

Positive or Negative? Exploring Graph-based Augmentation for Contrastive Learning in Knowledge Tracing

Soobin Ahn
Seoul National University
ahnsb0418@snu.ac.kr

Hyojun Lee
Seoul National University
gywnszz@snu.ac.kr

Shinyoung Lee
Independent Researcher
02.shin.00@gmail.com

Insub Shin
Seoul National University
iss9802@snu.ac.kr

ABSTRACT

Contrastive learning is a machine learning technique that optimizes the embedding space by pulling positive pairs closer and pushing negative pairs farther. Recently, knowledge tracing (KT) models have adopted contrastive learning to learn the representation of students' interaction sequences. Therefore, augmentation methods should preserve students' knowledge states when generating positive pairs and distort them when generating negative pairs. Which means that augmentation methods should take into account the pedagogical semantics beneath students' interaction sequences. However, random augmentations may ignore the educational context and consequently alter the underlying knowledge representations. To address this issue, we propose a graph-based augmentation method that leverages an interaction graph, where each node denotes a skill-response pair and each edge captures the conditional transition probability between interactions. The interaction graph is constructed using transition probabilities predicted by a trained DKT model for the next interaction. Each augmentation method generates learning paths by following transition probabilities in the interaction graph, enabling the construction of structurally plausible sequences. This design preserves pedagogically meaningful transitions within augmented sequences. Furthermore, we explored whether pairs generated by random augmentation methods and our graph-based augmentation methods function as positive or negative pairs. Our results show that graph-based augmentation can effectively generate sufficient and meaningful pairs. However, we find that knowledge state preservation differs between the human perspective and the model's perspective. Future work is needed to further explore and understand this difference.

Keywords

Data Augmentation, Contrastive Learning, Knowledge Tracing

Soobin Ahn, Shinyoung Lee, Lee Hyojun, and Shin Insub. Positive or Negative? Exploring Graph-based Augmentation for Contrastive Learning in Knowledge Tracing. In Anthony Botelho, Maria Mercedes T. Rodrigo, Adish Singla, Hiroaki Ogata, Hyojeong So, and Young Hoan Cho (eds.) Proceedings of the 19th International Conference on Educational Data Mining, Seoul, Republic of Korea, June, 2026, pp. 743–747. International Educational Data Mining Society (2026).

© 2026 Copyright is held by the author(s). This work is distributed under the Creative Commons Attribution NonCommercial NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license.
<https://doi.org/10.5281/zenodo.21040032>

ing

1. INTRODUCTION

In recent years, Knowledge Tracing (KT) has been actively studied as a fundamental task in Intelligent Tutoring Systems (ITS). In order to capture the current knowledge state within an ITS, KT estimates the probability of a correct response by taking the student's historical learning path and a target skill as inputs. The estimated knowledge state supports personalized instruction and adaptive learning systems.

To better capture the dynamics of knowledge states, various deep learning approaches have been proposed for KT. Among these, contrastive learning has gained attention as a representation learning framework. It trains models by pulling together semantically similar (positive) pairs while pushing apart semantically dissimilar (negative) pairs. Several studies have adopted contrastive learning in KT by applying data augmentation to construct these pairs from student interaction sequences. For instance, CL4KT[3] was proposed to improve representation learning under sparse interaction data by applying random augmentations to construct positive and negative pairs, encouraging semantically similar learning histories to have closer representations through a contrastive objective. This approach demonstrated the effectiveness of contrastive learning in enhancing knowledge state representations and improving KT performance.

However, in contrastive learning for KT, it remains unclear whether commonly used augmentation methods function as valid positive or negative pairs. Recent work such as QCKT[8] has highlighted that the criteria for selecting positive and negative samples remain insufficiently defined. In KT, semantic similarity reflects the consistency of the students' latent knowledge state. Therefore, in this work, positive pairs are defined as sequence pairs that preserve the students' knowledge state, and negative pairs as those that distort it. Based on this definition, we investigate whether each augmentation method acts as a positive or negative pair, based on its effect on the students' knowledge state.

Despite these successes, we identify several limitations of existing augmentation approaches. First, current models fo-

cus on sequence-level augmentations.[9, 1] However, these methods do not fully capture the structured learning path that emerges from the underlying connections between skills. Second, random augmentations may overlook the educational context of the data and risk distorting the underlying logic of knowledge acquisition.[5] For example, if a prerequisite question is randomly replaced or removed, it creates an invalid sequence that contradicts the actual logic of a students’ knowledge acquisition. Such augmentations may lead to sub-optimal contrastive objectives by providing the model with semantically invalid positive pairs.

