

Enhancing Knowledge Tracing via Breakpoint-Aware Sequence Augmentation

Chunli Huang^{ID}, Wenjun Jiang^{ID}, *Member, IEEE*, Kenli Li^{ID}, *Senior Member, IEEE*,
Ji Zhang^{ID}, *Senior Member, IEEE*, and Jie Wu^{ID}, *Fellow, IEEE*

Abstract—Knowledge Tracing (KT) aims to predict learners' future responses by modeling historical learning sequences. However, existing works often overlook the severe disruptions in temporal and interaction continuity arising from large time intervals between interactions, which leads to significant prediction errors. Therefore, we identify and validate this *breakpoint* effect through empirical studies, and try to address it via data augmentation (DA). Existing DA methods usually emphasize sequence diversity or similarity while neglecting the breakpoint effect in KT. Hence, we strive to enhance knowledge tracing via breakpoint-aware sequence augmentation and propose Collaborative Learning Path Augmentation (CLPA), a systematic three-phase method to enhance models' representational capacity and alleviate the breakpoint effects. First, identifying boundary interaction pairs that exhibit the temporal gap and response inconsistency. Second, leveraging cross-sequence collaborative information to infer functional pseudo-paths between identified boundaries. Third, reconstructing and inserting pseudo-learning paths into target sequences to restore sequence continuity; moreover, two refinements of the sampling quantity constraint and the distance-based sampling strategy are proposed to ensure augmentation quality. Comprehensive experiments on four real-world datasets validate the effectiveness of CLPA, which significantly enhances the target KT model's representational capacity and alleviates the breakpoint effects.

Index Terms—Knowledge tracing, breakpoint effects, data augmentation, collaborative learning path.

I. INTRODUCTION

KNOWLEDGE tracing (KT) [1] is a key task in smart education for predicting learners' future responses based on their historical interaction sequences. Temporal-agnostic KT models treat all interactions equally, neglecting their temporal impacts [2], [3], [4], [5]. Meanwhile, temporal-aware KT models incorporate temporal information to better trace knowledge

dynamics, by temporal feature embedding [6], temporal decay mechanism [7], [8], [9], explicit temporal modeling [10], [11], [12], or temporal kernel functions [13], [14]. However, existing KT models usually overlook the negative effects arising from disruptions in temporal and interaction continuity caused by large time intervals, leading to significant prediction errors. Therefore, in this paper, we identify and validate this *breakpoint* effect through empirical studies, and try to address it via data augmentation.

Fig. 1 illustrates a representative learning sequence S , consisting of several interactions of an online learner. It can be seen that there are significant variations in time intervals between adjacent interactions, which commonly exist in real learning scenarios. We define interactions with large time intervals from their preceding ones as *breakpoints* (e.g., s_{p_1} and s_{p_2} in Fig. 1), while the remaining interactions are termed *non-breakpoints*. In real-world scenarios, it is entirely natural for learners to experience temporal gaps due to regular breaks, schedule changes, or unrecorded offline learning activities. However, from a sequence modeling perspective, these large intervals cause abrupt unobservable shifts in learners' knowledge state (e.g., knowledge decay or external skill acquisition). The *breakpoint effect* emerges because these disruptions prevent KT models from leveraging the continuous sequential context to smoothly track knowledge dynamics, making accurate response prediction challenging at these specific positions. We conduct empirical studies in four KT datasets, and the results in Table III validate the breakpoint effect. It shows that although breakpoints constitute a small proportion of interactions, they consistently yield higher prediction errors than non-breakpoints. To mitigate this modeling challenge posed by breakpoints, we propose a novel data augmentation strategy. This approach encourages the model to learn smoother state transitions and enhances its representational capacity.

Existing data augmentation (DA) in KT primarily relies on perturbation-based strategies that modify original sequences to generate similar yet distinct sequences [9], [15]. In broader sequential domains, some methods enhance data diversity and density through behavior-aware perturbations [16], interval-aware operators for temporal uniformity [17], and the integration of cross-user patterns [18]. However, given the explicit prerequisite relationships among knowledge concepts, the learners' interaction exhibits strong sequentiality. Therefore, effective data augmentation must preserve both temporal continuity and interaction sequentiality. Although existing DA methods demonstrate promising results in various domains, they encounter several

Received 25 August 2025; revised 2 April 2026; accepted 1 June 2026. Date of publication 4 June 2026; date of current version 6 July 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62172149 and Grant 62172372 and in part by the Provincial Natural Science Foundation of Hunan under Grant 2026JJ50526. Recommended for acceptance by H. Hu. (*Corresponding author: Wenjun Jiang.*)

Chunli Huang, Wenjun Jiang, and Kenli Li are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: huangchunli@hnu.edu.cn; jiangwenjun@hnu.edu.cn; lkl@hnu.edu.cn).

Ji Zhang is with the School of Science, Engineering and Digital Technologies, University of Southern Queensland, Toowoomba, QLD 310012, Australia (e-mail: Ji.Zhang@unisu.edu.au).

Jie Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: jiewu@temple.edu).

Digital Object Identifier 10.1109/TKDE.2026.3700199

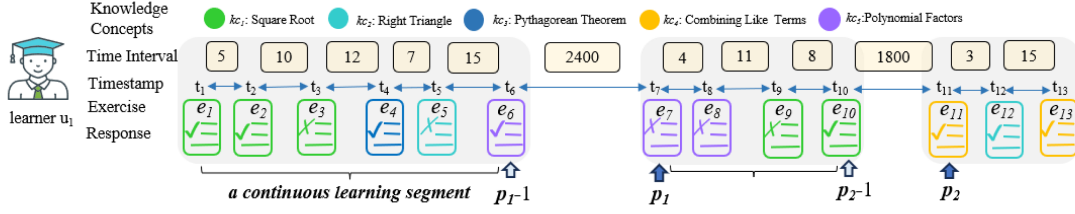


Fig. 1. Illustration of a learner's interaction sequence on an online learning platform. Each interaction s_i comprises four components: knowledge concept, timestamp, exercise, and response (correct or incorrect). Interactions in positions p_1 and p_2 feature substantial temporal gaps from their preceding interactions, thus s_{p_1} and s_{p_2} are identified as breakpoints.

fundamental limitations when applied to learning sequences for KT: 1) They mainly focus on enhancing sequence diversity or similarity, but they usually neglect the breakpoint effect as well as the sequence continuity and interaction sequentiality, which reduces their effectiveness in KT. 2) Many approaches lack robust mechanisms for accurately identifying optimal insertion positions and controlling the quantity and quality of augmentation, which may inject noise rather than meaningful augmentation.

Motivated by the breakpoint effect and the above limitations, we strive to enhance knowledge tracing via breakpoint-aware sequence augmentation and propose a systematic three-phase framework, Collaborative Learning Path Augmentation (CLPA) to enhance models' representational capacity, thereby alleviating the breakpoint effect.

Phase 1. Locating Boundary Position Pairs: CLPA employs dual-criteria detection to accurately identify the boundary positions exhibiting temporal gaps and response inconsistencies, thereby mitigating arbitrary filling. By precisely localizing discontinuous regions, it addresses the limitation of existing methods that overlook breakpoint identification.

Phase 2. Inferring Potential Learning Path: We infer the potential pseudo-paths between identified boundaries by leveraging cross-sequence collaborative information. This effectively provides a highly relevant functional proxy that bridges temporal discontinuities, directly addressing the absence of contextual imputation strategies.

Phase 3. Filling Pseudo-Learning Path: CLPA inserts the pseudo-learning paths into the original target sequence to bridge temporal gaps and restore sequence continuity. To ensure augmentation quality and control noise injection, we incorporate two refinement mechanisms: 1) a *sampling quantity constraint* that dynamically limits the inserted path length to prevent over-augmentation while ensuring adequate context restoration; 2) a *distance-based sampling strategy* that selects the most semantically relevant interactions and preserves the interaction sequentiality, thereby maximizing the educational value of augmented content while minimizing the potential side effects of noise.

Crucially, CLPA aims to provide a functional proxy to bridge temporal gaps, rather than reconstructing learners' actual, unrecorded offline activities. By imputing semantically relevant pseudo-paths for training, our framework forces the KT model to learn smoother state transitions, thereby enhancing its representational capacity. Empowered by this robust capability, the model maintains accurate predictions when encountering real, unaugmented breakpoints during inference to mitigate the

breakpoint effect. The main contributions of this work can be summarized as follows:

- 1) We empirically identify and validate the breakpoint effects through real-world data analysis. We demonstrate that temporal and interaction discontinuities arising from large time intervals trigger abrupt shifts in knowledge state, posing fundamental modeling challenges in KT and being of high necessity to be addressed. (Section III)
- 2) We propose the Collaborative Learning Path Augmentation (CLPA), a novel framework that leverages cross-sequence collaborative patterns for contextual imputation. It addresses the fundamental limitation of existing methods by providing systematic strategies for breakpoint identification, pseudo-path inference, and sequence continuity restoration. Particularly, it incorporates a principled quality control mechanism to prevent over-augmentation and maximize the educational value of the imputed paths. By augmenting the training data with semantically relevant pseudo-paths, CLPA enhances the KT model's representational capacity and mitigates the breakpoint effect. (Section IV)
- 3) We conduct extensive experiments on four real educational datasets. The results demonstrate that CLPA not only consistently enhances the target model's performance beyond state-of-the-art methods but also outperforms existing sequential augmentations. Furthermore, our experiments also validate that CLPA enhances the model's representational capacity, ensuring accurate knowledge tracking across continuous segments and breakpoints. (Section V)

II. RELATED WORK

A. Knowledge Tracing

Knowledge tracing (KT) aims to model learners' knowledge state evolution over time, based on their historical interactions. Existing KT methods can be categorized into two groups based on whether they incorporate temporal information: temporal-agnostic and temporal-aware KT models.

Temporal-agnostic KT models: They assume that all interactions are equally informative, ignoring the temporal distance between them. For instance, Bayesian Knowledge Tracing (BKT) [1] employs a Hidden Markov Model with time-invariant transition probabilities. Similarly, DKT [2] uses RNNs to encode knowledge states but still overlooks temporal intervals. To achieve finer-grained tracking, DKVMN explicitly maps

TABLE I
THE TEMPORAL MODELING STRATEGIES IN REPRESENTATIVE KT METHODS

Category	Representative Models	Temporal Modeling Mechanism	Description
Temporal-agnostic	BKT [1], DKT [2], DKVMN [3], DKT+ [19] SKVMN [20], IEKT [4], SparseKT [5]	ignoring temporal information	No explicit time variables
Temporal-aware	SAINT+ [6]	introducing two temporal features	enhancing KT with temporal feature embeddings
	AKT [7], Dtransformer [9], LefoKT [8], HiTSKT [21], QKT [22]	applying temporal decay in attention weights	a distance-aware temporal decay to model forgetting
	LPKT [10], DKT+forgetting [23], RKT [11], MF-DAKT [24], DGMN [12], ELPKT [25]	explicitly temporal modeling	model time-driven knowledge update and forgetting
	HawkesKT [13], MRT-KT [14]	propose temporal kernel functions	control the adaptive temporal evolution

exercises to underlying concepts, where SKVMN [20] further enhances sequential modeling via Hop-LSTM. Additionally, IEKT [4] accounts for diverse comprehension by estimating student cognition and question sensitivity. However, these approaches treat all interactions equally, neglecting the temporal impacts between interactions.

