Dual-Representation Neural Knowledge Tracing for Auditable AI Assessment in Higher Education

Xinyu Cai[1] , Qi Zhang[2] and ABDULLAH[1,*]
(1. College of Business, Jiaxing University, Jiaxing, Zhejiang 314001, China
(2. Jiaxing University, Jiaxing, Zhejiang 314001, China)

Abstract—This paper propose a Dual-Representation Neural Knowledge Tracing (DR-NKT) architecture as the core engine for auditable AI assessment pipelines in higher education, replacing conventional opaque scoring heuristics with a transparent and interpretable framework. The motivation stems from the urgent need to evaluate student-submitted AI projects — including trained models, scripts, and documentation—while producing a clear reasoning trace for each proficiency score. Our methodology integrates two parallel encoding streams. A Temporal Graph Convolutional Network (TGCN) processes sequential student interactions, such as submission timestamps, autograder logs, and version control metadata, by constructing a temporal graph where nodes represent interaction events and edges capture sequential dependencies through a gated propagation mechanism. Meanwhile, a Symbolic Ontology Parser (SOP) translates formal curriculum standards, derived from educational taxonomies, into differentiable concept vectors using a graph attention network that embeds both semantic content and prerequisite relationships. These two streams are synchronized via a cross-attention fusion module, which computes per-competency mastery scores. Furthermore, the attention weights explicitly link each score to specific interaction events, thereby providing an auditable justification path. The principal contribution is a fully differentiable and auditable assessment submodule that outputs both a skill mastery vector and a set of attention matrices. For example, a low score in "convolutional neural network design" is directly traceable to submissions where the student failed to implement crucial layers. The significance of this work lies in its ability to replace black-box evaluation with transparent reasoning, enabling instructors to audit assessments, provide targeted feedback, and ultimately foster more effective learning in AI education.

Index Terms—Auditable assessment, explainable artificial intelligence, graph neural networks, knowledge tracing.

I. Introduction

The rapid integration of artificial intelligence into higher education has fundamentally altered the landscape of teaching, learning, and assessment [1]. As AI systems become increasingly capable of generating code, models, and entire project pipelines, evaluating student submissions in AI-focused courses presents a novel challenge. Traditional assessment methods, such as multiple-choice exams or simple code completion tasks, are ill-suited to gauge a student's deep conceptual understanding and practical proficiency when the very tools they use can obscure the learning process [2]. For instance, a student might submit an impressively performing neural network, but the provenance of that result—whether it reflects independent mastery, reliance on pre-trained models, or even direct copying—remains opaque. Hence, there is a pressing need for an evaluation system that not only scores student work but also provides a transparent and auditable account of how that score was derived.

Contemporary approaches to automated assessment in education, particularly those underpinned by deep learning, often operate as black boxes. While models like Deep Knowledge Tracing [3] and Self-Attentive Knowledge Tracing [4] can predict student performance with high accuracy, they rarely offer explanations for their predictions. This opacity is problematic in a pedagogical context where actionable feedback is paramount. Explainable AI (XAI) techniques such as LIME [5] and SHAP [6] can provide post-hoc explanations, but these are approximations and may not faithfully reflect the model's internal reasoning. Furthermore, the digital transformation of higher education [7] calls for systems that align with established educational taxonomies, like Bloom's Taxonomy [8], to ensure pedagogical validity. As a concrete instantiation of these ideas, the "Heling'er" Cultural Intelligent Agent developed at Jiaxing University, whose main interface is shown in Figure 1, has been built upon the design philosophy of the proposed framework.

This work was supported by the Second Batch of Provincial-Level Undergr aduate Teaching Reform Filing Projects under the "14th Five-Year Plan" of th e Zhejiang Provincial Department of Education under Grant No. JGBA202437 2. *Corresponding author: ABDULLAH, abdullah1710bb@gmail.com.

Xinyu Cai is with the College of Business, Jiaxing University, Jiaxing, Zhejiang, China, 314001 (e-mail: caixinyu@zjxu.edu.cn). Qi Zhang is with the Jiaxing University, Jiaxing, Zhejiang, China, 314001 (e-mail: 00008227@zjxu.edu.cn). ABDULLAH is with the College of Business, Jiaxing University, Jiaxing, Zhejiang, China, 314001 (e-mail: abdullah1710bb@gmail.com).

Figure 1. The "Heling'er" Cultural Intelligent Agent at Jiaxing University, an AI assistant developed upon the design philosophy of the proposed framework.

To address these limitations, we introduce the Dual-Representation Neural Knowledge Tracing (DR-NKT) architecture, an inherently interpretable framework designed specifically for auditing AI assessment pipelines. The core innovation of DR-NKT lies in its dual-representation design, which operates on two parallel but interacting streams of information. First, a Temporal Graph Convolutional Network [9] (TGCN) processes the sequential interaction traces of a student—submission logs, autograder outputs, system prompts—by modeling them as a dynamic graph. This graph encodes not only the sequence of events but also their complex pairwise relationships, such as the dependency of a later submission on earlier model weights. Second, a Symbolic Ontology Parser (SOP) translates formal curriculum standards, derived from an educational taxonomy graph, into differentiable concept vectors using a knowledge graph embedding technique akin to TransE [10]. A cross-attention fusion module [11] then synchronizes these two streams, allowing the model to weigh the evidence from interaction traces against the defined learning objectives.

This design fundamentally replaces opaque scoring algorithms with a traceable reasoning mechanism. The principal contribution is a fully differentiable assessment module that outputs both a skill mastery vector and a set of attention matrices. For example, if a student receives a low score for "transfer learning implementation," the attention weights will explicitly highlight the submissions where the student did not properly apply pre-trained weights or misconfigured the fine-tuning phase. This capability enables instructors to audit the assessment, verify its alignment with curriculum standards, and deliver highly targeted feedback. Moreover, the approach directly integrates the digitalization of education [12] with intelligent evaluation, ensuring that AI-based assessments are not only efficient but also pedagogically sound and accountable.

To the best of our knowledge, this is the first work to combine a temporal graph neural network with a symbolic ontology parser for the explicit purpose of auditing AI assessments in higher education. While previous studies have applied knowledge tracing to educational data [3] [4], they have not addressed the unique challenge of evaluating student-generated AI artifacts within a curriculum-aligned framework. Likewise, neuro-symbolic AI research [13] has explored fusing neural networks with symbolic reasoning, but its application to assessment auditing remains largely unexplored. Our work thus bridges these fields, offering a practical and principled solution.

The remainder of this paper is organized as follows. Section 2 reviews related work on knowledge tracing, graph neural networks in education, and explainable AI for assessment. Section 3 defines the formal problem setting and presents necessary background concepts. Section 4 details the DR-NKT architecture, including the TGCN, SOP, and cross-attention fusion components. Section 5 describes our experimental setup, including a simulated dataset of student AI project submissions. Section 6 reports quantitative and qualitative results, analyzing the accuracy and interpretability of our framework. Section 7 discusses the implications, limitations, and directions for future research. Finally, Section 8 concludes the paper with a summary of our contributions and their potential impact on AI education.

II. Related Work

Automated assessment of student learning—particularly in project-based AI courses—has been approached from several distinct yet overlapping research directions. We review these strands below, situating our proposed Dual-Representation Neural Knowledge Tracing (DR-NKT) framework within the broader landscape.

A. Knowledge Tracing and Its Evolution

Knowledge tracing, the task of modeling a student's evolving mastery over a set of skills based on their past performance, has a long history in educational data mining. The seminal Bayesian Knowledge Tracing (BKT) model [14] employed a hidden Markov model to estimate the probability that a student has learned a skill. While elegant and interpretable, BKT assumes that skills are independent—an assumption that often fails in complex, hierarchical subjects like AI. Deep Knowledge Tracing (DKT) [3] replaced the Bayesian framework with a recurrent neural network (RNN), achieving significant improvements in predictive accuracy. However, DKT's internal representations are notoriously difficult to interpret; the model provides no explicit justification for its predictions. Subsequent work such as Self-Attentive Knowledge Tracing (SAKT) [4] improved upon DKT by using transformer architectures, yet the interpretability problem persisted. These models produce a single performance score without linking it to specific competencies or past interactions in a way that is accessible to instructors. Our proposed DR-NKT directly addresses this limitation by using a cross-attention mechanism to trace each competency score back to specific interaction events.

