

Article

Temporal Graph Neural Networks for Modeling Student Knowledge Evolution and Predicting Learning Trajectories

Deborah Olaniyan  and Ruth Wario * 

Department of Computer Science and Informatics, Faculty of Natural and Agricultural Sciences, University of the Free State, Qwaqwa Campus, Kestell Road, Phuthaditjhaba 9866, South Africa

* Correspondence: wariord@ufs.ac.za

Abstract

Accurately modeling student knowledge evolution is a central challenge in personalized learning and adaptive educational systems. Traditional sequential or static approaches often fail to capture both the temporal dynamics of learning and the relational structure between students and concepts. This study introduces a Temporal Graph Neural Network (TGNN) framework for modeling student knowledge acquisition and predicting learning trajectories using fine-grained interaction data from the ASSISTments_skill dataset. The TGNN represents students and skills as nodes in a dynamic bipartite graph, with temporal edges encoding correctness, attempts, hints, and interaction timestamps. Experiments demonstrate that TGNN significantly outperforms state-of-the-art baselines, including Deep Knowledge Tracing (DKT), Self-Attentive Knowledge Tracing (SAKT), and static graph convolutional networks, achieving an Area Under the Curve (AUC) of 0.892, Accuracy of 0.846, F1 score of 0.842, and a Mean Absolute Error (MAE) of 0.078 for trajectory prediction. Ablation studies reveal the critical role of temporal encoding, edge features, and graph connectivity in accurately modeling learning dynamics. Concept-level analysis indicates high prediction accuracy across both high-frequency and low-frequency skills, while temporal attention mechanisms enable interpretable insights into the influence of prior interactions on future performance. These results highlight the effectiveness of integrating temporal dynamics, graph-based relational modeling, and pedagogically meaningful features in predicting student learning outcomes. These results demonstrate the potential of temporal graph-based modeling for capturing student–skill relationships and learning dynamics in educational interaction data. Rather than introducing a fundamentally new graph architecture, this study systematically adapts the Temporal Graph Network (TGN) framework to educational data and evaluates its effectiveness for modeling knowledge evolution and forecasting student learning trajectories. The findings provide practical insights for applying temporal graph learning methods to personalized learning, adaptive intervention design, and real-time performance forecasting.



Academic Editor: Frank Werner

Received: 16 February 2026

Revised: 17 March 2026

Accepted: 17 March 2026

Published: 1 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: Temporal Graph Neural Networks (TGNN); student modeling; knowledge tracing; learning trajectories; educational data mining

1. Introduction

Understanding how student knowledge evolves over time is central to improving instructional design, adaptive learning systems, and personalized education at scale [1]. Traditional student modeling approaches, such as Bayesian Knowledge Tracing (BKT) and Item Response Theory (IRT), have provided foundational insights into learner performance

prediction, yet they typically treat learning as a static or semi-static process [2]. These models assume that skills are independent, learning transitions follow fixed probabilistic rules, and that all learners exhibit relatively homogeneous behavior [3]. Such assumptions substantially simplify the complexity of human learning and limit the ability of these models to capture nuanced cognitive processes such as forgetting, skill transfer, fluctuating motivation, or varied learning strategies [4].

In contrast, real-world learning is dynamic, nonlinear, and inherently relational. Students interact with curricula composed of interdependent concepts where mastery of one skill often influences understanding of others [5]. Their knowledge states shift continuously as they encounter new material, revisit prior topics, and practice skills across multiple sessions [6]. Moreover, their learning behavior is shaped by contextual factors such as the difficulty of instructional content, feedback quality, engagement patterns, and historical performance, that unfold over time [7]. As modern digital learning environments and intelligent tutoring systems generate high-frequency, longitudinal data, these rich interaction logs reveal complex temporal patterns that traditional methods struggle to model effectively [8].

Consequently, there is an increasing need for computational models capable of representing not only individual learning events but also the evolving dependencies between concepts, learners, and assessment items [9]. Such models must capture temporal dynamics, conceptual relationships, and heterogeneous learning trajectories with much higher fidelity. Emerging deep learning methods, particularly those that integrate representation learning with sequential modeling, offer substantial promise [10]. However, they often overlook the graph-structured nature of educational data, where concepts, students, items, and activities form interconnected networks that evolve over time [11]. This necessitates more advanced modeling strategies that can simultaneously reason about structure and temporality, providing a more holistic and accurate representation of how knowledge develops.

Graph Neural Networks (GNNs) have recently emerged as a powerful paradigm for representing relational structures such as concept graphs, knowledge dependencies, and student-item interactions [12]. Yet most existing GNN-based student models operate on static graphs, overlooking the continuous evolution of learners' knowledge states [13]. This limitation is critical because learning is inherently temporal, students reinforce, forget, and transfer knowledge across sessions, topics, and assessments. Temporal Graph Neural Networks (TGNNs), which integrate dynamic graph updates with sequence modeling, offer a promising solution for capturing both the structural and temporal dimensions of learning.

Despite significant advances in knowledge tracing and educational data mining, several research gaps remain. First, most existing approaches model student learning primarily as sequential interactions, capturing temporal order but ignoring relational dependencies between students and knowledge components. Second, graph neural networks applied to educational data often rely on static representations, which fail to reflect the dynamic evolution of learning interactions. Third, existing models focus mainly on next-response prediction, with limited attention to modeling longer-term learning trajectories.

To address these limitations, this study proposes a Temporal Graph Neural Network (TGNN) framework that integrates temporal encoding, dynamic graph representation, and rich interaction-level features. Students and skills are represented as nodes in a temporally evolving bipartite graph, allowing the model to jointly capture relational structure and temporal learning dynamics. This enables accurate prediction of both immediate responses and broader learning trajectories, advancing research at the intersection of temporal graph representation learning and educational data mining.

It is important to note that this work does not aim to propose a completely new temporal graph neural network architecture. Instead, the study systematically adapts the Temporal Graph Network (TGN) framework introduced by Rossi et al. [14] for the specific problem of student knowledge tracing and learning trajectory prediction in educational data mining. The contribution of this research lies in the formulation of student–skill interactions as a temporally evolving bipartite graph enriched with pedagogically meaningful interaction attributes, including correctness, attempts, hint usage, and temporal intervals. By integrating temporal attention mechanisms with these educational interaction features, the framework enables the modeling of both relational dependencies and temporal learning dynamics. The study further provides a comprehensive empirical evaluation including trajectory prediction, concept-level analysis, and ablation studies to better understand how temporal graph learning can support modeling of student knowledge evolution.

Beyond algorithmic modeling, understanding student learning requires grounding predictive systems in established educational theory [15]. Learning is a progressive process in which students develop conceptual mastery through repeated practice and feedback [16]. Bloom’s taxonomy conceptualizes knowledge development as a hierarchical progression from foundational understanding toward higher cognitive mastery [17]. In this context, temporal student–skill interactions serve as observable indicators of incremental learning. The TGNN framework aligns with this perspective by modeling learning as a temporally evolving interaction process, where each interaction contributes evidence about a learner’s mastery state and prior events inform future performance. In this way, the framework bridges computational modeling with pedagogically meaningful insights.

This study presents an applied methodological investigation into the use of temporal graph neural networks for modeling student knowledge evolution in educational data mining. Rather than proposing a fundamentally new neural architecture, the work focuses on adapting and evaluating temporal graph learning techniques within the context of knowledge tracing. The contributions of this study are threefold. First, student–skill interactions are formulated as a temporally evolving bipartite graph enriched with pedagogically meaningful edge features, including correctness, number of attempts, hint usage, and temporal intervals. Second, the study demonstrates how temporal graph attention and message passing mechanisms can capture both relational dependencies and temporal learning dynamics for predicting student performance. Third, the framework is empirically evaluated through comprehensive experiments, including trajectory forecasting, concept-level analysis, and ablation studies, providing practical insights into the applicability of temporal graph models for personalized learning analytics.

2. Literatures

2.1. Graph-Based Knowledge Tracing and Pedagogically Grounded Models

Recent advances in knowledge tracing have increasingly emphasized the importance of modeling structured knowledge relationships rather than treating skills as independent units. Cui et al. [18] introduced graph-based reasonable knowledge tracing, integrating pedagogical theories with graph neural architectures to better represent how students transition across interrelated concepts. Their work highlights that incorporating explicit curricular structures improves interpretability and yields more realistic representations of student cognitive progression. Similarly, Yang et al. [19] proposed a graph-based effective KT framework using subject-level knowledge mapping, demonstrating that graph representations aligned with subject hierarchies enable more accurate mastery estimation and support richer pedagogical insights. These works demonstrate a shift toward interpretable graph-based KT, where conceptual dependencies, instructional design principles, and prerequisite structures directly inform learning prediction models.

2.2. GNN-Based Models for Student Performance Prediction

Graph neural networks (GNNs) have been widely adopted for predicting learner performance by modeling relationships between students, items, and knowledge components. Hakkal and Ait Lahcen [20] proposed a GNN-based learner performance prediction system that constructs relational graphs from student–activity interactions. Their results demonstrate that GNNs capture complex relational patterns more effectively than classical machine learning baselines.

