Personalized Knowledge Tracing Based on Generative Models: Cognitive Exploration of Learning Preferences

Shiyu Zhu (herb711@email.swu.edu.cn)
Southwest University, Chongqing, China
Chongqing Institute of Engineering, Chongqing, China

Li Li(lily@swu.edu.cn)
Southwest University, Chongqing, China

Yichen Liu (04870@cqie.edu.cn)
Chongqing Institute of Engineering, Chongqing, China

Abstract

PLGAN is a generative model-based framework for personalized knowledge tracing, designed to explore and model individual learning preferences. By integrating a Personalized Attention Mechanism (P-Attn), PLGAN effectively extracts learners' distinct learning patterns and behavioral tendencies, addressing the limitations of traditional knowledge tracing models that assume homogeneous learner groups. Unlike conventional approaches, PLGAN dynamically adjusts the weighting of behavioral features, enabling a more nuanced representation of learning preferences and improving the accuracy of knowledge state predictions across diverse learning environments. Experimental results on multiple public datasets demonstrate that PLGAN achieves an average performance improvement of 3.5% in knowledge tracing tasks. Furthermore, the generative nature of PLGAN enhances its generalization and robustness, effectively capturing individualized learning dynamics. This work advances the study of personalized learning by leveraging generative models to model and analyze learner behavior, providing a cognitively informed approach to knowledge tracing.

Keywords: Knowledge Tracing; Personalized Learning; Generative Adversarial Networks; Attention Mechanism

Introduction

Knowledge Tracing (KT) is a key area in educational research, aiming to track learners' knowledge states by analyzing interaction data to optimize learning outcomes Abdelrahman et al. (2023). With the rise of deep learning techniques, models like SAKT Pandey & Karypis (2019), AKT Ghosh et al. (2020), and DTransformer Yin et al. (2023) have significantly advanced KT, achieving improved prediction performance. However, these models often treat learners as a homogeneous group, overlooking individual differences that limit personalized learning.

Learner engagement and cognitive processing often exhibit complex distribution patterns, such as bimodal or skewed distributions, which current KT models fail to address effectively. For example, mathematics scores may be bimodal, reflecting distinct cognitive strategies adopted by high- and low-performing learners Dweck (1986). Most KT studies overlook personalized learning preferences, reducing learner identity to student IDs, which results in insufficient characterization of individual cognitive tendencies and suboptimal knowledge state predictions, particularly in scenarios with short or fragmented interaction sequences.

To improve KT performance, it is essential to focus on extracting personalized features and modeling performance consistency, particularly in data-sparse scenarios with significant fluctuations F. Wang et al. (2023); Mao et al. (2023); Tomić et al. (2022); Shen et al. (2022). Although there is growing recognition of the need for personalization, effectively capturing and modeling such information remains a challenge. To address these challenges, we propose PLGAN, a generative adversarial network (GAN) framework that improves knowledge tracing models by generating features that capture learners' personalized characteristics. PLGAN integrates generative adversarial mechanisms with personalization modeling, focusing on extracting learners' preference features and demonstrating strong robustness and generalization. The main contributions are:

- **PLGAN for Personalized Feature Extraction**: PLGAN extracts personalized preference features from learners' interaction sequences using a cross-sequence attention mechanism, offering richer learner representations than traditional methods based on learner IDs, thus enhancing knowledge tracing performance.
- **Dynamic Personalized Associative Attention (P-Attn)**: PLGAN incorporates P-Attn, which dynamically adjusts learner behavior weights, emphasizing personalized traits in multi-stage interaction data, improving the model's ability to capture dynamic behaviors, especially in imbalanced data scenarios.
- **Comprehensive Evaluation**: PLGAN's performance is evaluated across multiple datasets, focusing on model stability, feature evolution, sequence length impact, and adaptability to different KT models. Results show that PLGAN significantly improves baseline models, outperforming existing methods in various scenarios.

The paper is structured as follows: Section II reviews related work, including advances in knowledge tracing, GANs, attention mechanisms, and personalized learning. Section III

In D. Barner, N.R. Bramley, A. Ruggeri and C.M. Walker (Eds.), *Proceedings of the 47th Annual Conference of the Cognitive Science Society*

defines the knowledge tracing problem, details the PLGAN architecture, cross-sequence attention mechanism, and multi-objective optimization. Section IV discusses experimental setup, datasets, baseline methods, and evaluation. Finally, Section V summarizes contributions and suggests future directions.

Related Work

Knowledge Tracing (KT) began with the Deep Knowledge Tracing (DKT) model based on Recurrent Neural Networks (RNNs) Piech et al. (2015), which captures learner behavior trajectories. However, DKT faces two key limitations: the long-term dependency problem and constrained feature modeling capacity. While Long Short-Term Memory (LSTM) Henaff et al. (2016) and Gated Recurrent Units (GRU) Chung et al. (2014) improve memory, they still struggle with long sequences and complex knowledge relationships. Additionally, DKT's reliance on one-hot encoded interaction data fails to model personalized learner characteristics Khajah et al. (2016); Ding & Larson (2019).