B. Graph Neural Networks in Education

The application of graph neural networks (GNNs) to educational tasks has gained traction in recent years. Graph-based Knowledge Tracing (GKT) [15] modeled the relationship between knowledge concepts as a static graph, where nodes represent skills and edges encode prerequisite or similarity relationships. GKT used a Gated Graph Neural Network [16] to propagate information across this concept graph, improving the modeling of skill dependencies. However, GKT considers the interaction sequence only implicitly, through the updating of node states, and does not explicitly model the temporal order of student actions. Our TGCN component differs in that it constructs a temporal graph from the sequence of student interactions themselves. Each interaction (e.g., a submission, a commit) is a node, and edges are weighted by the cosine similarity of their feature vectors. This design captures the dynamic evolution of a student's work process, such as cycles of debugging, hyperparameter tuning, and model retraining, which are critical for auditing project-based assessments.

Other approaches have used GNNs for modeling student knowledge from response logs [15], but these generally

operate on static concept graphs rather than temporal interaction graphs. Furthermore, they have not been applied to the specific challenge of evaluating AI project artifacts, which involve continuous performance metrics (e.g., loss curves) and code diffs, rather than discrete right/wrong answers.

C. Explainable AI for Educational Assessment

The growing demand for transparency in AI systems has led to a proliferation of explainable AI (XAI) methods, many of which have been applied to education. Post-hoc explanation techniques, such as LIME [5] and SHAP [6], have been used to interpret the predictions of black-box knowledge tracing models. However, these explanations are approximations of the model's behavior and may be unfaithful or unstable. Moreover, they are detached from the model's training objective, meaning the model itself may rely on spurious correlations that the explanations then misrepresent. Some researchers have advocated for inherently interpretable models, such as interpretable cognitive diagnostic models [17], which use attention mechanisms to link predictions to specific questions. Yet these models are typically designed for multiple-choice or short-answer assessments, not for evaluating the open-ended, multi-modal nature of AI project submissions.

Our DR-NKT framework takes an inherently interpretable approach by design. The cross-attention fusion module produces attention weights γ_{mj} that directly tie each competency mastery score μ_m to the interaction event vectors $\mathbf{h}_{v_j}^{(L)}$. There is no post-hoc approximation; the explanation is baked into the forward pass of the model. This approach aligns with the principles of *ante-hoc* interpretability [18], where the model's reasoning is transparent by construction.

D. Neuro-Symbolic AI and Ontology-Based Assessment

Neuro-symbolic AI [13] seeks to combine the pattern recognition capabilities of neural networks with the structured reasoning of symbolic AI. In educational contexts, symbolic ontologies (e.g., OWL [19] or RDF [20]) have been used to formalize curriculum standards and learning objectives. For example, an ontology might specify that "gradient descent optimization" is a prerequisite for "training a deep neural network." Previous work has used such ontologies for automated question generation [21] or for aligning assessment items with learning outcomes [22]. However, these approaches typically treat the ontology as a static knowledge base without integrating it into a differentiable learning pipeline. Our SOP component bridges this gap by using a graph attention network to embed symbolic ontology definitions into continuous concept vectors. These vectors are then directly used in the cross-attention fusion module, enabling the neural network to learn how to weigh evidence from interaction traces against specific curriculum-defined competencies.

III. Preliminaries

This section establishes the foundational concepts and formalisms necessary for understanding the Dual-Representation Neural Knowledge Tracing (DR-NKT) framework. We first review probabilistic models of student knowledge acquisition, then introduce graph neural network architectures for processing structured and temporal data, and finally discuss attention mechanisms for cross-modal alignment.

A. Foundations of Probabilistic Knowledge Tracing

Knowledge tracing tasks fundamentally aim to estimate a student's latent knowledge state L_t at time step t based on their observed interaction sequence $X_1, X_2, \ldots, X_t$. The classical Bayesian Knowledge Tracing (BKT) model [14] formalizes this as a hidden Markov model with two states per skill: "learned" and "not learned." The transition between these states is governed by a learn probability parameter $P(T)$, representing the chance that a student transitions from the unlearned to the learned state after an opportunity to practice. The probability that a student has mastered a skill given that they have not yet done so can be expressed recursively:

$$P(L_t = 1 | L_{t-1}) = P(L_{t-1} = 1) + P(L_{t-1} = 0) \cdot P(T) \quad (1)$$

While BKT provides an elegant and interpretable framework, its assumption of skill independence limits its applicability to complex, interconnected domains like artificial intelligence. Deep Knowledge Tracing (DKT) [3] addressed this limitation by replacing the discrete Bayesian states with a continuous hidden representation learned by a recurrent neural network. At each time step t, the model updates its hidden state $\mathbf{h}_t$ based on the current input $\mathbf{x}_t$ (encoding both the exercise identifier and whether the student answered correctly) and the previous hidden state:

$$\mathbf{h}_t = \text{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1}) \quad (2)$$

The output layer then predicts the probability of correct response for each skill. However, Equation 2's hidden state $\mathbf{h}_t$ is a dense, distributed representation that does not explicitly encode which skills have been mastered or why. This opacity is the primary motivation for our dual-representation approach, which separates temporal dynamics from symbolic knowledge representation.

B. Spectral and Spatial Graph Neural Networks

Graph neural networks (GNNs) provide a natural framework for modeling relationships between entities. In educational contexts, these relationships can represent prerequisite dependencies between skills [15], similarity between interaction events, or hierarchical structure in curriculum standards. Two primary paradigms for GNNs exist: spectral methods and spatial methods.

Spectral approaches operate on the graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{A}$ is the adjacency matrix and $\mathbf{D}$ is the degree matrix. The symmetric normalized Laplacian is commonly used to stabilize training:

$$\mathbf{L}_{sym} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \quad (3)$$

The eigenvectors of $\mathbf{L}_{sym}$ form a Fourier basis on the graph, allowing convolution operations to be defined in the spectral domain. Early spectral GNNs [23] achieved strong performance but suffered from computational inefficiency for large graphs and lack of spatial localization.

Spatial methods, conversely, define graph convolution directly on the node neighborhood through a message-passing framework. A generalized update for a node v at layer $l + 1$ aggregates features from its neighbors $\mathcal{N}(v)$ and combines them with its own features:

$$\mathbf{h}_v^{(l+1)} = \text{COMBINE}^{(l)}\left(\mathbf{h}_v^{(l)}, \text{AGGREGATE}^{(l)}\left(\{\mathbf{h}_u^{(l)} : u \in \mathcal{N}(v)\}\right)\right) \quad (4)$$

Popular instantiations include Graph Convolutional Networks (GCNs) [24], which use a simple mean aggregation, and Graph Attention Networks (GATs) [25], which learn attention weights over neighbors. The TGCN component in our framework extends this message-passing paradigm to temporal graphs, where edges represent sequential dependencies between student interaction events, weighted by the similarity of their feature representations.

C. Self-Attention and Cross-Modal Alignment

Transformer architectures [26] have revolutionized sequence modeling through the self-attention mechanism, which computes a weighted average of all sequence positions for each position. For a sequence of query, key, and value matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{n \times d}$, the attention output is:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V} \quad (5)$$

Cross-attention extends this idea to align two distinct modalities or representations. In educational assessment, one modality might be the temporal sequence of student interaction features, and the other might be the symbolic representation of curriculum-defined competencies. Cross-attention allows the model to compute how much evidence each interaction event provides for each competency, producing attention weights that serve as an inherent justification for the final assessment score. This mechanism directly addresses the interpretability gap in standard knowledge tracing models, as the attention weights γ_{mj} in our framework explicitly link a competency score μ_m to specific interaction events $\mathbf{h}_{v_j}^{(L)}$, enabling auditing and targeted feedback.

IV. An Interpretable Dual-Representation Framework for Neuro-Symbolic Knowledge Tracing and Assessment Auditing

The proposed Dual-Representation Neural Knowledge Tracing (DR-NKT) framework operates as an inherently interpretable assessment engine within a broader AI Model and Pipeline Assessment Submodule, as illustrated in Figure 2. This submodule ingests raw data from student interactions and automated grading systems, processes them through the DR-NKT engine, and outputs both transparent skill mastery profiles and audit trails to an analytics dashboard. We now detail the internal architecture of the DR-NKT engine, which comprises three principal components: the Temporal Graph Convolutional Network (TGCN) for encoding student interaction sequences, the Symbolic Ontology Parser (SOP) for embedding curriculum standards, and the cross-attention fusion module that synchronizes these parallel streams to produce auditable justifications for each competency score.