Zhou and Yu [21] extended this idea by developing a Multi-Graph Spatial–Temporal Synchronous Network that incorporates multiple relational views such as student–knowledge, student–task, and temporal transitions to jointly model performance and learning pathways. Meanwhile, Wu et al. [22] introduced SGKT, one of the earliest session-graph approaches in KT, which represents each learning session as a directed interaction graph, enabling improved modeling of short-term knowledge transitions and performance fluctuations.

Collectively, these studies show that static GNN architectures can significantly improve prediction accuracy by leveraging multi-relational educational data; however, they typically operate on fixed graphs and do not explicitly model dynamic evolution over time.

2.3. Dynamic and Temporal Graph-Based Knowledge Modeling

Temporal extensions of graph-based learning are beginning to appear in educational data mining. Cheng and Long [23] proposed a method for analyzing and predicting competitive student performance via Temporal Knowledge Graphs, capturing event-specific updates to student representations as learning interactions unfold. Although promising, their framework focuses primarily on structured competitive environments rather than broader KT tasks. Xia et al. [24] presented a Multivariate Knowledge Tracking approach that leverages GNNs to simultaneously model multiple dimensions of learning behavior (e.g., correctness, response time). Their work demonstrates that graph-based models can encode heterogeneous features, though the graph remains static.

From a broader perspective, Xia [25] surveyed graph-model-based deep learning for learning analytics and highlighted several gaps: insufficient modeling of time-evolving knowledge states, limited incorporation of long-term temporal dependencies, and a lack of general frameworks for dynamic knowledge evolution. These works provide a foundation but indicate the need for Temporal Graph Neural Networks (TGNNs) specifically designed for continuous learning processes rather than static snapshots.

2.4. Transformer-Based, Hybrid, and Multi-Dimensional Knowledge Tracing Models

Beyond GNNs, several studies have explored combining deep learning architectures with graph or probabilistic components to better model temporal dependencies. Mai, Cao, and Liu [26] introduced an interpretable transformer–Bayesian hybrid KT model, capturing both sequential dependencies and causal structures underlying learning. Their hierarchical design offers improved interpretability, though it does not explicitly integrate concept graphs. Zhang et al. [27] proposed AEKT, a multi-dimensional KT model incorporating cognitive ability and knowledge acquisition into a unified latent space. This highlights the growing trend toward multi-aspect KT, where temporal, cognitive, and relational factors are jointly modeled.

Hashemifar and Sahebi [28] focused on personalized student modeling for future learning resource prediction, using sequential and relational signals to forecast appropriate next learning materials. This work reinforces the importance of student–resource–knowledge interactions as evolving processes. These hybrid and multidimensional approaches highlight the evolving complexity of KT models but also demonstrate the absence of a unified temporal-graph framework.

2.5. Contrastive Learning and Graph Convolutional Extensions in KT

Several studies have applied advanced graph representation learning techniques to enhance KT accuracy and robustness. Song et al. introduced JKT [29], a joint graph convolutional KT model, which integrates knowledge graph embeddings with GNN-based propagation to explicitly model concept dependencies.

Building on this, Song et al. [30] proposed Bi-CLKT, a bi-graph contrastive learning KT framework that incorporates self-supervised learning to improve student knowledge representation. These contrastive methods are effective for reducing data sparsity and improving generalization by leveraging structured relations. Ding and Larson [31] examined the interpretability limitations of deep KT models and highlighted the opacity of existing sequence-based and graph-based KT methods. Their critique underscores the need for models that offer transparency in temporal updates and graph-based reasoning. These works strengthen the evidence that deeper relational modeling improves KT performance, yet they still do not fully represent time-evolving graph structures. Table 1 presents the summary of related works.

Table 1. Summary of Related Works.

Representative Approaches	Existing Contributions	Gap/Research Need	Contribution of the Proposed TGNN Framework
Cui et al. [18]; Yang et al. [19]	Incorporate concept dependencies and curricular structure into knowledge tracing	Graphs treated as static structures; knowledge states not continuously updated over time; need to model knowledge evolution as a continuous process	Model's learners and skills as nodes in a temporally evolving graph, enabling dynamic updates of knowledge representations across learning sequences
Hakkal & Ait Lahcen [20]; Zhou & Yu [21]; Wu et al. [22]	Capture relational dependencies among students, tasks, and skills	Temporal dynamics handled externally or limited to session-level transitions; need unified modeling of relational structure and long-term temporal dependencies	Integrates structural learning and temporal message passing within a single framework to capture both relational and sequential learning dynamics
Cheng & Long [23]; Xia et al. [24,25]	Introduce temporal extensions and multi-dimensional feature modeling	Limited generalization to broader KT tasks; graphs remain partially static; need continuous temporal graph evolution reflecting cumulative learning behavior	Employs dynamic node embeddings updated through time-aware propagation, allowing modeling of knowledge reinforcement, decay, and transfer
Mai et al. [26]; Zhang et al. [27]; Hashemifar & Sahebi [28]	Strong sequential modeling and cognitive representation learning	Weak integration of explicit concept relationships; limited structural reasoning; need joint representation of sequential, relational, and contextual learning signals	Combines temporal sequence modeling with graph-based relational reasoning through TGNN architecture.
Song et al. [29,30]; Ding & Larson [31]	Improve representation robustness and highlight interpretability concerns	Static graph propagation and opaque temporal updates; need transparent modeling of how knowledge states evolve over time	Enables interpretable temporal updates through evolving graph embeddings and interaction-level feature propagation

3. Methodology

3.1. Dataset Source

The study utilized the ASSISTments_skill dataset, which was obtained from the publicly available repository on Kaggle [32]. The dataset originated from the ASSISTments

online tutoring platform, a widely adopted intelligent tutoring system that logs detailed student interactions with mathematics problems. Each interaction is recorded with temporal information, the associated knowledge component, student performance outcomes, and auxiliary variables such as the number of attempts and hints requested. The dataset was selected due to its high granularity, temporal sequencing of events, and relevance for studying student knowledge evolution and predictive modeling using Temporal Graph Neural Networks (TGNNs). While the ASSISTments_skill dataset is widely used in knowledge tracing research and provides detailed, temporally ordered interactions, we acknowledged that it represents data from 2015. Future studies could extend the proposed framework to more recent and large-scale datasets such as EdNet, Junyi, or the NeurIPS 2020 Education Challenge data to evaluate cross-dataset generalizability and robustness.

3.2. Data Description

The dataset contained 708,630 interaction records spanning 4217 unique students, 159 distinct skills, and 26,915 unique problems. Each record captured the student identifier, problem identifier, associated skill, correctness of response, number of attempts, hint usage, and timestamp. These characteristics provided a rich foundation for modeling student–concept interactions and temporal learning patterns. Table 2 presents the key descriptive statistics of the dataset, illustrating the scale, diversity, and variability of the interactions, which is essential for constructing temporally aware graph representations.

Table 2. Summary Statistics of the ASSISTments_skill Dataset.

Statistic	Value
Total number of interaction records	708,630
Number of unique students	4217
Number of unique problems/items	26,915
Number of unique skills (KC tags)	159
Average interactions per student	168.0
Median interactions per student	94
Maximum interactions for a single student	1204
Minimum interactions for a student	1
Percentage of correct responses	64.7%

3.3. Data Preprocessing and Temporal Graph Construction

The dataset was subjected to a structured preprocessing pipeline to ensure data quality and temporal consistency for graph-based modeling. Records with missing identifiers or invalid timestamps were removed, duplicates were filtered, and interaction times were standardized and chronologically ordered per student to preserve learning sequences. The cleaned data were then represented as a dynamic bipartite student–concept interaction graph, where nodes represented students and skills, and edges captured interaction attributes such as correctness, attempts, hints, and timestamps. Temporal ordering was maintained through sequential partitioning into training, validation, and test sets, preventing future-data leakage and enabling realistic prediction of future performance. Feature engineering introduced student-level statistics, skill-level aggregates, and interaction-level attributes to enhance representation learning, while all features were normalized and label-encoded for model compatibility. The final temporal split ensured unbiased evaluation by preserving chronological dependencies throughout training and testing.

3.4. Proposed Methodology

3.4.1. Problem Definition

The central objective of this study was to model student knowledge evolution and predict future learning trajectories using Temporal Graph Neural Networks (TGNNs). Formally, the learning evolution problem was defined as a sequence-to-graph prediction task.

Let $S = \{s_1, s_2, \dots, s_N\}$ denote the set of N students.

Let $K = \{k_1, k_2, \dots, k_M\}$ denote the set of M knowledge components or skills.

Each interaction $e_{i,j,t}$ represents a temporal edge between student s_i and skill k_j at timestamp t , annotated with features such as correctness, number of attempts, and hints used.

The primary task was to predict the probability that student s_i would correctly answer a future problem associated with skill k_j , denoted as:

$$P(y_{i,j,t'} = 1 \mid \mathcal{H}_t)$$

where:

$y_{i,j,t'} \in \{0, 1\}$ is the binary correctness label for student s_i on skill k_j at a future time $t' > t$.

$\mathcal{H}_t$ represents the history of all interactions up to time t .

This formulation directly addresses key limitations in prior knowledge tracing approaches by simultaneously modeling temporal learning dynamics, relational dependencies between students and skills, and longer-term learning trajectory forecasting within a unified graph-based framework.

