

3DG: A Framework for Using Generative AI for Handling Sparse Learner Performance Data From Intelligent Tutoring Systems*

Liang Zhang^{1,2}, Jionghao Lin^{3,*}, Conrad Borchers³, Meng Cao⁴ and Xiangen Hu^{1,2,4}

¹*Institute for Intelligent Systems, University of Memphis, Memphis, TN 38152, USA*

²*Department of Electrical and Computer Engineering, University of Memphis, Memphis, TN 38152, USA*

³*Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, 15213, USA*

⁴*Department of Psychology, University of Memphis, Memphis, TN, 38152, USA*

Abstract

Learning performance data (e.g., quiz scores and attempts) is significant for understanding learner engagement and knowledge mastery level. However, the learning performance data collected from Intelligent Tutoring Systems (ITSs) often suffers from sparsity, impacting the accuracy of learner modeling and knowledge assessments. To address this, we introduce the 3DG framework (3-Dimensional tensor for DenSification and Generation), a novel approach combining tensor factorization with advanced generative models, including Generative Adversarial Network (GAN) and Generative Pre-trained Transformer (GPT), for enhanced data imputation and augmentation. The framework operates by first representing the data as a three-dimensional tensor, capturing dimensions of learners, questions, and attempts. It then densifies the data through tensor factorization and augments it using Generative AI models, tailored to individual learning patterns identified via clustering. Applied to data from an AutoTutor lesson by the Center for the Study of Adult Literacy (CSAL), the 3DG framework effectively generated scalable, personalized simulations of learning performance. Comparative analysis revealed GAN's superior reliability over GPT-4 in this context, underscoring its potential in addressing data sparsity challenges in ITSs and contributing to the advancement of personalized educational technology.

Keywords

Learning Performance Data, Data Sparsity, Intelligent Tutoring System, Generative Model, Generative Adversarial Network, Generative Pre-trained Transformer

1. Introduction

Intelligent Tutoring System (ITS) is a prototype of computer system designed to offer personalized and adaptive instructions through tracing and analyzing learning performance data such as quiz scores and question attempts [1, 2, 3, 4]. However, during the interaction between learners and ITS, learning performance data often exhibits data sparsity due to unexplored questions, insufficient attempts to master knowledge, and lacking variability in learning patterns [5, 6, 7, 8, 9].

Joint Proceedings of LAK 2024 Workshops, co-located with 14th International Conference on Learning Analytics and Knowledge (LAK 2024), Kyoto, Japan, March 18-22, 2024.

*Corresponding author.

✉ lzhang13@memphis.edu (L. Zhang); jionghao@cmu.edu (J. Lin); cborcher@cs.cmu.edu (C. Borchers); mcao@memphis.edu (M. Cao); xhu@memphis.edu (X. Hu)

🆔 0009-0002-0017-2569 (L. Zhang); 0000-0003-3320-3907 (J. Lin); 0000-0003-3437-8979 (C. Borchers); 0000-0002-1286-2885 (M. Cao); 0000-0001-9045-4070 (X. Hu)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Data sparsity can lead to biased analysis and modeling of learning data. This is particularly evident in the “Learner Model” component of ITS, which is crucial for tracking learning and predicting performance of individual learners [10, 11, 12]. Specifically, sparse performance data can lead to skewed or overfitted Knowledge Tracing models in “Learner Model”, which impedes accurately capturing learner knowledge states and may result in misleading predictions of learning performance [6, 9, 13, 14]. The scarcity of learning performance data significantly hampers the development of ITSs, particularly in cases where learners have not sufficiently engaged with certain instructional scenarios [15, 16].

Tackling data sparsity for ITSs presents a practical yet challenging research area. Informed by the machine learning literature [17, 18, 19, 20], the issue of data sparsity can be addressed by two principal ways: *data imputation* and *data augmentation*. Firstly, data imputation focuses on filling the gaps in missing data to ensure a comprehensive dataset [6, 8, 21]. Secondly, data augmentation aims to enrich and expand datasets where there are insufficient learning patterns, thus ensuring robustness in analysis, modeling, and even potential testing tasks for ITSs [9, 22]. Currently, limited efforts have been made in the field of ITSs to systematically address these data sparsity issues in learning performance data [5, 6, 21]. Driven by this, we propose the 3DG (3-dimensions, **D**ensification, and **G**eneration) simulation framework, a systematic approach leveraging generative models to handle sparse learning performance from ITS.