Temporal-aware KT models: These models integrate time intervals to better capture evolving knowledge states. For example, SAINT+ [6] explicitly incorporates handcrafted temporal features into response embeddings. A prominent line of work adapts attention mechanisms to implicitly capture temporal decay and emphasize recent interactions, as seen in AKT [7] and LefoKT [8]. DTransformer [9] refines this with question- and knowledge-level temporal estimators, while HiTSKT [21] and QKT [22] utilize distance-aware attention to highlight recent sessions. Alternatively, other methods explicitly formulate time-dependent learning and forgetting dynamics. LPKT [10] and RKT [11] calculate temporal-based knowledge gains and knowledge forgetting. Similarly, MF-DAKT [24] and ELPKT [25] apply forgetting functions based on recent factors and the intervals between sessions, respectively. Finally, temporal kernel functions offer fine-grained continuous tracking. HawkesKT [13] captures cross-concept effects via Hawkes processes, where MRT-KT [14] further extends them using relation-specific kernels. Despite their success, these temporal-aware KT models largely ignore the breakpoint effect, an omission that severely limits their performance in capturing knowledge state evolution. To better illustrate the distinctions, we summarize the temporal modeling strategies of representative KT methods in Table I.

B. Data Augmentation

Data augmentation (DA) expands training data by perturbing existing samples or generating synthetic data, which is widely adopted to enhance model generalization [34]. Given the sequential nature of KT modeling, our review focuses on sequential DA methods. These approaches are broadly categorized by their reliance on model training: 1) *model-based* DA, which integrates augmentation within the training process, and 2) *model-agnostic* DA, which operates independently.

Model-based DA employs trainable modules and auxiliary objectives to enhance data [35]. For example, Luo et al. [26] prompt large language models to insert predicted preferred items into sequences, while BARec [28] and ASRec [29] synthesize

TABLE II
THE REPRESENTATIVE DATA AUGMENTATION METHODS. “AUG.” IS THE ABBREVIATION OF AUGMENTATION.

Types	Methods	Aug. Level	Aug. Mode
Model-based augmentation	Llama4Rec [26]	item-level	single-sequence
	S^3 -Rec [27]	hybrid	cross-sequence
	BARec [28], ASRec [29], EC4SRec [30]	sequence-level	cross-sequence
Model-agnostic augmentation	SDA [15], MBASR [16], CL4SRec [31], TiCoSeRec [17]	hybrid	single-sequence
	RepPad [32]	sequence-level	single-sequence
	SDA_{inter} [18], BayMAN [33]	sequence-level	cross-sequence

pseudo-prior items to improve Transformer performance on short sequences. Furthermore, contrastive learning is widely adopted to enforce similarity between original and augmented data. Within this framework, Zhou et al. [27] model multi-level correlations (e.g., item-attribute, sequence-segment) to enrich representations, and Lei et al. [30] utilize explanation-guided augmentations for recommendation.

Model-agnostic DA focuses solely on transforming input sequences independently. Common sequence-level heuristic operators include interval-aware cropping and masking to uniformize sequences [17], as well as reusing original interactions as padding [32]. In knowledge tracing and recommendation, stochastic transformations like item masking and permutation are frequently employed to generate self-supervision signals [15], [31]. To further diversify training samples, other strategies incorporate cross-user sequential patterns [18] and behavior-aware perturbations [16]. Additionally, graph-based augmentations, such as replacing target points-of-interest (POIs) with similar ones in BayMAN [33], effectively enhance connectivity without model-in-the-loop optimization. Table II summarizes the representative DA methods.

In existing works, we note that no augmentation method specifically addresses the breakpoint effect, lacking strategies to bridge temporal gaps and enhance sequence continuity via learning path augmentation. However, as demonstrated in [36], well-designed data augmentation can match or even surpass contrastive learning in alleviating data sparsity, but with significantly lower computational overhead. Motivated by this, we

TABLE III
THE RESULTS OF EMPIRICAL STUDY. THE LOWEST PERFORMANCE IS **BOLDED**.

Data	Test Set	AKT				CL4KT			
		MAE ↓	RMSE ↓	AUC ↑	ACC ↑	MAE ↓	RMSE ↓	AUC ↑	ACC ↑
Assist2012	Ds_entire	0.3219	0.4089	0.7752	0.755	0.3468	0.4206	0.7444	0.7399
	Ds_break^c	0.3176	0.4044	0.7822	0.7612	0.3429	0.4163	0.753	0.7464
	Ds_break	0.328	0.4153	0.7652	0.7459	0.3526	0.4265	0.7319	0.7305
Assist2017	Ds_entire	0.3721	0.4324	0.7677	0.7124	0.4149	0.4548	0.7007	0.6788
	Ds_break^c	0.3669	0.428	0.7698	0.719	0.4097	0.451	0.7105	0.6854
	Ds_break	0.3816	0.4403	0.7612	0.7005	0.4244	0.4617	0.6985	0.6666
Junyi	Ds_entire	0.3134	0.4011	0.805	0.7681	0.3462	0.4216	0.7556	0.7423
	Ds_break^c	0.3014	0.395	0.8091	0.7772	0.3376	0.4138	0.7631	0.7549
	Ds_break	0.3216	0.4132	0.8007	0.7523	0.3627	0.4344	0.7511	0.7212
Ednet	Ds_entire	0.3704	0.4459	0.738	0.7005	0.4266	0.4641	0.6643	0.6594
	Ds_break^c	0.3666	0.4419	0.7392	0.706	0.4212	0.4609	0.6806	0.6654
	Ds_break	0.376	0.4518	0.7205	0.6922	0.4333	0.469	0.6384	0.6503

All experimental results are the average of 5 individual tests.

propose a novel, model-agnostic augmentation strategy tailored for knowledge tracing. Our approach explicitly resolves the breakpoint effect in existing KT modeling while maintaining high computational efficiency.

III. PROBLEM FORMULATION AND EMPIRICAL STUDY

A. Problem Formulation

In online learning platforms, suppose there is a learner set $\mathcal{U}$, an exercise set $\mathcal{E}$, and a knowledge concept set $\mathcal{C}$, respectively. Each learner $u \in \mathcal{U}$ is associated with a historical interaction sequence up to the time step t in chronological order $\mathcal{S} = \{s_1, s_2, \dots, s_j, \dots, s_t\}$. The interaction s_j is a quadruple, where $s_j = (e_j, c_j, r_j, t_j)$. It consists of the exercise e_j that the learner u answers at timestamp t_j , the knowledge concept c_j contained in e_j , and the corresponding response r_j . r_j equals 1 if u answers e_j correctly and otherwise 0.

Definition. Prediction error: Knowledge tracing aims at predicting learners' responses $\hat{r}_{t+1}$ on future exercises e_{t+1} based on the historical interaction sequences $\mathcal{S}$. The models' prediction error on each interaction denotes the absolute difference between the ground truth r_{t+1} and the prediction $\hat{r}_{t+1}$, which is formulated as:

$$error_{t+1} = |r_{t+1} - \hat{r}_{t+1}| \quad (1)$$

Problem: To reduce the breakpoint effects, we augment the original sequence $\mathcal{S}$ by inserting functional pseudo-paths to bridge temporal gaps and enhance sequence continuity. The augmented sequence $\tilde{\mathcal{S}}$ is then used to train the KT models in place of $\mathcal{S}$. The task can be formulated as follows:

$$\hat{r}_{t+1} = P\left(r_{t+1} = 1 \mid \left(\tilde{\mathcal{S}}, e_{t+1}\right)\right) \quad (2)$$

B. Empirical Study

We conduct empirical studies to validate breakpoint effects in real-world datasets, focusing on two key aspects: 1) whether KT models exhibit significant prediction errors at breakpoint positions, and 2) whether such errors propagate to their subsequent s interactions.

Specifically, we first train two representative KT models, AKT [7] and CL4KT [15], on four real-world educational datasets: Assist2012, Assist2017, Junyi, and Ednet, as introduced in Section V-A. Then, we evaluate the well-trained KT models on the corresponding test sets and calculate the prediction errors for each interaction based on (1). We further split the entire test set Ds_entire into two subsets: Ds_break and Ds_break^c . The subset Ds_break contains all temporal breakpoints (e.g., intervals > 30 min from the preceding interactions) and their subsequent k interactions ($k \geq s$) that are likely to be affected by breakpoints. The complementary subset Ds_break^c contains the remaining interactions, such that $Ds_entire = Ds_break \cup Ds_break^c$.

To explore the correlation between breakpoints and KT model performance, we evaluate both error metrics (MAE and RMSE) and accuracy metrics (AUC and ACC) across the entire test set Ds_entire and its two subsets (Ds_break and Ds_break^c). To ensure a fair comparison between Ds_break and Ds_break^c , we select k such that Ds_break and Ds_break^c contain an equal number of interactions. Specifically, k is determined by $n_b(1+k) = \lfloor \frac{N}{2} \rfloor$, where n_b denotes the number of breakpoints, and N is the total number of interactions in Ds_entire .

The results are shown in Table III. We observe that 1) both MAE and RMSE are the highest in Ds_break , indicating poorer prediction accuracy at breakpoints and their subsequent interactions; 2) both MAE and RMSE of the entire test set Ds_entire lie between those of Ds_break and Ds_break^c ; 3) the lowest MAE and RMSE are observed in Ds_break^c . AUC and ACC exhibit consistent opposite trends with MAE and RMSE. These results validate that: 1) although breakpoints constitute a small proportion of interactions, they consistently yield higher prediction errors than non-breakpoints; 2) the higher errors at breakpoints propagate to several subsequent interactions, causing their prediction degradation. These findings highlight the importance of improving KT models' performance by alleviating the breakpoint effect.