The advent of Transformer models marked a new phase in KT. The SAKT model Pandey & Karypis (2019) was the first to apply Transformers to KT, addressing long-term dependency issues. The SAINT model Choi et al. (2020) and enhancements by Pu et al. Pu et al. (2020) improved feature modeling and prediction accuracy. However, despite their ability to model complex knowledge relationships, Transformer models still struggle to capture personalized learner preferences, limiting their ability to model diverse learning trajectories.

To improve personalized modeling, Shen et al. Shen et al. (2020) proposed Convolutional Knowledge Tracing (CKT), which learns personalized prior knowledge and learning rates. Long et al. Long et al. (2021) introduced Individual Estimated Knowledge Tracing (IEKT), enhancing personalized predictions by modeling learners' knowledge sensitivity. Zhang et al. Zhang et al. (2022) and Wang et al. C. Wang & Sahebi (2023) addressed the dynamic nature of KT by introducing online updates and long-term learning adaptations. Despite these advancements, challenges such as sparse data and limited samples hinder model generalization.

Existing KT research faces challenges such as poor personalized feature modeling, data sparsity, and difficulty in adapting to dynamic learning processes. The PLGAN model addresses these issues by integrating Generative Adversarial Networks (GANs) with personalized feature extraction.

Methodology

Problem Definition

The task of Knowledge Tracing (KT) is to assess a learner's knowledge state dynamically by modeling their historical interactions and predicting future responses. A learner's learning process is represented as a sequence of interactive events $X = (x_1, x_2, \ldots, x_n)$, where each event $x = (e, r)$ consists of the following elements:

- e: the exercise completed by the learner,
- $r \in \{1, 0\}$: the learner's response, where 1 indicates a correct answer and 0 indicates an incorrect answer.

In this work, we extend the definition of interaction events by introducing two additional components, u_i and t, as model inputs:

- u_i: the learner identifier, which differentiates individual learners, enabling personalized modeling of their knowledge states and learning behaviors,
- t: the timestamp of the interaction event, capturing the temporal dependencies and dynamic changes in the learner's knowledge state over time.

With the inclusion of u_i and t, the interaction event (e, r) is redefined as (u_i, e, t, r), and the learner's learning process is represented as a series of independent sequential events. This extended representation facilitates personalized modeling and accounts for the temporal dynamics of knowledge states, which are essential for capturing the evolution of a learner's understanding over time.

Personalized Learning GAN Framework

This study extracts learners' personalized learning preference features based on the following hypothesis: **There is a direct correlation between the distribution of learners' interaction behavior data and their learning preferences and knowledge states**. Learning preferences reflect learners' long-term tendencies and strategic choices in completing tasks, including preferences for specific knowledge points, preferred learning methods, and response patterns to practice problems. In contrast, knowledge states represent learners' current mastery of knowledge points.

Traditional Generative Adversarial Networks (GANs) can effectively model global data distributions, but they struggle to capture individual differences, which limits their application in personalized learning, as shown in the experiments in Section of this study.

To address these issues, we propose the Personalized Learning GAN (PLGAN). As shown in Figure 1, the PLGAN framework consists of multiple modules, with a specially designed encoder-decoder architecture embedded within the generator. This architecture seamlessly combines interaction sequence generation with personalized learning feature extraction. The encoder plays a critical role in extracting meaningful personalized features from global learning representations and effectively captures learners' interaction behavior patterns, thus reflecting their personalized learning characteristics.

In the encoder, we construct a memory vector matrix U_{total}, which is used to store the learner's global learning style features and is updated during the generator's forward propagation.

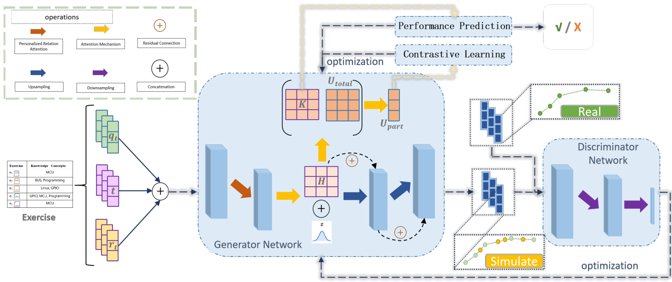

Figure 1: Structure of the PLGAN framework, showing the data flow and interactions between components. "Exercise" represents the learner's historical exercise data, including questions, timestamps, and results. "Simulate" represents the data generated by the generator network, while "Real" is a part of "Exercise," representing the actual learner interaction data.

In this process, U_{total} is first transformed through the P-Attn mechanism (see Section Personalized Relation Attention Mechanism), and combined with the input interaction sequence X to generate latent features H. The P-Attn mechanism adjusts the query and key vectors by adding the learner embedding u_{vec}, enabling the model to extract the personalized learning style features U_{part} from U_{total} that are relevant to X. This process is as follows: $Q_i = W_Q \cdot U_{\text{total}}, \quad K_j = W_K \cdot U_{\text{total}}$. Next, the learner embedding u_{vec} is added to adjust the query and key vectors further: $Q'_i = Q_i + W_u \cdot u_{\text{vec}}, \quad K'_j = K_j + W_u \cdot u_{\text{vec}}$.