The 3DG framework derived from its three core phases. In the *first phase*, a 3-dimensional tensor is constructed to represent learning performance data, with dimensions corresponding to learners, questions, and attempts. The *second phase* focuses on densifying the sparse tensor by tensor factorization. The *third phase* entails the generation of learning performance data based on generative models, tailored to the individual learning patterns of learners. The 3DG framework integrates the multidimensional learner model with generative models to facilitate scalable simulation sampling for individual learning patterns. The multidimensional learner model in our framework is derived from the Tensor Factorization method, a widely-used approach in predicting learner performance in many studies [21, 23, 24, 25]. Initially, learning performance values are represented in a three-dimensional tensor encompassing dimensions of learners, questions, and attempts. Specifically, learning performance indicators, such as binary responses from learners at problem-solving step attempts (with correct answers denoted as 1 and incorrect as 0), form the tensor entries, and they are arranged sequentially along the question queue in the learning process and sorted by attempts in ascending order. This constructed tensor exhibits data sparsity. Our study aims to perform data imputation and augmentation on the sparse tensor. Mathematically, the tensor factorization method addresses incomplete and missing performance values in factorization computations, serving as a form of tensor completion typically used in data imputation [21, 26, 27]). Inspired by the recent advancements of generative models [28, 29], which are capable of generating data based on patterns learned during training and have revolutionized simulation methodologies to be more flexible and cost-effective, our study delves into exploring their potential of addressing the data sparsity issue. We operate under the foundational assumption that, if learning patterns can be identified within the multidimensional learner model, they can be effectively simulated and generated using generative models, facilitating scalable data augmentation. Consequently, current research was guided by following two **Research Questions**:

- **RQ 1:** What is the most effective method for integrating tensor factorization and generative models to develop a systematic framework that proficiently imputes and augments sparse learning performance data?
- **RQ 2:** In the context of simulating learning performance data, how do Generative Adversarial Network (GAN) and Generative Pre-trained Transformer (GPT) models compare in terms of effectiveness and accuracy?

2. Methods

2.1. Dataset

Our study investigated a dataset derived from the AutoTutor ITS, focusing on learning performance in reading comprehension. This dataset originates from lessons developed for the Center for the Study of Adult Literacy (CSAL) [30, 31], specifically the 'Cause and Effect' lesson, involving 118 participants. The lesson design incorporates three levels of question difficulty: medium (M), easy (E), and hard (H). There are 9 medium-difficulty questions, 10 easy questions, and 10 hard questions. Notably, the distribution of learners across these difficulty levels varies within the lesson. Upon completing the medium difficulty level, learners are either advanced to the hard level or redirected to the easy level, depending on their performance, thus providing a tailored learning pathway.

2.2. The Systematic Simulation Framework

We propose a systematic simulation framework, 3DG, illustrated in Figure 1. This framework begins by structuring the initial learning performance data, sourcing from real-world learner-ITS interactions, into a three-dimensional tensor by dimensions of learners, questions, and attempts. As depicted in the sparse cube space in Figure 1, filled cubes represent recorded values of learning performance, while transparent cubes indicate missing values. Tensor completion

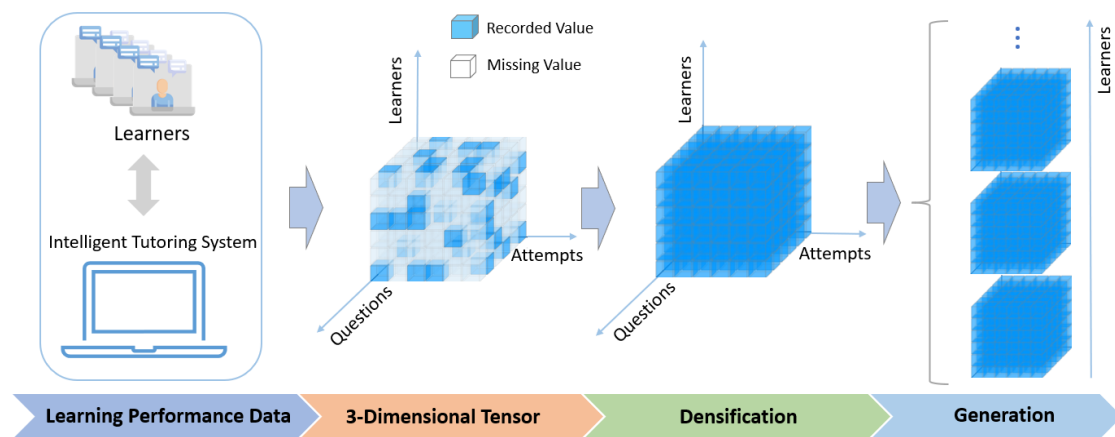


Figure 1: The 3-dimensions, Densification, and Generation (3DG) systematic simulation framework.

(based on tensor factorization) is then utilized, converting the sparse tensor to a densified one. The densified tensor provides invaluable information in identifying various learning patterns, which aids in dividing the tensor into sub-tensors by categorizing distinct learning patterns. Subsequently, generative models are harnessed to simulate additional data samples for enriching the original dataset based on each specific learning pattern. The entire operation is encapsulated for scalable simulation sampling and ultimately offers a comprehensive dataset incorporating both imputed and augmented data. This framework was developed to address **RQ 1**. More detailed methods within this framework are described in the following subsections.

