an original 180-minute algorithmic problem-solving plan followed by two interventions. At minute 60, low

assessment performance causes review and guided-loop activities to replace the unfinished suffix. At minute 120, a changed goal redirects the remaining time toward data preparation and visualization. Completed resources remain fixed, and only future activities are modified, which reduces learner disruption and keeps the revision auditable.



Fig. 12. Adaptive learning trajectory revision under changes in learner performance and goals.

The scenario demonstrates the distinction between state-triggered and goal-triggered revision. The first event changes the estimated mastery state while retaining the target, so remediation is inserted before advanced practice. The second event changes the target set, so the decoder selects a different feasible suffix under the remaining budget. A production evaluation should quantify the frequency of revision, the edit distance between original and revised suffixes, and whether revision improves subsequent mastery without causing excessive instability.

V. DISCUSSION

The proposed framework connects several strands of current educational artificial intelligence within one mathematical architecture. Its main contribution is not merely the use of a graph neural network, but the explicit coupling of dynamic interaction updates, calibrated knowledge-state prediction, prerequisite readiness, hard feasibility, and suffix-preserving replanning. The illustrative results are internally consistent with this design: temporal encoding improves all ranking measures, prerequisite gating improves both ranking and target coverage, and constrained decoding is essential for perfect compliance. The small coverage increase after removing constraints reveals why accuracy-only evaluation can reward invalid shortcuts.

A. Interpretation of the Architectural Contributions

Dynamic graph learning provides a stronger state representation than a static learner vector because the representation can preserve relation type, local structure, and temporal distance. A new problem attempt affects the learner node, the attempted resource, connected concepts, and prerequisite neighbors. Relation-specific attention can therefore distinguish evidence from an assessment, a video view, a content relation, and a prerequisite edge. This is particularly important in MOOCCubeX, where interaction types and resource types have different educational meanings.

The prerequisite readiness gate adds an inductive bias to the concept representation. Recent hierarchical knowledge-tracing methods show that multiple levels of knowledge structure can improve state estimation [37], and response-rebalancing research emphasizes that discriminative learner evidence should not be overwhelmed by frequent routine interactions [38]. Knowledge-associated memory models similarly indicate that structural associations can improve long-term state retention [39]. Graph-based subject mapping confirms that explicit concept structure remains useful even when strong sequence models are available [40]. TPG-Net extends these ideas from prediction to constrained multi-step planning.

Hard feasibility is the mechanism that turns a high-scoring sequence into a valid educational trajectory. A soft prerequisite penalty may improve average behavior, yet it cannot ensure that every emitted step is valid. By applying the mask before softmax and recomputing it after each projected transition, TPG-Net provides a formal compliance guarantee relative to the current graph, mastery estimates, threshold, and resource annotations. The guarantee is conditional rather than absolute: incorrect prerequisite edges, poor mastery calibration, or inaccurate duration estimates can still produce pedagogically

unsuitable recommendations. The system should therefore expose the prerequisite chain and confidence values to educators.

B. Comparison with Recent Approaches

Recent work offers several partial solutions to the same problem. Knowledge-network and graph-convolutional methods structure the content domain but often assume a fixed learner state [1], [2]. Multi-algorithm and ant-colony approaches address combinatorial path search, but their state representations may not update with every event [3], [6]. Dynamic knowledge-tracing and process-mining approaches represent progression more faithfully, although they do not always guarantee prerequisite-valid decoding [10], [15]. Graph reinforcement-learning approaches optimize long-term reward, but reward penalties may trade safety constraints against relevance [16]-[19]. TPG-Net is designed to preserve the benefits of these approaches while separating learned utility from non-negotiable feasibility.

The state-of-the-art table deliberately avoids a single cross-paper ranking. Published values are produced on different datasets, target definitions, and split protocols, so placing them in one performance column could imply false comparability. The supplied TPG-Net percentages are higher than the internal baselines in Fig. 7, which satisfies the proof-of-concept objective. A defensible external superiority claim would require reimplementing recent methods on the same MOOCCubeX partitions, using the same candidate universe, cutoff, prerequisite annotations, time budgets, negative samples, and random seeds.

C. Practical and Educational Implications

The architecture supports several forms of educational explanation. A learner can be told that a resource was selected because it addresses a current mastery gap, supports a target concept, fits the remaining time, and has satisfied prerequisites. An instructor can inspect which event changed the learner state and why a previous plan was revised. Platform designers can separately monitor ranking quality, calibration, prerequisite compliance, target coverage, and revision stability. This separation is preferable to a single opaque reward because it makes failure modes observable.