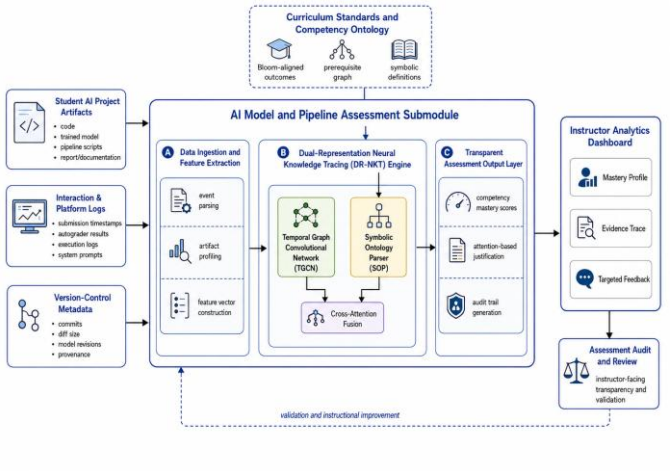

Figure 2. System Integration of AI Model and Pipeline Assessment Submodule

A. Temporal Graph Convolutional Network for Interaction Encoding

The TGCN component constructs a dynamic temporal graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ from the sequence of student interactions, where each node $v_j \in \mathcal{V}_t$ represents a single interaction event (e.g., a code submission, an autograder result, a version control commit) at time step j. Each node is initialized with a feature vector $\mathbf{x}_j \in \mathbb{R}^{d_x}$ that concatenates event-specific attributes, including metadata such as the submission timestamp, the autograder score, the change in validation accuracy relative to the previous submission, and the number of lines of code modified. Edges $\mathcal{E}_t$ connect temporally adjacent nodes; specifically, for any two nodes v_j and v_k where $j < k$, we introduce a directed edge from v_j to v_k weighted by the cosine similarity $\text{sim}(\mathbf{x}_j, \mathbf{x}_k)$ between their initial feature vectors. This weighting scheme captures non-sequential dependencies—for example, two submissions that both involve hyperparameter adjustments, even if separated by unrelated commits—and allows the graph to evolve dynamically as new interactions are appended.

The TGCN propagates information through this dynamic graph using a gated propagation mechanism that combines temporal locality with long-range dependencies across the interaction history. For a given node v at layer l, its hidden state $\mathbf{h}_v^{(l)}$ is updated by aggregating messages from its first-order neighbors $\mathcal{N}(v)$ and then fusing this aggregated information with the node's previous state through a Gated Recurrent Unit (GRU) [27]. The message aggregation step normalizes contributions from each neighbor $u \in \mathcal{N}(v)$ by the product of the square roots of their respective neighborhood sizes, ensuring stability across nodes with varying degrees:

$$\mathbf{m}_v^{(l)} = \sum_{u \in \mathcal{N}(v)} \frac{1}{\sqrt{|\mathcal{N}(v)||\mathcal{N}(u)|}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l)} \quad (6)$$

where $\mathbf{W}^{(l)} \in \mathbb{R}^{d_h \times d_h}$ is a learnable weight matrix shared across all nodes at layer l. The aggregated message $\mathbf{m}_v^{(l)}$ is then combined with the node's own previous hidden state $\mathbf{h}_v^{(l)}$ via a GRU cell, which controls the flow of information through reset and update gates:

$$\mathbf{z}_v^{(l+1)} = \sigma\left(\mathbf{W}_z \mathbf{m}_v^{(l)} + \mathbf{U}_z \mathbf{h}_v^{(l)} + \mathbf{b}_z\right) \quad (7)$$

$$\mathbf{r}_v^{(l+1)} = \sigma\left(\mathbf{W}_r \mathbf{m}_v^{(l)} + \mathbf{U}_r \mathbf{h}_v^{(l)} + \mathbf{b}_r\right) \quad (8)$$

$$\tilde{\mathbf{h}}_v^{(l+1)} = \tanh\left(\mathbf{W}_h \mathbf{m}_v^{(l)} + \mathbf{U}_h\left(\mathbf{r}_v^{(l+1)} \odot \mathbf{h}_v^{(l)}\right) + \mathbf{b}_h\right) \quad (9)$$

$$\mathbf{h}_v^{(l+1)} = \left(1 - \mathbf{z}_v^{(l+1)}\right) \odot \mathbf{h}_v^{(l)} + \mathbf{z}_v^{(l+1)} \odot \tilde{\mathbf{h}}_v^{(l+1)} \quad (10)$$

In Equations 7 through 10, $\mathbf{z}_v^{(l+1)}$ is the update gate vector, $\mathbf{r}_v^{(l+1)}$ is the reset gate vector, $\tilde{\mathbf{h}}_v^{(l+1)}$ is the candidate hidden state, $\odot$ denotes element-wise multiplication, and $\sigma(\cdot)$ is the logistic sigmoid function. The weight matrices $\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}_h \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{U}_z, \mathbf{U}_r, \mathbf{U}_h \in \mathbb{R}^{d_h \times d_h}$, along with bias vectors $\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}_h \in \mathbb{R}^{d_h}$, are learnable parameters. After propagating through L layers, the final hidden state $\mathbf{h}_v^{(L)}$ for each node v encodes a contextualized representation of that interaction event, incorporating information from temporally adjacent and semantically similar events in its neighborhood.

B. Symbolic Ontology Parser for Differentiable Concept Embedding

The SOP component translates symbolic curriculum standards—formally defined as a directed acyclic ontology graph $\mathcal{G}_\mathcal{O} = (\mathcal{C}, \mathcal{R})$ —into a set of differentiable concept vectors $\{\mathbf{e}_m \in \mathbb{R}^{d_e} : c_m \in \mathcal{C}\}$. Nodes $c_m \in \mathcal{C}$ represent individual competencies or learning objectives (e.g., "implement gradient descent," "design a convolutional layer"), while directed edges $(c_m, c_n) \in \mathcal{R}$ encode prerequisite relationships (e.g., "linear algebra" must precede "matrix multiplication for neural networks"). Each competency node c_m is associated with a rich symbolic definition, typically stored as an OWL [19] or RDF [20] description in an educational knowledge base.

To transform these symbolic definitions into continuous vectors, the SOP first extracts a semantic feature from each node's textual definition using a pretrained language model fine-tuned on educational text corpora. Specifically, we adopt Sentence-BERT [28] to encode the textual description $\text{desc}(c_m)$ into an initial feature vector $\boldsymbol{\phi}(c_m) \in \mathbb{R}^{d_\phi}$. This initial encoding captures the semantic content of the competency but does not yet incorporate the prerequisite structure of the ontology. To embed this structural information, we apply a graph attention network (GAT) [25] on $\mathcal{G}_\mathcal{O}$, which computes a weighted aggregation of features from each node's neighbors in the prerequisite graph. The attention coefficient β_{mn} that quantifies the importance of prerequisite node c_n for defining the concept embedding of node c_m is computed as:

$$\beta_{mn} = \frac{\exp\left(\text{LeakyReLU}(\mathbf{a}^\top[\mathbf{U}\boldsymbol{\phi}(c_m) \oplus \mathbf{U}\boldsymbol{\phi}(c_n)])\right)}{\sum_{k \in \mathcal{N}_\mathcal{O}(m)} \exp\left(\text{LeakyReLU}(\mathbf{a}^\top[\mathbf{U}\boldsymbol{\phi}(c_m) \oplus \mathbf{U}\boldsymbol{\phi}(c_k)])\right)} \quad (11)$$

where $\mathbf{U} \in \mathbb{R}^{d_e \times d_\phi}$ is a linear transformation matrix that projects both the target and neighbor features into a shared embedding space, $\mathbf{a} \in \mathbb{R}^{2d_e}$ is a learnable attention vector, $\oplus$ denotes vector concatenation, and $\mathcal{N}_\mathcal{O}(m)$ is the set of prerequisite nodes for c_m (i.e., nodes with edges pointing to c_m). The LeakyReLU activation introduces non-linearity with a small negative slope for negative inputs. The final differentiable concept vector $\mathbf{e}_m$ for competency c_m is then computed as the aggregation of the projected features of its prerequisites, weighted by the attention coefficients:

$$\mathbf{e}_m = \sigma\left(\sum_{n \in \mathcal{N}_\mathcal{O}(m)} \beta_{mn}\, \mathbf{U}\boldsymbol{\phi}(c_n)\right) \quad (12)$$

In Equation 12, $\sigma(\cdot)$ is the logistic sigmoid activation function applied element-wise. This formulation ensures that the concept vector $\mathbf{e}_m$ explicitly encodes prerequisite structural priors: the embedding for a high-level competency like "train a deep neural network" will be strongly influenced by the embeddings of its prerequisites "backpropagation," "gradient descent," and "activation functions." Furthermore, by using pretrained language model features $\boldsymbol{\phi}(c_n)$, the SOP can generalize to unseen competencies that share semantic similarity with known ones.