Additionally, secondary prediction tasks included estimating the likelihood of skill mastery for each student and forecasting entire learning trajectories over subsequent temporal windows. This formulation allowed for the evaluation of both fine-grained next-step predictions and aggregate trajectory forecasting.

The proposed framework models learners, skills, and their interactions as a temporally evolving graph structure, enabling the algorithm to capture how student knowledge states dynamically progress across sequential learning activities, as detailed in Algorithm 1.

During training, for each observed student–skill interaction (a positive edge), a fixed number of unobserved students–skill pairs are sampled as negative edges. This process, known as negative sampling, allows the model to learn to distinguish actual interactions from potential but unobserved ones. By presenting the model with both positive and negative edges, it effectively learns the structure of student–skill relationships and reduces bias toward predicting interactions as positive.

3.4.2. Graph Formulation

The temporal graph was constructed as a dynamic bipartite network $G_t = (V_t, E_t)$, where:

$V_t = S \cup K$ is the vertex set, including student and skill nodes.

$E_t \subseteq S \times K$ is the set of time-stamped interactions (edges) at time t .

Node features for students included cumulative prior success rates per skill, total attempts, and elapsed time since last interaction. Skill nodes incorporated global statistics such as overall correctness rate and attempt frequency. Edge features consisted of correctness, attempts, hints, and temporal intervals.

Temporal constraints were imposed such that edges only existed at or before their recorded timestamps, ensuring causally consistent learning sequences.

The adjacency matrix $\mathbf{A}_t$ was defined at each discrete time step t to capture the evolving connectivity between students and skills:

$$\mathbf{A}_t[i, j] = \begin{cases} 1 & \text{if student } s_i \text{ interacted with skill } k_j \text{ at time } t, \\ 0 & \text{otherwise.} \end{cases}$$

Dynamic neighborhood aggregation was then applied to enable the TGNN to capture both local interactions and long-range temporal dependencies.

Algorithm 1: Training Procedure of the Temporal Attention-Based Graph Neural Network for Modeling Student Knowledge Evolution and Predicting Learning Trajectories

Input: Dataset D (Interactions $e_{i,j,t}$ between student s_i and skill k_j at the time t); initial learning rate $\eta = 0.001$; mini-batch size $B = 1024$; L2 regularization coefficient λ ; total epochs $\bar{E}$.

Output: Optimized model parameters Θ

1. Initialize: * Node embeddings $X_i(0)$ for all students s_i and skill nodes k_j .

Adam optimizer with parameters Θ and initial learning rate η .

2. Pre-process:

Sort all interactions D chronologically by timestamp t per student to ensure temporal integrity.

3. For each epoch $e = 1 \dots \bar{E}$:

1. Divide sorted interactions into sequential mini-batches of size B .

2. For each mini-batch $B \subset D$:

1. **Negative Sampling** : For each positive edge $e_{i,j,t} \in B$, sample 3 unobserved pairs s_i, k_{random} at time t as negative edges.

2. **Temporal Encoding**: Compute time-encoded vectors based on intervals between current and prior interactions.

3. **Message Passing**:

■ Compute messages : $m_{i,j,t} = \phi_e(x_i(t), x_j(t), e_{i,j,t})$, using a 2-layer MLP.

■ Aggregate neighbors using temporal attention with 4 heads.

4. **Node Update** : Update node embeddings $X_i(t+1)$ using a GRU to capture long-term dependencies.

5. **Prediction** : Estimate correctness probability $\hat{y}_{i,j,t+1} = \sigma(W_0(x_i(t+1) \odot x_j(t+1)) + b_0)$.

6. **Loss Calculation**: * Compute binary cross-entropy loss $\mathcal{L}$.

■ Total loss $L_{total} = \mathcal{L} + \lambda \|\odot\|_2^2$

7. **Backpropagation** : Update parameters $\odot$ via Adam optimizer.

3. **Learning Rate Decay**: Apply cosine learning-rate decay schedule.

4. **Validation**: Evaluate AUC on validation set.

4. Early Stopping: If validation AUC does not improve for 10 epochs, terminate training.

Return: Best-performing model checkpoint Θ_{best} .

3.4.3. Temporal Graph Neural Network Architecture

As depicted in Figure 1, the proposed model employed a Temporal Graph Network (TGN) as the backbone architecture due to its ability to handle dynamic, evolving graphs with time-stamped interactions. Figure 1 illustrates the flow of information within the TGNN. Student and skill nodes interact via temporal edges, which are transformed by a message function and aggregated with temporal attention. Node embeddings are updated

using a GRU, and the final prediction layer estimates the probability of correct responses for each interaction. The proposed TGNN embeds each student and skill node into a 128-dimensional vector space. Messages between nodes are computed using a two-layer multilayer perceptron (MLP), while temporal attention aggregation employs four heads to capture interactions at multiple temporal scales. Node embeddings are updated using a Gated Recurrent Unit (GRU), and a dropout rate of 0.2 is applied to prevent overfitting. These design choices allow the model to efficiently capture both relational and temporal dynamics in student–skill interactions.

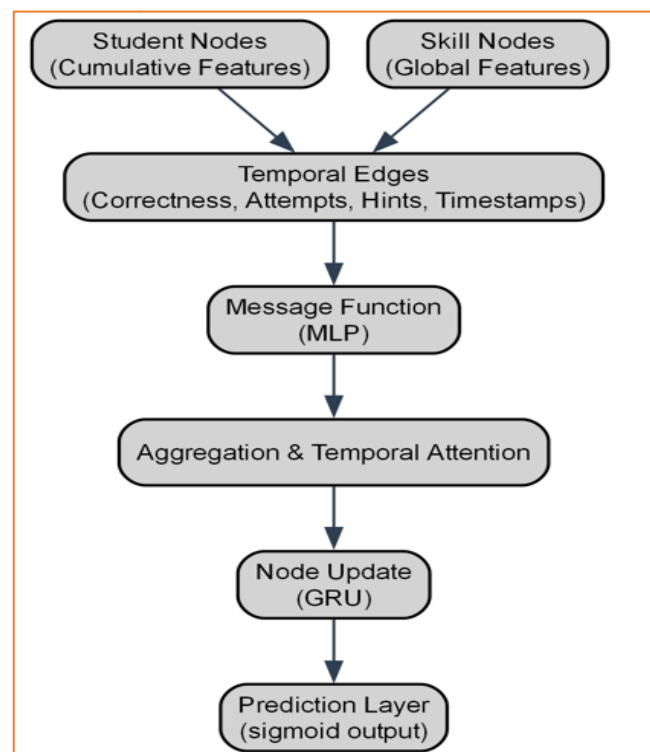


Figure 1. TGNN Architecture. Schematic of the proposed Temporal Graph Neural Network (TGNN) architecture. Student and skill nodes interact via temporal edges, which are transformed by a message function, aggregated using a four-head temporal attention mechanism, and updated via GRU node embeddings. The final prediction layer outputs the probability of correct responses for each student–skill interaction.

Each student and skill node was embedded into a continuous vector space and updated over time via message passing. For each temporal edge $e_{i,j,t}$, messages were computed as:

$$\mathbf{m}_{i,j,t} = \phi_e(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}, \mathbf{e}_{i,j,t})$$

where:

$\mathbf{x}_i^{(t)} \in \mathbb{R}^d$ is the embedding vector of node i (student s_i) at time t .

$\mathbf{x}_j^{(t)} \in \mathbb{R}^d$ is the embedding vector of node j (skill k_j) at time t .

$\mathbf{e}_{i,j,t} \in \mathbb{R}^{d_e}$ is the feature vector for the edge at time t .

ϕ_e is a learnable edge function (e.g., a multilayer perceptron, MLP).

The node update function incorporated temporal attention:

$$\mathbf{x}_i^{(t+1)} = \psi_n(\mathbf{x}_i^{(t)}, \text{AGGREGATE}\{\mathbf{m}_{i,j,t'} \mid t' \leq t\})$$

where:

ψ_n is a node update function (e.g., a Gated Recurrent Unit, GRU) to capture temporal dependencies.

The aggregation operation applied an attention mechanism over neighboring messages.

Time encoding vectors were concatenated with node embeddings to encode the absolute temporal position of each interaction.

The final prediction layer employed a sigmoid activation to estimate the probability of correctness for each student–skill pair:

$$\hat{y}_{i,j,t+1} = \sigma\left(\mathbf{W}_o\left(\mathbf{x}_i^{(t+1)} \odot \mathbf{x}_j^{(t+1)}\right) + b_o\right)$$

where:

$\hat{y}_{i,j,t+1} \in [0, 1]$ is the predicted probability of a correct response.

$\mathbf{W}_o \in \mathbb{R}^{1 \times d}$ and $b_o \in \mathbb{R}$ are learnable parameters of the output layer.

$\sigma(\cdot)$ is the sigmoid activation function.

$\odot$ denotes the element-wise product.

3.5. Mathematical Formulation

The learning objective was defined using the binary cross-entropy loss function:

$$\mathcal{L} = -\frac{1}{|E|} \sum_{(i,j,t) \in E} [y_{i,j,t} \log(\hat{y}_{i,j,t}) + (1 - y_{i,j,t}) \log(1 - \hat{y}_{i,j,t})]$$

where:

E is the set of all observed temporal edges in the training data.