2.3. Tensor Completion for Data Imputation

The three-dimensional tensor \mathcal{T} , representing the learning process, is defined as $\mathcal{T} \in R^{U \times N \times M}$, where the $U = \max(1, 2, 3, \dots, u)$ is the maximum number of learners, $N = \max(1, 2, 3, \dots, n)$ the maximum number of questions, and $M = \max(1, 2, 3, \dots, m)$ the maximum number of attempts. Each element τ_{uij} of \mathcal{T} indicates the performance variable of learner l_u on question q_i at the attempt a_j . For instance, in the CSAL AutoTutor context, a binary variable $\tau_{uij} = \{0, 1\}$ is used, where 1 signifies a correct answer and 0 denotes an incorrect one. We model the tensor \mathcal{T} as a factorization of two lower dimensional components: 1) a learner latent matrix \mathcal{U} of size $U \times K$ (K represents the set of latent features in tensor factorization), which captures learner-related latent features matrix/space (such as abilities and learning-related features); and 2) a latent tensor \mathcal{V} of size $K \times M \times N$, representing the learner knowledge in terms of latent features during question attempts. The approximated tensor $\hat{\mathcal{T}}$ is obtained by the following formula:

$$\hat{\mathcal{T}} \approx \mathcal{U} \times \mathcal{V} \quad (1)$$

where \mathcal{U} can be interpreted as the latent feature space encapsulating learner-related effects, reflecting characteristics such as individual abilities/features and learning preferences. On the other hand, the tensor \mathcal{V} represents the interaction between attempts and question-related (knowledge acquisition) effects, adapting to various learner features.

2.4. Scalable Simulation based on Generative Models for Data Augmentation

To answer **RQ 2**, we used two generative models, GAN (Generative Adversarial Network) and GPT (Generative Pre-trained Transformer), to facilitate scalable simulations that are tailored to individual learning patterns. According to [32, 33], GAN model is uniquely structured with a dual-network architecture comprising a generator and a discriminator. This architecture enables GAN model to excel in generating high-quality synthetic data. In comparison, GPT model is distinguished by its use of transformer architecture, which empowers it to generate data that is not only contextually relevant but also maintains a high degree of coherence [34, 35].

Before initiating the simulations, we employ a clustering algorithm (i.e., K-means++) to categorize individual learning patterns based on similarities in learners' performance. The learners-attempts matrix slice extracted from the $\hat{\mathcal{T}}$, encapsulates the probability-based knowledge states associated with the performance on the n th question q_n , for all U learners over M attempts. In our analysis, we employ the "power law learning curve", a model widely recognized in educational and training research [36, 37, 38], to fit the learning performance with increasing

attempts. In the power-law formula $Y = aX^b$, the Y represents the learning performance, quantified as the probability of producing correct answers, and X is the number of opportunities to practice a skill or attempt. The parameter a indicates the measurement of the learner's initial ability or prior knowledge, and b represents the learning rate at which the learner acquires knowledge through practice. We employ K-means++ [39, 40] to cluster the distribution of two model parameters (a and b), which assists in identifying distinct individual learning patterns.

As illustrated in Figure 2, the architecture of Generative Adversarial Network (GAN) consists of two distinct neural networks: the Generator and the Discriminator. The Generator, often a type of neural network like a convolutional neural network (CNN), is designed to create synthetic data samples. It is denoted as $G(\cdot)$. The Discriminator, typically another neural network which can also be a CNN (though its structure may vary based on the specific application), is tasked with evaluating whether the data samples are real (authentic data) or fabricated by the Generator. It is denoted as $D(\cdot)$. In the process (Figure 2), the Generator starts with a noise sample, usually drawn from a Gaussian distribution, which has dimensions compatible with the original data distribution. This noise sample serves as the initial input for the Generator, resulting in *Simulated Data*. Both this *Simulated Data* and the *Real Data* are then fed into the Discriminator $D(\cdot)$. The Discriminator's role is to discern whether each sample is real or simulated. Concurrently, the Generator $G(\cdot)$ is trained to progressively reduce the difference between the distributions of the real and simulated data through iterative tuning.