As shown in Equation 1, the attention weights are calculated, and the context representation H is generated:

$$\alpha_{ij} = \frac{\exp(Q'_i \cdot (K'_j)^T)}{\sum_k \exp(Q'_i \cdot (K'_k)^T)}, \quad H_i = \sum_j \alpha_{ij} V_j. \tag{1}$$

Subsequently, H is used as a latent feature and is decomposed through the attention mechanism into learning style features U_{total} and knowledge features K: $H = [U_{\text{total}}, K]$. Here, U_{total} represents the global features of the learner's overall learning style, and K reflects their mastery of specific knowledge points. Based on this decomposition, we again use the P-Attn mechanism to extract the learner-specific preferences U_{part} from U_{total}. To ensure that the extracted personalized learning style features U_{part} accurately reflect the learner's characteristics, a contrastive loss is introduced (see Section Loss Function Design).

The generator outputs H (learner features) combined with random noise z, which are passed to the decoder. The decoder is a multilayer convolutional network that processes the concatenated features F, generating simulated interaction sequences that closely resemble real interaction data:

$$F = \text{Concat}(H, U_{\text{part}}, K, z). \tag{2}$$

The overall architecture of PLGAN addresses the limitations of traditional GANs in capturing individual differences while ensuring the integrity of the extracted learner-specific preferences U_{part}.

Personalized Relation Attention Mechanism

The P-Attn mechanism plays a crucial role in extracting latent features H and personalized learning style features U_{part} by incorporating learner-specific information. This section provides a detailed description of the design and operations of the P-Attn mechanism.

As shown in Figure 2, the P-Attn mechanism consists of several key components and operations that work together to process the input sequence, adjust the attention mechanism, and extract personalized features.

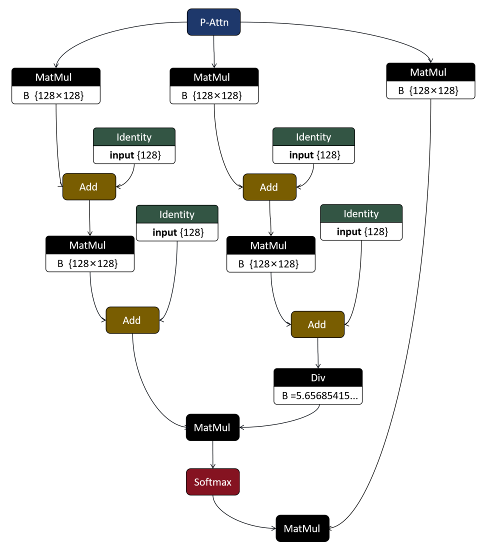

Figure 2: Flow of the Personalized Relation Attention (P-Attn) mechanism. **Identity** represents direct tensor transfer, **MatMul** performs linear transformations, **Add** implements residual connections, **Transpose** adjusts tensor dimensions, **Reshape** modifies tensor shapes, **Div** normalizes, **Softmax** generates attention probabilities, and the final **Transpose** prepares the output.

In the P-Attn mechanism, the input data $X \in \mathbb{R}^{B \times 128 \times 128}$ (where B represents the batch size and 128×128 represents the feature dimension) is first mapped through multiple linear transformations to obtain the query Q, key K, and value V. The specific transformations are performed via matrix multiplication with the weight matrices W_Q, W_K, and W_V: $Q = XW_Q, \quad K = XW_K, \quad V = XW_V$ Here, W_Q, W_K, and W_V are weight matrices for the query, key, and value, each of size 128×128. These transformations enable the model to learn the query, key, and value representations of the input data, which are crucial for the subsequent attention calculation.

To implement a personalized attention mechanism, P-Attn introduces user identity information (uID). By incorporating the user identity embedding $uID \in \mathbb{R}^{1 \times 128}$ into the queries Q and keys K, their representations are adjusted to ensure that the attention mechanism is tailored to each user's specific characteristics. This adjustment is achieved as follows:

$$Q' = Q + uIDW_u, \quad K' = K + uIDW_u \tag{3}$$

Where $W_u \in \mathbb{R}^{128\times128}$ is the weight matrix associated with the user identity uID. This adjustment ensures that the query and key representations for each user are personalized based on their unique identity, allowing the attention mechanism to be optimized for each user's needs.

Next, the queries Q' and keys K' are combined with the input data X through an addition operation, generating new queries Q'', keys K'', and values V'': $Q'' = Q' + X, \quad K'' = K' + X, \quad V'' = V' + X$ This addition operation implements residual connections, allowing the original information to be preserved in deeper layers of the network. This not only prevents information loss but also helps stabilize the model during training.