IV. COLLABORATIVE LEARNING PATH AUGMENTATION

We propose a systematic three-phase method named Collaborative Learning Path Augmentation (CLPA) to directly tackle the

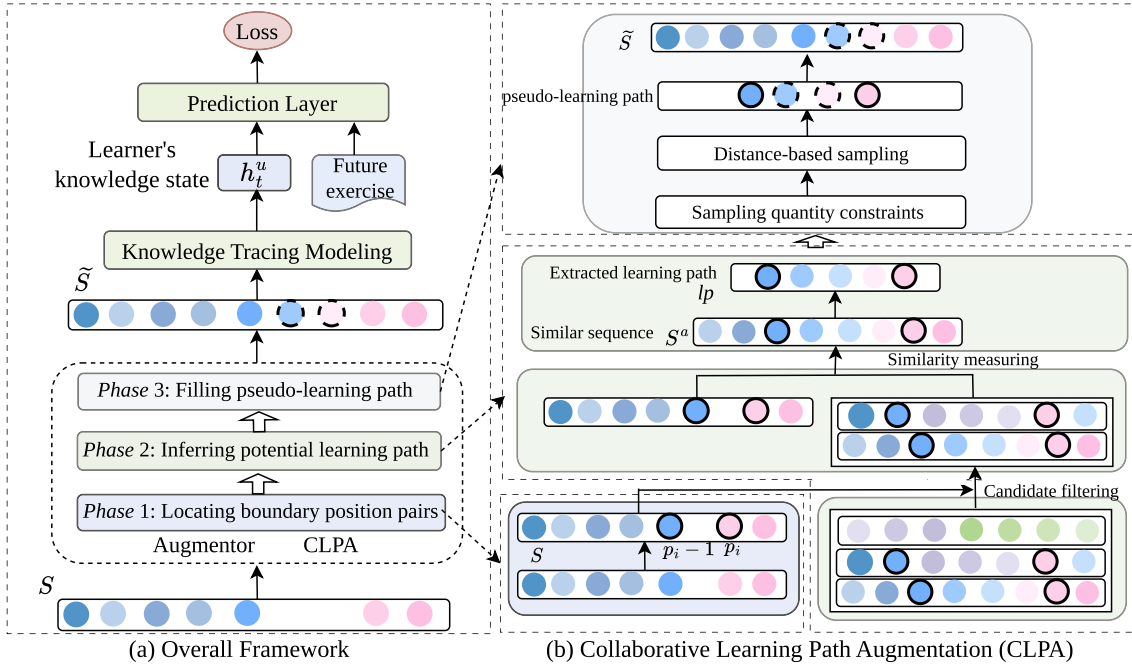


Fig. 2. The overall architecture of the proposed CLPA.

core issue of the breakpoint effects in KT. Fig. 2 illustrates the overview of our method. It takes the original sequence S as input and generates an augmented sequence $\tilde{S}$ through a three-phase learning path augmentor, including *Phase 1: Locating Boundary Position Pairs*, *Phase 2: Inferring Potential Learning Path*, and *Phase 3: Filling Pseudo-Learning Path*. The augmented $\tilde{S}$ is used as the input to the downstream KT task to achieve better prediction results. The details of our CLPA method are introduced as follows.

A. Phase 1: Locating Boundary Position Pairs

The breakpoints arise from temporal gaps within sequences and are characterized by knowledge state shifts, leading to higher prediction errors compared to non-breakpoints, as discussed in Section III-B. To address the breakpoint effect via meaningful augmentation, we employ a dual-criteria detection mechanism to accurately identify boundary position pairs that exhibit the temporal gap and response inconsistency. A pseudo-path is inserted only when a temporal gap is accompanied by response inconsistency, which strongly indicates a latent knowledge state shift that requires context bridging.

1) *Locating the Boundary Position Pairs With the Temporal Gap*: We consider the position pair $(p-1, p)$ with a temporal gap if the time interval between interactions s_{p-1} and s_p is greater than a threshold θ_1 . The position pair $(p-1, p)$ with a temporal gap in sequence S is formulated as:

$$P = \{(p-1, p) \mid t_p - t_{p-1} \geq \theta_1, s_{p-1}, s_p \in S\} \quad (3)$$

where P is the set of boundary position pairs representing temporal gaps exceeding θ_1 in sequence S , with t_{p-1} and t_p denoting the timestamps of interactions s_{p-1} and s_p .

2) *Refining the Boundary Position Pairs With Response Inconsistency (RiC) Detection*: Learners often exhibit stable performance during continuous learning periods, punctuated by fluctuations, e.g., knowledge forgetting or updating across temporal gaps. The learners' response inconsistencies reflect their knowledge state shifts, posing modeling challenges. To address this, boundary position pairs with response inconsistency should be supplemented with functional pseudo-paths to smooth the knowledge state evolution. Specifically, for boundary position pairs (p_i-1, p_i) in set P , we assess the learner's response inconsistency at position p_i relative to its preceding continuous learning segment, rather than solely comparing positions p_i-1 and p_i . For the pair (p_i-1, p_i) , the response inconsistency $RiC(\cdot)$ is computed as follows:

$$RiC(p_i-1, p_i) = \frac{1}{|S[p_{i-1} : p_i-1]|} \sum_{j=p_{i-1}}^{p_i-1} \mathbb{I}(r_{p_i} \neq r_j) \quad (4)$$

where $S[p_{i-1} : p_i-1]$ is the preceding learning segment of s_{p_i} , and $|\cdot|$ denotes the length of the segment. r_j is the learner's response (0 or 1) in s_j . $\mathbb{I}(\cdot)$ is an indicator function to judge whether r_{p_i} equals r_j . For example, to evaluate the RiC between p_2 and p_2-1 in Fig. 1, we calculate the RiC between p_2 and its prior continuous segment $[p_1, p_2-1]$ to determine the RiC between p_2 and p_2-1 . In this work, a boundary position pair (p_i-1, p_i) is considered to exhibit response inconsistency when $RiC(p_i-1, p_i)$ exceeds a threshold θ_2 . The set of such boundary pairs P_s is defined as follows:

$$P_s = \{(p_i-1, p_i) \mid RiC(p_i-1, p_i) \geq \theta_2, (p_i-1, p_i) \in P\} \quad (5)$$

B. Phase 2: Inferring Potential Learning Path

After identifying the boundary position pairs, we identify the potential learning paths between them to bridge temporal gaps and improve sequence continuity. Unlike learning path recommendations (e.g., [37], [38]) that suggest ordered learning resource sequences to enhance learning efficiency, this part aims to infer potential learning paths to complete the educational context and bridge the temporal and interaction discontinuity in target sequences. The process involves three main steps: filtering candidate sequences, measuring learning records' similarity, and extracting the potential pseudo-paths.

1) *Candidate Sequence Filtering Based on Boundary Position Pairs*: For each position pair $(p_i - 1, p_i)$ in P_s , the boundary exercises e_{p_i-1} and e_{p_i} define the endpoints of the potential learning path lp . Accordingly, the task of learning path inference is formulated as finding a path that connects the left boundary e_{p_i-1} to the right boundary e_{p_i} . To better align with learners' real-world trajectories on online learning platforms, we select sequences containing the boundary exercises e_{p_i-1} and e_{p_i} as candidates that may contain potential learning segments of the target sequences.

The filtering procedure selects candidate sequences for all boundary position pairs $(p_i - 1, p_i)$ in the training set. Naive loop iteration for matching boundary positions is computationally expensive. An alternative solution is the table join, whereas direct full-table joins incur excessive memory overhead. To trade off time and memory during candidate filtering, we adopt a *blocked table join* strategy. Specifically, we partition all boundary positions into multiple subtables. Each subtable is then joined with the complete training data table using boundary exercises e_{p_i-1} and e_{p_i} as keys, significantly accelerating the process while reducing memory consumption.

2) *Learning Records Similarity Measurement via Overlapping Exercises*: Overlapping exercises among learners provide direct evidence of synchronized learning trajectories. Building on prior work that infers target responses from similar learners [39], we explicitly leverage collaborative information from cross-learner interactions to augment target sequences.

Specifically, we extract the subsequence from similar learners as the potential learning path for the target learner within the boundary position $(p_i - 1, p_i)$. To identify collaborative peers with similar learning trajectories, we calculate the Intersection-over-Union (IoU) of their historical exercise sets. A high IoU reflects similar learning trajectories and cognitive levels, providing the context needed to bridge temporal gaps. There are some scenarios where the learners share similar knowledge states but practice low-overlapping exercises. The embedding-based measures (e.g., similarity of state representation) can capture such underlying semantic correlations, but they heavily depend on well-trained KT models and incur substantial computational overhead. Meanwhile, the temporal alignment metrics (e.g., longest common subsequence) lead to limited candidate paths and computational complexity. Conversely, IoU serves as a transparent, model-agnostic heuristic operating on raw data, offering an efficient approach to quantify learning records' similarity. We calculate the IoU of the exercise sets $\mathcal{E}$ and $\mathcal{E}^a$ of

the target sequence S and a candidate sequence S^a , serving as a measure of fillable confidence C . It is formulated as follows:

$$C(S, S^a) = \frac{|\mathcal{E} \cap \mathcal{E}^a|}{|\mathcal{E} \cup \mathcal{E}^a|} \quad (6)$$

where $\cup$ and $\cap$ denote set union and intersection, respectively.

3) *The Potential Learning Path Construction*: The subsequence lp extracted from S^a with the maximal fillable confidence C , connecting boundaries e_{p_i-1} and e_{p_i} , serves as a proxy learning path for target S . To construct a KT-compatible pseudo-path, we directly inherit responses from lp and linearly interpolate L equally spaced timestamps between t_{p_i-1} and t_{p_i} . The constructed path with length L is formulated as:

$$lp = [s_1^a, s_2^a, \dots, s_L^a] \quad (7)$$

where lp denotes the learning path bridging the boundary exercises e_{p_i-1} and e_{p_i} in S^a , and $s_L^a = (e_L^a, c_L^a, r_L^a, t_L^{a'})$ is the L -th interaction within lp . Constrained by the boundary exercises e_{p_i-1} and e_{p_i} , the constructed proxy path retains the temporal sequentiality. Consequently, the inserted pseudo-path ensures that the interaction sequentiality of the target sequence is preserved and utilized during augmentation.

C. Phase 3: Filling Pseudo-Learning Path

We insert the pseudo-learning path into the original target sequence to restore sequence continuity and enable KT models to enhance its representation capacity. For the boundary position pair $p_i - 1$ and p_i in the target sequence S , the augmented sequences $\tilde{S}$ can be formulated as:

$$\tilde{S} = [s_1, \dots, s_{p_i-1}, \underbrace{s_1^a, s_2^a, \dots, s_L^a}_{lp}, s_{p_i}, \dots] \quad (8)$$

where $lp = [s_1^a, s_2^a, \dots, s_L^a]$ is the pseudo-path constructed from the candidate sequence S^a , with L denoting its length.

Unlike item-level augmentation, inserting entire subsequences inherently risks introducing substantial noise. To ensure augmentation quality and reduce noise injection, we further propose two refinement strategies: 1) a sampling quantity constraint dynamically limits the length of the filled paths based on temporal gap size, which prevents over-augmentation while ensuring adequate context restoration; 2) a distance-based sampling strategy selects the relevant and informative interactions for constructing pseudo-paths, which maximizes the educational value of augmented content while minimizing the potential side effects of noise.