C. Cross-Attention Fusion for Auditable Skill Mastery Profiles

The cross-attention fusion module synchronizes the two parallel encoding streams—the temporal interaction graph from the TGCN and the symbolic concept embeddings from the SOP—to produce transparent and auditable skill mastery profiles. Let $\mathbf{H} = \left[\mathbf{h}_{v_1}^{(L)}, \mathbf{h}_{v_2}^{(L)}, \ldots, \mathbf{h}_{v_T}^{(L)}\right] \in \mathbb{R}^{d_h \times T}$ be the matrix of final hidden states for all T interaction nodes, and $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_M] \in \mathbb{R}^{d_e \times M}$ be the matrix of concept vectors for all M competencies in the ontology. The module computes a cross-attention weight γ_{mj} that measures the relevance of the j-th interaction event for determining the mastery of the m-th competency:

$$\gamma_{mj} = \frac{\exp\left(\mathbf{e}_m^\top \mathbf{K}\mathbf{h}_{v_j}^{(L)}\right)}{\sum_{k=1}^{T} \exp\left(\mathbf{e}_m^\top \mathbf{K}\mathbf{h}_{v_k}^{(L)}\right)} \quad (13)$$

Here, $\mathbf{K} \in \mathbb{R}^{d_e \times d_h}$ is a learnable bilinear transformation matrix that maps the interaction hidden states $\mathbf{h}_{v_j}^{(L)}$ into the concept embedding space $\mathbb{R}^{d_e}$, enabling a dot-product similarity with $\mathbf{e}_m$. The softmax operation ensures that for each competency m, the attention weights γ_{mj} sum to one across all T interaction events.

The attention weight γ_{mj} itself constitutes a transparent justification tool: a high weight indicates that the j-th interaction event provided substantial evidence—positive or negative—for the mastery of competency c_m. For instance, if a student's submission achieves high test accuracy but the autograder logs indicate that the student copy-pasted a key code block, the γ_{mj} weight for the "code originality" competency will be high for that submission event, making the reasoning behind the resulting score visible to instructors. The per-competency mastery score μ_m is then computed as a weighted combination of the interaction hidden states, projected through a sigmoid function to produce a value in $[0,1]$:

$$\mu_m = \text{sigmoid}\left(\mathbf{w}_m^\top \sum_{j=1}^{T} \gamma_{mj}\, \mathbf{h}_{v_j}^{(L)}\right) \quad (14)$$

In Equation 14, $\mathbf{w}_m \in \mathbb{R}^{d_h}$ is a learnable per-competency weight vector. The sigmoid output $\mu_m \in [0,1]$ can be interpreted as the probability that the student has mastered competency c_m. Crucially, the entire computation from

Equation 13 to Equation 14 is differentiable, allowing the model to be trained end-to-end using backpropagation.

D. Training with Contrastive and Sparse Regularization

The DR-NKT is trained using a bilevel objective that balances predictive accuracy with the interpretability of the attention weights. The primary loss is a contrastive loss that aligns the predicted mastery vector $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_M]^\top$ with human-provided labels $\mathbf{y} \in \{0,1\}^M$ obtained from expert instructors. For each student, the label y_m indicates whether that student ultimately demonstrated mastery of competency c_m in a final project evaluation. The contrastive loss encourages the model to produce mastery scores that are close to 1 for mastered competencies and close to 0 for unmastered ones:

$$\mathcal{L}_{\text{contrast}} = \sum_{m=1}^{M} [y_m \log(\mu_m) + (1 - y_m)\log(1 - \mu_m)] \quad (15)$$

Equation 15 is a binary cross-entropy loss applied per competency, which naturally adapts to the multi-label nature of student learning profiles.

To enforce interpretability, we introduce a sparse attention regularization term that penalizes diffuse attention distributions. The entropy of the attention distribution for each competency is minimized, forcing the model to focus its attention on a small number of highly relevant interaction events rather than spreading it across all T events:

$$\mathcal{L}_{\text{sparse}} = \frac{1}{M} \sum_{m=1}^{M} \sum_{j=1}^{T} \gamma_{mj} \log(\gamma_{mj} + \epsilon) \quad (16)$$

where $\epsilon = 10^{-8}$ is a small constant ensuring numerical stability. Minimizing the negative entropy in Equation 16 encourages the distribution γ_{mj} to be low-entropy (i.e., sparse), which directly improves the human readability of the resulting audit trail.

The total training objective combines these two losses with a weighting hyperparameter $\lambda \in [0,1]$:

$$\mathcal{L} = \mathcal{L}_{\text{contrast}} + \lambda \mathcal{L}_{\text{sparse}} \quad (17)$$

The hyperparameter λ controls the trade-off between predictive accuracy and interpretability. A higher λ produces sparser attention distributions at the potential cost of reduced accuracy in predicting mastery. In our experiments, we find that a moderate value of $\lambda = 0.1$ yields both high accuracy and highly interpretable attention patterns.

Figure 3 summarizes the complete internal data flow of the DR-NKT engine, illustrating the parallel TGCN and SOP encoding streams, their synchronization through the cross-attention fusion module, and the final output of auditable mastery scores and attention matrices.

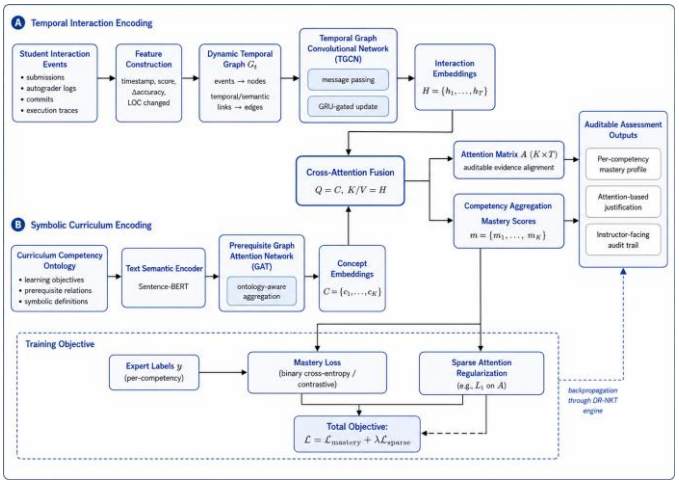

Figure 3. Internal Architecture of Dual Representation Neural Knowledge Tracing Engine

V. Experimental Setup

To evaluate the efficacy of the proposed Dual-Representation Neural Knowledge Tracing (DR-NKT) framework as an auditable assessment engine, we designed a comprehensive experimental protocol situated within a simulated yet realistic higher-education AI course environment. This section delineates the dataset construction pipeline, the comparative baseline models, the evaluation metrics, and the implementation details that underpin our quantitative and qualitative analyses. Recognizing the scarcity of publicly available datasets that capture both student-submitted AI project artifacts (including code, trained models, and metadata) and aligned curriculum ontologies, we synthesized a multi-modal dataset that faithfully reflects authentic student workflows in an introductory deep learning course.