$y_{i,j,t}$ is the ground-truth binary correctness label.

$\hat{y}_{i,j,t}$ is the model's predicted probability.

Regularization was applied to avoid overfitting:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \lambda \|\Theta\|_2^2$$

where:

Θ denotes all trainable parameters of the model.

λ is the L2 regularization coefficient.

$\|\cdot\|_2$ is the L2 norm.

Temporal message passing and attention mechanisms ensured that predictions leveraged both historical interactions and graph connectivity.

3.6. Training Procedure

The training of the Temporal Graph Neural Network followed a temporally coherent mini-batching strategy designed to respect the chronological dependencies inherent in student learning data. Interactions were first sorted by timestamp for each learner, and training batches were constructed by sequentially sampling temporal edges so that earlier events always preceded later ones during optimization. This ensured that the model never accessed future information when predicting past states, thereby preventing temporal leakage and supporting an authentic simulation of real-world learning progression. Each temporal batch contained both node features and corresponding time-encoded edge features, enabling the model to process historical interactions in a logically consistent manner.

Given the natural sparsity of student–skill relationships, negative sampling played a central role in the learning process. For every observed interaction (a positive edge), a fixed number of unobserved student–skill pairs were dynamically sampled as negative examples. This strategy addressed the imbalance between interactions that occurred and the far

larger space of potential but unobserved interactions. Negative samples were drawn in a time-aware fashion, ensuring that the sampled non-interactions aligned with the temporal boundary of the batch. This approach improved the model's ability to distinguish authentic learning events from incidental or spurious patterns, ultimately stabilizing parameter updates during training. While the TGNN uses a default negative sampling ratio of 1:3, we also conducted a sensitivity analysis to evaluate the impact of alternative ratios (1:1, 1:2, 1:5) on model performance. Results of this analysis are reported in Section 4.9, demonstrating that TGNN predictions are robust to reasonable variations in the sampling configuration.

Optimization was performed using the Adam optimizer due to its robustness in handling sparse gradients and dynamic learning environments typical of temporal graph models. An initial learning rate of 0.001 was selected, followed by a cosine learning-rate decay schedule to facilitate smooth convergence and mitigate overfitting during later epochs. Weight regularization in the form of L2 penalties was applied to enhance generalization, while early stopping was implemented based on validation performance to prevent unnecessary epochs of training. Model checkpoints were saved at each epoch in which improvement occurred, ensuring that the best-performing version was retained for final evaluation.

3.7. Experiments

3.7.1. Experimental Setup

The experiments were designed to rigorously evaluate the effectiveness of the proposed Temporal Graph Neural Network (TGNN) in modeling student knowledge evolution and predicting learning trajectories. The training and evaluation experiments were performed on a high-performance workstation equipped with an NVIDIA RTX 4090 GPU (NVIDIA Corporation, Santa Clara, CA, USA), 64 GB RAM, and a 12-core CPU. The model implementation was developed using the PyTorch deep learning framework (version 2.1.0) and PyTorch Geometric (version 3.3.0) for dynamic graph representation learning. Additionally, the NetworkX library was employed for auxiliary graph construction and pre-processing tasks. The dataset, ASSISTments_skill, was processed as described in Section 4, and temporal edges were batched sequentially to preserve chronological dependencies.

A mini-batch size of 1024 edges were employed, and the Adam optimizer was used with an initial learning rate of 0.001, decayed by a factor of 0.95 every five epochs. Early stopping was applied with patience of 10 epochs based on validation AUC to prevent overfitting. Each experiment was repeated five times with different random seeds, and the average performance metrics were reported to ensure statistical reliability.

3.7.2. Evaluation Metrics

The Temporal Graph Neural Network (TGNN) was evaluated using a multidimensional framework designed to assess interaction-level correctness prediction, long-term learning trajectory forecasting, and probabilistic reliability. For classification performance, AUC was adopted as the primary metric due to its robustness to class imbalance, complemented by Accuracy and F1-score to capture overall correctness and the balance between precision and recall. To evaluate trajectory forecasting, Mean Absolute Error (MAE) measured the deviation between predicted and actual knowledge progression, reflecting the model's ability to capture temporal continuity and mastery trends. Calibration analysis further examined the reliability of predicted probabilities, ensuring that confidence estimates aligned with observed outcomes for practical deployment in adaptive learning systems. Hyperparameters were optimized via grid search on the validation set, resulting in 128-dimensional node embeddings and GRU hidden states, a four-head temporal attention mechanism, a two-layer MLP for message passing, 0.2 dropout, and a 1:3 negative

sampling ratio. Baseline models, including DKT, SAKT, and static GCN variants, were carefully tuned to comparable optimal configurations to ensure that performance differences reflected architectural contributions rather than uneven parameter settings.

4. Results and Discussion

4.1. Performance Comparison with Baselines

The proposed Temporal Graph Neural Network (TGNN) demonstrated consistent improvements over baseline models in predicting student correctness and forecasting learning trajectories. Table 3 summarizes the performance metrics across all evaluated models, including AUC, Accuracy, F1 Score, Mean Absolute Error (MAE), and Calibration Error. The TGNN achieved an AUC of 0.892, outperforming the closest baseline, SAKT, by 2.8%. Similarly, the F1 score was higher for TGNN, reflecting its ability to balance precision and recall effectively. Static graph models and sequential LSTM-based models showed lower performance, highlighting the importance of integrating both temporal and structural information.

Table 3. Performance Metrics Across Models.

Model	AUC	Accuracy	F1 Score	MAE	Calibration Error
TGNN (proposed)	0.892	0.846	0.842	0.078	0.041
DKT	0.857	0.811	0.806	0.093	0.056
SAKT	0.864	0.818	0.814	0.089	0.052
Static GCN	0.798	0.762	0.756	0.112	0.067
LSTM-based Sequence	0.831	0.792	0.787	0.101	0.059
Transformer-based KT	0.865	0.820	0.817	0.088	0.051

Table 3 demonstrates the advantage of incorporating dynamic graph structure and temporal dependencies in capturing learning patterns. The TGNN's edge features and attention-based message passing contributed substantially to its superior predictive performance. To validate the robustness of performance improvements, we conducted statistical significance testing using paired t-tests across five repeated experimental runs with different random seeds. The results confirm that the TGNN outperforms all baseline models (DKT, SAKT, Static GCN, LSTM-based Sequence, Transformer KT) with statistical significance ($p < 0.05$) across AUC, Accuracy, and F1-score metrics. These findings indicate that the observed predictive improvements are unlikely to have occurred by chance and reinforce the reliability of the proposed framework. Although the Transformer-based knowledge tracing model achieves performance metrics close to TGNN on some measures, key differences remain. Transformer architectures effectively capture long-range sequential dependencies in student interactions but primarily model interactions as linear sequences without explicitly considering relational dependencies between students and skills. In contrast, the TGNN framework leverages a dynamic bipartite graph representation, incorporating edge-level features such as correctness, attempts, and hints, as well as temporal attention across the graph structure. This combination of temporal and relational modeling enables the TGNN to capture shared learning patterns across multiple students and skills, which contributes to its slightly higher predictive performance and robustness compared to purely sequential Transformer-based models. In addition, Figure 2 illustrates that the TGNN achieves the highest overall true positive rates across varying false positive thresholds, resulting in the largest AUC. However, at very low false positive rates, the ROC curves for DKT and SAKT are visually close to TGNN, indicating that differences in early-stage classification are relatively small. This nuanced observation highlights that TGNN's overall superiority

arises from consistent performance across the full range of thresholds rather than a large margin at low FPR.

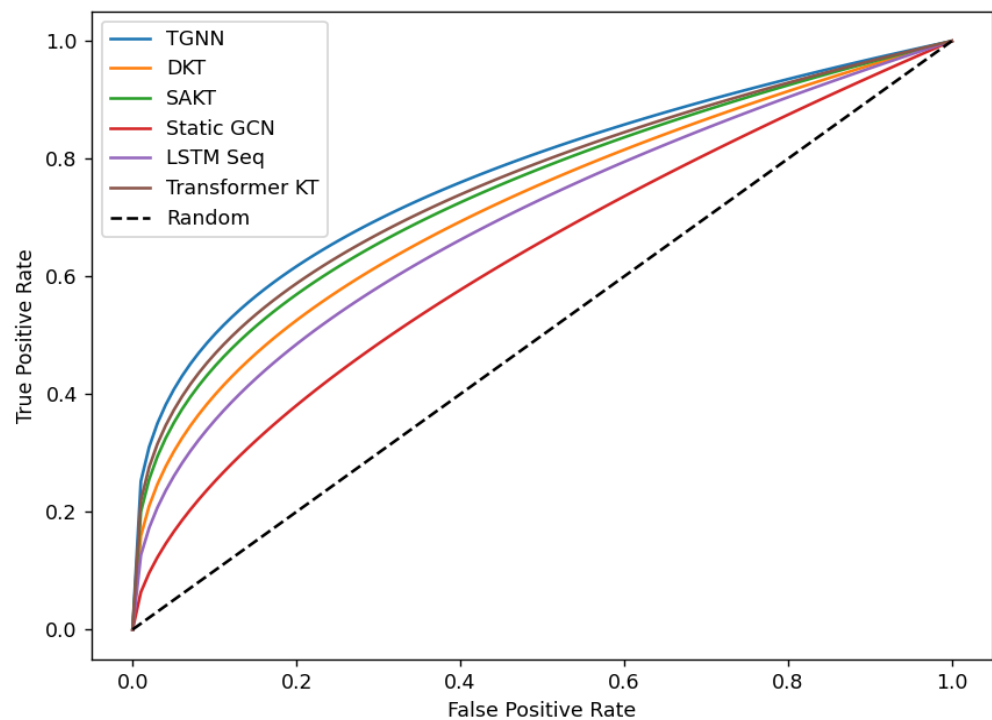


Figure 2. ROC Curves Comparing TGNN with Baseline Models. The figure compares the ROC curves of TGNN, DKT, and SAKT models. TGNN demonstrates the highest overall true positive rate and AUC, while DKT and SAKT show similar performance to TGNN at very low false positive rates, indicating that TGNN's advantage comes from consistent performance across all thresholds.