Incremental graph updates also reduce unnecessary computation. Only the affected neighborhood needs to be refreshed after an event, whereas a complete graph rebuild would be infeasible for hundreds of millions of records. Candidate retrieval can be restricted to targets, prerequisite ancestors, and aligned resources before constrained decoding. Approximate nearest-neighbor search can support retrieval, while cached concept embeddings and bounded temporal neighborhoods can control latency. These engineering choices should be evaluated without weakening past-only data handling.

D. Limitations and Threats to Validity

The most important limitation is the illustrative nature of the numerical results. Fig. 7-11 were supplied as simulated panels, and Fig. 8 explicitly states that no model was trained or evaluated. The manuscript therefore does not provide empirical evidence that TPG-Net reaches the reported percentages on

MOOCCubeX. It provides an auditable architecture, metric specification, and target result pattern for a future experiment. Training code, raw processed splits, checkpoint hashes, seed-level outputs, and confidence intervals are required before quantitative claims can be presented as measured findings.

The dataset presents additional risks. Interaction logs reflect platform exposure, self-selection, content popularity, and incomplete observation of learning outside the platform. Correctness is an imperfect proxy for mastery, and video watching does not imply comprehension. Prerequisite annotations may be incomplete, cyclic, or domain-dependent. The large concept vocabulary may contain duplicate or overly fine-grained concepts, while the graph can encode historical curricular choices rather than universally valid pedagogy. Preprocessing decisions must therefore be documented and subjected to sensitivity analysis.

Model-based risks also remain. The minimum prerequisite gate may be too strict when prerequisite relations are probabilistic or substitutable. Projected mastery gains may accumulate error across long rollouts. A hard threshold can introduce discontinuities, especially for poorly calibrated probabilities near theta. Learners with sparse histories may receive uncertain states, and feedback loops could repeatedly recommend popular or easy resources. Evaluation should include cold-start learners, long-tail resources, demographic and institutional subgroups where legally and ethically appropriate, and robustness to missing or noisy edges.

Finally, high ranking scores do not prove improved learning. The strongest evaluation would be a prospective study comparing TPG-Net with instructor-designed and algorithmic alternatives using delayed mastery, retention, time-to-goal, dropout, learner satisfaction, and prerequisite violations. Human oversight is necessary when recommendations affect progression, assessment, or access to advanced material. Explanations should support review rather than create unwarranted confidence in a probabilistic learner model.

VI. CONCLUSION

This study proposed TPG-Net, a Temporal Prerequisite-Gated Graph Network for prerequisite-constrained personalized learning trajectory recommendation. The framework represents learner interactions, educational resources, concepts, and directed prerequisites in a dynamic heterogeneous graph. Relation-specific temporal attention updates learner and concept embeddings; a recurrent state model estimates mastery probabilities; a readiness gate summarizes prerequisite status, and a hard-masked autoregressive decoder generates a feasible resource sequence within the available time. A closed feedback loop preserves completed activities and replans only the unfinished suffix when performance or goals change.

The supplied simulation-based panels show the intended behavior of the architecture. TPG-Net has the highest illustrative ranking scores, maintains 100% prerequisite compliance across time budgets, and exhibits coherent ablation effects. Removing constrained decoding increases nominal target coverage but produces prerequisite violations,

demonstrating that coverage and pedagogical validity must be evaluated separately. These values are not presented as results from trained MOOCCubeX checkpoints. Future work should implement the complete data pipeline, publish learner-safe partitions and code, repeat training across seeds, compare recent methods under a common protocol, and conduct prospective educational evaluation. The proposed formulation offers a transparent foundation for that empirical program and for learning systems that must optimize personalization without relaxing prerequisite structure.