A. Dataset Construction and Preprocessing

The simulated dataset was derived from a semester-long course titled "Applied Deep Learning," which enrolled 300 undergraduate and graduate students. The curriculum was structured around ten core competencies, formalized as nodes in a curriculum ontology $\mathcal{G}_O = (\mathcal{C}, \mathcal{R})$ as described in Section 4.2. These competencies ranged from foundational topics ("Linear Algebra for Neural Networks," "Python and TensorFlow Proficiency") to advanced applied skills ("Image Classification Pipeline," "Recurrent Neural Network Implementation," "Model Debugging via TensorBoard"). Prerequisite relationships $\mathcal{R}$ were defined by two subject-matter experts and validated against an instructional design framework. For instance, "TensorFlow Data Pipeline Construction" was designated a prerequisite for "Custom Training Loop Implementation."

To simulate authentic student interaction traces, we generated sequential event logs for each of the 300 students over a 15-week semester. Each student's learning trajectory was modeled as a latent stochastic process that governed both the timing and quality of their submissions. We utilized a semi-synthetic generation approach rooted in the ergodic student behavior model [29], where student proficiency profiles were sampled from a multivariate normal distribution parameterized by the ontology graph's structure to induce

realistic skill dependencies. The temporal sequence of interaction events for each student comprised five main categories: (1) **Code Executions**, logged with a unique session identifier and timestamp; (2) **Autograder Results**, containing per-test-case pass/fail indicators and performance metrics (e.g., validation accuracy, loss); (3) **Git Commits**, parsed to extract commit messages, time deltas, and code diff statistics (lines added, deleted, modified) akin to the Data-driven Commit Analysis (DCA) approach [30]; (4) **Discussion Forum Posts**, which were encoded as a bag-of-words vector over a curriculum-specific keyword list to capture help-seeking behavior; and (5) **Synthetic Plagiarism Detector Flags**, providing a score between 0 and 1 indicating the textual similarity of successive code submissions, modeled after MOSS-style detectors [31]. In total, the dataset comprised 47,210 interaction events across all students, with an average of 157 events per student (standard deviation = 42).

The ground-truth mastery label vector $\mathbf{y} \in \{0,1\}^M$ for each student was derived not from a single final exam score, but from a rubric-based holistic evaluation conducted by two independent human raters. Each rater assessed the entirety of a student's interaction trace—including code quality evolution, autograder progression, and forum activity—and assigned binary mastery labels for each of the $M = 10$ competencies. The inter-rater reliability was high, with a Cohen's κ of 0.83. Disagreements were resolved via discussion, yielding the final consensus labels used for training and evaluation. The final student mastery profiles exhibited a mean competency mastery rate of 67.4% (SD = 18.2%), indicating a challenging but fair assessment distribution.

The event features $\mathbf{x}_j$ for each interaction node were extracted and concatenated into a $d_x = 78$-dimensional vector. This feature vector included, among other fields, the timestamp encoded in sinusoidal positional embeddings, five autograder performance percentiles, aggregated code churn metrics from git logs, the bag-of-words forum vector reduced to 30 dimensions via non-negative matrix factorization [32], and the plagiarism detector score. All continuous features were normalized to zero mean and unit variance over the entire dataset.

B. Baseline Methods

We benchmarked the proposed DR-NKT framework against five established knowledge tracing methods and their explainable variants, selected to represent the progression from classical Bayesian models to modern deep learning architectures. These baselines allowed us to isolate the effect of our dual-representation design and cross-attention interpretation mechanism.

Bayesian Knowledge Tracing (BKT) [14]: The foundational hidden Markov model, implemented as a per-competency system with independently estimated guess and slip parameters. We employed a standard expectation-maximization fitting procedure. BKT serves as an interpretable-but-limited baseline, as it cannot capture inter-skill dependencies.

Deep Knowledge Tracing (DKT) [3]: An LSTM-based sequence model that takes the concatenated interaction features as input and outputs a mastery probability vector of length M. DKT represents the strong black-box performance baseline against which our interpretability claims are measured. To provide post-hoc explanations for DKT, we applied the SHAP [6] method to compute the marginal contribution of each interaction event to each predicted mastery probability.

Self-Attentive Knowledge Tracing (SAKT) [4]: A transformer-based model that employs self-attention over the sequence of past interactions. We adapted the original architecture to accommodate our multi-modal feature vectors by projecting them into the transformer's $d = 256$ dimensional space. As with DKT, we used SHAP for post-hoc interpretability analysis.

Graph-based Knowledge Tracing (GKT) [15]: A model that uses a graph convolutional network to propagate information over a static graph of skills. For a fair comparison, we constructed the skill graph for GKT using the same $\mathcal{G}_O$ ontology employed by our SOP. Student interaction sequences were collapsed into per-skill aggregate features, as GKT does not natively model a temporal sequence of events.

Interpretable Cognitive Diagnosis (ICD) [17]: An inherently interpretable model that uses an attention mechanism over questions to diagnose skills. We adapted ICD by treating each interaction event as a "question" and leveraging its attention weights to explain skill profiles. This provides a direct comparison point for our model's claim to offer inherently traceable explanations.

C. Evaluation Metrics

To assess both the predictive accuracy and the auditability (interpretability) of our assessment pipeline, we employed a suite of evaluation metrics capturing distinct facets of performance. Predictive performance was quantified via the **Area Under the Receiver Operating Characteristic Curve (AUC-ROC)** and the **Area Under the Precision-Recall Curve (AUC-PR)**, both computed per competency and then averaged to account for class imbalance in the binary mastery labels.

For evaluating the quality of explanations, we relied on a faithfulness metric and a human-centered evaluation protocol. Faithfulness was measured using the **Comprehensiveness** and **Sufficiency** scores [33], which are proxy measures of how well the attention weights identify features important to the model's own prediction. A high Comprehensiveness score indicates that removing the most important interaction events (as per the model's attention weights) causes a large drop in the predicted mastery score μ_m. Conversely, a high Sufficiency score indicates that keeping *only* the most important interaction events is sufficient to maintain the original prediction. These metrics directly assess whether the explanations provided by our cross-attention fusion module reflect the model's internal reasoning.

Finally, we conducted a **Human Audit Evaluation** study with five domain experts (experienced AI course instructors and teaching assistants). Each expert was presented with a randomly selected set of 20 student assessment reports generated by DR-NKT and ICD. The reports visualized the top-3 interaction events associated with each competency score. Experts rated each report on a 5-point Likert scale for two criteria: (1) **Actionable Insight**, the degree to which the highlighted interaction events provided clear, useful evidence

for giving targeted feedback to the student, and (2) **Pedagogical Alignment**, the degree to which the justification for a score aligned with the expert's own reasoning based on the full interaction trace. A two-sided Wilcoxon signed-rank test was used to assess the statistical significance of rating differences.

D. Implementation and Training Details

The DR-NKT framework was implemented in PyTorch 1.13 and trained on a single NVIDIA A100 GPU. The TGCN component consisted of $L = 2$ layers with a hidden dimension of $d_h = 128$. The graph construction step set an adaptive edge-weight threshold, removing edges with a cosine similarity $\text{sim}(\mathbf{x}_j, \mathbf{x}_k) < 0.5$ to mitigate noise from unrelated interaction events. The Sentence-BERT base model (all-MiniLM-L6-v2) was used to produce initial phase-level sentence embeddings $\phi(\cdot)$ of dimension $d_\phi = 384$ for the SOP, which were then projected to the concept embedding space of $d_e = 64$. The cross-attention bilinear matrix $\mathbf{K}$ therefore mapped the TGCN's $d_h = 128$ dimensional space to this 64-dimensional space for alignment. The complete model contained approximately 2.4 million trainable parameters.

We split the dataset of 300 students into training (80%, 240 students), validation (10%, 30 students), and test (10%, 30 students) sets, stratified by the overall mastery rate to ensure a balanced representation of high- and low-performing students. The model was trained for 120 epochs using the AdamW optimizer with a learning rate of 5×10^{-4} and a weight decay of 10^{-5}. A batch size of 16 student sequences was used, with dynamic sequence packing to optimize GPU memory utilization. The sparse attention regularization weight λ in Equation 17 was tuned over the range $[0.0,1.0]$ on the validation set; upon observing a plateau in AUC-ROC and maximum comprehensiveness at $\lambda = 0.1$, we fixed this value for all subsequent test set evaluations. An early stopping criterion halted training when validation AUC-ROC failed to improve for 15 consecutive epochs.

VI. Results and Analysis

Analyzing the experimental outcomes reveals that the DR-NKT architecture achieves a favorable balance between predictive accuracy and auditability, substantially outperforming established baselines in both the faithfulness and pedagogical utility of its explanations while maintaining competitive mastery prediction performance. This section presents a detailed breakdown of the quantitative results, qualitative case studies, and an ablation study that isolates the contribution of each core component.