To address this limitation, we propose graph-based augmentation methods that leverage an interaction graph over student interactions. Rather than applying random perturbations, our approach generates augmented sequences by following transition-weighted paths in the graph, enabling structure-aware augmentations that reflect meaningful pedagogical transitions.

To systematically analyze the role of augmentation methods in contrastive learning for KT, we evaluate both existing random augmentations and the proposed graph-based augmentations in terms of their suitability as positive or negative pairs. Specifically, we assess their suitability based on AUC gain, where higher performance indicates better alignment with the assigned role.

The following research questions are investigated in this study:

RQ1. How do random augmentation methods behave when assigned as positive or negative pairs in contrastive learning for KT?

RQ2. How do graph-based augmentation methods behave when assigned as positive or negative pairs in contrastive learning for KT?

RQ3. How does the performance of the full random augmentation setting compare to that of the full graph-based augmentation setting?

2. METHOD

2.1 Graph-based Augmentation

2.1.1 Definition of interaction graph

Graph-based Knowledge Tracing (GKT) [6] has attempted to capture structured skill relationships and educational context by modeling relationships between skills as graph structures. However, existing studies focus primarily on skill-to-skill correlations.[2, 4, 10] Consequently, there is still a lack of research that utilizes interaction nodes to fully capture the directionality of a students’ knowledge state. To address this gap, we employed DKT graph. Given $|S|$ skills, the graph contains $N = 2|S|$ interaction nodes, and each interaction (s, r) with $r \in \{0, 1\}$ is mapped to a node index $v = s + |S| \cdot r$. We define a weighted interaction graph $G = (V, E)$ and its corresponding adjacency matrix $A_{ij} \in \mathbb{R}^{N \times N}$. To capture various pedagogical aspects, we construct an adjacency matrix based on a DKT graph. The DKT graph is a model-based interaction graph constructed from the conditional prediction probabilities of a trained DKT model [7]. Given a student interaction (s_t, r_t) at time step t , we estimate the probability that a student answers a target skill s_{t+1} correctly at the next step. Specifically, the trained DKT model

provides

$$p_1 = \Pr(r_{t+1} = 1 \mid s_t, r_t, s_{t+1})$$

which represents the likelihood of correct response on skill s_{t+1} given interaction (s_t, r_t) . The probability of an incorrect response is defined as

$$p_0 = \Pr(r_{t+1} = 0 \mid s_t, r_t, s_{t+1}) = 1 - p_1.$$

Based on these predictions, response r_t is determined as 1 if $p_1 \geq p_0$ and 0 otherwise. To ensure reliable transitions, a threshold τ is defined and only transitions satisfying $\Delta = |p_1 - p_0| > \tau$ are retained, with τ set to 0.5 in practice to preserve only transitions with sufficiently meaningful differences. To determine the edge weights, we define $w = \max(p_1, p_0)$, selecting the higher probability between a correct and an incorrect response at the next step. We then retain the top- k transitions with highest w values for each node, defining the weighted adjacency matrix as $A_{ij} = w$.

2.1.2 Graph-based augmentation method

Using the DKT graph defined above, we propose six graph-based augmentation methods. All augmentations leverage the weighted adjacency matrix A to ensure that generated sequences reflect meaningful transitions between interaction nodes.

Graph-based Masking Masking candidates are restricted to interactions that have at least one neighbor within the same sequence with a transition weight above threshold τ . Only these eligible interactions are subject to masking, ensuring that masked positions remain inferable from their structural context, thereby preventing excessive distortion of the students’ knowledge state.

Graph-based Crop Transition weights between consecutive interaction pairs are computed using the adjacency matrix A , reflecting how strongly each pair of interactions is connected. Based on these weights, the position with the weakest connection (i.e., the minimum transition weight) is identified and used as the boundary for cropping. The subsequence closer to the target crop length is then retained, ensuring that the remaining segment consists of strongly connected interactions, thereby preserving a consistent and meaningful learning context.