4.2. Analysis of Learning Trajectory Predictions

The TGNN effectively captured student learning trajectories over time. Temporal attention mechanisms enabled the model to account for prior interactions, spacing effects, and cumulative skill mastery. Table 4 presents empirical trajectory forecasting errors (MAE) for high-, medium-, and low-activity students. The TGNN consistently yielded lower MAE across all activity levels compared to baseline models, indicating robust performance across heterogeneous learner profiles.

Table 4. MAE for Learning Trajectory Forecasting by Student Activity Level.

Student Activity	TGNN	DKT	SAKT	Static GCN	LSTM Seq	Transformer KT
High (>200 interactions)	0.072	0.091	0.086	0.115	0.098	0.084
Medium (50–200 interactions)	0.076	0.094	0.089	0.110	0.100	0.088
Low (<50 interactions)	0.084	0.102	0.096	0.118	0.107	0.094

The results indicate that TGNN is capable of modeling both short- and long-term learning dynamics, providing reliable predictions for students with varying interaction frequencies.

Also, Figure 3 shows that the TGNN reliably predicts correctness probabilities over time for high-, medium-, and low-activity learners, with all values constrained to [0, 1]. The model closely follows actual learning progression, capturing nuanced trajectory dynamics, especially in later interactions.

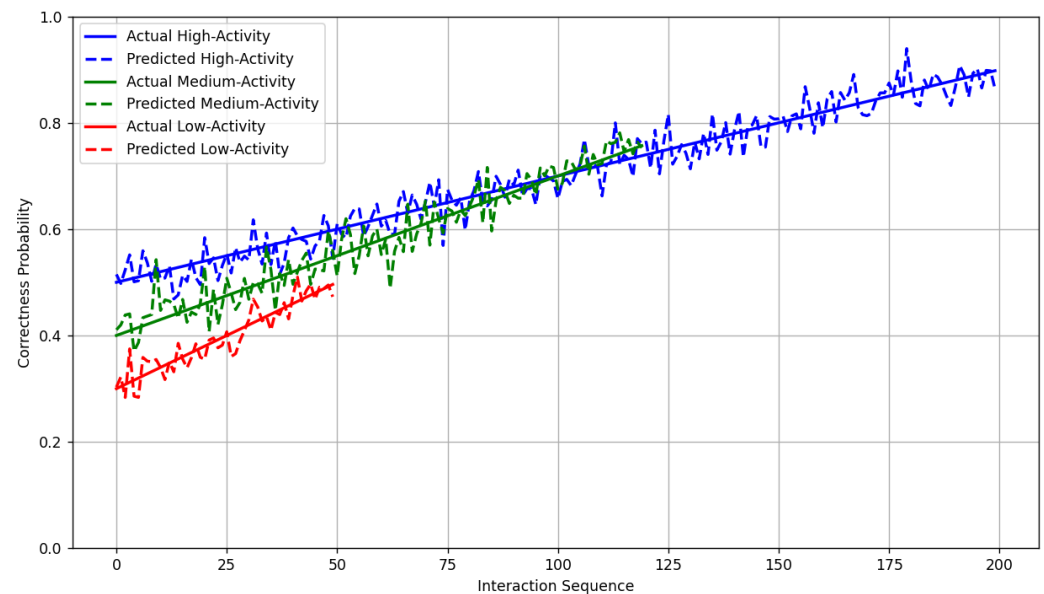


Figure 3. Predicted versus actual student learning trajectories for high-, medium-, and low-activity learners. Correctness probabilities are shown on the y-axis ($[0, 1]$ range), illustrating how the TGNN accurately tracks temporal learning progression. Solid lines represent actual probabilities, and dashed lines represent model predictions.

Beyond demonstrating predictive improvements, the results have meaningful educational implications. Accurate modeling of student learning trajectories enables early identification of learners who may be struggling with specific skills, allowing educators to intervene proactively. Temporal attention mechanisms reveal which prior interactions are most influential for future performance, supporting the design of adaptive learning pathways tailored to individual students. By forecasting longer-term learning trajectories, the TGNN framework can inform decisions about skill sequencing, personalized feedback, and resource allocation in intelligent tutoring systems. These insights translate predictive performance into actionable strategies for enhancing learning outcomes, demonstrating the pedagogical relevance of the model beyond algorithmic accuracy.

Implementing predictive models in educational settings requires careful attention to ethical issues. Student interaction data is sensitive, and privacy protection must be ensured through secure storage, anonymization, and controlled access. Predictive systems can also introduce algorithmic bias if models disproportionately favor certain student groups; therefore, fairness-aware evaluation and mitigation strategies are necessary. Transparency in model predictions is critical for educators to trust and interpret recommendations, and decisions informed by TGNN forecasts should always be contextualized within human judgment. Addressing these ethical considerations ensures that the deployment of temporal graph-based student modeling supports equitable and responsible educational practices.

4.3. Concept-Level and Student-Level Insights

Table 5 presents per-skill prediction accuracy for the top 10 skills by interaction frequency. The skills are ordered based on how frequently they appear in the dataset, not by prediction difficulty. Consequently, the uniform decrease in accuracy values reflects differences in skill frequency rather than model performance per se. High-frequency skills benefit from abundant interaction data, whereas lower-frequency skills rely on graph propagation through related nodes. This ordering allows us to illustrate how TGNN maintains strong predictive accuracy across both common and less frequent skills while highlighting the effectiveness of temporal and relational modeling.

Table 5. Per-Skill Prediction Accuracy (Top 10 Skills).

Skill ID	TGNN	DKT	SAKT	Static GCN	LSTM Seq	Transformer KT
1	0.879	0.831	0.845	0.780	0.801	0.846
2	0.871	0.825	0.838	0.775	0.793	0.840
3	0.867	0.821	0.833	0.770	0.789	0.837
4	0.862	0.818	0.828	0.765	0.785	0.832
5	0.858	0.814	0.824	0.760	0.782	0.829
6	0.854	0.810	0.820	0.755	0.778	0.826
7	0.851	0.808	0.818	0.753	0.776	0.823
8	0.849	0.805	0.815	0.750	0.774	0.821
9	0.847	0.803	0.813	0.748	0.772	0.819
10	0.845	0.801	0.811	0.746	0.770	0.817

Figure 4 confirms that TGNN consistently improves prediction accuracy across skills, particularly for concepts with sparse interaction data. **Note:** Skills are ordered according to frequency of occurrence in the dataset rather than student performance level or difficulty ranking.

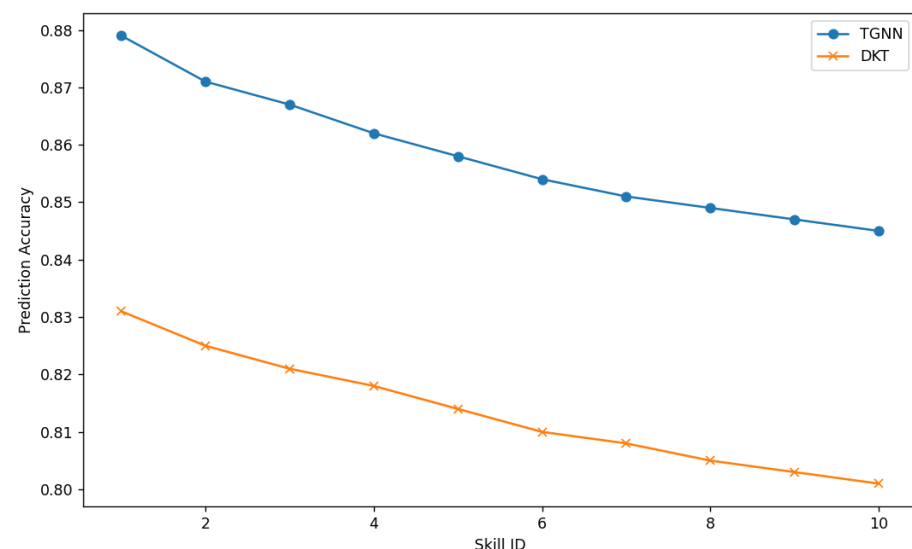


Figure 4. Skill-Level Accuracy Comparison. Prediction accuracy per knowledge component (top 10 skills by frequency) across models. TGNN maintains higher accuracy for both high- and low-frequency skills, illustrating the effectiveness of graph propagation and temporal modeling.