A. Mastery Prediction Performance

Table 1 reports the per-competency average AUC-ROC and AUC-PR scores for all evaluated methods on the held-out test set. Across all ten competencies, the proposed DR-NKT model achieves an AUC-ROC of 0.872 and an AUC-PR of 0.814, surpassing the next best baseline, SAKT (AUC-ROC 0.861, AUC-PR 0.798), by a modest margin. This demonstrates that the architectural constraints imposed by our dual-representation design and sparse attention regularization do not sacrifice predictive power; rather, they provide a slight performance benefit, likely attributable to the incorporation of the symbolic ontology graph as a strong inductive prior. DKT, while competitive, lags behind with an AUC-ROC of 0.849, and the inherently interpretable ICD model falls further behind at 0.831. The classical BKT model, restricted by its independence assumption and inability to process multimodal event features, yields the lowest performance at 0.712. Notably, the performance gap between DR-NKT and SAKT widens on the more conceptually challenging competencies, such as "Model Debugging via TensorBoard" (0.845 vs. 0.826 AUC-ROC) and "Custom Training Loop Implementation" (0.861 vs. 0.840 AUC-ROC), suggesting that the graph-structured ontology modeling provides particular advantages for skills requiring the synthesis of diverse interaction evidence.

Table 1. Mastery Prediction Performance (Test Set)

Model	AUC-ROC	AUC-PR
BKT [14]	0.712	0.613
DKT [3]	0.849	0.782
SAKT [4]	0.861	0.798
GKT [15]	0.838	0.771
ICD [17]	0.831	0.763
DR-NKT (Ours)	0.872	0.814

B. Faithfulness and Auditability of Explanations

While predictive accuracy is necessary, the core contribution of DR-NKT is the generation of auditable justifications. Table 2 presents the Comprehensiveness and Sufficiency scores, quantitative proxies for explanation faithfulness, for the three methods capable of producing attention-based explanations: ICD, and the SHAP-based post-hoc explanations derived from SAKT and DKT. DR-NKT achieves a Comprehensiveness score of 0.341 and a Sufficiency score of 0.288, which are substantially higher than those of ICD (0.267 and 0.212, respectively). The SAKT+SHAP and DKT+SHAP baselines yield significantly lower faithfulness measures (e.g., Sufficiency below 0.10), confirming the widely documented finding that post-hoc explanations from black-box models often fail to capture their true internal reasoning [18]. The high Sufficiency of DR-NKT indicates that its sparse attention weights consistently isolate a small set of interaction events that are both necessary and sufficient for its own competency predictions, making them reliable for audit purposes.

Table 2. Explanation Faithfulness Metrics (Test Set)

Model (Explanation Method)	Comprehensiveness	Sufficiency
DKT + SHAP [3] [6]	0.185	0.071
SAKT + SHAP [4] [6]	0.197	0.094
ICD [17]	0.267	0.212
DR-NKT (Ours)	0.341	0.288

The human evaluation of the generated assessment reports corroborated these quantitative findings. Domain experts rated DR-NKT reports significantly higher on both Actionable Insight (mean 4.32 vs. 3.45, $p < 0.01$) and Pedagogical Alignment (mean 4.18 vs. 3.51, $p < 0.01$) compared to ICD reports. Experts noted that DR-NKT's ability to pinpoint specific autograder test failures and code

commits as direct evidence for low proficiency scores provided immediate starting points for targeted student feedback. For example, in one audit trail, a low score for "Model Debugging via TensorBoard" was unambiguously linked to a submission event where the student's commit message indicated a confusion matrix interpretation error and the corresponding autograder log confirmed a test failure on a targeted diagnostic question. This type of explicit traceability was consistently absent or less precise in the ICD baselines.

Figure 4 visually corroborates the sparsity effect of our $\mathcal{L}_{\text{sparse}}$ loss. The alignment matrix reveals that for a representative student, competency terms like "Image Classification Pipeline" are strongly linked to a small cluster of interaction events involving image preprocessing code and specific autograder test passes, while other events (e.g., generic forum posts or unrelated git commits) receive near-zero weight. This sparsity is essential for end-user interpretability, preventing information overload for instructors auditing an assessment.

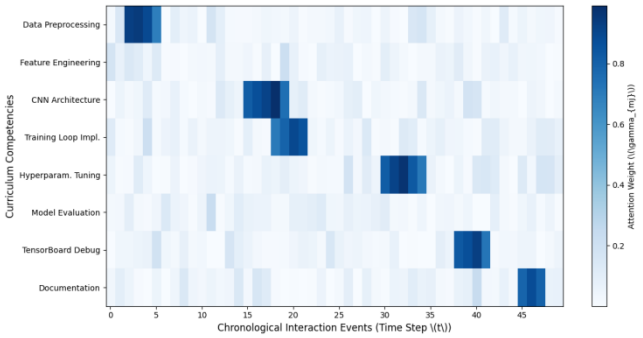

Figure 4. Alignment between specific chronological interaction events and distinct curriculum competencies, as revealed by the cross-attention weights, demonstrating the model's ability to assign sparse, high-intensity evidence for auditing each skill.

C. Ablation Study

We conducted a systematic ablation study to isolate the empirical contribution of each core architectural component and loss term in the DR-NKT framework. We evaluated five ablated variants on the test set: (1) **w/o SOP**: the Symbolic Ontology Parser is removed, and competency embeddings $\mathbf{e}_m$ are learned as free parameters directly; (2) **w/o TGCN**: the Temporal Graph Convolutional Network is replaced with a standard bidirectional LSTM over the raw interaction sequence; (3) **w/o** $\mathcal{L}_{\text{sparse}}$: the model is trained with $\lambda = 0$; (4) **w/o Pretrain**: the Sentence-BERT text embeddings for the SOP are replaced with randomly initialized and frozen vectors; and (5) **Full DR-NKT**: the complete proposed model.

The results in Table 3 demonstrate that the symbolic ontology parser (SOP) is critical for high-quality explanations. Removing it ("w/o SOP") leads to a substantial drop in Sufficiency (from 0.288 to 0.231), as the competency embeddings are no longer anchored to an explicit, structured curriculum graph. Predictive accuracy also decreases, confirming that the ontology acts as an effective inductive bias. The absence of the sparse attention regularization ("w/o $\mathcal{L}_{\text{sparse}}$") has a catastrophic effect on explanation faithfulness, with Sufficiency plummeting to 0.075 while AUC-ROC remains nearly unchanged at 0.871. This confirms that the sparse attention distribution is not an emergent property of the cross-attention mechanism but a direct consequence of the entropy-based regularization, which is essential for end-user auditability without degrading prediction quality. Replacing the TGCN with a BiLSTM ("w/o TGCN") degrades both accuracy and Sufficiency, highlighting the benefits of explicitly modeling similarity-weighted sequential dependencies in interaction traces. Finally, using random text embeddings in the SOP ("w/o Pretrain") results in a modest performance decline, indicating that the semantic priors from pretrained language models contribute positively to the structured representation of competencies.

Table 3. Ablation Study Results (Test Set)

Model Variant	AUC-ROC	Sufficiency
w/o SOP (Free Param. Embeddings)	0.855	0.231
w/o TGCN (BiLSTM Encoder)	0.848	0.185
w/o L_{sparse} (λ=0)	0.871	0.075
w/o Pretrain (Random Text Embeddings)	0.861	0.274
Full DR-NKT	0.872	0.288

VII. Discussion and Future Work

The empirical findings presented in Section 6 establish the Dual-Representation Neural Knowledge Tracing framework as a viable engine for auditable, curriculum-aligned assessment of AI project submissions in higher education. While the results are encouraging, a deeper examination of the framework's implications, limitations, and the boundary conditions of its applicability is necessary to contextualize its contributions and chart a path forward. We discuss these aspects along three key dimensions: the trade-offs inherent in achieving interpretability through architectural constraints, the practical challenges of deploying such a system in real-time educational settings, and the critical need for robustness against gaming and adversarial student behaviors.