Graph-based Extension For each interaction (s, r) , a weighted candidate set is constructed by sampling neighboring interactions (s', r') from the row-normalized adjacency matrix A , excluding same-skill nodes. Using this candidate set, two augmented views \mathbf{x}^{+1} and \mathbf{x}^{+2} are generated via prior and posterior extension, respectively. In prior extension, (s', r') is appended before the head of the sequence; in posterior extension, it is appended after the tail. If the sequence exceeds the maximum length, the opposite end is truncated accordingly.

Graph-based Branching Path Sequence pairs are constructed to share the same initial interactions and diverge from a branching point. From this point, the remaining interactions are generated by traversing the graph, sampling from the top- k neighbors based on transition strength and proximity to a target endpoint. The final interaction is constrained to match the original skill but with a flipped response, ensuring

divergence in the terminal knowledge state while preserving overall structural similarity.

Graph-based Path Substitution The first and last interactions of a sequence are preserved, while an intermediate segment is replaced with an alternative path generated from the interaction graph. Specifically, the intermediate segment is selected and replaced by traversing the adjacency matrix A to construct a new path between the two fixed anchor points. During this process, interactions from the original segment are penalized to encourage the generation of diverse alternative paths, while the initial and final interactions remain consistent.

2.2 Experiment Settings

2.2.1 Benchmark Datasets

To evaluate the performance of the proposed method, we utilized the ASSISTment 2009 dataset. To ensure data integrity, we applied several preprocessing steps before model training. Specifically, users with fewer than five interactions were excluded, and interactions without explicit concept tags were removed to maintain clear concept mapping.

ASSISTments (2009). These datasets were obtained from an online mathematics tutoring platform. Specifically, we used the ASSISTment 2009-2010 "Skill Builder" set version.

Table 1: Statistics of the Datasets

Dataset	Users	Items	Skills	Interactions	Avg. Seq Length
ASSISTments 09	3,695	17,728	112	282,071	76.34

2.2.2 Experimental design

To investigate whether each augmentation method is more suitable as a positive or negative pair, each augmentation is evaluated under two distinct roles within the InfoNCE loss. When evaluated as a positive, the augmentation is applied to both views \mathbf{x}^{+1} and \mathbf{x}^{+2} with masking. When evaluated as a negative, the augmentation is used to generate the hard negative z_k , while both positive views consist of masking only. We use masking alone as the base positive, as it is considered the most conservative augmentation that is least likely to distort the students' knowledge state. We evaluated two augmentation methods: random augmentations adopted from CL4KT [3], including crop, replace, and permute; and our proposed graph-based augmentations, including prior extension, posterior extension, graph-based crop, graph-based branching path, and path substitution. Each augmentation is tested in both roles, and performance is compared to determine whether it is more appropriate as a positive or negative pair. We also include two additional baselines: No CL, where the model is trained without any contrastive objective, and Full, which represents the best configuration for each augmentation type.

2.2.3 Implementation details

The embedding dimension and hidden dimension were fixed to $d = 64$ for all experiments. Student interaction sequences

were sorted chronologically for each user. To standardize sequence lengths, we retained up to the most recent 50 interactions per student; longer sequences were truncated, while shorter ones were zero-padded. All models were evaluated using 5-fold cross-validation, with the Area Under the ROC Curve (AUC) serving as the primary metric. All experiments share masking probability of $r_{\text{mask}} = 0.1$, and the target augmentation method in each condition is applied with a transformation ratio of 0.2. When evaluated as a negative, r_{neg} is set to 1.0; otherwise, it is set to 0. The Full configuration applies both $r_{\text{mask}} = 0.1$ and $r_{\text{crop}} = 0.2$ as positive views, with $r_{\text{neg}} = 1.0$ for hard negative generation. All models were trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 64. We applied a learning rate scheduler that decayed the learning rate by a factor of 0.5 when the validation AUC did not improve for five consecutive epochs, and early stopping was applied if no improvement was observed for 5 consecutive epochs.

3. RESULTS

3.1 RQ 1. Effect of Random Augmentation on Positive and Negative Pairs

Table 2 presents the performance of random augmentation methods when used as either positive or negative pairs. As shown in the table, random augmentations generally achieve higher performance when used as positive pairs than as negative pairs. In particular, replace shows a noticeable performance drop when used as a negative pair, while crop and permute also exhibit a moderate gap. Based on these results, crop, replace, and permute tend to be more suitable when used as positive pairs. From a knowledge state perspective, this provides partial evidence that these random augmentations may better maintain consistency with the original learning sequence when used as positive pairs. These findings highlight that the behavior of random augmentations depends on how they are incorporated within the contrastive learning framework, and that their role assignment can influence their effectiveness.