To further ensure the stability of the computation, the P-Attn mechanism normalizes the results after the addition operation. Specifically, the inner product of the queries and keys is scaled to obtain the normalized attention scores:

$$\text{Scaled Attention} = \frac{Q''K''^T}{\sqrt{d_k}}$$

Where d_k is the dimensionality of the keys. The scaled attention scores are then passed through a Softmax function to compute the importance of each input feature, and these weights are combined with the values V'' to generate the final personalized output:

$$Y = \text{Softmax}\left(\frac{Q''K''^T}{\sqrt{d_k}}\right)V'' \tag{4}$$

In this way, the P-Attn mechanism dynamically adjusts the attention weights based on each user's identity information, generating personalized weighted outputs to meet the individual needs of each user.

Loss Function Design

The optimization objective of PLGAN consists of three components: the GAN optimization, learning style features U_{part} optimization using contrastive learning, and knowledge features K optimization via interaction prediction. These objectives enable PLGAN to effectively extract personalized features while ensuring the alignment of data distribution and improving the model's generation capability.

GAN Optimization Goals The first component of PLGAN's optimization is to align the generated data distribution with real data using a standard GAN loss function. This helps train the generator and discriminator, ensuring that generated interaction sequences are close to real data. The GAN loss function is defined as:

$$\min_G \max_D \mathbb{E}_{X\sim p_{\text{data}}}[\log D(X)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z,H)))], \tag{5}$$

where $D(X)$ represents the discriminator's output (real data probability), and $G(z,H)$ is the generated sequence based on latent learner features.

Optimization of Learning Style Features U_{part} For personalized learning preferences, we use contrastive learning to minimize the distance between U_{part} vectors from different sequences of the same learner. The contrastive loss is defined as:

$$\mathcal{L}_{\text{c}} = \mathbb{E}_{(X_i,X_j)\sim p_{\text{data}}}\left[\|U_{\text{part}}^{(i)} - U_{\text{part}}^{(j)}\|_2^2\right], \tag{6}$$

This loss ensures consistency in U_{part} across different sequences of the same learner, stabilizing personalized feature modeling.

Optimization of Knowledge Features K The knowledge feature K represents the learner's mastery of knowledge. Its optimization involves minimizing the error in predicting interaction outcomes. The knowledge feature loss is given by:

$$\mathcal{L}_{\text{k}} = \mathbb{E}_{X,r\sim p_{\text{data}}}\left[\|f(X,K) - r\|_2^2\right], \tag{7}$$

where $f(X,K)$ predicts the outcome r based on X and K. By minimizing this loss, K becomes a more accurate representation of the learner's knowledge state.

Comprehensive Optimization Objective The total loss function combines the three components: GAN optimization, contrastive learning for U_{part}, and prediction-based optimization for K. The overall loss function is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{GAN}} + \lambda_1\mathcal{L}_{\text{c}} + \lambda_2\mathcal{L}_{\text{k}}, \tag{8}$$

where λ_1 and λ_2 balance the contributions of each component.

Experiments and Analysis

This section evaluates the effectiveness and generalizability of the proposed PLGAN model through experiments on four publicly available knowledge tracing datasets: Algebra2005 (AL05), ASSISTment 2009 (AS09), ASSISTment 2017 (AS17), and STATICS (STC). To further assess PLGAN's ability to capture personalized learner behaviors, we conducted a visualization study on the dynamic evolution of features and learner differentiation. Detailed descriptions of the datasets and experimental configurations are provided in the Appendix.

Experimental Design and Research Questions

The experiments are designed to address the following key research questions (RQs):

- **RQ1:** Can PLGAN significantly improve the performance of baseline models (e.g., SAKT, AKT, and DTransformer) in knowledge tracing tasks?
- **RQ2:** Can PLGAN's generative features maintain stability and generalization across different datasets?

Table 1: Performance Comparison Across Datasets

Model	Type	AL05		AS09		AS17		STC		AUC improv.
		AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	
SAKT	Original	77.96	79.40	75.24	71.78	69.06	67.35	80.29	78.87	2.88%
	uID	77.98	79.42	75.26	71.77	69.08	67.37	80.30	78.88	
	uID+PE	78.25	79.62	75.49	72.10	69.51	67.74	80.83	79.19	
	PLGAN	**79.87**	**80.22**	**77.48**	**73.95**	**72.47**	**71.53**	**81.47**	**80.88**	
AKT	Original	77.64	78.89	75.28	71.60	75.55	70.71	79.38	77.57	2.91%
	uID	77.71	78.95	75.30	71.65	75.63	70.80	79.41	77.63	
	uID+PE	79.75	80.49	76.32	72.58	76.53	71.49	80.61	78.53	
	PLGAN	**80.35**	**81.68**	**77.03**	**75.27**	**77.18**	**72.28**	**82.23**	**80.03**	
DTrans-former	Original	79.46	84.24	81.46	76.32	75.06	70.78	83.82	79.62	2.48%
	uID	79.31	84.23	81.45	76.24	75.12	70.85	83.90	79.67	
	uID+PE	80.98	85.31	82.08	77.25	76.12	71.48	84.87	80.52	
	PLGAN	**82.05**	**85.99**	**82.68**	**78.08**	**77.03**	**74.53**	**85.98**	**82.64**	
DKT+	Original	71.38	83.37	78.53	74.38	65.40	66.33	78.07	77.27	5.56%
	uID	73.15	83.55	78.76	74.45	66.56	67.45	78.25	77.35	
	uID+PE	73.96	84.09	78.98	75.08	68.53	68.82	78.72	77.81	
	PLGAN	**77.70**	**85.02**	**80.11**	**76.02**	**71.18**	**71.25**	**80.74**	**79.22**	

- **RQ3:** How do varying sequence lengths impact the feature extraction and performance of PLGAN?