Considering the limitations of using purely numerical values for interoperability and the enhanced semantic understanding that detailed descriptions provide, we have developed a mixed-based prompt approach for GPT-4 (illustrated in Figure 3). The prompt strategy integrates original matrix data with interpretive text, thereby enriching the context and interpretability of the data. Additionally, it incorporates the Chain-of-Thought (CoT) prompting technique [41], which involves appending guiding phrases such as 'Let's think step by step' at the end of the prompt to facilitate a more structured analytical process. Specifically, the constructed prompt includes comprehensive elements such as the reading material being analyzed, detailed information about the questions (including their answers), and the learners-attempts matrix data, complete with descriptive information about both its format and entries. Subsequently, a simulation request prompts GPT-4 to integrate the numerical and textual data in a coherent and

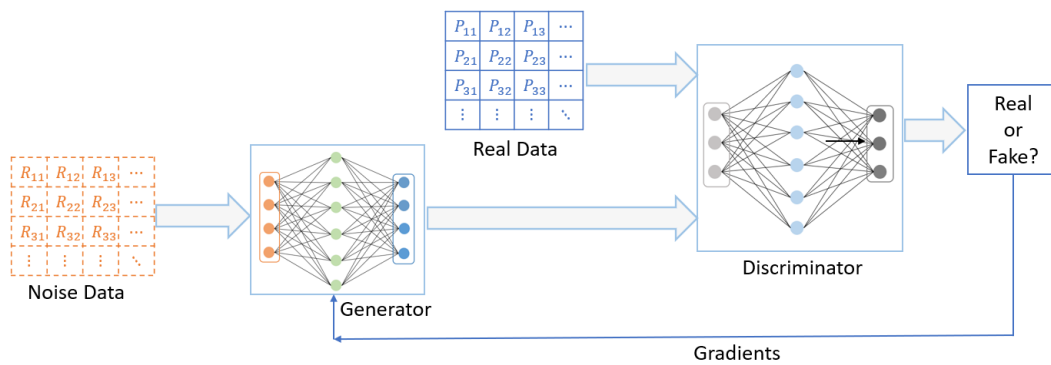


Figure 2: Diagram of using generative adversarial network (GAN) model for data simulation.

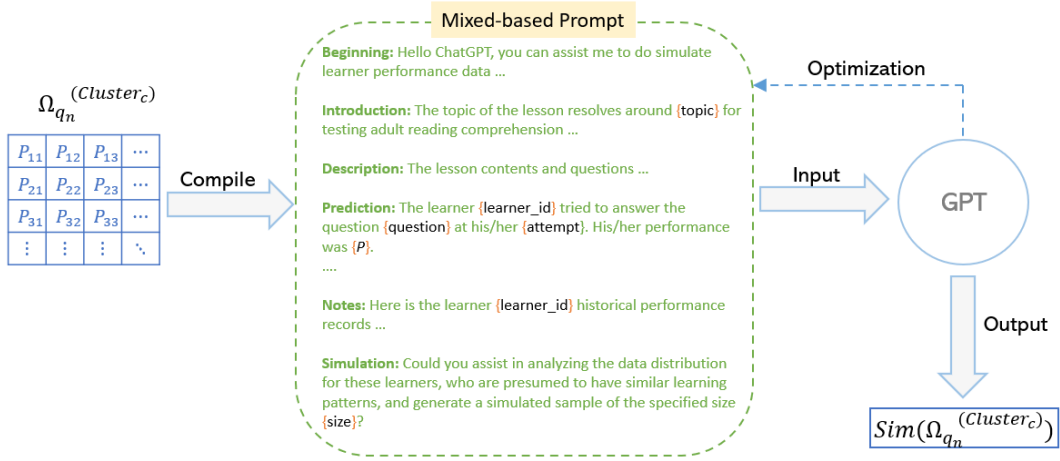


Figure 3: Diagram of using generative pre-trained transformer (GPT) model for data simulation.

insightful manner, ultimately driving the execution of a simulation. During the optimization process, these prompts are iteratively refined and adjusted to efficiently yield results that align with our specified objectives.

3. Results

As illustrated in Table 1, the original dataset exhibits sparsity levels ranging from 80% to 85% (as determined by calculating the proportion of missing values to the total number of entries). By iteratively tuning the latent feature range [1, 20] in tensor factorization algorithms, we identified the optimal number of latent features (K) as 6 for both Lesson (M) and Lesson (H), and 4 for Lesson (E). The optimal K value was derived by averaging results from multiple trainings with optimized K values in tensor factorization.

These findings suggest that tensor completion (based on tensor factorization) can efficiently impute missing values in the original sparse performance data, notably for unexplored questions and attempts. This enhancement is crucial for facilitating more comprehensive analysis and modeling in Intelligent Tutoring Systems (ITSs). The latent features, closely associated with learner-specific characteristics during the learning process, are captured with nuanced detail,