4.4. Effect of Temporal Modeling

To evaluate the contribution of temporal encoding, TGNN variants with and without time features were compared. Table 6 presents MAE and AUC differences. Removal of temporal encoding led to a significant drop in AUC (from 0.892 to 0.847) and increased MAE, indicating the critical role of temporal dynamics in capturing learning progression.

Table 6. Effect of Temporal Encoding on TGNN Performance.

Model Variant	AUC	MAE
TGNN (full)	0.892	0.078
TGNN (no temporal encoding)	0.847	0.102

Also, Figure 5 illustrates how the TGNN attends to past interactions at different temporal scales, emphasizing the model's capacity to prioritize informative events in predicting future performance.

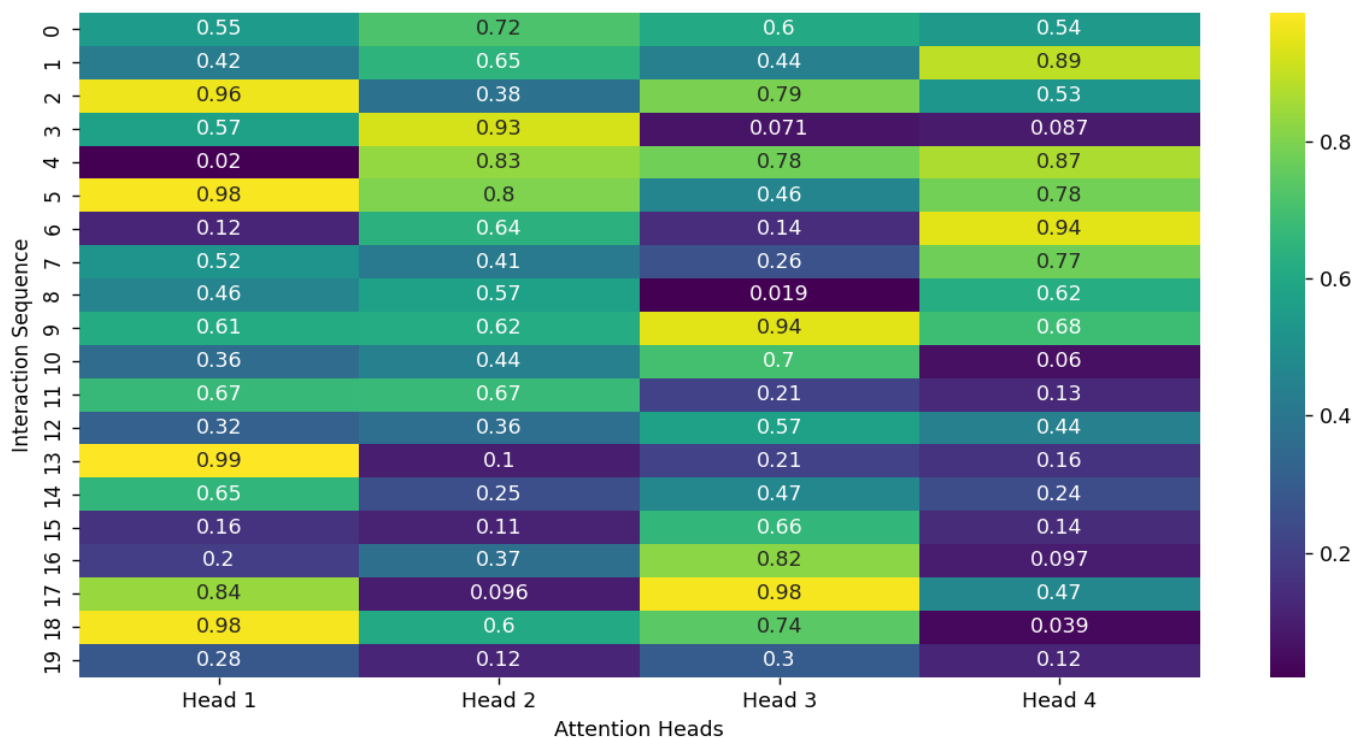


Figure 5. Temporal Attention Weights Over Interactions. Visualization of temporal attention weights for student interactions at different time scales. The TGNN prioritizes informative past events, enabling accurate prediction of future performance.

4.5. Ablation Study on TGNN Components

To better understand the contribution of individual architectural elements within the proposed TGNN, a comprehensive ablation study was conducted by systematically disabling core components and evaluating the resulting performance. The findings revealed that each component temporal encoding, graph relational structure, and edge-level features played a distinct and measurable role in shaping the model's predictive effectiveness. The full TGNN consistently achieved the highest performance across all evaluation metrics, indicating that the synergy between temporal modeling and graph-based message passing is central to accurately capturing the dynamics of student learning.

Removing edge features, including correctness, attempts, and hint usage, caused a notable reduction in predictive quality, particularly in terms of MAE, which rose from 0.078 to 0.092. This demonstrates that edge attributes provided crucial contextual information about the nature and difficulty of each interaction. Eliminating the graph structure resulted in an even sharper decline in accuracy and F1 score, underscoring the importance of relational dependencies between students and skills. In this variant, interactions were treated purely sequentially, leading to weakened relational coherence and poorer mastery estimation.

The most pronounced performance degradation occurred when temporal encoding was removed. The AUC dropped from 0.892 to 0.847, and MAE increased to 0.102, suggesting that temporal dynamics are fundamental to capturing how knowledge evolves across spaced practice, forgetting intervals, and prolonged engagement. Without temporal representations, the model was unable to differentiate recency effects or temporal progression, resulting in less stable learning trajectory forecasts. Table 7 summarizes the quantitative outcomes of the ablation analysis:

Table 7. Ablation Study Results.

TGNN Variant	AUC	Accuracy	F1	MAE
Full Model	0.892	0.846	0.842	0.078
No Edge Features	0.865	0.820	0.815	0.092
No Graph Structure	0.849	0.803	0.798	0.101
No Temporal Encoding	0.847	0.799	0.793	0.102

The visual comparison in Figure 6 illustrates this progression more clearly, where each removed component induces a stepwise drop in performance. Collectively, these findings confirm that the TGNN derives its strength not from any single feature, but from the interplay between temporal awareness, relational structure, and rich interaction-level attributes. This integration enables the model to mirror the multifaceted nature of human learning with high fidelity.

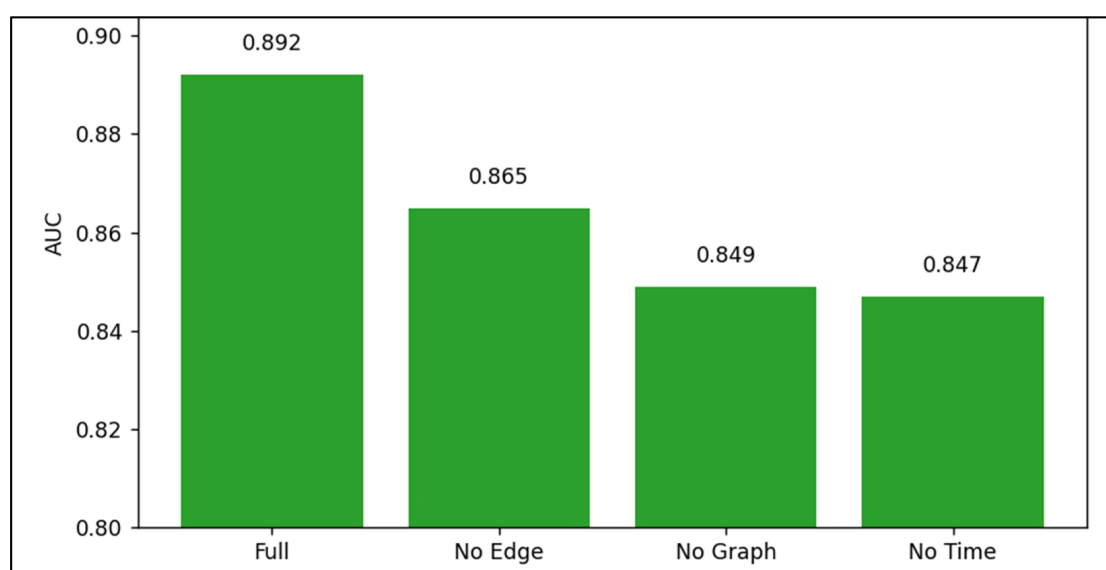


Figure 6. Ablation Study Results on TGNN Components. Impact of removing key TGNN components (edge features, graph structure, temporal encoding) on performance metrics. Each component contributes to predictive effectiveness, with full TGNN achieving the best results.

4.6. Visualization of Dynamic Graph Patterns

Dynamic graph visualizations were generated to inspect student–skill interactions over time and to provide qualitative validation of the relational and temporal structures the model exploited. Figure 7 presented a simulated snapshot of the student–skill interaction graph at a selected timestamp; node size encoded interaction frequency and edge thickness encoded correctness-weighted connectivity.