- **RQ4:** How do PLGAN's generative features evolve during dynamic training, and can their feature aggregation reveal differences among learners?

Experimental Results and Analysis

Overall Performance (RQ1) The main goal of Knowledge Tracing (KT) is to track learners' dynamic knowledge states, which cannot be directly observed. Model effectiveness is evaluated by its ability to predict the accuracy of a learner's next interaction. PLGAN, while not a standalone KT model, enhances baseline models by generating personalized learner features. In the experiments, baseline models were progressively enhanced by adding learner identity (uID) and sequence positional encoding (uID+PE) to assess the contribution of each module and the performance boost from PLGAN.

The experimental results in Table 1 show that PLGAN significantly improves the performance of baseline models, as measured by AUC and ACC across four knowledge tracing datasets (ASSISTment 2005 (AL05), ASSISTment 2009 (AS09), ASSISTment 2017 (AS17), and Student Response Prediction (STC)). PLGAN enhances model adaptability and stability, driving notable performance gains.

PLGAN notably improves baseline models, including attention-based models like SAKT Pandey & Karypis (2019), AKT Ghosh et al. (2020), and DTransformer Yin et al. (2023), achieving an average AUC gain of 2.48% in DTransformer, despite its already strong baseline. It also demonstrates strong cross-model generalization, delivering a 5.56% AUC increase for non-attention models like DKT+ Yeung & Yeung (2018) on the AL05 dataset, showing PLGAN's adaptability across diverse architectures.

Additionally, PLGAN consistently improves performance across different datasets, even in more complex scenarios. On the STC dataset, PLGAN boosts AUC by at least 2%, highlighting its robustness across varying task complexities.

In conclusion, the experimental results validate PLGAN's ability to significantly enhance KT model performance by capturing personalized learner behaviors and integrating seamlessly with various model architectures, providing a robust solution for consistent improvement in KT tasks.

Contrastive Analysis of GAN Model Feature Generation (RQ2) To assess PLGAN's effectiveness in knowledge tracing, we compared it with SeqGAN Yu et al. (2017), TextGAN Liang et al. (2021), and TGAN Yoon et al. (2019), which represent classical sequence generation, text generation, and time-series modeling approaches, respectively. These models used the same feature generation and integration method as PLGAN, with the generated learner features fed into the DTransformer model for performance evaluation. The results are shown in Figure 3.

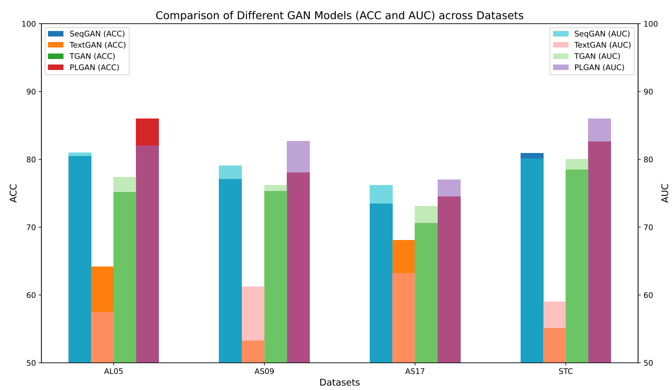

Figure 3: Performance comparison of four GAN models (SeqGAN, TextGAN, TGAN, and PLGAN) on four knowledge tracing datasets (AL05, AS09, AS17, and STC) using AUC and ACC metrics.

The results show that PLGAN outperforms the other models on all datasets and metrics (AUC and ACC). Notably, on the complex STC dataset, PLGAN achieves an AUC of 85.98, demonstrating its superior performance. PLGAN's success reflects its ability to effectively extract personalized learner features, which enhances its suitability for knowledge tracing tasks. Unlike other GAN models, PLGAN's design optimizes feature generation specifically for knowledge tracing, highlighting its unique contribution to personalized learning.

Sequence Length Adaptation Analysis (RQ3) This experiment examines the impact of sequence length on feature extraction. Learner features were generated from interaction sequences of varying lengths and integrated into the DTransformer model, evaluated using the AUC metric. The results, shown in Figure 4, reveal that PLGAN outperforms other models across most sequence lengths.