Table 2: Effect of Random Augmentation on Positive and Negative Pairs

	Positive	Hard negative	Δ
Crop	0.74135	0.73906	+0.00229
Replace	0.73179	0.71676	+0.01503
Permute	0.72977	0.72717	+0.0026
Full random	0.72487		

3.2 RQ 2. Effect of Graph-based Augmentation on Positive and Negative Pairs

Table 3 shows the performance of each graph-based augmentation as a positive or negative pair. Overall, graph crop, branching path, and path substitution tend to perform better as positive pairs, suggesting that these methods are more aligned with the role of positive pairs from a knowledge state perspective. In contrast, prior and posterior extension show slightly better performance as negative pairs, although the differences compared to their positive counterparts are not substantial. For augmentation methods that do not exhibit a clear performance gap between roles, it is less straightforward to determine their appropriate assignment within the

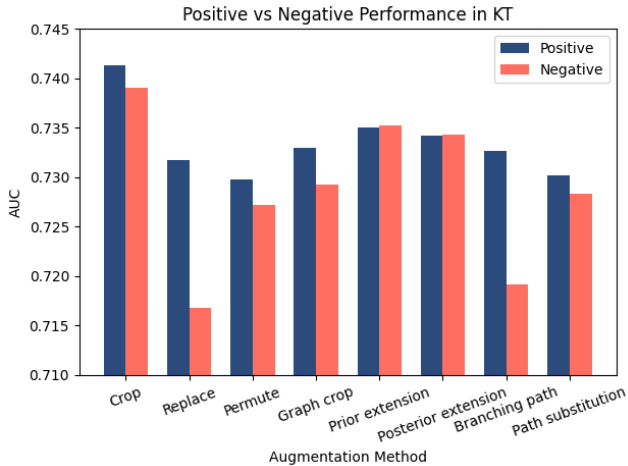


Figure 1: Comparison of AUC performance for each augmentation method under positive and negative pair assignments in knowledge tracing.

contrastive framework. These methods can be interpreted as producing pairs with only minor semantic differences, rather than clearly positive or strongly negative ones. Additionally, in the case of branching path, although it was originally designed as a hard negative by transitioning from the original sequence to a node with a different response, it shows one of the most pronounced performance differences when used as a positive pair. This observation suggests that its behavior is not fully aligned with its intended role, requires further discussion regarding its underlying cause. The results of RQ1 and RQ2 are further illustrated in Figure 1, where the differences between augmentation methods across roles can be clearly compared.

Table 3: Effect of Graph-based Augmentation on Positive and Negative Pairs

	Positive	Hard negative	Δ
Graph crop	0.73299	0.72922	+0.00377
Prior ext	0.73502	0.73526	-0.00024
Posterior ext	0.73418	0.73428	-0.00010
Branching path	0.73271	0.71912	+0.01359
Path subst	0.73020	0.72834	+0.00186
Full graph	0.73477		

3.3 RQ 3. Comparison of Full Random and Full Graph-based Augmentation

We evaluate the Full configuration for both random and graph-based augmentation settings. In the random setting, all augmentation strategies—masking, crop, permute, and replace—are employed to construct positive views, while negative samples are drawn from the interaction histories of other students within the same mini-batch. For the graph-based setting, the Full configuration is constructed based on the observed performance of each method, where graph mask, graph crop, branching path, and path substitution are assigned as positive pairs, and prior and posterior ex-

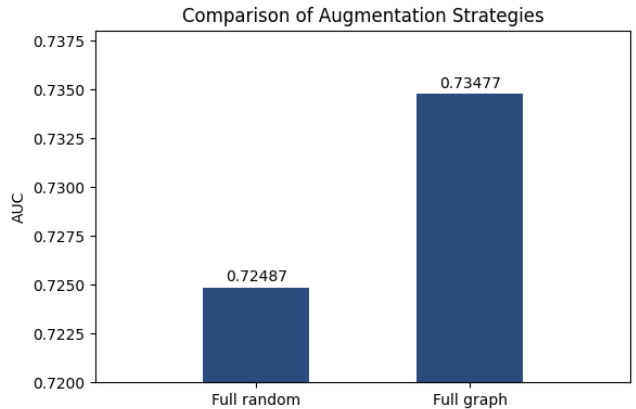


Figure 2: Performance Comparison between Full Random and Graph-based Augmentation Settings