1) *Sampling Quantity Constraints*: Determining the optimal imputation size across large time gaps is inherently challenging. To address this, we introduce a dynamic quantity constraint that jointly evaluates the constructed pseudo-path length L and the temporal gap $t_{p_i} - t_{p_i-1}$. The length M of the imputed pseudo-path is constrained as follows:

$$M = \begin{cases} \mathcal{U}\left(\left\lfloor \frac{t_{p_i} - t_{p_i-1}}{\theta_1} \right\rfloor, L\right), & \text{if } \left\lfloor \frac{t_{p_i} - t_{p_i-1}}{\theta_1} \right\rfloor < L \\ \lfloor \alpha \cdot L \rfloor, & \text{otherwise} \end{cases} \quad (9)$$

where $\mathcal{U}$ denotes discrete uniform sampling of an integer. $\frac{t_{p_i} - t_{p_i-1}}{\theta_1}$ represents the minimum filling quantity to bridge the

temporal gap between breakpoint s_{p_i} and its preceding interaction $s_{p_{i-1}}$. $\lfloor \cdot \rfloor$ is floor operation. L is the length of pseudo-path lp , and α is a scaling factor when L is insufficient.

The logic for the sampling quantity constraint is formulated under two distinct scenarios: 1) *Sufficient Length*: When the pseudo-path length L is adequate to cover the time interval $t_{p_i} - t_{p_{i-1}}$ (i.e., $\frac{t_{p_i} - t_{p_{i-1}}}{\theta_1} < L$), we employ a flexible padding strategy that stochastically determines the sampling size. 2) *Insufficient Length*: This scenario occurs when the constructed pseudo-paths, despite their substantial length, remain insufficient to cover the temporal gap between $s_{p_{i-1}}$ and s_{p_i} . To balance contextual imputation against the risk of noise from over-filling, we sample a controlled fraction α of the pseudo-path interactions to bridge the temporal gap $t_{p_i} - t_{p_{i-1}}$. We determine α for each dataset by assessing the model's performance when trained on the augmented data.

2) *Distance-Based Sampling*: To construct a high-quality pseudo-path, we select top M most informative interactions. The recency effect [40] suggests that the recent interactions have greater impacts on current performance [7], [22]. We hypothesize that interactions proximal to the right boundary (i.e., the breakpoint) possess stronger semantic relevance. Consequently, these recent interactions are prioritized to impute educational context and bridge temporal gaps. Following the approach in [41], we employ a distance-based sampling strategy that assigns higher scores to interactions closer to the right boundary (i.e., the breakpoint), thereby maximizing the sampling of semantically relevant interactions for context imputation. The score function $f(\cdot)$ is defined as:

$$f(m) = \beta^{L-m} \quad (10)$$

where $\beta \in (0, 1)$ regulates positional influence, while L represents lp 's length. m is the index of an interaction within lp . The sampling probability of the m -th interaction $p(m)$ is proportional to the score function $f(\cdot)$:

$$p(m) = \frac{f(m)}{\sum_{j=1}^L f(j)} \quad (11)$$

By sampling M high-probability interactions from lp and preserving their relative sequentiality, we construct a pseudo-path, serving as a functional proxy for contextually imputing the target breakpoint.

D. Time and Space Complexity Analysis

As an offline preprocessing step, we evaluate CLPA's computational overhead using the following notations: total training interactions N_Q , identified boundary pairs N_B ($N_B \ll N_Q$), average candidate sequences per breakpoint N_C , average sequence length L_S , extracted path length L , and sampled filling length M . Notably, they satisfy the constraint $M < L < L_S < L_{max}$ (L_{max} being the maximum training length).

Phase 1 (Breakpoint Identification) takes $\mathcal{O}(N_Q)$ time to linearly scan all interactions and the preceding segments of breakpoints. Storing the identified boundary position pairs incurs a minimal space complexity of $\mathcal{O}(N_B)$.

Phase 2 (Potential Learning Path Inference), the most resource-intensive phase, comprises three steps. 1) Candidate sequence filtering by our blocked table join: Partitioning the N_B boundary pairs into blocks of size D for table joining limits the peak memory overhead to $\mathcal{O}(D + N_Q + D \cdot F_{max})$, where F_{max} is the maximum interaction frequency of an exercise in training set. By bounding D , we effectively mitigate the Out-of-Memory (OOM) bottlenecks caused by Cartesian product explosions during full-table join. This yields a time complexity of $\mathcal{O}(\frac{N_B}{D} \cdot (D + N_Q))$. 2) Learning records similarity measurement: Computing the exercise IoU across N_C candidates for all N_B breakpoints yields a time complexity of $\mathcal{O}(N_B \cdot N_C)$, while storing the result requires $\mathcal{O}(N_B)$ space. (3) Potential learning path extraction: Extracting and storing the subsequence of length L for N_B breakpoints takes $\mathcal{O}(N_B \cdot L)$ in both time and space complexity.

Phase 3 (Pseudo-Path Filling) reconstructs sequences while controlling noise. 1) Sampling quantity constraints: Calculating the filling length M for N_B breakpoints, yielding $\mathcal{O}(N_B)$ time. 2) Distance-based sampling: Computing positional scores and sampling M interactions requires $\mathcal{O}(N_B \cdot L)$. 3) Target sequence augmentation: Inserting the M sampled interactions into original sequences (average length L_S) incurs $\mathcal{O}(N_B \cdot (L_S + M))$ in both time and space complexity.

In summary, guided by the length constraint $M < L < L_S$, the overall time complexity of CLPA reduces to $\mathcal{O}(N_Q + \frac{N_B \cdot N_Q}{D} + N_B \cdot N_C)$. The overall space complexity peaks at $\mathcal{O}(N_Q + D \cdot F_{max} + N_B \cdot L_S)$, which can be approximated as $\mathcal{O}(N_Q + D \cdot F_{max})$. This ensures that the offline augmentation remains highly scalable and executable on standard hardware without Out-of-Memory (OOM) risks.

V. EXPERIMENTS

To comprehensively evaluate our method, we aim to answer the following research questions: *RQ1*: To what extent does CLPA enhance base KT models compared against state-of-the-art (SOTA) baselines? *RQ2*: How does CLPA perform compared to SOTA sequential data augmentation methods? *RQ3*: How do the core components of CLPA contribute to its overall effectiveness? *RQ4*: How do varying CLPA hyperparameters affect the predictive performance of the base model trained on the corresponding augmented datasets? *RQ5*: Can CLPA effectively bridge temporal gaps in original sequences? *RQ6*: How do the predictions of the model differ when trained on the original versus the CLPA-augmented datasets? *RQ7*: Can CLPA enhance KT performance on breakpoint-affected test subsets? *RQ8*: What is the computational overhead of CLPA?

A. Datasets

The experiments are conducted on four real-world educational datasets that are widely used for KT tasks, as follows:

- **Assist2012**¹ and **Assist2017**² are collected from the ASSISTments,³ an online tutoring system established in 2004.

¹<https://sites.google.com/site/assistmentsdata/home/2012-13-school-data-withaffect>

²<https://sites.google.com/view/assistmentsdatamining/dataset>

³<https://new.assistments.org/>

TABLE IV
STATISTICS OF THE PROCESSED DATASETS

Statistics	Dataset			
	Assist2012	Assist2017	Junyi	Ednet
Learners	17,264	1704	4000	6000
Exercises	36,867	2521	681	11,757
Concepts	252	97	39	142
Interactions	2,504,117	939,989	517,206	845,202
Breakpoints rate	0.08	0.01	0.03	0.06
positive rate	0.70	0.37	0.67	0.62

They record the learning interaction of students answering math exercises during the academic year 2012-2013 and 2004-2007, respectively.

- **Junyi**⁴ is collected from Junyi Academy,⁵ a Chinese e-learning platform in 2015 [42]. It is a large-scale dataset. Considering the capacity of computation resources, we randomly sample 4000 learners' data for experiments.
- **Ednet**⁶ is the dataset of all student interactions collected over 2 years by Santa, a multi-platform AI tutoring service with more than 780 K users in Korea. It consists of students' question-solving logs and is also a large-scale dataset. To make the computation resources affordable, we randomly sample 6,000 learners.

Each interaction in these datasets contains learner ID, exercise ID, knowledge concept (or skill), and timestamp. For each dataset, incomplete records are discarded. Aligning with [10], we use only the first concept for multi-concept exercises. To illustrate breakpoint sparsity, we report their proportions across datasets under an illustrative $\theta_1 = 30$ minutes threshold. Table IV shows statistics of the processed datasets.

B. Baselines

To evaluate the effectiveness of our CLPA, we integrate it into a representative KT model, AKT [7], and compare its performance against two groups of baselines: 1) representative KT models trained on the original sequences and 2) the state-of-the-art data augmentation approaches, as follows:

1) Representative KT models trained on the original sequences: *i*) *Temporal-agnostic KT models* include DKT [2], DKVMN [3], SKVMN [20], CL4KT [15], SparseKT [5], and QIKT [43]. *ii*) *Temporal-aware KT models* include LPKT [10], AKT [7], Dtransformer [9], LefoKT [8], and Folibikt [44].

2) The state-of-the-art data augmentation approaches, including RepPad [32], MBASR [16], and SDA, which is a stochastic data augmentation built in CL4KT [15]. To align with the subsequences segment in MBASR, we split learners' sequences into subsequences by the interval threshold (i.e., 30 minutes). Additionally, we introduce two popularity-based imputation strategies: Pop_item fills breakpoints with top M globally popular exercises (where M follows (9); Pop_path bridges boundary exercises $e_{p_{i-1}}$ and e_{p_i} using the highest-probability transition

path, assigning the average correct rate as the response label of exercise nodes, with its length and sampling positions determined by (9) and (11).

C. Experimental Setup

We conduct all experiments using standard 5-fold cross-validation and report the average performance on the test sets. For each dataset, 20% of learners were used as the test set, and the remaining 80% were split as the training set (80%) and validation set (20%). Data augmentation methods are only applied to the training learners. We use a mini-batch size of 32, a fixed input sequence length of 100 (with a minimum of 3), and an embedding size of 128 for all methods.

For the baselines, we adopt hyperparameters as reported in their original papers or carefully tune them, reporting the best results. For our CLPA, we tune the core boundary detection thresholds (θ_1, θ_2) and sampling parameters (α, β) across pre-defined search spaces, as introduced in Section V-G. All models are trained for up to 300 epochs, with early stopping applied if the validation AUC does not improve for 10 consecutive epochs. We conduct all experiments on a Linux server with an NVIDIA GeForce RTX 4090 GPU. Our code is available at <https://github.com/HuangCL22/CLPA>.

D. Overall Performance Improvement (RQ1)

To validate the effectiveness of the proposed CLPA, we integrate it with AKT [7] and compare its performance against two groups of KT models. Table V presents the comparison results on four datasets.

Overall, our CLPA significantly enhances the performance of AKT across all four datasets. Trained on CLPA-augmented data, AKT consistently outperforms its vanilla version and all other representative KT baselines, validating the effectiveness of our CLPA. By providing collaborative functional proxies to bridge temporal gaps, CLPA helps the model learn robust state transitions. This fundamentally mitigates the model's representational disruptions caused by the breakpoint effect.