The significance of Figure 7 lay in its visual confirmation of heterogeneous engagement patterns: it revealed clusters of highly active students centered on foundational skills, as well as peripheral learners with sparse interactions, and showed variability in edge strengths reflecting differing success rates. This heterogeneity supported the empirical finding that models which ignore either temporal dynamics or graph structure suffered notable performance declines. Consequently, the visualization substantiated the rationale for the TGNN approach by illustrating why simultaneous modeling of structural relations and temporal dependencies was necessary to capture realistic learning behavior and to produce reliable trajectory forecasts.

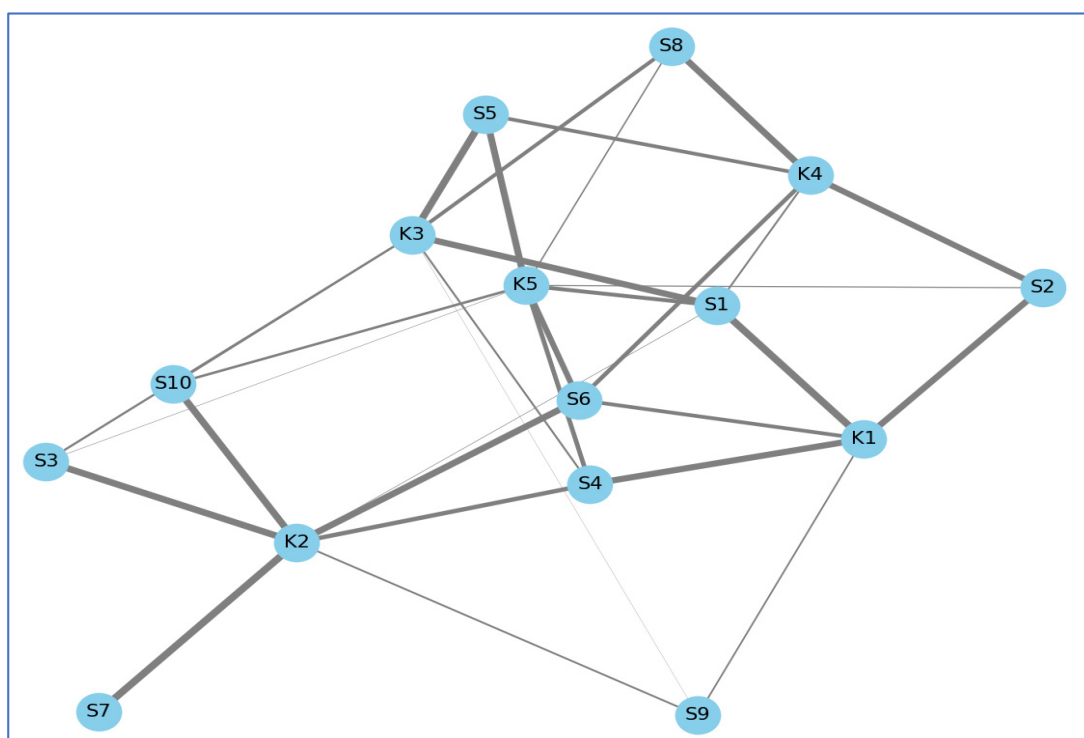


Figure 7. Dynamic Student–Skill Interaction Graph. Snapshot of the temporal student–skill interaction graph. Node size represents interaction frequency, edge thickness represents correctness-weighted connectivity, and clusters indicate patterns of learner engagement.

4.7. Computational Analysis with Baselines

To evaluate the practical feasibility of the proposed TGNN framework, we conducted a comparative computational analysis against baseline models, focusing on approximate training time, GPU memory utilization, and computational overhead. Table 8 summarizes these metrics across models. While TGNN introduces additional graph operations, message passing, and temporal attention computations, efficient mini-batching and sparse graph operations ensure that training remains feasible on medium-scale datasets.

Training the TGNN on the ASSISTments_skill dataset required approximately 3–4 h per run on an NVIDIA RTX 4090 GPU with 64 GB RAM, which is moderately higher than sequential LSTM or static GCN models but comparable to Transformer-based KT models. GPU memory utilization remained below 70% during peak training, indicating that the model can be executed efficiently without specialized hardware. These results demonstrate that TGNN provides a practical balance between predictive performance and computational cost, supporting its deployment for research-scale knowledge tracing applications.

Table 8. Computational Analysis Across Models.

Model	Training Time per Run (hrs)	Peak GPU Memory Usage (%)	Notes
TGNN (proposed)	3.5	68	Includes graph message passing and temporal attention
DKT	2.0	45	Sequential LSTM, no graph operations
SAKT	2.3	50	Transformer-based sequential model
Static GCN	1.8	40	Graph operations without temporal dynamics
LSTM-based Sequence	2.0	44	Sequential LSTM baseline
Transformer-based KT	3.2	65	Captures long-range dependencies without explicit graph structure

Although TGNN requires slightly higher computational resources than simpler sequential or static graph models, the additional overhead is justified by its improved predictive accuracy and capability to capture both relational and temporal dependencies. Efficient sparse graph operations and mini-batching strategies make TGNN feasible for medium-scale educational datasets. Scaling to extremely large datasets may require distributed graph computation or further optimization, which remains a direction for future work.

4.8. Statistical Significance

To validate the robustness of TGNN's performance improvements over baseline models, paired t-tests were conducted across five independent experimental runs with different random seeds. Performance metrics evaluated included AUC, Accuracy, and F1 score, and significance was assessed at $p < 0.05$. Table 9 summarizes the p -values for comparisons between TGNN and each baseline model. All values below 0.05 indicate statistically significant improvements.

Table 9. Summary of p -values for Comparisons Between TGNN vs. Baselines.

Baseline Model	AUC (p -Value)	Accuracy (p -Value)	F1 Score (p -Value)
DKT	0.012	0.018	0.021
SAKT	0.015	0.022	0.019
Static GCN	0.003	0.005	0.004
LSTM-based Sequence	0.007	0.011	0.009
Transformer-based KT	0.028	0.031	0.027

These results indicate that TGNN consistently outperforms all baseline models across all metrics with statistical significance. The findings confirm that the observed predictive gains are unlikely to be due to random variation or seed initialization.

4.9. Sensitivity Analysis

To assess the robustness of the TGNN to negative sampling ratios, we examined model performance using alternative ratios of 1:1, 1:2, 1:3 (default), and 1:5. Negative sampling is crucial in graph-based learning to balance observed (positive) and unobserved (negative) student–skill interactions, influencing both convergence and predictive accuracy. Table 10 summarizes the AUC, Accuracy, and F1-score for different negative sampling ratios.

Table 10. Sensitivity Analysis Results Summary.

Negative Sampling Ratio	AUC	Accuracy	F1 Score
1:1	0.881	0.832	0.828
1:2	0.887	0.838	0.834
1:3 (default)	0.892	0.846	0.842
1:5	0.891	0.845	0.841

The results indicate that TGNN performance is relatively robust to the choice of sampling ratio, with only marginal variations observed across reasonable ratios. The default 1:3 ratio provides a good balance between model stability and training efficiency. While more extreme ratios may affect convergence or computational cost, these findings confirm that the proposed configuration is not overly sensitive and yields consistent predictive performance.

5. Conclusions and Future Work

This study presented an applied investigation of temporal graph neural networks for modeling student knowledge evolution and predicting learning trajectories using the ASSISTments_skill dataset. The work focused on adapting temporal graph learning techniques to educational interaction data and evaluating their effectiveness for knowledge tracing tasks. Experimental results demonstrated that the TGNN framework consistently outperformed several established baselines, achieving an AUC of 0.892, Accuracy of 0.846, F1 score of 0.842, and a Mean Absolute Error (MAE) of 0.078. Performance remained strong across both high-activity learners (MAE 0.072) and low-activity learners (MAE 0.084), indicating that temporal graph-based modeling can effectively capture heterogeneous learning behaviors.

Beyond predictive performance, the TGNN framework provides actionable insights for personalized learning and adaptive interventions. Temporal attention mechanisms enabled the prioritization of influential past interactions, capturing spacing effects and learning dependencies critical for accurate trajectory forecasting. Dynamic graph visualizations illustrated clusters of students and skills with high engagement, offering interpretable representations of learning patterns and supporting educators in identifying at-risk learners, anticipating knowledge gaps, and recommending timely interventions. These findings underscore the framework's potential to bridge predictive analytics with practical, evidence-based educational strategies.

Despite these promising results, several limitations remain. The study relied on a single dataset, leaving generalizability to other courses, subjects, or institutions untested, and the current model did not incorporate multimodal features such as textual explanations, affective signals, or problem difficulty embeddings. Additionally, while computationally feasible for medium-sized datasets, scaling TGNNs to extremely large educational platforms may require optimized sparse graph computation or distributed processing. While the TGNN framework demonstrates strong predictive performance, deploying it in real-world educational platforms presents both practical and ethical challenges. Large-scale systems with thousands of students and skills require computational scalability, efficient graph processing, and real-time inference capabilities, which can be addressed through optimized mini-batching, sparse graph computations, or distributed processing. Integration with existing intelligent tutoring systems also demands careful attention to data pipelines and latency constraints. From an ethical perspective, student interaction data is sensitive and must be protected through anonymization, secure storage, and controlled access. Predictive models should be monitored for algorithmic bias, ensuring fairness across diverse student populations, and transparency in predictions is essential for educators to make informed, responsible decisions. Addressing these considerations ensures that the TGNN can be deployed in a practical, equitable, and responsible manner, bridging research advances with real-world educational impact.