tension are used as hard negatives. As shown in Figure 2, the graph-based Full configuration achieves the best overall performance (AUC = 0.73477), outperforming the random augmentation-based Full configuration (AUC = 0.72487). This comparison highlights the effectiveness of graph-based augmentation in the full setting. However, it is also observed that the AUC obtained from the Full configurations is lower than the best performance achieved by individual augmentation methods (e.g., random crop and prior extension). This suggests that simply combining multiple augmentation strategies does not necessarily lead to further performance gains, and that further consideration is needed on how to effectively combine them.

4. CONCLUSION

In this work, we investigated graph-based augmentations from the perspective of knowledge state preservation and distortion, and compared them with commonly used random augmentations. To construct structure-aware augmentations, we build an interaction graph using a trained DKT model, where edges are defined by conditional prediction probabilities. This results in a weighted graph that reflects meaningful learning paths, based on which we generate augmented sequences along valid transitions. To evaluate their effectiveness, we analyze how each augmentation method contributes to contrastive learning when assigned as either a positive or negative pair, by examining performance differences across roles. The results show that random augmentations are generally more suitable for positive pairs, while in the graph-based setting, most augmentation methods, except for extension-based approaches, show similar tendencies. In particular, prior and posterior extension exhibit only marginal differences between roles, indicating low sensitivity to role assignment. In contrast, the proposed branching path method, originally designed as a hard negative, demonstrates a clear AUC gain when used as a positive pair. This suggests that its empirical behavior does not fully align with its intended role. Additionally, in this study, the semantic role of each augmentation method was discussed only through downstream AUC performance, which provides an indirect perspective on whether an augmented learning path is truly semantically equivalent to the original sequence. Future work should therefore develop quantitative measures for

semantic equivalence and revisit whether knowledge state preservation and distortion from a human perspective are aligned with those learned by contrastive learning models. Furthermore, it remains unclear how augmentation methods with minor performance differences across roles should be incorporated into the contrastive learning framework. These methods may instead be better suited for data augmentation or regularization, and we suggest this as a direction for future work.

5. REFERENCES

- [1] H. Dai, Y. Yun, Y. Zhang, R. An, W. Zhang, and X. Shang. Self-paced contrastive learning for knowledge tracing. *Neurocomputing*, 609:128366, 2024.
- [2] W. Han, W. Lin, L. Hu, Z. Dai, Y. Zhou, M. Li, Z. Liu, C. Yao, and J. Chen. Contrastive cross-course knowledge tracing via concept graph guided knowledge transfer. *arXiv preprint arXiv:2505.13489*, 2025.
- [3] W. Lee, J. Chun, Y. Lee, K. Park, and S. Park. Contrastive learning for knowledge tracing. In *Proceedings of the ACM web conference 2022*, pages 2330–2338, 2022.
- [4] Y. Liu, Q. Liu, Y. Tian, C. Wang, Y. Niu, Y. Song, and C. Li. Concept-aware denoising graph neural network for micro-video recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 1099–1108, 2021.
- [5] S. Mao, G. Li, and Y. Jiang. Boosting knowledge tracing with structure-preserving adaptive contrastive learning. *IEEE Transactions on Consumer Electronics*, 2025.
- [6] H. Nakagawa, Y. Iwasawa, and Y. Matsuo. Graph-based knowledge tracing: modeling student proficiency using graph neural network. In *IEEE/WIC/aCM international conference on web intelligence*, pages 156–163, 2019.
- [7] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. *Advances in neural information processing systems*, 28, 2015.
- [8] X. Shen, F. Yu, Y. Liu, R. Liang, Q. Wan, T. Yang, M. Shi, and J. Sun. Enhancing knowledge tracing with question-based contrastive learning. *Knowledge-Based Systems*, 325:113899, 2025.
- [9] X. Shen, F. Yu, Q. Wan, R. Liang, and J. Sun. Multi-level contrastive learning for knowledge tracing. *ACM Transactions on Knowledge Discovery from Data*, 19(8):1–29, 2025.
- [10] X. Song, J. Li, Y. Tang, T. Zhao, Y. Chen, and Z. Guan. Jkt: A joint graph convolutional network based deep knowledge tracing. *Information Sciences*, 580:510–523, 2021.