Going deeply into the representative temporal-aware KT models, we analyze the possible reasons for the results. LPKT achieves strong performance by modeling temporal continuity, while it overlooks interaction continuity. Methods like AKT, DTransformer, and lefoKT employ temporal decay mechanisms to emphasize recent interactions. However, this strategy may lead to over-reliance on recent interactions that could become unreliable or uninformative when disrupted by large time intervals. While DTransformer further captures stable knowledge states within a learner's sequence, it overlooks the unobservable context that cause potential instability in knowledge evolution. By inserting the potential learning paths into the original sequences, CLPA supplies informative and reliable contextual interactions for breakpoints, which enables "AKT +CLPA" to achieve the best performance.

E. Compare With Representative DA Methods (RQ2)

To validate the advantages of our CLPA, we compare state-of-the-art DA methods, using two representative KT models: AKT

⁴<https://pslclatashop.web.cmu.edu/DatasetInfo?datasetId=1275>

⁵<https://www.junyiacademy.org/>

⁶<https://github.com/rriid/ednet>

TABLE V

PERFORMANCE COMPARISONS OF ALL METHODS. THE BEST RESULT IN EACH ROW IS IN **BOLD**. THE BEST RESULT IN EACH KT GROUP IS UNDERLINED. Imp_i REPRESENTS THE IMPROVEMENT OF OUR METHOD COMPARED WITH THE BEST BASELINE OF EACH CATEGORY.

Dataset	Metric	Temporal-agnostic KT models						imp_1	Temporal-aware KT models					imp_2	AKT +CLPA
		DKT	DKVMN	SKVMN	CL4KT	QIKT	SparseKT		AKT	LPKT	Dtransformer	Folbikt	lefoKT		
Assist2012	AUC	0.7301	0.7377	0.7009	0.7444	0.7612	0.7667	3.07%	0.7752	0.7785	0.7765	0.7784	0.7788	1.46%	0.7902
	ACC	0.7344	0.7431	0.7224	0.7399	0.7469	<u>0.7507</u>	1.61%	0.7550	0.7552	0.7559	<u>0.758</u>	0.7567	0.63%	0.7628
	RMSE	0.4249	0.4216	0.4335	0.4206	0.415	<u>0.4132</u>	2.18%	0.4089	<u>0.4073</u>	0.4096	0.4082	0.4092	1.15%	0.4042
	MAE	0.3591	0.3507	0.3739	0.3468	0.343	<u>0.3275</u>	5.07%	0.3219	0.3206	0.3219	0.3223	<u>0.3186</u>	2.42%	0.3109
Assist2017	AUC	0.7196	0.7283	0.7221	0.7007	<u>0.7767</u>	0.7578	1.26%	0.7677	<u>0.7802</u>	0.7533	0.7651	0.7469	0.81%	0.7865
	ACC	0.6848	0.6915	0.6833	0.6788	0.7199	0.7070	1%	0.7124	<u>0.7193</u>	0.7035	0.7113	0.7008	1.14%	0.7275
	RMSE	0.4496	0.4375	0.4491	0.4548	<u>0.4285</u>	0.4368	2.93%	0.4324	<u>0.4311</u>	0.4379	0.4329	0.4406	1.65%	0.4240
	MAE	0.3995	0.3861	0.3997	0.4149	<u>0.3616</u>	0.3725	0.72%	0.3721	<u>0.3710</u>	0.3767	0.3719	0.3904	3.23%	0.3590
Junyi	AUC	0.7653	0.7630	0.7512	0.7556	<u>0.8084</u>	0.8003	1.03%	0.8050	0.8081	0.8069	<u>0.8102</u>	0.8084	0.80%	0.8167
	ACC	0.7466	0.7461	0.7395	0.7423	<u>0.7673</u>	0.7639	1.03%	0.7681	0.7702	0.7687	<u>0.7713</u>	0.7707	0.51%	0.7752
	RMSE	0.4175	0.4180	0.4227	0.4216	<u>0.4000</u>	0.4032	1.1%	0.4011	0.3995	0.4013	<u>0.3987</u>	0.3998	0.78%	0.3956
	MAE	0.3482	0.3491	0.3554	0.3462	<u>0.3141</u>	0.3207	4.01%	0.3134	0.3075	0.3083	0.306	<u>0.3042</u>	0.89%	0.3015
Ednet	AUC	0.6645	0.6662	0.6353	0.6643	0.7276	<u>0.7304</u>	6.50%	0.7380	<u>0.7415</u>	0.7352	0.7388	0.7402	4.91%	0.7779
	ACC	0.6597	0.6605	0.6477	0.6594	0.6914	<u>0.6940</u>	4.52%	0.7005	<u>0.7138</u>	0.6956	0.6981	0.7002	1.63%	0.7254
	RMSE	0.4636	0.4633	0.4702	0.4641	0.4473	<u>0.4464</u>	4.35%	0.4459	<u>0.4356</u>	0.4447	0.4431	0.4427	1.97%	0.4270
	MAE	0.4280	0.4268	0.4394	0.4266	0.3961	<u>0.3858</u>	10.16%	0.3704	<u>0.3681</u>	0.3853	0.3845	0.3806	5.84%	0.3466

“A +B” denotes the vanilla model “A” trained on datasets augmented by “B”, without any architectural modifications.

TABLE VI

PERFORMANCE COMPARISON BETWEEN CLPA AND REPRESENTATIVE SEQUENTIAL AUGMENTATION METHODS WITH THE BACKBONE MODELS AKT AND CL4KT. THE BEST RESULT IS **BOLDED**. THE SECOND ONE IS UNDERLINED.

Dataset	Metric	AKT							CL4KT					
		original	+SDA	+RepPad	+MBASR	+Pop_path	+Pop_item	+CLPA	original (SDA)	+RepPad	+MBASR	+Pop_path	+Pop_item	+CLPA
Assist2012	AUC	0.7752	0.7763	<u>0.78</u>	0.7785	0.7781	0.7295	0.7902	0.7444	<u>0.7571</u>	0.7463	0.7538	0.7232	0.7658
	ACC	0.755	0.7562	<u>0.7576</u>	0.7572	0.7569	0.7307	0.7628	0.7399	<u>0.7438</u>	0.7405	0.7416	0.6884	0.7485
	RMSE	0.4089	0.4107	<u>0.4081</u>	0.4097	0.4088	0.4261	0.4042	0.4206	<u>0.4156</u>	0.4196	0.4193	0.4487	0.4135
	MAE	0.3219	0.3196	<u>0.3178</u>	0.3213	0.3162	0.3557	0.3109	0.3468	<u>0.3392</u>	0.3462	0.3459	0.3978	0.3358
Assist2017	AUC	0.7677	0.7689	<u>0.7736</u>	0.7695	0.7699	0.7529	0.7865	0.7007	<u>0.7157</u>	0.7015	0.7017	0.6972	0.7204
	ACC	0.7124	0.7147	<u>0.7185</u>	0.7152	0.7164	0.7018	0.7275	0.6788	<u>0.6841</u>	0.6784	0.6828	0.6796	0.6856
	RMSE	0.4324	0.4316	<u>0.4291</u>	0.4311	0.4316	0.4383	0.4240	0.4548	<u>0.4513</u>	0.4526	0.4535	0.4612	0.4498
	MAE	0.3721	0.3713	<u>0.3675</u>	0.3732	0.3703	0.3792	0.3590	0.4149	<u>0.4001</u>	0.4133	0.4057	0.4028	0.3989
Junyi	AUC	0.8050	0.8083	<u>0.8107</u>	0.8087	0.8101	0.8022	0.8167	0.7556	<u>0.7569</u>	0.7562	0.7559	0.7555	0.7595
	ACC	0.7681	0.7692	<u>0.7711</u>	0.7704	0.7011	0.7662	0.7752	0.7423	<u>0.7435</u>	0.7429	0.7432	0.7401	0.7447
	RMSE	0.4011	0.4002	<u>0.3993</u>	0.4017	0.3996	0.4026	0.3956	<u>0.4216</u>	<u>0.4223</u>	0.4229	0.4221	0.4230	0.4210
	MAE	0.3134	0.3128	0.3042	<u>0.3023</u>	0.3057	0.3148	0.3015	0.3462	<u>0.3457</u>	0.3471	0.3453	0.3462	0.3423
Ednet	AUC	0.738	0.7415	<u>0.7609</u>	0.7428	0.7581	0.7047	0.7779	0.6643	<u>0.6679</u>	0.6654	0.6657	0.6596	0.6763
	ACC	0.7005	0.7078	<u>0.7146</u>	0.7104	0.7137	0.6801	0.7254	0.6594	<u>0.6613</u>	0.6607	0.6610	0.6561	0.6649
	RMSE	0.4459	0.4431	<u>0.4324</u>	0.4411	0.4426	0.4541	0.427	0.4641	<u>0.4626</u>	0.4672	0.4635	0.4659	0.4605
	MAE	0.3704	0.3674	<u>0.3585</u>	0.3649	0.3654	0.4069	0.3466	0.4266	0.4251	0.4268	<u>0.4240</u>	0.4261	0.4235

and CL4KT. The comparative results, presented in Table VI, reveal the following key findings:

Our CLPA outperforms the state-of-the-art DA methods, demonstrating its effectiveness. This primarily stems from three factors: 1) By imputing collaborative functional proxies at breakpoints, CLPA effectively mitigates the representational disruptions caused by temporal gaps, enabling smoother and accurate modeling of knowledge state transitions; 2) Unlike existing single-sequence methods, CLPA leverages cross-sequence collaborative priors to enrich the context of breakpoints, enhancing the model’s capacity to trace complex knowledge transitions; and 3) Compared to the two popularity-based augmentation methods, Pop_item and Pop_path, the superiority of CLPA validates that our performance gains are driven by the semantic relevance of the collaborative paths, rather than merely from

the regularization effect of bridging temporal gaps with high-frequency interactions.

CLPA achieves greater performance gains for AKT compared to CL4KT. This may be because that CL4KT generates positive/negative samples from augmented sequences of the same/other learners and optimizes a contrastive loss to align semantically similar pairs and separate dissimilar ones. Since our CLPA inserts subsequences from other learners into the target sequence, CL4KT may misclassify the pseudo-subsequences as negatives. This disrupts contrastive loss convergence and limits its performance improvement.

RepPad performs the best among the baselines. The reason may be that RepPad pads the original sequence itself, which increases the training data density and improves models’ performance on short sequences for its implicit oversampling.