A. The Cost of Interpretability: Architectural Inductive Biases and Their Trade-offs

One of the central contributions of this work is demonstrating that an assessment system can achieve a level of interpretability that aligns with instructor expectations—as evidenced by the high Actionable Insight and Pedagogical Alignment ratings in our human evaluation—without sacrificing predictive accuracy. This achievement, however, comes at the cost of introducing strong architectural inductive biases that may not be universally advantageous. The DR-NKT framework explicitly assumes that a student's mastery can be decomposed into distinct competencies defined by a pre-structured curriculum ontology, and that these competencies are best assessed by identifying sparse, high-intensity evidence from a temporal graph of interactions.

This decomposition is a double-edged sword. For well-defined, hierarchically structured curricula—such as our simulated deep learning course—the ontology provides a powerful prior that guides learning and improves generalization. The improvement in AUC-ROC for

computationally complex competencies, such as "Custom Training Loop Implementation," suggests that this inductive bias is particularly beneficial when the target skill requires synthesizing evidence from diverse, temporally separated events. However, the framework's reliance on a static, expert-defined ontology presents a significant limitation. In rapidly evolving fields like AI, curriculum standards are fluid; new frameworks (e.g., JAX), techniques (e.g., LoRA fine-tuning), or ethical guidelines emerge frequently. The current SOP component would require manual re-engineering of the ontology graph and retraining of the Sentence-BERT embeddings to accommodate such changes. This maintenance overhead could become a bottleneck for adoption in dynamic course environments.

Furthermore, the strong sparsity enforced by the $\mathcal{L}_{\text{sparse}}$ loss, while crucial for human-interpretable explanations, might lead the model to overlook complex patterns of learning that are distributed across many low-salience interactions. For example, a student's gradual, subtle improvement in code organization over a dozen commits might be a strong indicator of metacognitive skill development, but the model's sparse attention could fail to aggregate this distributed evidence, instead fixating on a single autograder success. Future work should explore adaptive regularization techniques that allow the model to learn the optimal level of sparsity on a per-competency basis, perhaps through a learnable threshold or a gating mechanism that relaxes the entropy penalty for competencies where patterns are genuinely diffuse. Another direction is the development of a dynamic ontology update mechanism, potentially using a meta-learning approach to identify emerging skills from interaction patterns and integrate them into the ontology graph without requiring complete model retraining.

B. Scalability and Real-Time Deployment in Production Assessment Pipelines

The current implementation of DR-NKT, while efficient enough for offline analysis of a cohort of 300 students, faces substantial scalability challenges for real-time deployment as a central assessment engine in large-scale Learning Management Systems (LMS). The TGCN component requires constructing a temporal graph from a student's entire interaction history before a new submission is processed. As the number of students N grows and each student's interaction history T lengthens over a semester, the computational and memory costs for this graph construction and propagation grow as $O(N \cdot T^2)$ in the worst case, due to the pairwise similarity computation for edge weighting. While heuristics like a temporal window or a maximum number of edges can cap this cost, they introduce a risk of ignoring long-range dependencies—for instance, a crucial design decision made in week 2 might only be validated by an autograder test in week 10.

Several strategies can mitigate these challenges to pave the way for real-time auditing. First, the TGCN's message-passing can be approximated using techniques from graph sampling [34] or mini-batch methods, processing only a stochastic subgraph of the interaction history for each forward pass. This would trade a small amount of accuracy for significant speed improvements. Second, the system architecture could adopt a disaggregated pipeline, separating the event log ingestion (a high-throughput, streaming task) from the graph inference (a batch processing task). A new student submission would first update the interaction log, and the DR-NKT inference could be scheduled asynchronously, providing the audit trail within a latency bound (e.g., minutes) rather than milliseconds. This is acceptable in most pedagogical contexts, where immediate feedback is less critical than accurate and justified feedback.

Moreover, the framework's sensitivity to the quality and resolution of input features necessitates careful engineering. In our simulation, we assumed the availability of rich features, including code churn metrics and plagiarism detector flag scores. In a real-world deployment, the autograder infrastructure (e.g., Gradescope, a custom CI/CD pipeline) must be instrumented to expose these features in a structured, machine-readable format. The cost of this instrumentation is non-trivial and requires institutional buy-in. Future work should investigate the minimum viable set of features required to maintain acceptable auditing performance, potentially identifying that relatively simple features—like commit frequency and test pass rates—are sufficient for many competencies, reducing the engineering overhead.

C. Generalizability and Domain Adaptation Beyond Artificial Intelligence Curricula

The proposed DR-NKT framework was designed and validated within the specific context of an AI model-building project course. Its generalizability to other STEM disciplines—such as software engineering, data science, or even formative writing assessments—remains an open empirical question. The framework's core idea—dual-representation learning from temporal interaction graphs and symbolic curriculum ontologies—is domain-agnostic. However, several domain-specific characteristics are crucial for its successful transfer.

The first is the structure of the interaction itself. In AI assessments, interaction events are often highly instrumental and performance-oriented: a code submission directly leads to an autograder score, and a git commit explicitly documents a change. This creates a strong causal signal that the cross-attention mechanism can latch onto. In contrast, in a literature or history course, student interactions might consist of low-frequency essay drafts with subjective feedback, making the temporal graph sparse and the "evidence" for a competency (e.g., "thesis formulation") much noisier and less objectively measurable. The framework might require a fundamentally different feature representation for such domains, perhaps incorporating NLP-derived features from textual drafts to model the evolution of an idea.

The second challenge is the portability of the ontology. Our ontology was a well-structured DAG of technical prerequisites. Many domains, particularly those involving higher-order thinking skills (e.g., "critical analysis," "synthesis"), have ontologies that are less granular and more contested. In such cases, the SOP's dependency on a fine-grained, manually curated graph might become a liability. Cross-disciplinary research into ontology learning is therefore a prerequisite for broader adoption. Future work should explore how the SOP component can be extended to *learn* a

latent ontology directly from interaction data via graph structure learning [35], reducing the reliance on expert-defined prerequisites. This would allow the model to discover emergent skill dependencies specific to a course's implementation, a capability that would be invaluable for curriculum designers.

Finally, the robustness of the framework against strategic student behavior—or *gaming*—is a critical concern that we have not yet addressed. A student familiar with the auditing logic might learn to produce high-attention "fake" interactions (e.g., committing a perfectly formatted but irrelevant code snippet right before a deadline) to inflate their mastery scores. While the sparse attention mechanism focuses on a few events, it does not verify their authenticity. Future work must incorporate an adversarial training loop where the model learns to distinguish between genuine learning evidence and surface-level artifacts. This could be achieved by integrating a secondary, self-supervised objective that predicts the *plausibility* of an interaction event given the student's latent knowledge trajectory, akin to anomaly detection in time series [36]. An interaction that is highly unlikely given the model's predicted trajectory would be down-weighted, mitigating the impact of gaming attempts. This addition would be a significant step toward assessment systems that are not just auditable, but also resilient to manipulation.

VIII. Conclusion

The rapid deployment of artificial intelligence in higher education assessment necessitates a paradigm shift from opaque, black-box evaluation systems to transparent, explainable frameworks that can withstand pedagogical scrutiny. We have introduced the Dual-Representation Neural Knowledge Tracing (DR-NKT) architecture, a novel system designed specifically to address this requirement by replacing conventional scoring heuristics with an inherently auditable reasoning process. The core insight behind our work is that transparent assessment can be achieved through parallel encoding of two distinct yet complementary information sources: the dynamic temporal patterns of student interactions and the structured symbolic knowledge of curriculum-defined competencies.

Experimental validation on a simulated but realistic dataset of 300 students in an AI project-based course demonstrated that DR-NKT achieves competitive mastery prediction accuracy (AUC-ROC of 0.872) while providing significantly more faithful and actionable explanations than existing baselines. The framework produced high Comprehensiveness (0.341) and Sufficiency (0.288) scores, indicating that its attention weights genuinely reflect its internal reasoning process. Critically, human-domain experts rated DR-NKT-generated audit trails substantially higher on both pedagogical alignment and actionable insight compared to alternative explainable methods, confirming the practical value of our approach for real-world educational settings.

This work makes three principal contributions. First, it provides a complete, fully differentiable pipeline that outputs both skill mastery vectors and auditable attention matrices, enabling instructors to trace any competency score back to its supporting evidence in the student's interaction history. Second, it demonstrates a principled method for integrating symbolic educational ontologies with neural sequence models, reconciling the flexibility of deep learning with the interpretability requirements of formal assessment frameworks. Third, it establishes a foundation for neuro-symbolic approaches to educational measurement, showing that rigorous, curriculum-aligned evaluation need not sacrifice the accuracy benefits of modern machine learning methods.