REFERENCES

- [1] H. Li et al., "Research on personalized learning path planning model based on knowledge network," *Neural Computing and Applications*, vol. 35, pp. 8809-8821, 2023. doi: 10.1007/s00521-022-07658-8.
- [2] X. Zhang, S. Liu, and H. Wang, "Personalized Learning Path Recommendation for E-Learning Based on Knowledge Graph and Graph Convolutional Network," *International Journal of Software Engineering and Knowledge Engineering*, vol. 33, no. 01, pp. 109-131, 2023. doi: 10.1142/S0218194022500681.
- [3] Y. Ma, L. Wang, J. Zhang, F. Liu, and Q. Jiang, "A Personalized Learning Path Recommendation Method Incorporating Multi-Algorithm," *Applied Sciences*, vol. 13, no. 10, art. 5946, 2023. doi: 10.3390/app13105946.
- [4] M. Mu and M. Yuan, "Research on a personalized learning path recommendation system based on cognitive graph with a cognitive graph," *Interactive Learning Environments*, vol. 32, no. 8, pp. 4237-4255, 2024. doi: 10.1080/10494820.2023.2195446.
- [5] N.S. Raj and V.G. Renumol, "An improved adaptive learning path recommendation model driven by real-time learning analytics," *Journal of Computers in Education*, vol. 11, no. 1, pp. 121-148, 2024. doi: 10.1007/s40692-022-00250-y.
- [6] Y. Zheng, D. Wang, Y. Xu, and Y. Li, "A Multigranularity Learning Path Recommendation Framework Based on Knowledge Graph and Improved Ant Colony Optimization Algorithm for E-Learning," *IEEE Transactions on Computational Social Systems*, vol. 12, no. 2, pp. 586-607, 2025. doi: 10.1109/TCSS.2024.3488373.
- [7] X. Wu et al., "Meta concept recommendation based on knowledge graph," *Discover Computing*, vol. 27, no. 1, art. 30, 2024. doi: 10.1007/s10791-024-09467-0.
- [8] R. Pelánek, T. Effenberger, and P. Jarušek, "Personalized recommendations for learning activities in online environments: a modular rule-based approach," *User Modeling and User-Adapted Interaction*, vol. 34, no. 4, pp. 1399-1430, 2024. doi: 10.1007/s11257-024-09396-z.
- [9] F. Wang, M. Wu, and Y. Liu, "A Novel Personalized Learning Path Mining Based on Steiner Tree and Knowledge Graph," *Procedia Computer Science*, vol. 242, pp. 305-312, 2024. doi: 10.1016/j.procs.2024.08.229.
- [10] F. Zhang, X. Feng, and Y. Wang, "Personalized process-type learning path recommendation based on process mining and deep knowledge tracing," *Knowledge-Based Systems*, vol. 303, art. 112431, 2024. doi: 10.1016/j.knosys.2024.112431.
- [11] S. Ying and Y. Fu, "Knowledge Graph Construction and Personalised Learning Path Generation Mechanism for Online English Teaching and Learning," *Applied Mathematics and Nonlinear Sciences*, vol. 9, no. 1, art. 20243609, 2024. doi: 10.2478/amns-2024-3609.
- [12] Q. Xiao, Y.W. Zhang, X.Q. Xin, and L.W. Cai, "Sustainable Personalized E-Learning through Integrated Cross-Course Learning Path Planning," *Sustainability*, vol. 16, no. 20, art. 8867, 2024. doi: 10.3390/su16208867.
- [13] X. Xie and X. Feng, "Personalized Learning Path Recommendation Based on Learner Profile and Knowledge Graph," *Computing and Informatics*, vol. 44, no. 4, pp. 983-1008, 2025. doi: 10.31577/cai_2025_4_983.