Table 1

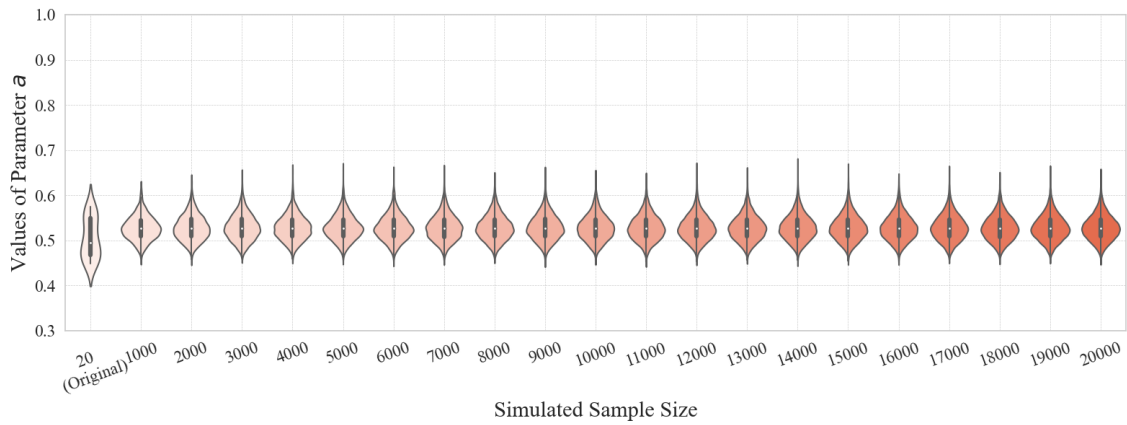
Results about the sparsity levels and latent features by tensor completion. **M**, **H**, and **E** denote **Medium**, **Hard**, and **Easy** lesson levels, respectively.

| Dataset | Sparsity Level (Original) | K (Latent Features) |
|------------|---------------------------|-----------------------|
| Lesson (M) | 84.02% | 6 |
| Lesson (H) | 85.45% | 6 |
| Lesson (E) | 81.25% | 4 |

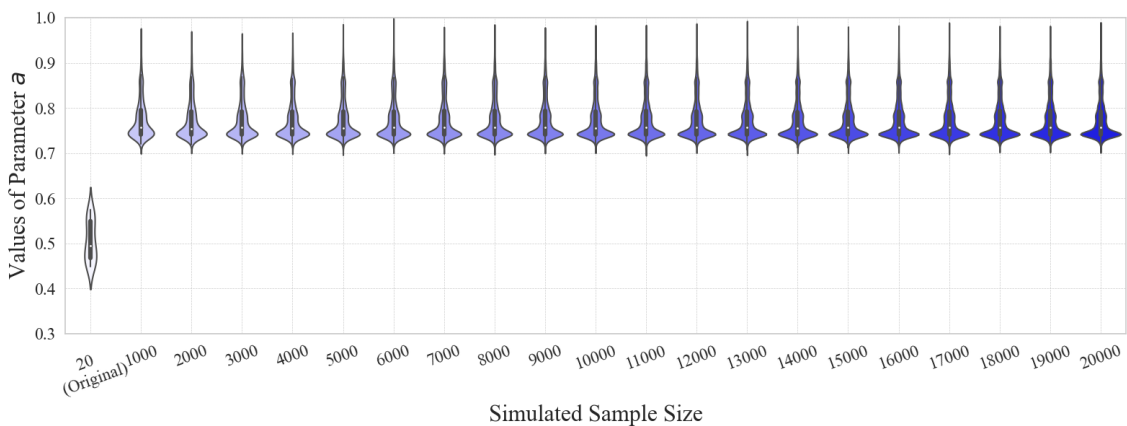
particularly in the context of reading comprehension. Further research is imperative to fully understand the underlying physical essence of these latent features.

The distributions of parameters a and b are illustrated in Figure 4 and Figure 5, respectively. These figures visualize the parameter distributions from an example cluster data set with an original size of 20, and exhibit simulations in increments of 1000, with total sizes ranging from 1000 to 20000.

Figure 4 demonstrates the distributions of parameters a , which is used to represent the learner's initial ability or prior knowledge. Figure 4a shows the distribution of the parameter a obtained by GAN simulation. As the sample size increases, the range of parameter a from the simulation sample mostly falls within the original range of parameter a , although it exhibits a longer tail distribution extending beyond the original maximum value of parameter a . The distribution of the parameter a obtained by GPT-4 simulation is illustrated in Figure 4b. Unlike those obtained from GAN simulation, the range of parameter a values here extends beyond the original range, which is particularly evident as the simulated sample size increases. This



(a) Distribution of parameter a by GAN simulation.