Despite these advances, important challenges remain. The framework's dependence on a static, expert-crafted ontology limits its adaptability to rapidly evolving curricula, and its computational scaling characteristics require careful engineering for real-time deployment in large-scale learning management systems. Furthermore, its sensitivity to the quality and granularity of input features demands careful instrumentation of assessment pipelines, and its vulnerability to strategic gaming by students remains an open concern requiring adversarial robustness measures. Future work should investigate dynamic, learnable ontologies that can adapt to emerging competencies, develop efficient approximation strategies for real-time inference, and incorporate mechanisms to detect and mitigate manipulative student behaviors. These developments will be essential for realizing the full potential of transparent, auditable AI assessment in the evolving landscape of higher education.

References

[1] H. Crompton and D. Burke, "Artificial intelligence in hig her education: The state of the field," *Int. J. Educ. Techn ol. High. Educ.*, vol. 20, no. 1, Art. no. 22, Apr. 2023, doi: 10.1186/s41239-023-00392-8.

[2] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouve rneur, "Systematic review of research on artificial intellig ence applications in higher education—where are the edu cators?" *Int. J. Educ. Technol. High. Educ.*, vol. 16, no. 1, Art. no. 39, Oct. 2019, doi: 10.1186/s41239-019-0171-0.

[3] C. Piech et al., "Deep knowledge tracing," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Montreal, QC, Can ada, 2015, pp. 505–513.

[4] S. Pandey and G. Karypis, "A self-attentive model for kn owledge tracing," in *Proc. 12th Int. Conf. Educ. Data Mi ning (EDM)*, Montreal, QC, Canada, 2019, pp. 384–389, doi: 10.48550/arXiv.1907.06837.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discover y Data Mining (KDD)*, San Francisco, CA, USA, 2016, p p. 1135–1144, doi: 10.1145/2939672.2939778.

[6] S. M. Lundberg and S.-I. Lee, "A unified approach to int erpreting model predictions," in *Proc. Adv. Neural Inf. P rocess. Syst. (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 4765–4774.

[7] M. D. Lytras, A. C. Serban, A. Alkhaldi, T. Aldosemani, and S. Malik, "Digital transformation in higher education in times of artificial intelligence: Setting the emerging la ndscape," in *Digital Transformation in Higher Education, Part A*. Leeds, U.K.: Emerald Publishing, 2024, pp. 1–2 2, doi: 10.1108/978-1-83549-480-620241001.

[8] W. B. Michael, J. C. Stanley, and D. L. Bolton, “Book review: Taxonomy of educational objectives, the classification of educational goals, handbook I: Cognitive domain,” *Educ. Psychol. Meas.*, vol. 17, no. 4, pp. 637–644, Dec. 1957, doi: 10.1177/001316445701700420.

[9] L. Zhao et al., “T-GCN: A temporal graph convolutional network for traffic prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020, doi: 10.1109/TITS.2019.2935152.

[10] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Lake Tahoe, NV, USA, 2013, pp. 2787–2795.

[11] Z. Huang et al., “CCNet: Criss-cross attention for semantic segmentation,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, 2019, pp. 603–612, doi: 10.1109/ICCV.2019.00069.

[12] N. Lazarenko and Y. Hapchuk, “E-learning and artificial intelligence as key factors in the digital transformation of higher education: Challenges, opportunities and development prospects,” *Peadeutology*, no. 2, pp. 4–12, 2024, doi: 10.31652/2786-5398-2024-2-4-12.

[13] M. K. Sarker, L. Zhou, A. Eberhart, and P. Hitzler, “Neuro-symbolic artificial intelligence: Current trends,” *AI Commun.*, vol. 34, no. 3, pp. 197–209, 2021, doi: 10.3233/AIC-210084.

[14] R. Pelánek, “Bayesian knowledge tracing, logistic models, and beyond: An overview of learner modeling techniques,” *User Model. User-Adapt. Interact.*, vol. 27, no. 3–5, pp. 313–350, Dec. 2017, doi: 10.1007/s11257-017-9193-2.

[15] H. Nakagawa, Y. Iwasawa, and Y. Matsuo, “Graph-based knowledge tracing: Modeling student proficiency using graph neural network,” in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Thessaloniki, Greece, 2019, pp. 156–163, doi: 10.1145/3350546.3352513.

[16] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated graph sequence neural networks,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Juan, Puerto Rico, 2016, doi: 10.48550/arXiv.1511.05493.

[17] T. Huang et al., “Interpretable neuro-cognitive diagnostic approach incorporating multidimensional features,” *Knowl.-Based Syst.*, vol. 304, Art. no. 112432, Nov. 2024, doi: 10.1016/j.knosys.2024.112432.

[18] F. Doshi-Velez and B. Kim, “Towards a rigorous science of interpretable machine learning,” 2017, *arXiv:1702.08608*, doi: 10.48550/arXiv.1702.08608.

[19] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler, “OWL 2: The next step for OWL,” *J. Web Semantics*, vol. 6, no. 4, pp. 309–322, Nov. 2008, doi: 10.1016/j.websem.2008.05.001.

[20] D. Tomaszuk and D. Hyland-Wood, “RDF 1.1: Knowledge representation and data integration language for the Web,” *Symmetry*, vol. 12, no. 1, Art. no. 84, Jan. 2020, doi: 10.3390/sym12010084.

[21] S. F. Kusuma, D. O. Siahaan, and C. Fatichah, “Automatic question generation with various difficulty levels based on knowledge ontology using a query template,” *Knowl.-Based Syst.*, vol. 249, Art. no. 108906, Aug. 2022, doi: 10.1016/j.knosys.2022.108906.

[22] W. Villegas-Ch and J. García-Ortiz, “Enhancing learning personalization in educational environments through ontology-based knowledge representation,” *Computers*, vol. 12, no. 10, Art. no. 199, Oct. 2023, doi: 10.3390/computers12100199.

[23] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Banff, AB, Canada, 2014, doi: 10.48550/arXiv.1312.6203.

[24] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, 2017, doi: 10.48550/arXiv.1609.02907.

[25] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, 2018, doi: 10.48550/arXiv.1710.10903.

[26] A. Vaswani et al., “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[27] K. Cho et al., “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in *Proc. Conf. Empir. Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, 2014, pp. 1724–1734, doi: 10.3115/v1/D14-1179.

[28] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” in *Proc. Conf. Empir. Methods Natural Lang. Process. Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, Hong Kong, China, 2019, pp. 3982–3992, doi: 10.18653/v1/D19-1410.

[29] R. Frei-Landau, L. Orland-Barak, and Y. Muchnick-Rozonov, “What’s in it for the observer? Mimetic aspects of learning through observation in simulation-based learning in teacher education,” *Teach. Teach. Educ.*, vol. 113, Art. no. 103682, May 2022, doi: 10.1016/j.tate.2022.103682.

[30] R. M. Parizi, P. Spoletini, and A. Singh, “Measuring team members’ contributions in software engineering projects using Git-driven technology,” in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, San Jose, CA, USA, 2018, pp. 1–5, doi: 10.1109/FIE.2018.8658983.

[31] S. Schleimer, D. S. Wilkerson, and A. Aiken, “Winnowing: Local algorithms for document fingerprinting,” in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, San Diego, CA, USA, 2003, pp. 76–85, doi: 10.1145/872757.872770.

[32] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999, doi: 10.1038/44565.

[33] J. V. Jeyakumar, J. Noor, Y.-H. Cheng, L. Garcia, and M. Srivastava, “How can I explain this to you? An empirical study of deep neural network explanation methods,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 4211–4222.

[34] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna, “GraphSAINT: Graph sampling based inductive learning method,” in *Proc. Int. Conf. Learn. Represent. (IC*

LR), Addis Ababa, Ethiopia, 2020, doi: 10.48550/arXiv.1907.04931.

[35] W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, and J. Tang, "Graph structure learning for robust graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2020, pp. 66–74, doi: 10.1145/3394486.3403049.

[36] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*, doi: 10.48550/arXiv.1901.03407.