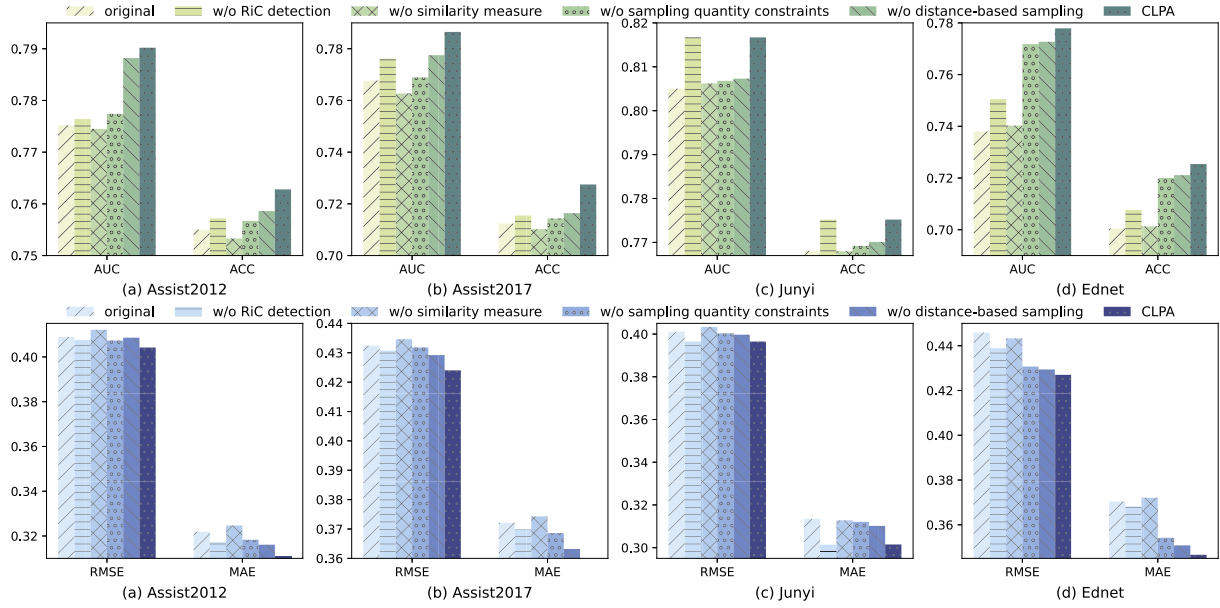


Fig. 3. Predictive performance of AKT trained on datasets augmented by various CLPA variants.

Meanwhile, both SDA and MBASR add perturbations within individuals' sequences, enhancing the generalization ability of KT models. Compared to other methods, RepPad avoids introducing additional noise by only padding the original sequence itself, making it a competitive baseline.

F. Ablation Study (RQ3)

To assess the contributions of components in CLPA, we conduct an ablation study on the following variants: 1) *w/o similarity measure*: randomly selects learners' subsequences as the extracted learning path. 2) *w/o RiC detection*: applies learning path augmentations to all boundary position pairs with temporal gaps, without response inconsistency detection. 3) *w/o sampling quantity constraints*: directly inserts the extracted learning paths into the given boundary position pairs. 4) *w/o distance-based sampling*: randomly selects M interactions from the extracted learning paths to construct the pseudo-learning path. From Fig. 3, we observe that:

Removing any component of CLPA leads to a performance decline in AKT, indicating that all components jointly contribute to its effectiveness. Notably, removing the similarity measure causes the most severe performance drop across all four datasets. This validates that sequences of similar learners provide valuable context for inferring knowledge states at breakpoints, allowing their subsequences to serve as valid functional proxies to augment target sequences.

Both response inconsistency detection and sampling quantity constraints can effectively mitigate the noise introduced. Compared to "w/o response inconsistency detection", our CLPA improves the AUC of AKT by 1.78%, 1.33%, and 3.65% on Assist2012, Assist2017, and EdNet. On Junyi, we set θ_2 for response inconsistency in CLPA to 0, which yields identical performance to this variant. Compared to "w/o the sampling quantity constraint", our CLPA improves the AUC of AKT by

1.65%, 2.29%, 1.23%, and 0.79% on Assist2012, Assist2017, Junyi, and EdNet, respectively. These results validate that response inconsistency detection and sampling quantity constraint effectively suppress noise during augmentation, leading to notable performance gains.

When removing distance-based sampling in CLPA, the performance of AKT drops. It demonstrates that the pseudo-learning paths constructed by distance-based sampling provide effective contexts for modeling breakpoints.

G. Hyperparameter Sensitivity Analysis (RQ4)

As our main goal is to construct high-quality pseudo-paths filling in appropriate positions, we carefully tune the boundary-detection thresholds θ_1 (3) and θ_2 (5), alongside sampling parameters α (9) and β (10). To select the optimal parameters, we evaluate the performance of "AKT +CLPA", i.e., AKT on the augmented training set, from the following search spaces: $\theta_1 \in \{10, 30, 60, 90, 120\}$, $\theta_2, \alpha \in \{0, 0.1, \dots, 0.9\}$, and $\beta \in \{0.1, 0.2, \dots, 0.9\}$. The experimental results on Ednet are shown in Fig. 4.

The thresholds θ_1 and θ_2 jointly control the filling positions by detecting breakpoints. As shown in Fig. 4(a) and (b), as these two parameters increase, the performance of the "AKT +CLPA" exhibits a trend of initially rising and then gradually declining overall. The results validate that too small values cause over-augmentation by misclassifying pauses or minor response fluctuations as breakpoints, thereby disrupting inherent sequence continuity. Conversely, overly large values cause under-augmentation, failing to capture severe temporal gaps and critical state shifts. The optimal settings achieve this trade-off, accurately locating the knowledge state shifts without disrupting sequence continuity.

Parameters α and β control the length and semantic quality of the imputed functional proxies. As shown in Fig. 4(c),

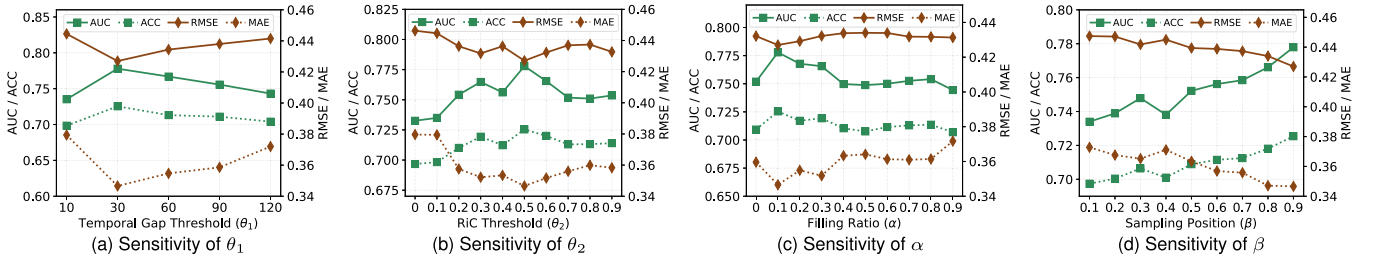


Fig. 4. Hyperparameter sensitivity analysis of “AKT +CLPA” on the Ednet dataset.

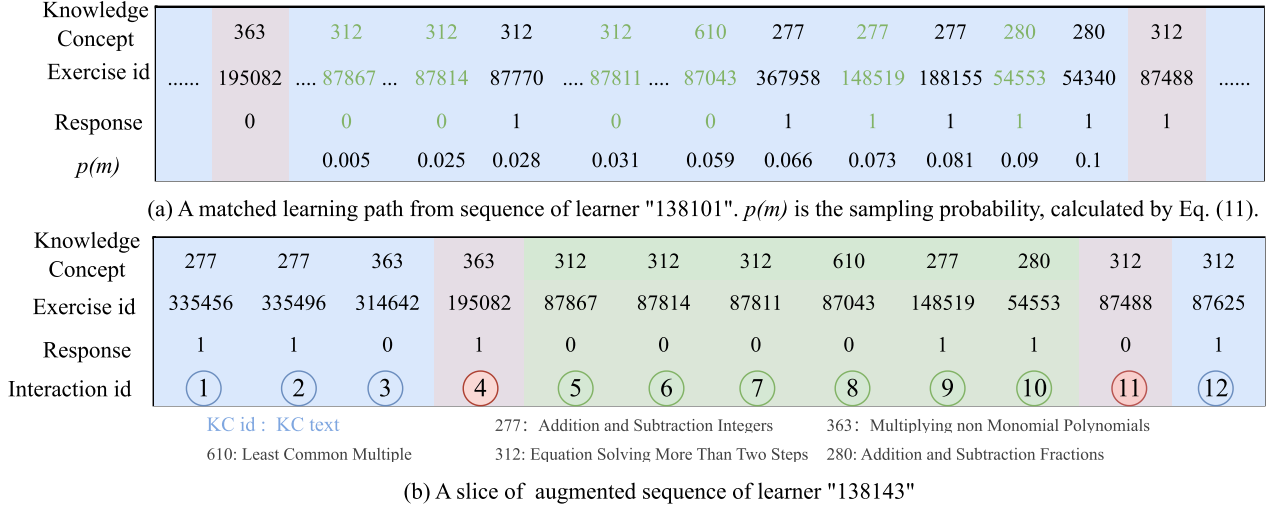


Fig. 5. A slice of a similar sequence and our augmented sequence. The original sequence is marked in blue. The red marks the two endpoints of the extracted learning path. In (b), the pseudo-learning path in green is sampled from (a).

“AKT +CLPA” peaks at $\alpha = 0.1$, followed by subsequent performance degradation. This validates that when extracted paths cannot fully cover temporal gaps, short but highly relevant interactions form better pseudo-paths than excessive, noise-introducing padding. Conversely, performance increases monotonically with the sampling position β (Fig. 4(d)), peaking at $\beta = 0.9$. This validates that interactions adjacent to the breakpoint (i.e., larger β) exhibit stronger semantic relevance to the unobserved target context.

H. Visualization of an Augmented Sequence (RQ5)

To demonstrate that CLPA can enhance sequence continuity, we visualize an augmented sequence from the Assist2012 dataset. In Fig. 5(a), the learning path connecting boundary exercises “195082” and “87488” is extracted from the sequence of learner “138101”, which has a total length of 372. The extracted learning path has a length of 67. Fig. 5(b) shows a segment of the augmented sequence for learner “138143”, where the green interactions represent the pseudo-learning path sampled from learner “138101”. The sequence intersection-over-union (IoU) between learners “138101” and “138143” is 0.3. In the original sequence of learner “138143”, a 4-day temporal gap exists between adjacent interactions “④” and “⑪”. To bridge the temporal gap between interactions “④” and “⑪”, we construct the pseudo-learning path with a length of 6 from Fig. 5(a),

calculated by (9). The pseudo-path for filling is determined by (10) and (11).

From Fig. 5(b), we observe that CLPA effectively resolves the severe four-day temporal gap between interactions ④ and ⑪ in the original sequence of target learner “138143”. In the original sequence, inferring the target state at ⑪ relies solely on distant historical context. By filling this temporal gap with a 6-step proxy path, CLPA introduces highly relevant intermediate interactions, specifically focusing on the target KC “312” and its prerequisites. This dual enhancement of structural continuity and semantic richness provides downstream models with fine-grained, reliable clues to accurately trace the learner’s evolving knowledge state.