- [14] B. Hou et al., "KG-PLPPM: A Knowledge Graph-Based Personal Learning Path Planning Method Used in Online Learning," *Electronics*, vol. 14, no. 2, art. 255, 2025. doi: 10.3390/electronics14020255.
- [15] C. Tong and C. Ren, "Deep knowledge tracing and cognitive load estimation for personalized learning path generation using neural network architecture," *Scientific Reports*, vol. 15, no. 1, art. 24925, 2025. doi: 10.1038/s41598-025-10497-x.
- [16] S. Haldar, S. Sengupta, and A.K. Das, "Personalized Learning Path Recommendation using Graph Reinforcement Learning," *Procedia Computer Science*, vol. 258, pp. 3480-3489, 2025. doi: 10.1016/j.procs.2025.04.604.
- [17] L.Y. Zhou and Y.Y. Wang, "Simulation of personalized English learning path recommendation system based on knowledge graph and deep reinforcement learning," *Scientific Reports*, vol. 15, no. 1, art. 34554, 2025. doi: 10.1038/s41598-025-17918-x.
- [18] Z. Fu, "Integrating Reinforcement Learning with Dynamic Knowledge Tracing for personalized learning path optimization," *Scientific Reports*, vol. 15, no. 1, art. 40202, 2025. doi: 10.1038/s41598-025-23900-4.
- [19] R. Gu, "Personalized learning path based on graph attention mechanism deep reinforcement learning research on recommender systems," *Journal of Computational Methods in Sciences and Engineering*, vol. 25, no. 3, pp. 2411-2426, 2025. doi: 10.1177/14727978241313260.
- [20] Mukhamedaly, S., Kabekeyeva, K., Mussabekova, G., Kuralbayeva, A., Toibekova, B., Makashkulova, G., & Omarov, B. (2026). A Pedagogical Framework for Ethical Skill Development in Higher Education within Smart Learning Environments. *International Journal of Modern Education and Computer Science*, 18(2), 1.
- [21] Y. Ma et al., "DORIS: Personalized course recommendation system based on deep learning," *PLOS ONE*, vol. 18, no. 6, art. e0284687, 2023. doi: 10.1371/journal.pone.0284687.
- [22] G. Abdelrahman and Q. Wang, "Learning data teaching strategies via knowledge tracing," *Knowledge-Based Systems*, vol. 269, art. 110511, 2023. doi: 10.1016/j.knosys.2023.110511.
- [23] D. Han, D. Kim, M. Kim, K. Han, and M.Y. Yi, "Temporal enhanced inductive graph knowledge tracing," *Applied Intelligence*, vol. 53, no. 23, pp. 29282-29299, 2023. doi: 10.1007/s10489-023-05083-5.
- [24] X. Wang, L. Chen, and M. Zhang, "Deep Attentive Model for Knowledge Tracing," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 8, pp. 10192-10199, 2023. doi: 10.1609/aaai.v37i8.26214.
- [25] T. Wu and Q. Ling, "Self-supervised heterogeneous hypergraph network for knowledge tracing," *Information Sciences*, vol. 624, pp. 200-216, 2023. doi: 10.1016/j.ins.2022.12.075.
- [26] J. Sun et al., "Weighted Heterogeneous Graph-Based Three-View Contrastive Learning for Knowledge Tracing in Personalized e-Learning Systems," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2838-2847, 2024. doi: 10.1109/TCE.2023.3293953.
- [27] Omarov, B., & Zhumanov, Z. (2023). Bidirectional long short-term memory with attention mechanism for emotion analysis in textual content. *International Journal of Advanced Computer Science and Applications*, 14(6).
- [28] Y. Han et al., "Dynamic heterogeneous graph contrastive networks for knowledge tracing," *Applied Soft Computing*, vol. 166, art. 112194, 2024. doi: 10.1016/j.asoc.2024.112194.
- [29] Y. Wu et al., "Programming knowledge tracing based on heterogeneous graph representation," *Knowledge-Based Systems*, vol. 300, art. 112161, 2024. doi: 10.1016/j.knosys.2024.112161.
- [30] X. Sun et al., "Target hierarchy-guided knowledge tracing : Fine-grained knowledge state modeling," *Expert Systems with Applications*, vol. 251, art. 123898, 2024. doi: 10.1016/j.eswa.2024.123898.
- [31] H. Dai et al., "Adaptive meta-knowledge dictionary learning for incremental knowledge tracing," *Engineering Applications of Artificial Intelligence*, vol. 132, art. 107969, 2024. doi: 10.1016/j.engappai.2024.107969.
- [32] Y. Pu et al., "ELAKT: Enhancing Locality for Attentive Knowledge Tracing," *ACM Transactions on Information Systems*, vol. 42, no. 4, pp. 1-27, 2024. doi: 10.1145/3652601.
- [33] S. Park, D. Lee, and H. Park, "Enhancing knowledge tracing with concept map and response disentanglement," *Knowledge-Based Systems*, vol. 302, art. 112346, 2024. doi: 10.1016/j.knosys.2024.112346.
- [34] M. Li, Z. Li, C. Huang, Y. Jiang, and X. Wu, "EduGraph: Learning Path-Based Hypergraph Neural Networks for MOOC Course Recommendation," *IEEE Transactions on Big Data*, vol. 10, no. 6, pp. 706-719, 2024. doi: 10.1109/TBDATA.2024.3453757.
- [35] S. Wu, J. Wang, and W. Zhang, "Contrastive Personalized Exercise Recommendation With Reinforcement Learning," *IEEE Transactions on Learning Technologies*, vol. 17, pp. 691-703, 2024. doi: 10.1109/TLT.2023.3326449.
- [36] J. Yu et al., "MOOCCubeX: A Large Knowledge-centered Repository for Adaptive Learning in MOOCs," *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 4643-4652, 2021. doi: 10.1145/3459637.3482010.
- [37] Q. Li et al., "HKT: Hierarchical structure-based knowledge tracing," *Information Processing & Management*, vol. 62, no. 5, art. 104206, 2025. doi: 10.1016/j.ipm.2025.104206.
- [38] J. Cui et al., "Rebalancing Discriminative Responses for Knowledge Tracing," *ACM Transactions on Information Systems*, vol. 43, no. 3, pp. 1-25, 2025. doi: 10.1145/3716821.
- [39] J. Li, Y. Deng, S. Mao, Y. Qin, and Y. Jiang, "Knowledge-Associated Embedding for Memory-Aware Knowledge Tracing," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 3, pp. 4016-4028, 2024. doi: 10.1109/TCSS.2023.3306909.
- [40] Z. Yang, J. Hu, S. Zhong, L. Yang, and G. Min, "Graph-based effective knowledge tracing via subject knowledge mapping," *Education and Information Technologies*, vol. 30, no. 7, pp. 9813-9840, 2025. doi: 10.1007/s10639-024-13069-0.