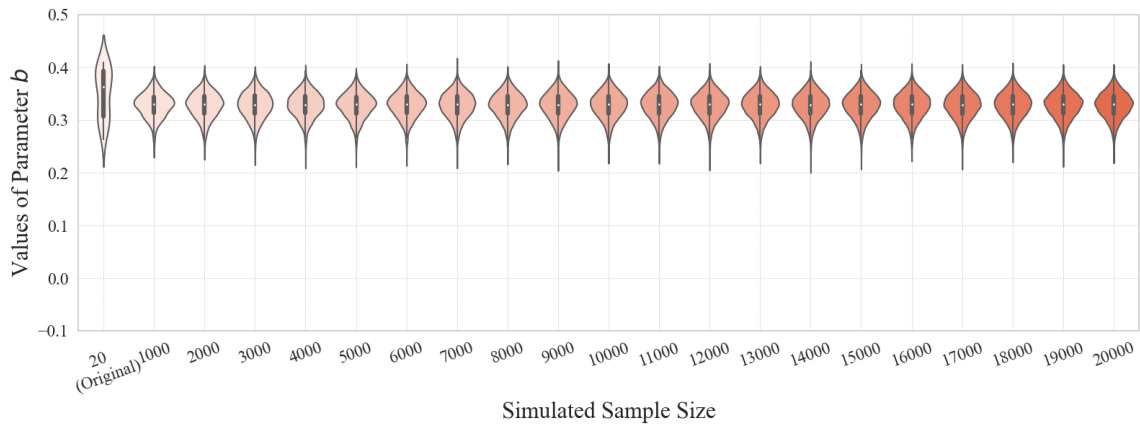


(b) Distribution of parameter a by GPT-4 simulation.

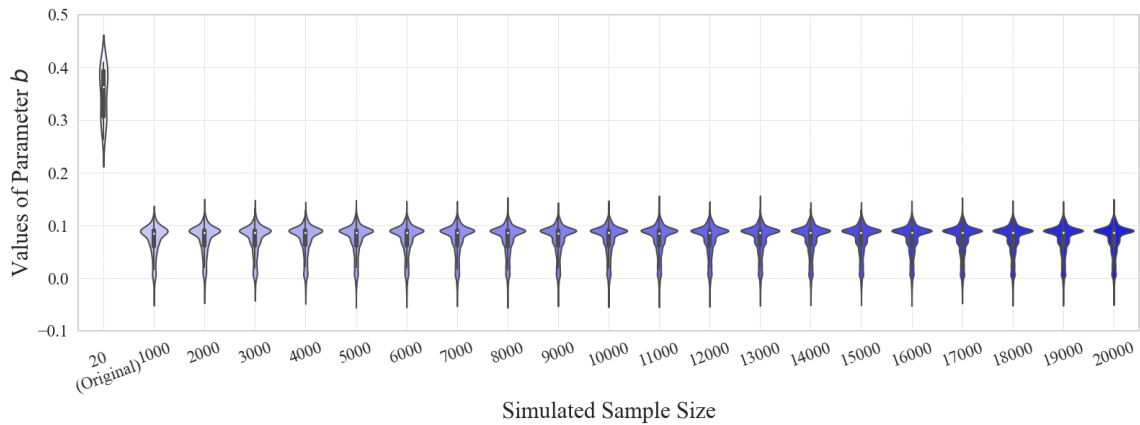
Figure 4: Distribution of parameter a by simulation.

suggests that the initial learning ability in GPT-4 simulated samples exhibits more variability and divergence from the original data compared to those from GAN simulation.

Then, Figure 5 demonstrates the distributions of parameters b as derived from both GAN and GPT-4 simulations. The parameter b represents the learning rate, which reflects how quickly a learner acquires knowledge through practice. The GAN simulation produces a narrower range of parameter b values, especially in terms of maximum and minimum values, when compared to the original dataset, as depicted in Figure 5a. With increasing sample size, this range generally maintains a consistent pattern. On the other hand, the GPT-4 simulation, as shown in Figure 5b, demonstrates a broader range for parameter b , extending beyond the original scope. This contrast suggests that GPT-4 simulation may capture a wider variability in learning rates compared to GAN simulation.



(a) Distribution of parameter b by GAN simulation.



(b) Distribution of parameter b by GPT-4 simulation.

Figure 5: Distribution of parameter b by simulation.

4. Discussion and Conclusion

This paper proposed the 3DG systematic simulation framework based on generative models (particularly GAN and GPT) to address data sparsity challenges in learning performance data within intelligent tutoring systems (ITS). The framework involves representing learner data from problem-solving step attempts as a three-dimensional tensor with the axes of learners, questions, and attempts. Tensor completion, based on tensor factorization, is then utilized to impute missing performance data entries, generating a dense tensor. Such imputation computation leverages the similarities in learner performance across various questions and attempts, capturing the sequential and temporal dynamics of learning [42, 43]. We have demonstrated the integration of generative models, including GAN and GPT-4 for creating scalable, individualized learning simulations aimed at enhancing learner models for personalized instruction. Our comparative analysis reveals that GAN surpasses GPT-4 in terms of reliability for scalable simulations.