PLGAN excels in medium-length sequences (e.g., length 100), achieving an AUC of 85.98 on the STC dataset, far outperforming SeqGAN (80.10), TGAN (80.05), and TextGAN (59.00). TextGAN, optimized for short text, consistently performs poorly, with AUC below 57.50. SeqGAN and TGAN

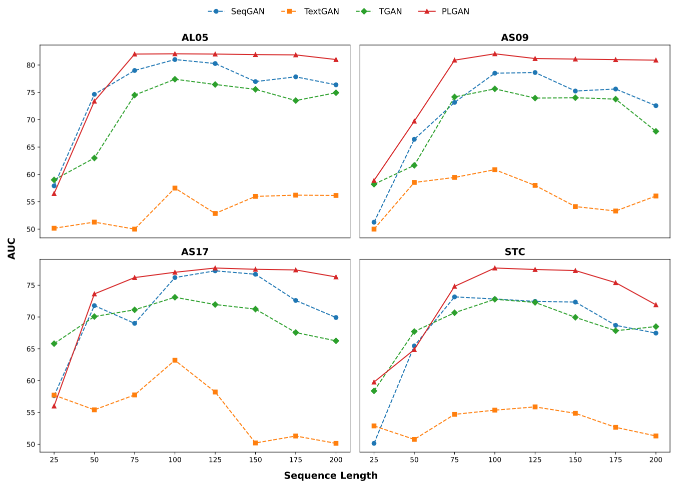

Figure 4: Performance comparison of GAN models (SeqGAN, TextGAN, TGAN, and PLGAN) on four datasets (AL05, AS09, AS17, and STC) for knowledge tracing, evaluated with AUC.

show some stability across varying lengths, but PLGAN surpasses them, e.g., in the AS09 dataset where PLGAN reaches an AUC of 82.68 at length 100, compared to TGAN's 76.18.

However, PLGAN shows slight degradation in long-sequence scenarios (e.g., length 200). On the STC dataset, its AUC drops from 85.98 to 79.00 as sequence length increases. This may be due to the contrastive learning mechanism's reduced effectiveness with longer sequences and increased noise. Nevertheless, PLGAN still outperforms other models.

In conclusion, PLGAN demonstrates strong adaptability and stability, especially for short and medium-length sequences. To mitigate performance decline in long sequences, future work could improve contrastive learning or introduce regularization techniques to enhance performance.

Visualization Analysis of Generated Features (RQ4) To address **RQ4**, PCA was used to reduce the dimensionality of the personalized feature vectors U_{part} generated by PLGAN and visualize their distribution across different training epochs. Figure 5 shows the evolution of feature distributions from the 1st to the 38th epoch in AS17.

As seen in the figure, features are dispersed in the early training stages (1st and 7th epochs), indicating the model's developing ability to capture personalized behavioral patterns. As training progresses (13th and 18th epochs), features begin to cluster into distinct groups, signifying improved differentiation of learner behaviors. In the later stages (26th and 38th epochs), the clustering becomes more pronounced, reflecting the model's stability in representing personalized behavioral patterns.

This dynamic evolution highlights PLGAN's ability to not only distinguish diverse behavioral patterns but also optimize learner differentiation over time. The more structured

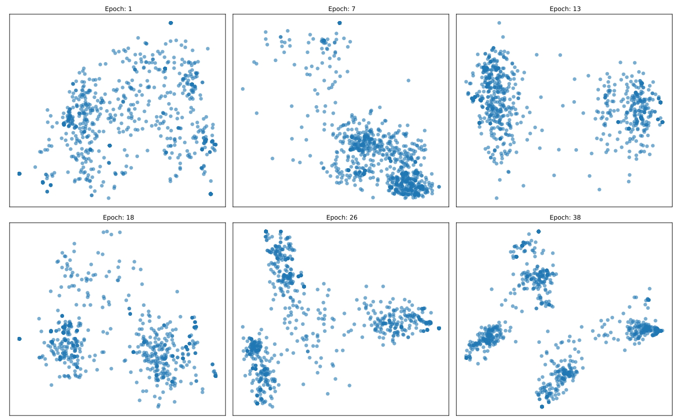

Figure 5: PCA visualization of the personalized feature vectors U_{part} generated by PLGAN at different epochs (1st, 7th, 13th, 18th, 26th, and 38th), showing feature distribution evolution.

and discriminative feature distributions in later stages provide stronger inputs for downstream knowledge tracing tasks.

In conclusion, the PCA results validate PLGAN's feature generation ability and its effectiveness in capturing learner differences, with improved clustering leading to more precise learner representations for enhanced knowledge tracing performance.

Conclusion and Future Work

This paper presents PLGAN, a novel Generative Adversarial Network (GAN) designed to improve the modeling of learners' personalized features in knowledge tracing. Experimental evaluations across various models and datasets demonstrate PLGAN's effectiveness, with the following key findings:

PLGAN enhances performance in knowledge tracing, outperforming traditional methods in AUC and ACC. Its generated features capture deeper behavioral patterns, and the model shows strong adaptability to different tasks, datasets, and data distributions.