I. Comparison of Test Predictions: AKT Trained on Original vs CLPA Augmented Data (RQ6)

To validate whether CLPA improves KT models’ representational capacity and breakpoint performance, we visualize a learner’s prediction sequence from the Assist2012 dataset. Fig. 6 compares predictions from AKT and “AKT +CLPA”, i.e., AKT trained on the original and augmented training data, respectively. The breakpoint “⑥” in the sequence is labeled by a blue box, which has 3.7 days with its previous interaction “⑤”. From Fig. 6, we observe that:

Knowledge Concept	breakpoint											
	580	580	580	75	75	75	75	75	75	7	7	7
Exercise id	341939	341981	341962	342018	342001	341994	342011	342009	342012	341715	341738	341705
Response	1	1	1	0	1	0	1	1	1	1	1	0
Interaction id	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫
Ori+Prediction	1	1	0.947	0.421	0.499	0.78	0.046	0.402	0.944	0.174	0.811	0.551
Original PE	0	0	0.053	0.421	0.501	0.78	0.954	0.598	0.056	0.826	0.189	0.551
Aug+Prediction	0.997	0.997	0.827	0.399	0.725	0.334	0.62	0.574	0.893	0.276	0.881	0.135
Augmented PE	0.003	0.003	0.173	0.399	0.275	0.334	0.38	0.426	0.107	0.724	0.119	0.135

KC id : KC text 580: Exponents 75: Square Root 7: Sampling Techniques

Fig. 6. The predictions on the sequence of learner “176429”. In the figure, the breakpoint is highlighted with a blue box. “Ori+Prediction” and “Aug+Prediction” denote the probability of response correctly predicted by AKT and “AKT+CLPA”, respectively. The corresponding prediction errors, calculated by (1), are labeled as “Original PE” and “Augmented PE”. Incorrect predictions are marked with red boxes.

The AKT exhibits strong sequential dependency in prediction. As illustrated in Fig. 6, for a subsequence involving concept “75”, the original AKT predicts an incorrect response for interaction ⑦” based on the learner’s prior incorrect response to interaction ⑥”. After the learner answers correctly to interactions ⑦” and ⑧”, the original AKT substantially increases the predicted probability of the learner’s correct response to interaction “⑨”. A similar pattern is observed for the subsequence involving concept “7”.

Due to the over-reliance on the recent interactions, the AKT shows significant prediction errors at the breakpoint. As shown in Fig. 6, when predicting the breakpoint “⑥” involving concept “75”, the AKT assigns a high probability of 0.78 only based on the learner’s correct solving interaction “⑤”, ignoring the temporal gap between them. This gap renders the recent interaction “⑤” an unreliable reference for predicting interaction “⑥”, introducing significant prediction bias. It highlights the original model’s limitation in temporal modeling through the dependency mechanisms.

“AKT+CLPA” demonstrates improved representational capacity and robust state transition. As observed in Fig. 6, “AKT+CLPA” achieves accurate predictions before reaching the breakpoint (e.g., interactions “④” and “⑤”). This phenomenon demonstrates that by leveraging imputed collaborative paths, the model enhances its representational capacity, enabling accurate tracking of knowledge state across continuous learning segments. Crucially, when encountering the raw breakpoint “⑥”, “AKT+CLPA” avoids over-reliance on the outdated interaction “⑤” and assigns a highly accurate probability (0.334) aligning with the actual response (0). This validates that CLPA equips the model to learn robust state transitions during training, effectively mitigating the breakpoint effects during inference.

Large-span knowledge concept switch poses significant modeling challenges. In Fig. 6, both AKT and “AKT+CLPA” models struggle with the knowledge concept switch. For example, although the learner had no prior attempts on concept “75”, both models correctly predict interaction “④” due to the strong correlation between concept “75” (square root) and “580” (exponents). However, when facing a larger-span concept switch, e.g., from concept “75” (square root) to concept “7”

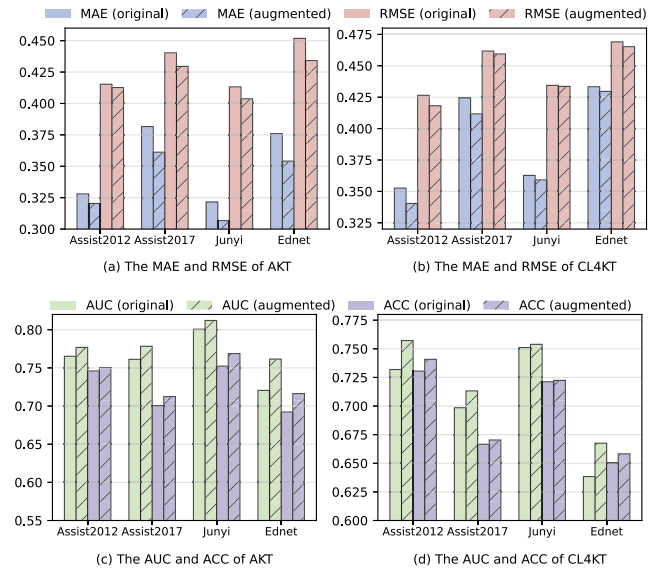


Fig. 7. Performance comparison of base models trained on original versus CLPA-augmented datasets across the Ds_break subsets of Assist2012, Assist2017, Junyi, and Ednet. In the figure, metrics such as MAE (original) and MAE (augmented) denote the performance yielded by the original and augmented training sets, respectively (similarly for RMSE, AUC, and ACC).

(sampling techniques), both models fail to predict interaction “⑩” accurately, indicating limitations in modeling large-span concept switches and the necessity of considering the semantic gap in the future.

J. Performance on Breakpoint-Affected Interactions: Original Vs. CLPA-Augmented Training (RQ7)

To further assess CLPA’s effectiveness on breakpoint-affected interactions, we evaluate the performance of base models trained on original versus CLPA-augmented data, by using Ds_break (the targeted subsets defined in Section III-B) across four datasets. Fig. 7 presents the comparison results on Ds_break from four datasets: Assist2012, Assist2017, Junyi, and Ednet. As shown, CLPA consistently improves performance across all Ds_break . Specifically, base models trained on

TABLE VII
THE EFFICIENCY ANALYSIS OF CLPA

Dataset	Metric	Assist2012	Assist2017	Junyi	Ednet
AKT	Runtime (s/epoch)	10.13	3.3	2.35	4.75
	GPU memory (MB)	1444	1714	1682	1728
AKT +CLPA	Runtime (s/epoch)	14.5	3.6	2.62	4.82
	GPU memory (MB)	2266	1731	1695	1740
CLPA	Total time (s)	650.5	78.0	144.0	352.6

CLPA-augmented data achieve lower MAE and RMSE compared to those trained on the original datasets. For AKT, CLPA reduces MAE by 2.32%, 5.35%, 4.63%, and 5.82% across the four Ds_break , respectively. For CL4KT, the reductions are 3.46%, 2.99%, 0.99%, and 0.85%. AUC and ACC of KT models are also consistently improved with CLPA. The AUC gains for AKT are 1.52%, 2.25%, 1.42%, and 5.69%, and CL4KT shows improvements of 3.31%, 2.1%, 0.37%, and 4.46%. These results demonstrate that our CLPA effectively helps base models to mitigate the breakpoint effect, thereby achieving improved performance on these affected interaction subsets.

K. Analysis of Efficiency and Feasibility of CLPA (RQ8)

To assess the practical feasibility of CLPA, we evaluate its computational overhead across the preprocessing, training, and inference stages. As shown in Table VII, CLPA incurs marginal additional time and GPU memory costs during both the offline data augmentation and the subsequent model training, demonstrating its efficiency and practical feasibility. While the offline preprocessing time naturally scales up on massive datasets like Assist2012 (requiring 650.5 s), this remains a strictly one-time offline cost. A more granular time and space complexity analysis regarding this scalability bottleneck and our mitigation strategies is detailed in the Discussion (Section IV-D). During training, the slight GPU memory overhead for “AKT +CLPA” (Table VII) stems from the enhanced data density after filling, which is easily accommodated by standard GPUs. Furthermore, since CLPA preserves the base model’s architecture, it introduces zero additional latency during inference, validating its suitability for industrial deployment.

L. Discussion

We discuss two critical aspects of the CLPA framework: 1) the retrieval mechanism of our CLPA and 2) matching coverage of our collaborative candidates.

1) *The Retrieval Mechanism of Our CLPA:* In the KT field, the learners’ similar learning trajectories are not affected by absolute timestamps, for example, CoKT [39] retrieves similar question-answering experiences across the entire dataset, disregarding absolute recording time. Following these established retrieval practices, we allow newer collaborative paths to augment older breakpoints, which enriches the candidate path pool. Standard DA methods operate only within the training set and commonly relax absolute chronological constraints in the augmentation process (e.g., [15], [16], [17], [18], [31], [32]). In our experiments, we confine all augmentations to the training

set. This rigorous data isolation leaves the test set excluded from the augmentation process and entirely unmodified, preventing ground-truth exposure.

2) *Matching Coverage of Our Collaborative Candidates:* Our CLPA matches similar learning records for identified breakpoints, achieving success rates of 69%, 98%, 91%, and 68% across Assist2012, Assist2017, Junyi, and Ednet, respectively. Importantly, these matching strategies benefit long-tail learners, as CLPA can integrate active peers’ learning paths into their sequences by aligning boundary exercises. Furthermore, to guarantee augmentation quality, CLPA adopts a conservative imputation strategy, i.e., breakpoints lacking relevant collaborative candidates are left unaugmented. As demonstrated by popularity-based imputation baselines (Section V-E), imputing with high-frequency interactions instead of semantically relevant collaborative contexts degrades performance. Thus, despite incomplete matching coverage, CLPA avoids injecting noise and still yields significant overall performance gains, validating the importance of high-quality collaborative paths. Future work will explore alternative methods to synthesize reliable proxy contexts for unmatchable breakpoints.

VI. CONCLUSION

In this paper, we identify and address the breakpoint effects in knowledge tracing (KT). Through empirical studies, we validate the existence of breakpoint effects, showing that the breakpoints introduce significant prediction errors and degrade the KT models’ performance. To tackle the core issue of state-transition disruptions at breakpoints in KT, we strive to enhance knowledge tracing via breakpoint-aware sequence augmentation and propose a systematic three-phase method, Collaborative Learning Path Augmentation (CLPA). First, identifying boundary position pairs that exhibit the temporal gap and response inconsistency from learners’ sequences. Then, inferring functional pseudo-paths between the identified positions by leveraging cross-sequence collaborative patterns. Finally, reconstructing and inserting the pseudo-learning paths into the target sequences to bridge the temporal gaps and enhance sequence continuity. Furthermore, two refinement strategies for sampling quantity constraints and distance-based sampling strategy are designed to mitigate noise introduction and ensure augmentation quality. The experimental results validate that CLPA enhances the model’s representational capacity and outperforms existing sequential data augmentation methods, achieving accurate knowledge tracking across continuous segments and breakpoints.

For future work, we plan to design an adaptive gap detection mechanism that integrates temporal and semantic signals, coupled with a priority-based augmentation strategy for varying gap severities. We are also interested in establishing evaluation metrics for augmented paths [45] and investigating the framework’s generalizability across diverse KT architectures [46] and broader domains like sequential recommenders.

REFERENCES

- [1] A. T. Corbett and J. R. Anderson, “Knowledge tracing: Modeling the acquisition of procedural knowledge,” *User Model. User-Adapted Interaction*, vol. 4, pp. 253–278, 1994.