Overall, the GAN simulations demonstrate a narrower and more consistent range of values for parameters a and b , indicating higher reliability for scalable simulations compared to the broader value range exhibited by the GPT-4 simulations. The mechanism for the GPT-4 simulations, refined through iterative optimization of GPT-4 prompts, involves selecting random values from a flat array of the original data. These values are then adjusted to match base probabilities, preserving the overall data distribution while facilitating the creation of an expanded dataset. Although valuable in computational simulations, this method generally underperforms in numerical computing compared to deep learning models, as demonstrated by the GAN's performance in this study.

Our findings shed light on the potential use of GPT-4 in simulating learner performance represented through numerical values in future research. *Firstly*, employing mixed-based prompts improves interoperability with numerical data, thus enhancing the efficiency of subsequent modeling and simulation computations. *Secondly*, the Chain-of-Thought (CoT) prompting technique delineates the steps for the simulation task, effectively directing GPT-4 in its reasoning process. This includes a structured approach comprising: **Understanding the Existing Matrix**, **Distribution Analysis**, **Clustering Information**, and **Simulation Process**. *Thirdly*, the computational power of GPT-4 in modeling and simulation is attributed to its capabilities in self-search, self-programming, and self-computing, all of which are facilitated by prompt engineering. This significantly enhances its utility in data analysis and modeling for future research endeavors. However, integrating GPT-4 with numerical computation presents fundamental challenges, as we discuss in the following section.

5. Limitations and Future Works

The capability of GPT-4 in performing deep learning tasks involving numerical computations remains insufficient, primarily due to the intrinsic limitations of large language models and platform constraints. Future research could productively explore the integration of GAN with GPT models, aiming to improve their interoperability and computational capabilities. Furthermore, the degree of sparsity in the original performance data, particularly when formatted as a tensor, significantly impacts the performance of generative models. Therefore, investigating the

sensitivity and robustness of tensor completion methods in response to different levels of data sparsity presents an important avenue for future studies. Such investigations are crucial for better integrating large language models within Intelligent Tutoring Systems (ITSs), potentially leading to more refined and effective educational tools.

6. Acknowledgements

We extend our sincere gratitude to Prof. Philip I. Pavlik Jr. from the University of Memphis and Prof. Shaghayegh Sahebi from the University at Albany - SUNY for their expert guidance on tensor factorization method.

References

- [1] J. R. Anderson, C. F. Boyle, B. J. Reiser, Intelligent tutoring systems, *Science* 228 (1985) 456–462.
- [2] A. T. Corbett, K. R. Koedinger, J. R. Anderson, Intelligent tutoring systems, in: *Handbook of human-computer interaction*, Elsevier, 1997, pp. 849–874.
- [3] K. VanLehn, The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems, *Educational psychologist* 46 (2011) 197–221.
- [4] A. C. Graesser, X. Hu, R. Sottolare, Intelligent tutoring systems, in: *International handbook of the learning sciences*, Routledge, 2018, pp. 246–255.
- [5] T.-N. Nguyen, Predicting student performance in an intelligent tutoring system, Ph.D. thesis, Stiftung Universität Hildesheim, 2011.
- [6] S. Pandey, G. Karypis, A self-attentive model for knowledge tracing, *arXiv preprint arXiv:1907.06837* (2019).
- [7] N. Zhang, Y. Du, K. Deng, L. Li, J. Shen, G. Sun, Attention-based knowledge tracing with heterogeneous information network embedding, in: *Knowledge Science, Engineering and Management: 13th International Conference, KSEM 2020, Hangzhou, China, August 28–30, 2020, Proceedings, Part I 13*, Springer, 2020, pp. 95–103.
- [8] C. Wang, S. Sahebi, S. Zhao, P. Brusilovsky, L. O. Moraes, Knowledge tracing for complex problem solving: Granular rank-based tensor factorization, in: *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, 2021, pp. 179–188.
- [9] X. Wang, S. Zhao, L. Guo, L. Zhu, C. Cui, L. Xu, Graphca: Learning from graph counterfactual augmentation for knowledge tracing, *IEEE/CAA Journal of Automatica Sinica* 10 (2023) 2108–2123.
- [10] P. I. Pavlik Jr, L. G. Eglington, L. Zhang, Automatic domain model creation and improvement., *Grantee Submission* (2021).
- [11] L. G. Eglington, P. I. Pavlik Jr, How to optimize student learning using student models that adapt rapidly to individual differences, *International Journal of Artificial Intelligence in Education* 33 (2023) 497–518.
- [12] L. Zhang, P. I. Pavlik Jr, X. Hu, J. L. Cockroft, L. Wang, G. Shi, Exploring the individual differences in multidimensional evolution of knowledge states of learners, in: *International Conference on Human-Computer Interaction*, Springer, 2023, pp. 265–284.