Furthermore, PLGAN's feature evolution, visualized through dynamic clustering, reveals its capacity to model personalized learner behavior. The clustering process ensures stability and provides high-quality representations, validating PLGAN's robustness and convergence.

Future work will explore expanding PLGAN's applications to cross-disciplinary knowledge tracing and multimodal data analysis.In conclusion, PLGAN offers an advanced solution for feature generation in knowledge tracing, improving model performance and opening new avenues for the application of GANs in educational data modeling. This work contributes to personalized learning and precision education, both theoretically and technically.

Acknowledgments

The research was supported by the National Natural Science Foundation of China (Grant No.61877051) and the Natural Science Foundation of Chongqing, China (Grant No.CSTB2022NSCQ-MSX1419). We are grateful to Prof. Liangrui Peng of Tsinghua University for offering insights and suggestions that improved the rigor and coherence of this paper. Their perspectives expanded our research scope and were instrumental in forming the structure and content of the manuscript.

References

Abdelrahman, G., Wang, Q., & Nunes, B. (2023). Knowledge tracing: A survey. *ACM Computing Surveys*, *55*(11), 1–37.

Chalnick, A., & Billman, D. (1988). Unsupervised learning of correlational structure. In *Proceedings of the tenth annual conference of the cognitive science society* (pp. 510–516). Hillsdale, NJ: Lawrence Erlbaum Associates.

Choi, Y., Lee, Y., Cho, J., Baek, J., Kim, B., Cha, Y., ... Heo, J. (2020). Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the seventh acm conference on learning@ scale* (pp. 341–344).

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Ding, X., & Larson, E. C. (2019). Why deep knowledge tracing has less depth than anticipated. *International Educational Data Mining Society*.

Dweck, C. S. (1986). Motivational processes affecting learning. *American psychologist*, *41*(10), 1040.

Feigenbaum, E. A. (1963). The simulation of verbal learning behavior. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought*. New York: McGraw-Hill.

Feng, M., Heffernan, N., & Koedinger, K. (2009). Addressing the assessment challenge with an online system that tutors as it assesses. *User modeling and user-adapted interaction*, *19*(3), 243–266.

Ghosh, A., Heffernan, N., & Lan, A. S. (2020). Context-aware attentive knowledge tracing. In *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining* (pp. 2330–2339).

Henaff, M., Szlam, A., & LeCun, Y. (2016). Recurrent orthogonal networks and long-memory tasks. In *International conference on machine learning* (pp. 2034–2042).

Hill, J. A. C. (1983). A computational model of language acquisition in the two-year old. *Cognition and Brain Theory*, *6*, 287–317.

Khajah, M., Lindsey, R. V., & Mozer, M. C. (2016). How deep is knowledge tracing? *arXiv preprint arXiv:1604.02416*.

Koedinger, K. R., d. Baker, R. S. J., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. (2010). A data repository for the edm community: The pslc datashop. *Handbook of educational data mining*, *43*, 43–56.

Liang, G., On, B.-W., Jeong, D., Heidari, A. A., Kim, H.-C., Choi, G. S., ... Chen, H. (2021). A text gan framework for creative essay recommendation. *Knowledge-Based Systems*, *232*, 107501.

Long, T., Liu, Y., Shen, J., Zhang, W., & Yu, Y. (2021). Tracing knowledge state with individual cognition and acquisition estimation. In *Proceedings of the 44th international acm sigir conference on research and development in information retrieval* (pp. 173–182).

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Mao, S., Zhan, J., Wang, Y., & Jiang, Y. (2023). Improving knowledge tracing via considering two types of actual differences from exercises and prior knowledge. *IEEE Transactions on Learning Technologies*.

Matlock, T. (2001). *How real is fictive motion?* Doctoral dissertation, Psychology Department, University of California, Santa Cruz.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Ohlsson, S., & Langley, P. (1985). *Identifying solution paths in cognitive diagnosis* (Tech. Rep. No. CMU-RI-TR-85-2). Pittsburgh, PA: Carnegie Mellon University, The Robotics Institute.

Pandey, S., & Karypis, G. (2019). A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*.

Patikorn, T., Heffernan, N. T., & Baker, R. S. (2018). Assistments longitudinal data mining competition 2017: A preface. In *Proceedings of the workshop on scientific findings from the assistments longitudinal data competition, international conference on educational data mining.*

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

Pu, S., Yudelson, M., Ou, L., & Huang, Y. (2020). Deep knowledge tracing with transformers. In *Artificial intelligence in education: 21st international conference, aied 2020, ifrane, morocco, july 6–10, 2020, proceedings, part ii 21* (pp. 252–256).

Shen, S., Chen, E., Liu, Q., Huang, Z., Huang, W., Yin, Y., ... Wang, S. (2022). Monitoring student progress for learning process-consistent knowledge tracing. *IEEE Transactions on Knowledge and Data Engineering*.