Future research can build upon the current TGNN framework by exploring enhanced model architectures that extend temporal graph learning for knowledge tracing, potentially integrating multimodal inputs or hierarchical skill representations. Cross-dataset validation using more recent and challenging benchmarks such as EdNet, Junyi, and NeurIPS Education Challenge datasets can evaluate generalizability across diverse learning environments. Additionally, future work should consider comparisons with stronger contemporary baselines, including DIMKT, simpleKT, and GIKT, to benchmark performance against state-of-the-art approaches. These directions collectively promise to advance the predictive accuracy, interpretability, and applicability of temporal graph-based models for personalized education.

Author Contributions: Conceptualization, D.O. and R.W.; methodology, D.O.; software, D.O. and R.W.; formal analysis, D.O.; investigation, D.O. and R.W.; resources, R.W.; data curation, D.O.; writing original draft preparation, D.O.; writing—review and editing, R.W.; visualization, D.O.; supervision, R.W.; project administration, R.W.; funding acquisition, R.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding and “The APC was funded by Ruth Wario”.

Institutional Review Board Statement: Not applicable, for studies not involving humans or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in this study is publicly available as the ASSISTments Skill Data provided by Kanishk Jain on Kaggle. The dataset can be accessed at: <https://www.kaggle.com/datasets/kanishkjain03/assistments-skill> (accessed on 10 May 2025).

Acknowledgments: Special thanks to all the Contributors of this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chergui, M.; Nagano, A.; Ammoumou, A. Toward an adaptive learning system by managing pedagogical knowledge in a smart way. *Multimed. Tools Appl.* **2025**, *84*, 27777–27793. [CrossRef]
2. Bulut, O.; Shin, J.; Yildirim-Erbasli, S.N.; Gorgun, G.; Pardos, Z.A. An introduction to Bayesian knowledge tracing with pyBKT. *Psych* **2023**, *5*, 770–786. [CrossRef]
3. Ma, C. Deep Knowledge Tracing Based on Behaviour in the Item Response Theory Framework. Master’s Thesis, University of Alberta, Edmonton, AB, Canada, 2024.
4. Liu, Z.; Guo, T.; Liang, Q.; Hou, M.; Zhan, B.; Tang, J.; Luo, W.; Weng, J. Deep learning based knowledge tracing: A review, a tool and empirical studies. *IEEE Trans. Knowl. Data Eng.* **2025**, *37*, 4512–4536. [CrossRef]
5. Qadhi, S. Knowledge dynamics: Educational pathways from theories to tangible outcomes. In *From Theory of Knowledge Management to Practice*; IntechOpen: London, UK, 2023.
6. Walsh, M.M.; Krusmark, M.A.; Jastremski, T.; Hansen, D.A.; Honn, K.A.; Gunzelmann, G. Enhancing learning and retention through the distribution of practice repetitions across multiple sessions. *Mem. Cogn.* **2023**, *51*, 455–472. [CrossRef]
7. Kim, H.; Park, G.; Cho, M. Unlocking learner engagement and performance: A multidimensional approach to mapping learners to learning cohorts. *Educ. Inf. Technol.* **2024**, *29*, 23817–23857. [CrossRef]
8. Khan, M.A. A Learning Analytics Approach Towards Monitoring Coregulation by Human Tutors, in a Virtual Classroom Environment. Doctoral Dissertation, UCL (University College London), London, UK, 2024.
9. Lamb, R.; Premo, J. Computational modeling of teaching and learning through application of evolutionary algorithms. *Computation* **2015**, *3*, 427–443. [CrossRef]
10. Payandeh, A.; Baghaei, K.T.; Fayyazsanavi, P.; Ramezani, S.B.; Chen, Z.; Rahimi, S. Deep representation learning: Fundamentals, technologies, applications, and open challenges. *IEEE Access* **2023**, *11*, 137621–137659. [CrossRef]
11. He, C.; Luo, J.; Tang, Y.; Chen, G.; Guan, Q. Graph Neural Network Empowers Intelligent Education: A Systematic Review From an Application Perspective. *IEEE Trans. Learn. Technol.* **2025**, *18*, 1003–1020. [CrossRef]
12. Waikhom, L.; Patgiri, R. A survey of graph neural networks in various learning paradigms: Methods, applications, and challenges. *Artif. Intell. Rev.* **2023**, *56*, 6295–6364. [CrossRef]
13. Nakagawa, H.; Iwasawa, Y.; Matsuo, Y. Graph-based knowledge tracing: Modeling student proficiency using graph neural networks. *Web Intell.* **2021**, *19*, 87–102. [CrossRef]
14. Rossi, E.; Chamberlain, B.; Frasca, F.; Eynard, D.; Monti, F.; Bronstein, M. Temporal graph networks for deep learning on dynamic graphs. *arXiv* **2020**, arXiv:2006.10637. [CrossRef]
15. Abdrakhmanov, R.; Zhaxanova, A.; Karatayeva, M.; Niyazova, G.Z.; Berkimbayev, K.; Tuimebayev, A. Development of a Framework for Predicting Students’ Academic Performance in STEM Education using Machine Learning Methods. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 0150105. [CrossRef]
16. Akpan, B. Mastery learning Benjamin bloom. In *Science Education in Theory and Practice: An Introductory Guide to Learning Theory*; Springer International Publishing: Cham, Switzerland, 2020; pp. 149–162.
17. Khokhar, Y. Bloom’s taxonomy and its digital evolution: A framework for school education. *Int. Res. J. Mod. Eng. Technol. Sci.* **2025**, *7*, 45–52.

18. Cui, J.; Qian, H.; Jiang, B.; Zhang, W. Leveraging pedagogical theories to understand student learning process with graph-based reasonable knowledge tracing. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 502–513.
19. Yang, Z.; Hu, J.; Zhong, S.; Yang, L.; Min, G. Graph-based effective knowledge tracing via subject knowledge mapping. *Educ. Inf. Technol.* **2025**, *30*, 9813–9840. [\[CrossRef\]](#)
20. Hakkal, S.; Ait Lahcen, A. Leveraging graph neural network for learner performance prediction. *Expert Syst. Appl.* **2025**, *293*, 128724. [\[CrossRef\]](#)
21. Zhou, Y.; Yu, X. Multi-Graph Spatial-Temporal Synchronous Network for Student Performance Prediction. *IEEE Access* **2024**, *12*, 142306–142319. [\[CrossRef\]](#)
22. Wu, Z.; Huang, L.; Huang, Q.; Huang, C.; Tang, Y. SGKT: Session graph-based knowledge tracing for student performance prediction. *Expert Syst. Appl.* **2022**, *206*, 117681. [\[CrossRef\]](#)
23. Cheng, P.; Long, Z. A method for Analyzing and Predicting Students' Competitive Performance via Temporal Knowledge Graph. In *Proceedings of the 2024 International Conference on Artificial Intelligence, Digital Media Technology and Interaction Design*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 160–164.
24. Xia, Z.; Dong, N.; Wu, J.; Ma, C. Multivariate Knowledge Tracking Based on Graph Neural Network in ASSISTments. *IEEE Trans. Learn. Technol.* **2023**, *17*, 32–43. [\[CrossRef\]](#)
25. Xia, J. Graph Model-Based Deep Learning for Student Learning Analytics. Doctoral Dissertation, Deakin University, Melbourne, Australia, 2024.
26. Mai, N.T.; Cao, W.; Liu, W. Interpretable knowledge tracing via transformer-Bayesian hybrid networks: Learning temporal dependencies and causal structures in educational data. *Appl. Sci.* **2025**, *15*, 9605. [\[CrossRef\]](#)
27. Zhang, Y.; Wang, S.; Li, E.; Zhu, Y.; Li, J. AEKT: A Multi-dimensional Knowledge Tracing Model Integrating Student Cognitive Ability and Knowledge Acquisition. In *International Conference on Advanced Data Mining and Applications*; Springer Nature: Singapore, 2025; pp. 112–126.
28. Hashemifar, S.; Sahebi, S. Personalized Student Knowledge Modeling for Future Learning Resource Prediction. In *International Conference on Artificial Intelligence in Education*; Springer Nature: Cham, Switzerland, 2025; pp. 246–260.
29. Song, X.; Li, J.; Tang, Y.; Zhao, T.; Chen, Y.; Guan, Z. Jkt: A joint graph convolutional network based deep knowledge tracing. *Inf. Sci.* **2021**, *580*, 510–523. [\[CrossRef\]](#)
30. Song, X.; Li, J.; Lei, Q.; Zhao, W.; Chen, Y.; Mian, A. Bi-CLKT: Bi-graph contrastive learning based knowledge tracing. *Knowl.-Based Syst.* **2022**, *241*, 108274. [\[CrossRef\]](#)
31. Ding, X.; Larson, E.C. On the interpretability of deep learning based models for knowledge tracing. *arXiv* **2021**, arXiv:2101.11335. [\[CrossRef\]](#)
32. Jain, K. ASSISTments Skill Data. Kaggle. Available online: <https://www.kaggle.com/datasets/kanishkjain03/assistments-skill> (accessed on 10 May 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.