- [2] C. Piech et al., "Deep knowledge tracing," in *Proc. Conf. Neural Inf. Process. Syst.*, 2015, pp. 505–513.
- [3] J. Zhang, X. Shi, I. King, and D.-Y. Yeung, "Dynamic key-value memory networks for knowledge tracing," in *Proc. ACM World Wide Web*, 2017, pp. 765–774.
- [4] T. Long, Y. Liu, J. Shen, W. Zhang, and Y. Yu, "Tracing knowledge state with individual cognition and acquisition estimation," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 173–182.
- [5] S. Huang, Z. Liu, X. Zhao, W. Luo, and J. Weng, "Towards robust knowledge tracing models via K-sparse attention," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2023, pp. 2441–2445.
- [6] D. Shin, Y. Shim, H. Yu, S. Lee, B. Kim, and Y. Choi, "SAINT: Integrating temporal features for ednet correctness prediction," in *Proc. Int. Learn. Anal. Knowl. Conf.*, 2021, pp. 490–496.
- [7] A. Ghosh, N. Heffernan, and A. S. Lan, "Context-aware attentive knowledge tracing," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 2330–2339.
- [8] Y. Bai, X. Li, Z. Liu, Y. Huang, M. Tian, and W. Luo, "Rethinking and improving student learning and forgetting processes for attention based knowledge tracing models," in *Proc. AAAI Conf. Artif. Intell.*, 2025, pp. 27822–27830.
- [9] Y. Yin et al., "Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer," in *Proc. ACM Web Conf.*, 2023, pp. 855–864.
- [10] S. Shen et al., "Learning process-consistent knowledge tracing," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 1452–1460.
- [11] S. Pandey and J. Srivastava, "RKT: Relation-aware self-attention for knowledge tracing," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1205–1214.
- [12] G. Abdelrahman and Q. Wang, "Deep graph memory networks for forgetting-robust knowledge tracing," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 7844–7855, Aug. 2023.
- [13] C. Wang et al., "Temporal cross-effects in knowledge tracing," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2021, pp. 517–525.
- [14] J. Cui, Z. Chen, A. Zhou, J. Wang, and W. Zhang, "Fine-grained interaction modeling with multi-relational transformer for knowledge tracing," *ACM Trans. Inf. Syst.*, vol. 41, pp. 1–26, 2023.
- [15] W. Lee, J. Chun, Y. Lee, K. Park, and S. Park, "Contrastive learning for knowledge tracing," in *Proc. ACM Web Conf.*, 2022, pp. 2330–2338.
- [16] J. Xiao, W. Pan, and Z. Ming, "A generic behavior-aware data augmentation framework for sequential recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2024, pp. 1578–1588.
- [17] Y. Dang et al., "Uniform sequence better: Time interval aware data augmentation for sequential recommendation," in *Proc. the AAAI Conf. Artif. Intell.*, 2023, pp. 4225–4232.
- [18] Y. Jiao, F. Yang, Y. Chen, Y. Gao, J. Liu, and Y. Sun, "Rethinking sequential relationships: Improving sequential recommenders with inter-sequence data augmentation," in *Proc. ACM Web Conf.*, 2024, pp. 641–645.
- [19] C. Yeung and D. Yeung, "Addressing two problems in deep knowledge tracing via prediction-consistent regularization," in *Proc. ACM Conf. Learn. Scale*, 2018, pp. 5:1–5:10.
- [20] G. Abdelrahman and Q. Wang, "Knowledge tracing with sequential key-value memory networks," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 175–184.
- [21] F. Ke et al., "HiTSKT: A hierarchical transformer model for session-aware knowledge tracing," *Knowl.-Based Syst.*, vol. 284, 2024, Art. no. 111300.
- [22] S. Shen et al., "Quiz-based knowledge tracing," 2023, *arXiv:2304.02413*.
- [23] K. Nagatani, Q. Zhang, M. Sato, Y.-Y. Chen, F. Chen, and T. Ohkuma, "Augmenting knowledge tracing by considering forgetting behavior," in *Proc. World Wide Web Conf.*, 2019, pp. 3101–3107.
- [24] M. Zhang, X. Zhu, C. Zhang, Y. Ji, F. Pan, and C. Yin, "Multi-factors aware dual-attentional knowledge tracing," in *Proc. Conf. Inf. Knowl. Manage.*, 2021, pp. 2588–2597.
- [25] C. Huang, W. Jiang, K. Li, J. Wu, and J. Zhang, "Enhancing learning process modeling for session-aware knowledge tracing," *Knowl.-Based Syst.*, vol. 309, 2025, Art. no. 112740.
- [26] S. Luo et al., "Integrating large language models into recommendation via mutual augmentation and adaptive aggregation," 2024, *arXiv:2401.13870*.
- [27] K. Zhou et al., "S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *Proc. Conf. Inf. Knowl. Manage.*, 2020, pp. 1893–1902.
- [28] J. Jiang et al., "Improving sequential recommendations via bidirectional temporal data augmentation with pre-training," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 5, pp. 2652–2664, May 2025.
- [29] Z. Liu, Z. Fan, Y. Wang, and P. S. Yu, "Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 1608–1612.
- [30] L. Wang, E.-P. Lim, Z. Liu, and T. Zhao, "Explanation guided contrastive learning for sequential recommendation," in *Proc. Conf. Inf. Knowl. Manage.*, 2022, pp. 2017–2027.
- [31] X. Xie et al., "Contrastive learning for sequential recommendation," in *Proc. IEEE 38th Int. Conf. Data Eng.*, 2022, pp. 1259–1273.
- [32] Y. Dang et al., "Repeated padding for sequential recommendation," in *Proc. 18th ACM Conf. Recommender Syst.*, 2024, pp. 497–506.
- [33] J. Xia, Y. Yang, S. Wang, H. Yin, J. Cao, and P. S. Yu, "Bayes-enhanced multi-view attention networks for robust poi recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 2895–2909, Jul. 2024.
- [34] C. Cao, F. Zhou, Y. Dai, J. Wang, and K. Zhang, "A survey of mix-based data augmentation: Taxonomy, methods, applications, and explainability," *ACM Comput. Surv.*, vol. 57, pp. 37:1–37:38, 2025.
- [35] Y. Dang et al., "Data augmentation for sequential recommendation: A survey," 2024, *arXiv:2409.13545*.
- [36] P. Zhou et al., "Is contrastive learning necessary? A study of data augmentation vs contrastive learning in sequential recommendation," in *Proc. Int. Conf. World Wide Web*, 2024, pp. 3854–3863.
- [37] H. Zhang et al., "Item-difficulty-aware learning path recommendation: From a real walking perspective," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2024, pp. 4167–4178.
- [38] Q. Li et al., "Graph enhanced hierarchical reinforcement learning for goal-oriented learning path recommendation," in *Proc. Conf. Inf. Knowl. Manage.*, 2023, pp. 1318–1327.
- [39] T. Long et al., "Improving knowledge tracing with collaborative information," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2022, pp. 599–607.
- [40] M. Glanzer and A. R. Cunitz, "Two storage mechanisms in free recall," *J. Verbal Learn. Verbal Behav.*, vol. 5, pp. 351–360, 1966.
- [41] A. Petrov and C. Macdonald, "Effective and efficient training for sequential recommendation using recency sampling," in *Proc. 16th ACM Conf. Recommender Syst.*, 2022, pp. 81–91.
- [42] H. Chang, H. Hsu, and K. Chen, "Modeling exercise relationships in E-learning: A unified approach," in *Proc. 8th Int. Conf. Educ. Data Mining*, 2015, pp. 532–535.
- [43] J. Chen, Z. Liu, S. Huang, Q. Liu, and W. Luo, "Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations," in *Proc. 37th AAAI Conf. Artif. Intell.*, 2023, pp. 14196–14204.
- [44] Y. Im, E. Choi, H. Kook, and J. Lee, "Forgetting-aware linear bias for attentive knowledge tracing," in *Proc. Conf. Inf. Knowl. Manage.*, 2023, pp. 3958–3962.
- [45] J. A. Martínez-Carrascal, J. Muñoz-Gama, and T. Sancho-Vinuesa, "Evaluation of recommended learning paths using process mining and log skeletons: Conceptualization and insight into an online mathematics course," *IEEE Trans. Learn. Technol.*, vol. 17, pp. 555–568, 2024.
- [46] M. Yin et al., "Dataset regeneration for sequential recommendation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2024, pp. 3954–3965.



Chunli Huang is currently working toward the PhD degree with Hunan University, supervised by Prof. Wenjun Jiang. Her research interests include behavior analysis and intelligent tutoring systems. She has several publications in top international journals such as ACM Transactions on Internet Technology and Knowledge-based Systems.



Wenjun Jiang (Member, IEEE) received the PhD degree in computer software and theory from Central South University, in 2014. She is a professor and doctoral supervisor with Hunan University. Her research interests include recommendation systems and smart education. She is a senior member of CCF. She has published more than 60 papers, including top international journals and conferences such as *ACM Computing Surveys*, *IEEE Transactions on Computers*, *IEEE Transactions on Parallel and Distributed Systems*, *ACM Transactions on Database Systems*, *WWW*, *INFOCOM*, *AAAI*, and so on.



Ji Zhang (Senior Member, IEEE) is currently a full professor in computer science with the University of Southern Queensland (UniSQ), Australia. He is an RSA life fellow, IET fellow, BCS fellow, IAAST Fellow, Australian Endeavour Fellow, Queensland International Fellow (Australia), and Izaak Walton Killam Scholar (Canada). His research interests include span Big Data analytics, data intelligence, intelligent computing, computational intelligence and data privacy. He has published more than 360 papers in major peer-reviewed international journals and conferences, including *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Knowledge and Data Engineering*, *ACM Transactions on Knowledge Discovery from Data*, *IEEE Transactions on Dependable and Secure Computing*, *IEEE Transactions on Information Forensics and Security*, *IEEE Transactions on Intelligent Transportation Systems*, *AAAI*, *IJCAI*, *VLDB*, *CIKM*, *SIGKDD*, *ICDE*, *ICDM*, *CVPR* and *WWW*.



Kenli Li (Senior Member, IEEE) received the PhD degree in computer science from the Huazhong University of Science and Technology, in 2003. He is a professor and doctoral supervisor with Hunan University. His research interests include high-performance computing, and applications. He has published more than 160 research papers in international conferences and journals such as the *IEEE Transactions on Computers*, *IEEE Transactions on Parallel and Distributed Systems*, and so on. He serves on the editorial board of the *IEEE Transactions on Computers*.



Jie Wu (Fellow, IEEE) received the PhD degree in computer engineering from Florida Atlantic University, in 1989. He is the director of the Center for Networked Computing and a Laura H. Carnell professor with Temple University. His research interests include distributed algorithms, recommendation systems, and other machine learning applications. He has served as an IEEE Computer Society Distinguished Visitor, an ACM Distinguished Speaker, and the Chair for the *IEEE Technical Committee on Distributed Processing*.