Shen, S., Liu, Q., Chen, E., Wu, H., Huang, Z., Zhao, W., ... Wang, S. (2020). Convolutional knowledge tracing: Modeling individualization in student learning process. In *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval* (pp. 1857–1860).

Shrager, J., & Langley, P. (Eds.). (1990). *Computational models of scientific discovery and theory formation*. San Mateo, CA: Morgan Kaufmann.

Stamper, J., Niculescu-Mizil, A., Ritter, S., Gordon, G., & Koedinger, K. (2010). *Algebra i 2005-2006 and bridge to algebra 2006-2007: Development data sets from kdd cup 2010 educational data mining challenge.*

Tomić, B. B., Kijevčanin, A. D., Ševarac, Z. V., & Jovanović, J. M. (2022). An ai-based approach for grading students' collaboration. *IEEE Transactions on Learning Technologies*.

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

Wang, F., Huang, Z., Liu, Q., Chen, E., Yin, Y., Ma, J., & Wang, S. (2023). Dynamic cognitive diagnosis: An educational priors-enhanced deep knowledge tracing perspective. *IEEE Transactions on Learning Technologies*.

Yeung, C.-K., & Yeung, D.-Y. (2018). Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the fifth annual acm conference on learning at scale* (pp. 1–10).

Yin, Y., Dai, L., Huang, Z., Shen, S., Wang, F., Liu, Q., ... Li, X. (2023). Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In *Proceedings of the acm web conference 2023* (pp. 855–864).

Yoon, J., Jarrett, D., & Van der Schaar, M. (2019). Time-series generative adversarial networks. *Advances in neural information processing systems*, *32*.

Yu, L., Zhang, W., Wang, J., & Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 31).

Zhang, W., Zhang, Y., Liu, S., & Shang, X. (2022). Online deep knowledge tracing. In *2022 ieee international conference on data mining workshops (icdmw)* (pp. 292–297). IEEE.

Appendix

Datasets

This study utilizes four widely used knowledge tracing datasets: Algebra 2005-2006 (AL05), ASSISTments 2009-2010 (AS09), AS17, and STATICS (STC). These datasets capture learners' interaction records across various exercise sequences, including information such as exercise questions, response timestamps, and correctness. Table 2 provides statistical details for each dataset:

1. **Algebra 2005-2006 (AL05)**: Introduced by the KDD Cup 2010 Educational Data Mining Challenge, this dataset contains student responses to algebra problems from 2005 to 2006 (Stamper et al., 2010). For more details, refer to link.

Table 2: Statistics of the datasets used in this study

Statistic	**AL05**	**AS09**	**AS17**	**STC**
Number of learners	574	4,217	1,079	331
Number of questions	1,084	26,688	3,162	-
Number of skills	138	123	102	154
Number of interactions	809,694	346,860	942,816	142,124
Average sequence length	1,410.62	82.25	873.79	429.38

2. **ASSISTments 2009-2010 (AS09)**: This dataset originates from the ASSISTments online tutoring system established in 2004. It utilizes the updated *skill-build* version, which addresses data modeling issues and removes duplicate records (Feng et al., 2009). More information is available at link.

3. **ASSISTments 2017-2018 (AS17)**: Derived from the same source as AS09, this dataset features longer learner sequences, allowing multiple attempts on the same question and highlighting cumulative learning effects (Patikorn et al., 2018). More details are provided at link.

4. **STATICS (STC)**: This dataset comprises student interaction records from a Fall 2011 Carnegie Mellon University engineering statics course, providing an example of higher education learning processes (Koedinger et al., 2010). For further information, see link.

Implementation Details

The PLGAN model is implemented using the PyTorch deep learning framework and Python programming language. The gradient optimization utilizes the Adam optimizer (Loshchilov & Hutter, 2017) with a learning rate of 0.0002 and hyperparameters set to $\beta_1 = 0.4$ and $\beta_2 = 0.9$. The model is trained with a sequence length of 100 for 100 epochs, and the best-performing epoch is selected for analysis.

In this study, features generated by PLGAN are integrated into baseline knowledge tracing models (e.g., DTransformer). The dataset is divided into standard train-validation-test splits, with 60% for training, 20% for validation, and 20% for testing. Early stopping is employed to prevent overfitting, and experiments are repeated five times to mitigate the impact of randomness.

Evaluation Metrics and Methodology

The models are evaluated using three key metrics: AUC (Area Under the Curve), ACC (Accuracy), and RMSE (Root Mean Square Error). Higher AUC and ACC values indicate better prediction performance, while lower RMSE values reflect reduced errors. The best and second-best results for each metric are highlighted in **bold** and <u>underlined</u>, respectively.

A 5-fold cross-validation method is employed, with the final results representing the average scores across all folds. In each fold, 20% of the data is used for testing, 20% for validation, and 60% for training. This approach ensures the robustness of the results.

Experimental Environment

The experiments are conducted on a 64-bit Linux operating system equipped with a quad-core CPU and an NVIDIA Tesla V100 GPU with 32 GB of GPU memory. The main memory capacity is also 32 GB, providing a stable computational environment for deep learning experiments.