- [13] T. Wang, F. Ma, J. Gao, Deep hierarchical knowledge tracing, in: Proceedings of the 12th international conference on educational data mining, 2019.
- [14] W. Lee, J. Chun, Y. Lee, K. Park, S. Park, Contrastive learning for knowledge tracing, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 2330–2338.
- [15] A. Kossiakoff, W. N. Sweet, S. J. Seymour, S. M. Biemer, Systems engineering principles and practice, volume 83, John Wiley & Sons, 2011.
- [16] M. Baudin, A. Dutfoy, B. Iooss, A.-L. Popelin, Openturns: An industrial software for uncertainty quantification in simulation, in: Handbook of uncertainty quantification, Springer, 2017, pp. 2001–2038.
- [17] E. Acar, D. M. Dunlavy, T. G. Kolda, M. Mørup, Scalable tensor factorizations for incomplete data, *Chemometrics and Intelligent Laboratory Systems* 106 (2011) 41–56.
- [18] C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, *Journal of big data* 6 (2019) 1–48.
- [19] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, O. Tabona, A survey on missing data in machine learning, *Journal of Big Data* 8 (2021) 1–37.
- [20] T. Liu, J. Fan, Y. Luo, N. Tang, G. Li, X. Du, Adaptive data augmentation for supervised learning over missing data, *Proceedings of the VLDB Endowment* 14 (2021) 1202–1214.
- [21] N. Thai-Nghe, L. Drumond, T. Horváth, A. Krohn-Grimberghe, A. Nanopoulos, L. Schmidt-Thieme, Factorization techniques for predicting student performance, in: *Educational recommender systems and technologies: Practices and challenges*, IGI Global, 2012, pp. 129–153.
- [22] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems* 25 (2012).
- [23] S. Sahebi, Y.-R. Lin, P. Brusilovsky, Tensor factorization for student modeling and performance prediction in unstructured domain., *International Educational Data Mining Society* (2016).
- [24] T.-N. Doan, S. Sahebi, Rank-based tensor factorization for student performance prediction, in: *12th International Conference on Educational Data Mining (EDM)*, 2019.
- [25] S. Zhao, C. Wang, S. Sahebi, Modeling knowledge acquisition from multiple learning resource types, *arXiv preprint arXiv:2006.13390* (2020).
- [26] N. Thai-Nghe, T. Horváth, L. Schmidt-Thieme, et al., Factorization models for forecasting student performance., in: *EDM*, Eindhoven, 2011, pp. 11–20.
- [27] X. Chen, Z. He, Y. Chen, Y. Lu, J. Wang, Missing traffic data imputation and pattern discovery with a bayesian augmented tensor factorization model, *Transportation Research Part C: Emerging Technologies* 104 (2019) 66–77.
- [28] M. Jovanovic, M. Campbell, Generative artificial intelligence: Trends and prospects, *Computer* 55 (2022) 107–112.
- [29] D. Baidoo-Anu, L. Owusu Ansah, Education in the era of generative artificial intelligence (ai): Understanding the potential benefits of chatgpt in promoting teaching and learning, Available at SSRN 4337484 (2023).
- [30] A. C. Graesser, Z. Cai, W. O. Baer, A. M. Olney, X. Hu, M. Reed, D. Greenberg, Reading comprehension lessons in autotutor for the center for the study of adult literacy, in: *Adaptive educational technologies for literacy instruction*, Routledge, 2016, pp. 288–293.
- [31] Y. Fang, K. Shubeck, A. Lippert, Q. Chen, G. Shi, S. Feng, J. Gatewood, S. Chen, Z. Cai,

- P. Pavlik, et al., Clustering the learning patterns of adults with low literacy skills interacting with an intelligent tutoring system., Grantee Submission (2018).
- [32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, *Advances in neural information processing systems* 27 (2014).
- [33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, *Communications of the ACM* 63 (2020) 139–144.
- [34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, *Advances in neural information processing systems* 30 (2017).
- [35] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training (2018).
- [36] A. Newell, P. S. Rosenbloom, Mechanisms of skill acquisition and the law of practice., Technical Report, CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1980.
- [37] H. Cen, K. Koedinger, B. Junker, Learning factors analysis—a general method for cognitive model evaluation and improvement, in: *International conference on intelligent tutoring systems*, Springer, 2006, pp. 164–175.
- [38] R. DeKeyser, Skill acquisition theory, in: *Theories in second language acquisition*, Routledge, 2020, pp. 83–104.
- [39] D. Arthur, S. Vassilvitskii, K-means++ the advantages of careful seeding, in: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035.
- [40] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, S. Vassilvitskii, Scalable k-means++, *arXiv preprint arXiv:1203.6402* (2012).
- [41] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, *Advances in Neural Information Processing Systems* 35 (2022) 24824–24837.
- [42] C. M. Conway, M. H. Christiansen, Sequential learning in non-human primates, *Trends in cognitive sciences* 5 (2001) 539–546.
- [43] C. M. Conway, *Sequential Learning*, Springer US, Boston, MA, 2012, pp. 3047–3050.